SILICON LABS

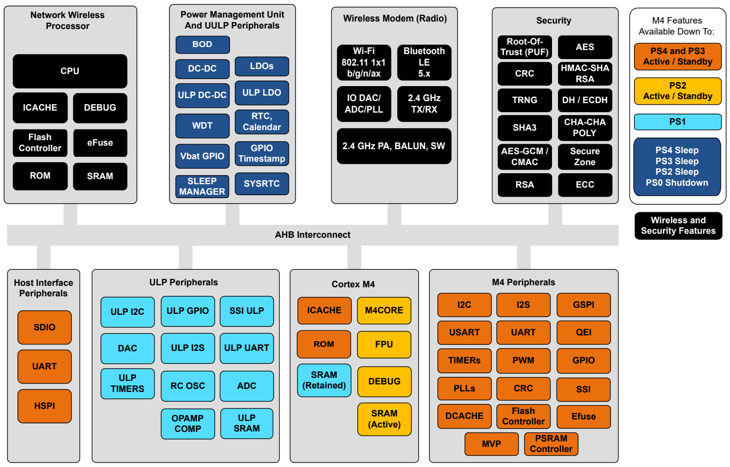
SiWx917 Family Reference Manual

Silicon Labs SiWG917 SoC is our lowest power Wi-Fi 6 plus Bluetooth LE 5.4 SoC, ideal for ultra-low power IoT wireless devices using Wi-Fi®, Bluetooth, Matter, and IP networking for secure cloud connectivity. It is optimal for developing battery operated devices that need long battery life. SiWG917 SoC includes an ultra-low power Wi-Fi 6 plus Bluetooth Low Energy (LE) 5.4 wireless CPU subsystem and an integrated microcontroller (MCU) application subsystem, security, peripherals, and power management subsystem all in a single 7x7 mm QFN package. The wireless subsystem consists of a multi-threaded Network Wireless Processor (NWP) running up to 160 MHz, baseband digital signal processing, analog front end, 2.4 GHz RF transceiver and integrated power amplifier. The application subsystem consists of an ARM® Cortex®-M4 running up to 180 MHz, embedded SRAM, FLASH, ultra-low power sensor hub, and matrix vector processor. The ARM® Cortex®-M4 is dedicated for peripheral and application-related processing, while the NWP runs the wireless and networking stacks on independent threads, thus providing a fully integrated solution that is ready for a wide range of embedded wireless IoT applications.

SiWG917x applications include:

- Smart Home
- Security Cameras
- HVAC
- Smart Sensors
- Smart Appliances
- Health and Fitness
- Pet Trackers
- Smart Cities
- Smart Meters
- Industrial Wearables
- Smart Buildings
- Asset Tracking
- Smart hospitals

SiWG917 applications include:



KEY FEATURES

- Wi-Fi 6 Single Band 2.4 GHz 20 MHz 1x1 stream IEEE 802.11 b/g/n/ax
- Bluetooth LE 5.4
- Single chip Matter Over Wi-Fi Solution
- ARM® Cortex® M4 Processor with FPU subsystem up to 180 MHz with rich set of Digital and Analog Peripherals.
- Wi-Fi 6 Benefits: TWT for improved efficiency and longer battery life, MU-MIMO/OFDMA for Higher Throughput, network capacity and low latency
- Best in Class Device and Wireless Security
- WLAN Tx power up to +19.5 dBm with integrated PA
- Bluetooth LE Tx power up to +19 dBm with integrated PA
- WLAN Rx sensitivity as low as -97.5 dBm
- Wi-Fi Standby Associated mode current: 65 µA @ 1-second listen interval
- Embedded Flash option up to 8 MB/ optional external Flash up to 16 MB
- Embedded PSRAM option up to 8 MB/ optional external PSRAM option up to 16 MB
- Ultra-low power sensor hub peripherals
- Matrix Vector Processor (MVP)
- Embedded Wi-Fi, Bluetooth LE, Matter, and networking stacks supporting wireless coexistence
- Three software-configurable MCU application memory options for sharing the RAM between the wireless, system, and application (192/256/320 KB)
- Operating temperature: -40 to +85 ºC
- Single or dual-supply operation:
  - Single supply: 3.3 V
  - Dual supply: 3.3 V and 1.8 V

Table of Contents

1. Introduction

This document contains reference material for the SiWG917 SoC devices. All modules and peripherals in the SiWG917 devices are described in general terms. Not all modules are present in all devices and the feature set for each device might vary. Such differences, including specific device pinouts, are covered in the device data sheets.

1.1 Related Documentation

Further documentation on the SiWG917 SoC devices can be found at the following web pages:

- Github link for 917 Release: https://github.com/SiliconLabs/wiseconnect/
- Software reference manual : https://github.com/SiliconLabs/wiseconnect/blob/master/docs/software-reference/manuals/siwx91x-software-reference-manual.md
- MCU API Reference: https://github.com/SiliconLabs/wiseconnect/blob/master/docs/software-reference/manuals/siwx91x-software-reference-manual.md

2. Feature List

- **Microcontroller**
  - ARM® Cortex® M4 core with up to 180 MHz, 225 DMIPS performance
  - Integrated Floating Point Unit (FPU), and Nested Vectored Interrupt Controller (NVIC)
  - Serial Wire Debug (SWD) and Joint Test Action Group (JTAG) debug options
  - Internal and external oscillators with Phase Locked Loops (PLLs)
  - In-Flash Application Programming (IAP), In-System Programming (ISP), and Over-the-Air (OTA) Wireless Firmware Upgrade
  - Power-On Reset (POR), Brown-Out and Black-out Detect (BOD) with separate thresholds
  - M4 has 2 dedicated Quad Serial Peripheral Interface (QSPI) controllers for Pseudo Static Random Access Memory (PSRAM) and Flash
- **Matrix Vector Processor (MVP)**
  - Co-processor for offloading matrix math operations
  - Delivers faster Machine Learning (ML) inference with lower power consumption
  - Performs Real and Complex Matrix and Vector operations, providing manifold computing efficiency
- **Memory**
  - Embedded SRAM (Static Random Access Memory) up to 672 KB total for Application and Wireless Processor
  - On-chip SRAM of 192, 256, or 320 KB for M4 Application Processor based on the memory configuration
  - Support for Flash up to 8 MB (in-package), or Optional External Flash up to 16 MB[2]
  - Support for PSRAM option up to 8 MB (in-package), Optional External PSRAM up to 16 MB[2]
- **Digital Peripherals**
  - Secure Digital Input Output (SDIO) 2.0 secondary
  - 1x Universal Synchronous/Asynchronous Receiver Transmitter (USART)
  - 2x Universal Asynchronous Receiver Transmitter (UART)
  - 3x Synchronous Serial Interface (SSI)
  - 1x General SPI (GSPI)
  - 1x Host SPI (HSPI) [2]
  - 3x Inter-Integrated Circuit (I2C)
  - 2x Inter-IC Sound Bus (I2S)
  - Pulse Width Modulation (MCPWM)
  - Quadrature Encoder Interface (QEI)
  - Timers: 4x 16/32-bit, 1x 24-bit, Watchdog Timer (WDT), Real Time Counter (RTC)
  - Up to 45 General Purpose Input Outputs (GPIOs) with GPIO multiplexer
- **Analog Peripherals**
  - 12-bit 16-ch, 2.5 Msps Analog to Digital Converter (ADC)
  - 10-bit Digital to Analog Converter (DAC)
  - 3x Op-amps[2]
- **Security**
  - Secure Boot
  - Secure firmware upgrade through boot-loader, Secure OTA
  - Secure Key storage and HW device identity with PUF
  - Secure Zone
  - Encrypted Execute in Place (XiP) from flash/ PSRAM
  - Secure Attestation
  - Hardware Accelerators: Advanced Encryption Standard (AES) 128/256/192, Secure Hash Algorithm (SHA) 256/384/512, Hash Message Authentication Code (HMAC), Random Number Generator (RNG), Cyclic Redundancy Check (CRC), SHA3, AES-Galois Counter Mode (GCM), Cipher based Message Authentication Code (CMAC), ChaCha-poly, True Random Number Generator (TRNG)
  - Software Implementation: RSA, ECC
  - Anti Rollback
  - Debug Lock
- **Wi-Fi**
  - Compliant to single-spatial stream IEEE 802.11 b/g/n/ax with single band (2.4 GHz) support
  - Support for 20 MHz channel bandwidth for 802.11n and 802.11ax.
  - Operating Modes: Wi-Fi 4 STA, Wi-Fi 6 (802.11ax) STA, Wi-Fi 4 AP, Enterprise STA, Wi-Fi 6 STA + Wi-Fi 4 AP, Wi-Fi STA + BLE
  - Wi-Fi 6 Features: Individual Target wake-up time (iTWT), Broadcast TWT (bTWT)[2], Intra PPDU power save[2], SU extended range (ER), DCM (Dual Carrier Modulation), DL MU-MIMO, DL/UL OFDMA, MBSSID[2], BFRP, Spatial Re-use[2], BSS Coloring[2], and NDP feedback up to 4 antennas
  - Transmit power up to +19.5 dBm with integrated PA
  - Receive sensitivity as low as -97.5 dBm
  - Data Rates: 802.11b: 1, 2, 5.5, 11; 802.11g: 6, 9, 12, 18, 24, 36, 48, 54 Mbps; 802.11n: MCS0 to MCS7; 802.11ax: MCS0 to MCS7
  - Operating Frequency Range: 2412 MHz – 2484 MHz
  - PTA Coexistence with Zigbee/Thread/Bluetooth
- **Bluetooth**
  - Transmit power up to +19 dBm with integrated PA
  - Receive sensitivity — LE 1 Mbps: -96 dBm, LR 125 kbps: -107 dBm
  - Operating Frequency Range — 2.402 GHz - 2.480 GHz
  - Support LE (1 Mbps & 2 Mbps) and LR (125 kbps & 500 kbps) rates
  - Advertising extensions
  - Data length extensions

- **Analog Peripherals (cont.)**
  - 2x Comparators
  - Temperature Sensor
- **Embedded Bluetooth Stack**
  - Support GAP profile
  - Support GATT profile
  - Support SMP
  - Support LE L2CAP
- **WiSeConnect SDK Features**
  - Simplified and Unified DX for Wi-Fi API and Platform APIs
  - Simplifies application development and presents clean and standardized APIs
  - Universal Configurator (UC) enables componentization, simplifying configuration of peripherals and examples
  - BSD and ARM IoT-compliant socket API
  - Available through Simplicity Studio and Github
- **Intelligent Power Management**
  - Power optimizations leveraging multiple power domains and partitioned sub systems
  - Many system-, component-, and circuit-level innovations and optimizations
  - Multiple Power states: PS4 / PS3 / PS2 / PS1[2] / PS0
  - Voltage & Frequency Scaling for MCU
  - Application-based Gear Shifting (switches from one power state to another based on processing requirements) for MCU
  - Deep sleep mode with only timer active – with and without RAM retention
- **Ultra Low Power Sensor Hub System**[2]
  - Offloads Sensor data collection without a need for MCU to be active
  - Extends battery life and recharging interval for IoT Sensors
- **MCU Sub-System Power Consumption**
  - Active current as low as 40.4 µA/MHz @ 20 MHz in low-power mode
  - Active current as low as 70.5 µA/MHz @ 180 MHz in high performance mode
  - Deep sleep mode current: ~2.5 µA
  - Voltage & frequency scaling
  - Deep sleep mode with only timer active – with and without RAM retention
- **Wireless Sub-System Power Consumption**
  - - Wi-Fi 4 Standby Associated mode current: 65 µA @ 1-second beacon listen interval
    - Wi-Fi 1 Mbps Listen current: 16.5 mA
    - Wi-Fi LP chain Rx current: 22 mA
    - Deep sleep current ~2.5 µA, Standby current (352 KB RAM retention) ~10 µA
- **Operating Conditions**
  - Single or dual-supply operation:
    - Single supply: 3.3 V
    - Dual supply: 3.3 V and 1.8 V
  - Operating temperature: -40 to +85 ºC
- **Bluetooth (cont.)**
  - LL privacy
  - LE dual role
  - BLE acceptlist
  - 2 Simultaneous BLE Connections (2 Peripheral, 2 Central, or 1 Central & 1 Peripheral)
- **Ultra-Low Power (ULP) Peripherals**
  - ULP I2C
  - ULP I2S
  - ULP UART
  - ULP GPIO
  - ULP Timers
  - ULP ADC
  - ULP DAC
  - ULP UDMA
  - ULP SSI Primary
- **RF & Modem Features**
  - Integrated baseband processor with calibration memory
  - Integrated RF transceiver, high-power amplifier, balun and T/R switch
- **Embedded Wi-Fi Stack**
  - Support for Embedded Wi-Fi STA mode, Wi-Fi access point mode, and concurrent (AP+STA) mode
  - Supports advanced Wi-Fi security features: WPA personal, WPA2 personal, WPA3 personal, WPA/WPA2 enterprise in STA mode
  - Networking: Integrated IPv4/IPv6 stack, TCP, UDP, ICMP, ICMPv6, ARP, DHCP Client/Server, DHCPv6 Client/Server, DNS Client, SSL3.0/TLS1.3 Client, SNTP, mDNS, SNI
  - Applications: HTTP/s Client, HTTP/s Server[2], MQTT/s Client, AWS Client, Azure Client
  - Sockets: BSD sockets, IoT sockets
  - Over-the-Air (OTA) firmware update
  - Provisioning using Wi-Fi AP or BLE
- **Software and Regulatory Certifications**
  - Wi-Fi Alliance: Wi-Fi 4, Wi-Fi 6
  - Matter Certification
  - Bluetooth SIG Qualification
  - Regulatory pre-certifications (FCC, IC, RED, UKCA, MIC)[1]
- **Advanced Software Features**
  - FreeRTOS, Zephyr[2]
  - Amazon AWS Cloud Connectivity, Microsoft Azure Cloud Connectivity
  - SensorHub (SensorHub framework which enables easier integration of new sensors)
  - SoC communication to external host via Co-Processor Communication (CPC) - Supported host interfaces are SDIO[2]
  - Dual-Host: Support both embedded TCP-IP and TCP-IP bypass simultaneously[2]

Note:

1. For latest certification information, refer to regulatory app notes or contact Silicon Labs for availability.
2. For information about software roadmap features, and lists of available features and profiles, contact Silicon Labs or refer to Release Notes and Reference Manuals.

3. System Processor

3.1 Introduction

The SIWx917 family include ARM® Cortex®-M4 processor for user application. The Cortex-M4 processor is a high performance 32-bit processor designed for the microcontroller market. It offers significant benefits to developers, including:

- Outstanding processing performance combined with fast interrupt handling
- Enhanced system debug with extensive break-point and trace capabilities
- Efficient processor core, system and memories
- Ultra-low power consumption with integrated sleep modes
- Platform security robustness.

3.1.1 Features

- Harvard architecture
- Separate data and program memory buses (No memory bottleneck as in a single bus system)
- 3-stage pipeline
- Thumb-2 instruction set
- Enhanced levels of performance, energy efficiency, and code density
- Code-patch ability for ROM system updates
- IEEE754-compliant single-precision FPU
- 24-bit System Tick Timer for Real Time OS
- Power control optimization of system components
- Integrated sleep modes for low power consumption
- Deterministic, high-performance interrupt handling for time-critical applications
- Optimized for low latency, nested interrupts
- Extensive debug and trace capabilities: Serial Wire Debug and Serial Wire Trace reduce the number of pins required for debugging, tracing, and code profiling.

3.2 Functional Description

For a full functional description of the ARM Cortex®-M4 implementation in the SIWx917 family, the reader is referred to the ARM Cortex®-M4 documentation (https://developer.arm.com/documentation/100166/0001/).

3.3 Instruction Cache Controller

3.3.1 General Description

The Instruction Cache (ICACHE) Controller controls the instruction fetching from External Memory into Local Cache for access to the processor. The ICACHE provides fast access to recently executed instructions, improving both speed and power consumption of code execution. ICACHE is disabled by default and it has to be enabled by setting icache_enable bit of ICACHE_CTRL_REG

3.3.2 Features

- Bypass cache mode
- 32 or 128 bit line usage mode
- AHB wrap transfer mode
- Cache enable/disable mode
- 16k/8k access mode
- 4 Ways set associative, 4KB for each way
- There is a direct path for cortex I and D port to access icache.

3.3.3 Functional Description

The Cache controller controls the processor access to the External memory by loading 16 bytes of program data into the local cache every time a cache miss occurs.

The main functions of the ICC are as follows:

- Whenever there is a fetch request from the CPU, ICC reads the data and tag rams and gives grant to CPU, if controller wins the arbitration with AHB port for data and tag rams.
- In the next cycle, it checks whether it's a hit or miss.
- Whenever there is a hit, the ICC sends the instruction data required for the CPU, and gives fetch ready signal.
- Whenever there is a miss, the ICC sends a request to AHB master for getting data from the memory and copies into the local buffer. It also gives fetch miss signal to CPU. And asserts line_busy signal as soon as it gives trigger to AHB.
- When the hready for the required data is present ICC gives line ready signal.
- Line busy signal will be deasserted when AHB transfer is finished.
- CPU has to re- request for that address when line ready is given.
- ICC can serve hits under miss but not miss under miss cases.

Table 3.1. Address Mapping

AHB address bits			Block accessed
31:26	15:14	13	
000001	xx	x	Arm Cache Req Gen (accessible only through cortex I-port)
xxxxx0	00	x	Control Registers
xxxxx0	01	0	trams set1
xxxxx0	01	1	trams set2
xxxxx0	10	x	drams set1
xxxxx0	11	x	drams set2

Ram	Offset Address Range	Size (in Bytes)
Dram set1	32’h0000_8000 – 32’h0000_BFFF	16K
Dram set2	32’h0000_C000 – 32’h0000_FFFF	Reserved
Tram set1	32’h0000_4000 – 32’h0000_5FFF	2K
Tram set2	32’h0000_6000 – 32’h0000_7FFF	Reserved

3.3.4 Register Summary

Base Address for M4 Instance : 0x2028_0000
Base Address for NWP Instance : 0x0080_0000

Register Name	Offset	Description
Section 3.3.5.1 RAM_CTRL_REG	0x4	RAMs control register
Section 3.3.5.2 ICACHE_CTRL_REG	0x14	Icache control register
ADDR_TRANSLATE_SEG1_CTRL_REG	0x24	Address Translate Value Segment Register_1
ADDR_TRANSLATE_SEG2_CTRL_REG	0x28	Address Translate Value Segment Register_2
ADDR_TRANSLATE_SEG3_CTRL_REG	0x2C	Address Translate Value Segment Register_3
ADDR_TRANSLATE_SEG4_CTRL_REG	0x30	Address Translate Value Segment Register_4
ADDR_TRANSLATE_SEG5_CTRL_REG	0x34	Address Translate Value Segment Register_5
ADDR_TRANSLATE_SEG6_CTRL_REG	0x38	Address Translate Value Segment Register_6
ADDR_TRANSLATE_SEG7_CTRL_REG	0x3C	Address Translate Value Segment Register_7
ADDR_TRANSLATE_SEG8_CTRL_REG	0x40	Address Translate Value Segment Register_8
ADDR_TRANSLATE_SEG9_CTRL_REG	0x44	Address Translate Value Segment Register_9
ADDR_TRANSLATE_SEG10_CTRL_REG	0x48	Address Translate Value Segment Register_10
ADDR_TRANSLATE_SEG11_CTRL_REG	0x4C	Address Translate Value Segment Register_11
ADDR_TRANSLATE_SEG12_CTRL_REG	0x50	Address Translate Value Segment Register_12
ADDR_TRANSLATE_SEG13_CTRL_REG	0x54	Address Translate Value Segment Register_13
ADDR_TRANSLATE_SEG14_CTRL_REG	0x58	Address Translate Value Segment Register_14
ADDR_TRANSLATE_SEG15_CTRL_REG	0x5C	Address Translate Value Segment Register_16

Register Name	Offset	Description
Section 3.3.5.4 THREAD_WAY_ALLOCATION_VECTOR_REG	0xD4	Thread way allocation vector register

3.3.5 Register Description

3.3.5.1 RAM_CTRL_REG

Table 3.2. Rams_Ctrl_Register

Bit	Access	Function	Reset Value	Description
0	Rams_ownership of second set	R/W	0	This bit controls the ownership of second set of RAMs when number of ways is 4 and 32k memory is enabled. 0-> controller cannot use set2. It is available on AHB. 1-> controller gets access of dram set2 when cache is enabled When the number of ways is 8, the ownership of RAMs is always with controller.(when cache enabled).This bit is ignored

3.3.5.2 ICACHE_CTRL_REG

Table 3.3. ICache_Ctrl_Register

Bit	Access	Function	Reset Value	Description
30:8	Reserved	R	0	Reserved
7	disable_fetch_256bit_lb	R/W	0	When set, disables fetching from line buffer which is present in 256-bit mode prefetch module. This will help in saving two cycles, if a request results in miss but present in prefetch line buffer. Even if the request served by prefetch buffer, line busy. line ready signals from icc will be asserted. Impact of these assertion on NWP is not clear and hence disable is added. These assertions shouldn't have any problem in M4
6	mode_256bit_line	R/W	0	0 -> One 128 bit line buffer is used. Four beat AHB transaction is initiated with the external memory 1 -> Two 128 bit line buffers are used. Eight beat AHB transaction is initiated with the external memory
5	lru_8ways	R/W	0	0-> 8 ways logic is disabled in controller 1-> 8 ways logic is disabled in controller
4	icache_line_buf_invalid	R/W	0	0-> line buffer valid for icc ‘n’ 1-> line buffer invalid for icc ‘n’ This is a self clearing bit
3	icache_ahb_wrap_mode	R/W	0	0-> wrap mode is disabled for icc ‘n’. 1-> wrap mode is enabled for icc ‘n’
2	mode32_128bit_line	R/W	0	0-> 128 bit mode enabled for icc ‘n’. AHB requests will be 128 bit 1-> 32 bit mode is enabled for icc ‘n’. AHB requests will be 32 bit The above is valid only when cache is disabled
1	bypass_cache	R/W	0	0-> Fetch Requests are served through ICache RAMs 1-> Fetch Requests are served via AHB, bypassing the ICache RAMs The above is valid only when cache is enabled.
0	icache_enable	R/W	0	0 -> ICache is not enabled for icc ‘n’ 1-> ICache is enabled and cache access can take place via ICC ‘n’

3.3.5.3 ADDR_TRANSLATE_SEG N_CTRL_REG

Table 3.4. Address_Translate register

Bit	Access	Function	Reset Value	Description
31:21	Segment Address Value	R/W	0x0	NWP - Hold the 11 bit segment translate address when icache is enabled. These 31:21 bits directly map to 31:21 bits of ahb address. M4 - For address translation functionality is disabled. In M4SS Addr_translate_value_seg1_ctrl1[22:21] are used for below purpose 21 - icache output will be registered and given to processor. This bit has to be set above 120MHz 22 - this is applicable only when above bit is set. If this bit is set, data will be served through unregistered path if there is hit to cache buffer line.
20:0	Reserved	R	0x0	Reserved

3.3.5.4 THREAD_WAY_ALLOCATION_VECTOR_REG

Table 3.5. Thread_way_allocation register

Bit	Access	Function	Reset Value	Description
(31: num_threads*num_ways)	Reserved	R	0x0	Reserved
(num_threads* num_ways) -1:0	Th_access	R/W	0x0	Thread way allocation vector: thread_alloc_vec[n] = th_access[((n+1)*num_ways-1:n*num_ways]

4. MCU Bus Matrix

4.1 General Description

The High Performance MCU AHB ICM (Inter-connect Matrix) is a multi-layer interconnect implementation of the AHB protocol designed for higher performance and higher frequency systems. The ICM operates as the same frequency as the processor. A 14 Primary x 14 Secondary AHB ICM is used. Multilayer bus matrix enables simultaneous access of peripherals by different primaries. This chapter discusses features, high level architecture and register interface details.

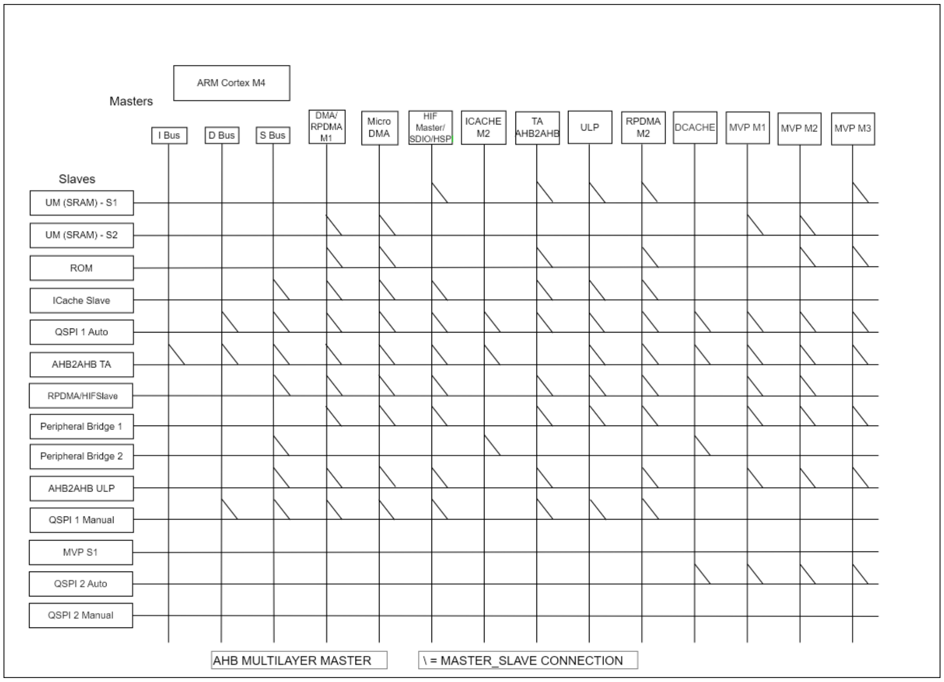
4.2 Features

- Multilayer interconnect matrix for high performance
- 14 Primaries and 14 Secondaries
- Concurrent accesses allowed to secondaries on different layers
- Operation at same frequency as Cortex M4 processor

4.3 Functional Description

4.3.1 Overview

The following diagram shows the interconnect configuration/connections possible between primaries and secondaries.



Masters	ARM Cortex M4													
Slaves	I Bus	D Bus	S Bus	DMA/ RPDMA M1	Micro DMA	HIF Master/ SDIO/HSP	ICACHE M2	TA AHB2AHB	ULP	RPDMA M2	DCACHE	MVP M1	MVP M2	MVP M3
UM (SRAM) - S1						\		\	\	\				\
UM (SRAM) - S2				\	\							\	\	
ROM				\	\			\		\			\	\
ICache Slave			\	\	\	\		\	\	\				
QSPI 1 Auto		\	\	\	\	\	\	\	\	\	\	\	\	\
AHB2AHB TA	\	\	\	\	\	\	\		\	\	\	\	\	\
RPDMA/HIFSlave			\	\	\	\		\	\	\		\	\	
Peripheral Bridge 1				\	\	\		\	\	\		\	\	\
Peripheral Bridge 2			\				\				\			
AHB2AHB ULP			\	\	\	\		\		\		\	\	\
QSPI 1 Manual		\	\	\	\	\		\	\	\				
MVP S1														
QSPI 2 Auto											\	\	\	\
QSPI 2 Manual														

Note: Refer to the product-specific data sheet to identify available peripherals in your part.

4.3.2 APB

- The APB is part of the AMBA 3 protocol family.
- It provides a low-cost interface that is optimized for minimal power consumption and reduced interface complexity.
- The APB interfaces to any peripherals that are low-bandwidth and do not require the high performance of a pipelined bus interface.
- The APB has unpipelined protocol.
- All signal transitions are only related to the rising edge of the clock to enable the integration of APB peripherals easily into any design flow.
- Every transfer takes at least two cycles.
- It can be used to provide access to the programmable control registers of peripheral devices.

4.3.3 Primary and Secondary Details

The following are the AHB primaries connected to ICM.

Primary Name	Description
Cortex M4 I Bus	Processor's Instruction Bus
Cortex M4 D Bus	Processor's Data Bus
Cortex M4 S Bus	Processor's System Bus
GPDMA Primary – 1	Multi channel GPDMA has two parallel Primary channels into the AHB bus matrix. Both DMA channels can independently access the bus matrix without any arbitration. This helps in ensuring that the latency is minimized for latency-critical peripherals.
GPDMA Primary – 2	
UDMA Primary	DMA Primary for the Micro DMA (UDMA).
ICache Primary	I-Cache fetches instructions from external NOR flash through this primary.
SDIO/ HSPI	SDIO/ HSPI's DMA.
NWP AHB to AHB Bridge	Used for accesses coming from NWP subsystem.
ULPSS AHB to AHB Bridge	Used for accesses coming from ULP MCU subsystem.
DCache Primary	D-Cache fetches data from external PSRAM through this primary.
MVP Primary - 1	Used to fetch operands from memories.
MVP Primary - 2	Used to fetch operands from memories.
MVP Primary -3	Used to load the Result to memories.

The following are the AHB secondaries connected to ICM.

Secondary	Description
Unified Memory - 1	Memory can be accessed through two secondaries. DMAs(GPDMA/UDMA) use secondary 2 and rest of the primaries use secondary 1. This helps in reducing the wait cycles when these two groups of primaries are accessing different banks in the on-chip SRAM (bank size is 16K). Not that when the two primaries are accessing same bank, then there will be wait states. Also note that processor buses have tightly coupled path to the memory and don't go through bus matrix.
Unified Memory - 2	
AHB to APB Bridge - 1	All primaries except the processor access the peripherals through this bridge.
AHB to APB Bridge - 2	Peripheral bridge port that is dedicated to the processor. This ensures minimal latency for peripheral accesses from processor.
ROM	ROM is a separate secondary.
Icache Secondary	Icache controller configuration is done through this secondary.
QSPI 1 Automode	Flash is presented as a memory mapped device and can be accessed through this secondary.
QSPI 1 Manualmode	This channel can be used to configure the QSPI Flash controller and do manual mode writes and reads to flash.
NWP AHB to AHB Bridge Secondary	Accesses to NWP subsystem go through this secondary.
ULPSS AHB to AHB Bridge Secondary	Accesses to ULP subsystem secondaries go through this secondary.
SDIO/HSPI/GPDMA Configuration Secondary	Accesses to SDIO/HSPI/GPDMA configuration registers go through this secondary.
MVP Secondary 1	Used to configure the MVP registers.
QSPI 2 Automode	PSRAM is presented as a memory mapped device and can be accessed through this secondary.

Secondary	Description
QSPI 2 Manualmode	This channel can be used to configure the QSPI controller and do manual mode writes and reads to PSRAM.

4.3.4 Address Mapping

The following table has the base addresses of memories and high-speed peripherals.

Table 4.1. MCU AHB Secondaries Address Mapping

	Module Name	Size	Start Address
Memories			
	LP SRAM	1 MB	0x0000_0000 **Note:** Note: Add 0x0050_0000 to the SRAM addresses for access from outside M4SS
	ROM	1 MB	0x0030_0000
AHB Peripherals			
	QSPI 1 Auto Mode	32 MB	0x0800_0000
	QSPI 1 Manual Mode	256 KB	0x1200_0000
	QSPI 2 Auto Mode	32 MB	0x0A00_0000
	QSPI 2 Manual Mode	256 KB	0x1204_0000
	SDIO/HSPI Secondary	1KB	0x2020_0000
	Icache Secondary	64 KB	0x2028_0000
	GPDMA Secondary	512 KB	0x2108_0000
	ULPSS AHB Bridge Secondary	256 KB	0x2404_0000
	APB Bridge	64 MB	0x4400_0000
	NWP AHB Bridge Secondary	512 MB	0x0010_0000 / 0x0040_0000 / 0x0060_0000 / 0x0400_0000 / 0x1000_0000 / 0x2010_0000 / 0x2040_0000 / 0x2100_0000 / 0x2200_0000 / 0x4000_0000
	MVP Secondary 1	256 KB	0x2400_0000

The following table has the base addresses of all low speed peripherals.

Table 4.2. MCU APB Peripherals Address Mapping

Peripheral	Base Address
PERIPHERAL Power Domain	
UART0	0x4400_0000

Peripheral	Base Address
USART0	0x4400_0100
I2C0	0x4401_0000
SSI_MST	0x4402_0000
UDMA	0x4403_0000
DCACHE	0x4404_0000
SSI_SLV	0x4501_0000
UART1	0x4502_0000
GSPI_1	0x4503_0000
CONFIG_TIMER	0x4506_0000
CRC	0x4508_0000
HWRNG	0x4509_0000
SGPIO	0x4700_0000
I2C1	0x4704_0000
I2S0	0x4705_0000
QEI*	0x4706_0000
PWM	0x4707_0000
Peripherals part of ALWAYS ON Domain	
VIC	0x4611_0000
ROM_PATCH	0x4612_0200
EGPIO	0x4613_0000
REG_SPI	0x4618_0000
PMU	0x4600_0000
PAD_CFG	0x4600_4000
MISC_CFG	0x4600_8000
EFUSE	0x4600_C000

The following table has the base addresses of all low speed ULP MCU peripherals.

Table 4.3. ULP MCU APB Peripherals Address Mapping

Peripheral	Starting Address
ULP I2C	0x2404_0000
ULP I2S	0x2404_0400
ULP SSI	0x2404_0800
ULP Config	0x2404_1400
ULP UART	0x2404_1800
ULP TIMER	0x2404_2000
AUX ADC DAC Controller	0x2404_3800
NPSS_APB	0x2404_8000

Peripheral	Starting Address
ULP EGPIO	0x2404_C000
IPMU Reg Access SPI	0x2405_0000
ULP Memory	0x2406_0000
ULP UDMA	0x2407_8000

5. Memory Architecture

5.1 General Description

This section describes the memory architecture of SiWx917 chip. It has on-chip ROM, RAM and off-chip FLASH connectivity. Sizes of ROM/RAM/FLASH/PSRAM will vary depending on the chip configuration.

5.2 Features

Highlights:

- Unified memory architecture - software can partition the memory between code and data usage
- Multiport - RAMs support multiport access - allowing simultaneous access from different primaries (I, D, DMAs) to non overlapping regions without any cycle penalty
- ROM/RAMs are tightly coupled to the processor I/D buses to reduce the latency and power

The Cortex-M4F processor has following memory:

- On-chip SRAM of 192K/256K/320Kbytes based on the chip configuration
- 8 Kbytes is present in the Ultra-low-power peripheral subsystem. This memory is present on the S-bus of the Cortex-M4 and is primarily used by the ULP MCU peripherals like ULP I2S, etc.
- 64 Kbytes of ROM used by bootloader and peripheral drivers.
- 16 Kbytes of Instruction cache enabling eXecute In Place (XIP) with external quad SPI SDR flashes.
- eFuse of 32 bytes (available for customer applications)
- 16 Kbytes of Data cache enabling data fetching with PSRAM

The NWP has following memory:

- On-chip SRAM of 480K/416K/352Kbytes based on chip configuration.
- 448 Kbytes of ROM which holds the Secure primary bootloader, Network Stack, Wireless stacks and security functions.
- 16 Kbytes of Instruction cache enabling eXecute In Place (XIP) with quad SPI flash memory.
- Based on the package configuration up to 8 MBytes of "in-package" Quad SPI flash is available for the NWP. This flash can be shared with Cortex-M4 in common flash mode (See Common Flash).
- eFuse of 1024 bytes (used to store primary boot configuration, security and calibration parameters)

5.3 Functional Description

The following diagram shows the memory architecture.

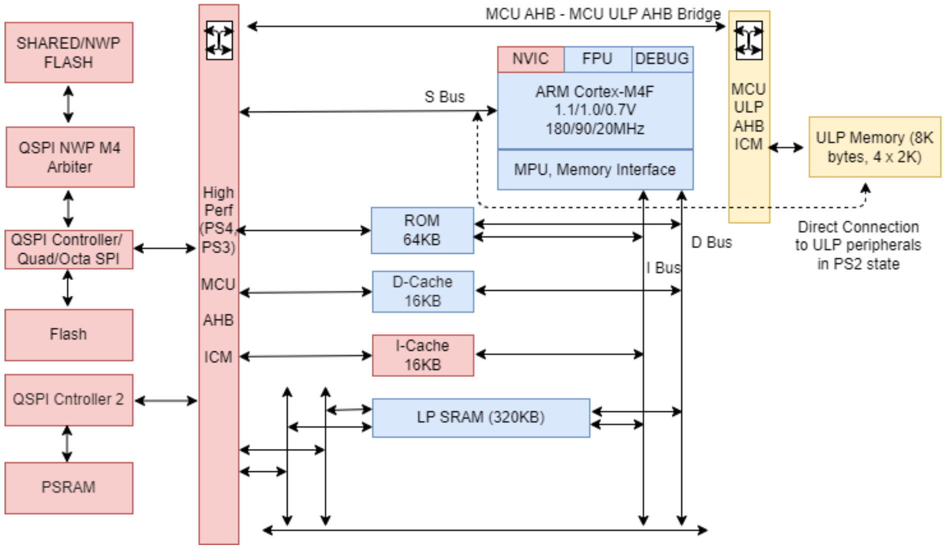


Figure 5.1. Memory Architecture

5.4 Unified Memory Architecture and Multiport

The MCU have unified memory architecture. There is no hardware level partitioning between code and data sections. Software can partition according to the application requirement. Memory is divided into 16K physical banks. For best performance, it is better to partition the memory at 16K granularity. LP SRAM supports four ports. Two of the ports are connected I/D buses of Cortex-M4. The other two ports are connected to ICM and meant to be used by other masters on bus like GPDMA, UDMA, and high-speed peripheral DMAs.

The amount of SRAM which is available for use varies between different power states (as described in Section 9. Power Architecture).

5.4.1 LP SRAM

LP SRAM is of size 320 KB. This can be accessed in PS4/PS3/PS2 states. This memory is available in all chip configurations. This memory supports four ports.

5.4.2 ROM

The ROM is of size 64 KB. This can be accessed in PS4/PS3/PS2 states. This will contain application boot code and some of the peripheral APIs. This memory is available in all chip configurations.

5.4.3 ULP Memory

ULP memory is of size 8 KB. This can be accessed in PS4/PS3/PS2/PS1* states. This is mainly used for sensor data collection from ULP peripherals. In PS2 state, processor will be accessing this memory directly without going through ICM and AHB2AHB bridge. This will reduce the cycles required to access this memory.

5.4.4 ICache Memory

There is a 16 KB of instruction cache memory. This can be accessed in PS4/PS3 states. This memory is also accessible through S bus at icache secondary base address. When instruction cache is not used, this memory can be used as general purpose memory.

5.4.5 DCache Memory

There is a 16 KB of data cache memory. This can be accessed in PS4/PS3 states.

5.4.6 Flash Memory

Flash memory is accessed through the QSPI controller. This can be accessed in PS4/PS3 states. All accesses to flash memory from I-Bus will be routed to icache memory. If there is a miss in icache memory, then icache controller will fetch it from external flash through QSPI controller in auto mode. The firmware has to configure the QSPI controller for auto mode before issuing any accesses from I-Bus. D-Bus accesses to flash address space will be routed to QSPI controller. Flash memory can also be accessed through S Bus by configuring the QSPI controller in manual mode.

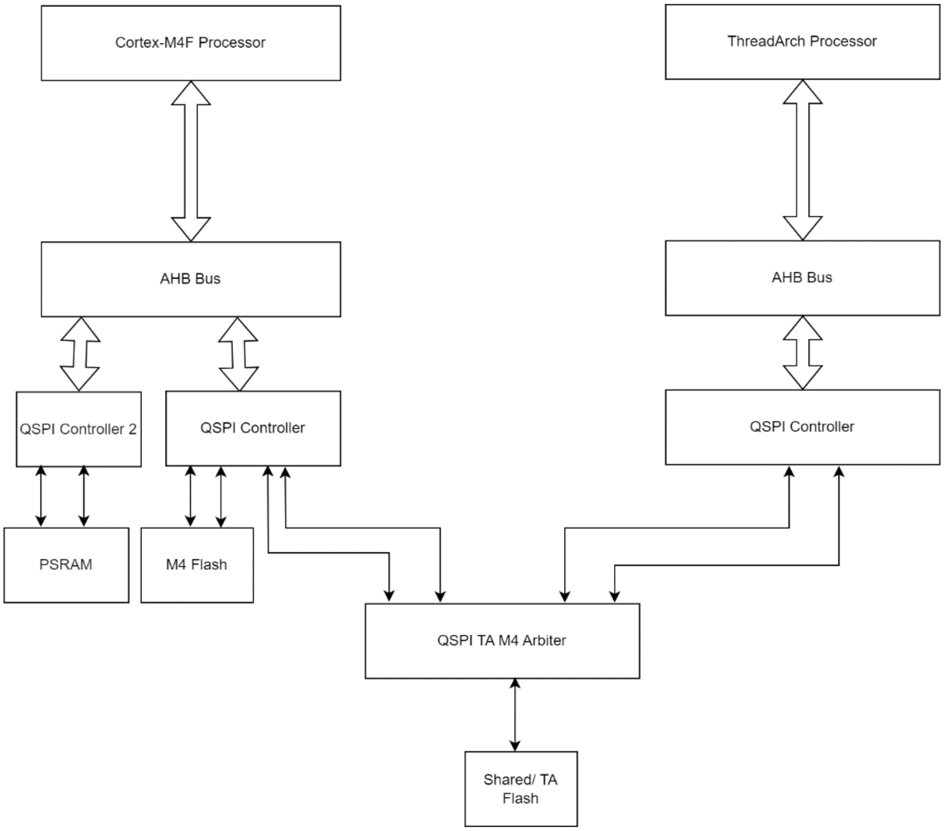


Figure 5.2. Flash Memory

Common Flash

Common Flash feature is a single flash memory that is shared by NWP and Cortex-M4 processors. As the memory is shared by both the processors, arbitration is present between the flash accesses. While both processors are executing from flash simultaneously, performance of each of the processors is impacted compared that of a dual flash where each processor has its own flash dedicated to it.

Common flash it connected NWP flash pin set. GPIO programming needs to done by assigning the flash pins to NWP. NWP QSPI power domain should be ON for M4 to access common flash.

6. Clock Architecture

6.1 General Description

The Clock Architecture describes how to configure the on-chip clocks (ULP Clock Oscillators, High-Frequency PLL) and the clocks to Processor, High Speed Interfaces, and Peripherals (includes MCU HP, MCU ULP and UULP Vbat). The Clock subsystem enables the software to vary the clock source/frequency for different functionalities to achieve lower power consumption based on the application.

6.2 Features

- Multiple high frequency clocks generated by PLLs
  - High Frequency Clock from 1MHz - 180MHz (SOC_PLL_CLK)
  - High Frequency Interface Clock from 1MHz - 180MHz (INTF_PLL_CLK)
  - Defined frequencies for I^{2}S Interface (I2S_PLL_CLK)
- Multiple clocks generated by ULP Clock Oscillators. These are low-power clock oscillators
  - External Crystal clock (XTAL_CLK)
  - RC 32MHz Clock (RC_32MHZ_CLK)
  - RO High-Frequency clock (RO_HF_CLK)
  - Doubler Clock (DOUBLER_CLK)
  - RC 32kHz Clock (RC_32KHZ_CLK)
  - RO 32kHz Clock (RO_32KHZ_CLK)
  - XTAL 32kHz clock (XTAL_32KHZ_CLK)
- Configurable independent division factors for varying the frequencies of different functional blocks
- Configurable independent clock gating for different functional blocks

6.3 Functional Description

6.3.1 Overview

The figure below depicts the top level overview of clocks being used for different domains.

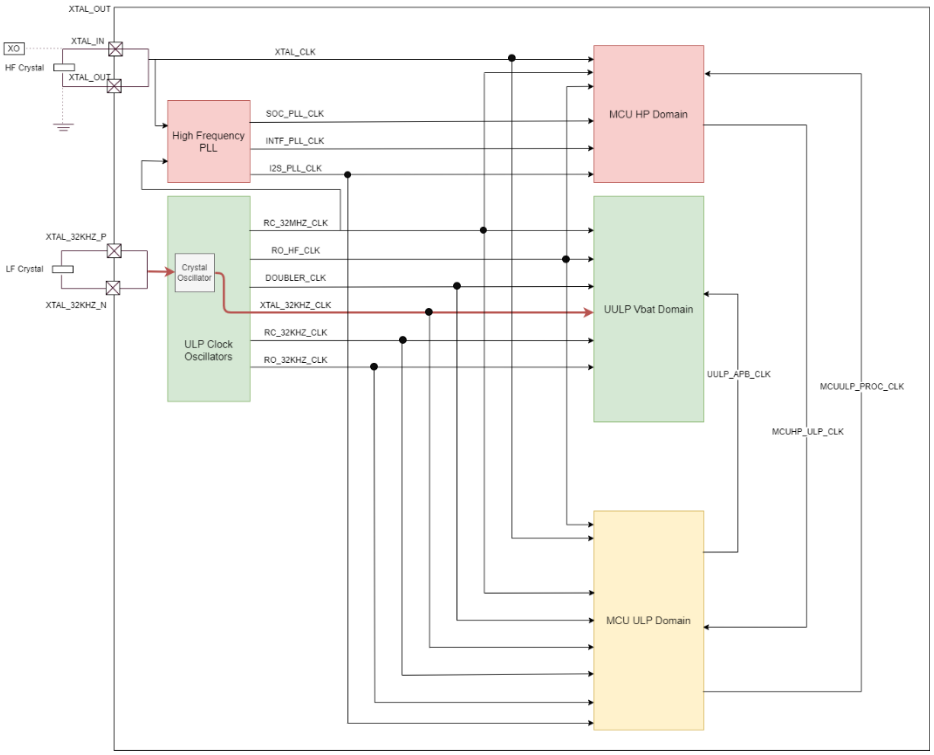


Figure 6.1. Clocking Diagram

- MCU HP Domain includes the Processor, FPU, Debugger, MCU High speed interfaces , MCU HP peripherals and MCU HP DMA, which are part of power domains active during the PS-4/3 power states.
- MCU ULP Domain includes MCU ULP AHB Inter-Connect-Matrix, MCU ULP Peripherals and the direct AHB Interface with the Processor, which are part of power domains active during PS-4/3/2/1 power states.
- UULP Vbat Domain MCU ULP VBATT Peripherals which are active during the PS-4/3/2/1/0 power states.

For detailed info regarding above domains refer to Section 9. Power Architecture.

6.4 HF Crystal Clock

Connect the external reference clock source on XTAL_IN and XTAL_OUT pins.

The list below provides the different features supported for the High-Frequency Crystal Clock. Please refer to the "Clock Configuration" section of the "siwx91x-software-reference-manual" for more details.

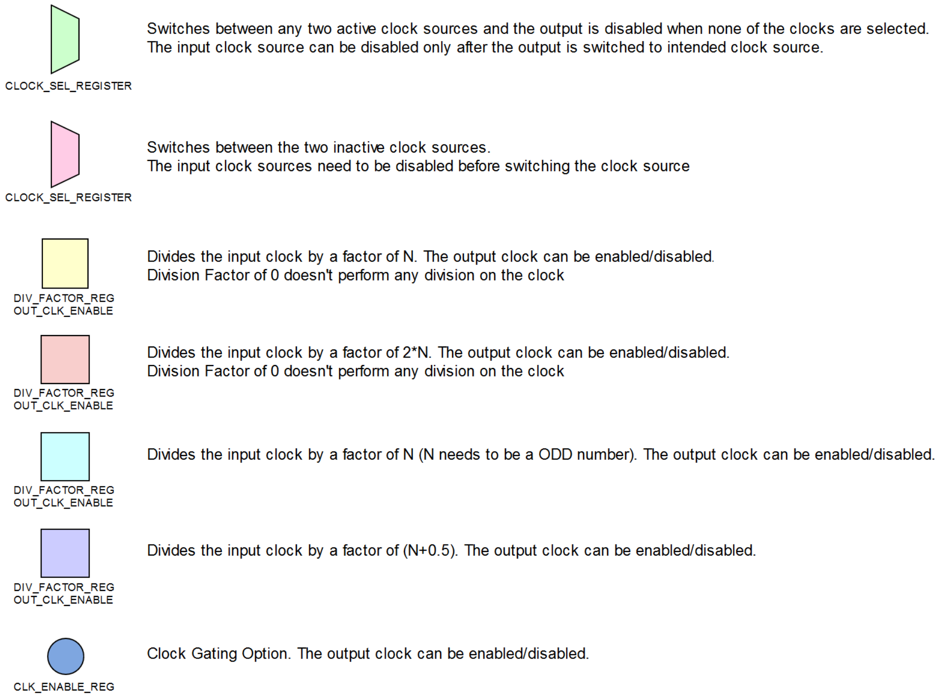
1. Ability to Operate on Battery Voltage or the DC-DC 1.35 Supply.
    a. Applications like calibration of Low-Frequency RC / XTAL Clocks during sleep requires the HF-Crystal to be operating on Battery Voltage.

6.5 Naming Convention

Table 6.1. Clocking Diagram Signal Descriptions

S.No	CLK_NAME	Source	Default Frequency	Frequency Range	Default State	Description
1	XTAL_CLK	External	-	Defined Frequencies	Enabled	This is the external reference clock for PLLs
2	SOC_PLL_CLK	SoC PLL	180 MHz	1 MHz - 180 MHz	Enabled	High Frequency Clock from PLL
3	INTF_PLL_CLK	Interface PLL/ SoC PLL	180 MHz	1 MHz - 180 MHz	Enabled	High Frequency Interface Clock from PLL
4	I2S_PLL_CLK	I2S PLL/ SoC PLL	6.144MHz	256kHz to 24.576MHz	Enabled	I^{2}S Interface Clock from PLL
5	RC_32MHZ_CLK	ULP CLOCK OSCILLATORS	32MHz	15MHz-65MHz	Enabled	Low power RC High Frequency clock source.
6	RO_HF_CLK	ULP CLOCK OSCILLATORS	20MHz	1MHz to 50MHz	Disabled	Low power RO High Frequency clock source.
7	DOUBLER_CLK	ULP CLOCK OSCILLATORS	Double the frequency of RO_HF_CLK/ RC_32MHZ_CLK		Disabled	Frequency Doubler Clock.
8	XTAL_32KHZ_CLK	ULP CLOCK OSCILLATORS	32KHz	32KHz-20ppm - 32KHz+20ppm	Enabled	Low power XTAL Low Frequency clock source.
9	RC_32KHZ_CLK	ULP CLOCK OSCILLATORS	32KHz	16kHz-128kHz	Enabled	Low power RC Low Frequency clock source.
10	RO_32KHZ_CLK	ULP CLOCK OSCILLATORS	32KHz	16kHz to 64kHz	Enabled	Low power RO Low Frequency clock source.

The following legend illustrates the components used in the clocking diagram's for different functional domains/peripherals listed in this section. The text appearing below each component refers to the register bits for configuring and/or controlling the component.



Switches between any two active clock sources and the output is disabled when none of the clocks are selected.
The input clock source can be disabled only after the output is switched to intended clock source.

Switches between the two inactive clock sources.
The input clock sources need to be disabled before switching the clock source

Divides the input clock by a factor of N. The output clock can be enabled/disabled.
Division Factor of 0 doesn't perform any division on the clock

Divides the input clock by a factor of 2*N. The output clock can be enabled/disabled.
Division Factor of 0 doesn't perform any division on the clock

Divides the input clock by a factor of N (N needs to be a ODD number). The output clock can be enabled/disabled.

Divides the input clock by a factor of (N+0.5). The output clock can be enabled/disabled.

Clock Gating Option. The output clock can be enabled/disabled.

Figure 6.2. Clocking Legend

6.6 Reference Clock

There are reference clock generated in MCU-HP and MCU ULP Domain which are reused in generation of clocks for peripherals/ modules in respective domains.

The clock source selection for these reference clocks will be retained during sleep mode and hence need not be re-configured on each wakeup.

6.6.1 MCU HP

The source for reference clock is configured through MCUHP_REF_CLK_SEL in MCU_REF_CLK_CONFIG Register.

The Clock switching status can be read through MCUHP_M4_SOC_CLK_SWITCHED in MCUHP_PLL_STATUS_REG Register as described in the Section 6.12 MCU HP Clock Architecture.

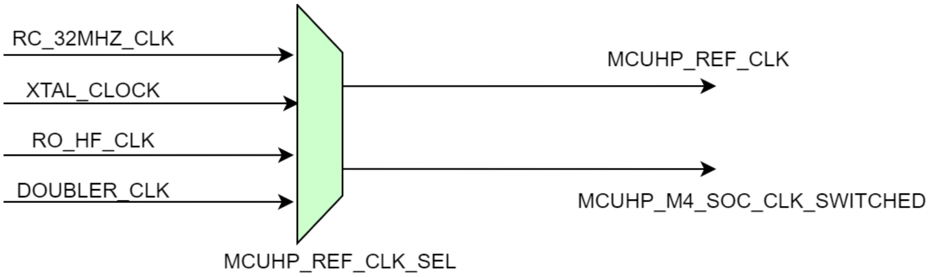


Figure 6.3. MCU-HP Reference Clock Generation

6.6.2 MCU ULP

The source for reference clock is configured through MCUULP_REF_CLK_SEL in MCU_REF_CLK_CONFIG Register.

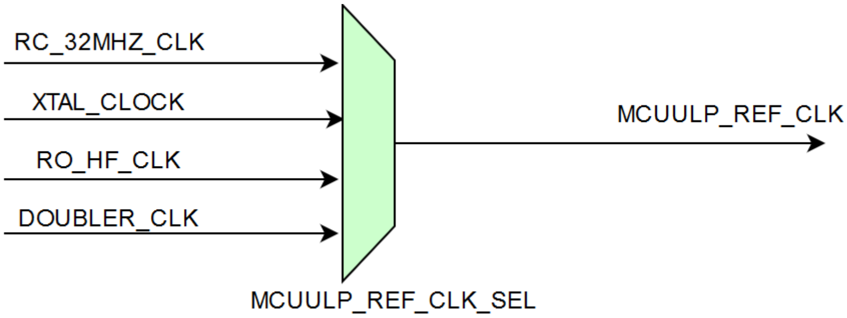


Figure 6.4. MCU-ULP Reference Clock Generation

6.7 Clocking Schemes

The list below shows the organization of different clock generation mechanisms:

1. The generation of clocks from ULP Clock Oscillators are described in the Section 6.15 ULP Clock Oscillators.
2. The generation of clocks from HIGH-Frequency PLL is described in the Section 6.11 High-Frequency PLL.
3. The generation of clocks for MCU HP Peripherals, MCU HP High speed Interface and the Cortex-M4F Processor are described in the Section 6.12 MCU HP Clock Architecture.
4. The generation of clocks for MCU ULP Peripherals and the MCU ULP AHB are described in the Section 6.13 MCU ULP Clock Architecture.
5. The generation of clocks for MCU UULP Vbat Peripherals are described in the Section 6.14 MCU ULP VBAT Clock Architecture.

6.8 Register Summary

Base Address: 0x2404_8100

Table 6.2. List of Registers

Register Name	Offset	Description
Section 6.9.1 MCU_REF_CLK_CONFIG	0x1C	Reference Clock Configuration Register

6.9 Register Description

6.9.1 MCU_REF_CLK_CONFIG

Table 6.3. MCU_REF_CLK_CONFIG Description

Bit	Access	Function	Default Value	Description
31:19	-	Reserved	-	It is recommended to retain the contents by using read/modify write to this register.
18:16		MCUULP_REF_CLK_SEL	1	Specifies the clock source to be used for MCU ULP Reference Clock. 0 - Output Clock is disabled 1 - RC_32MHZ_CLK 2 - Reserved 3 - XTAL_CLK 4 - Reserved 5 - RO_HF_CLK 6 - DOUBLER_CLK 7 - Reserved Note: Use RO & RC clock only if accuracy is not crucial. Otherwise use crystal clock
15:3	-		-	It is recommended to retain the contents by using read/modify write to this register.
2:0	R/W	MCUHP_REF_CLK_SEL	1	Specifies the clock source to be used for MCU HP Reference Clock. 0 - Output Clock is disabled 1 - RC_32MHZ_CLK 2 - Reserved 3 - XTAL_CLK 4 - Reserved 5 - RO_HF_CLK 6 - DOUBLER_CLK 7 - Reserved

6.10 Clock Distribution

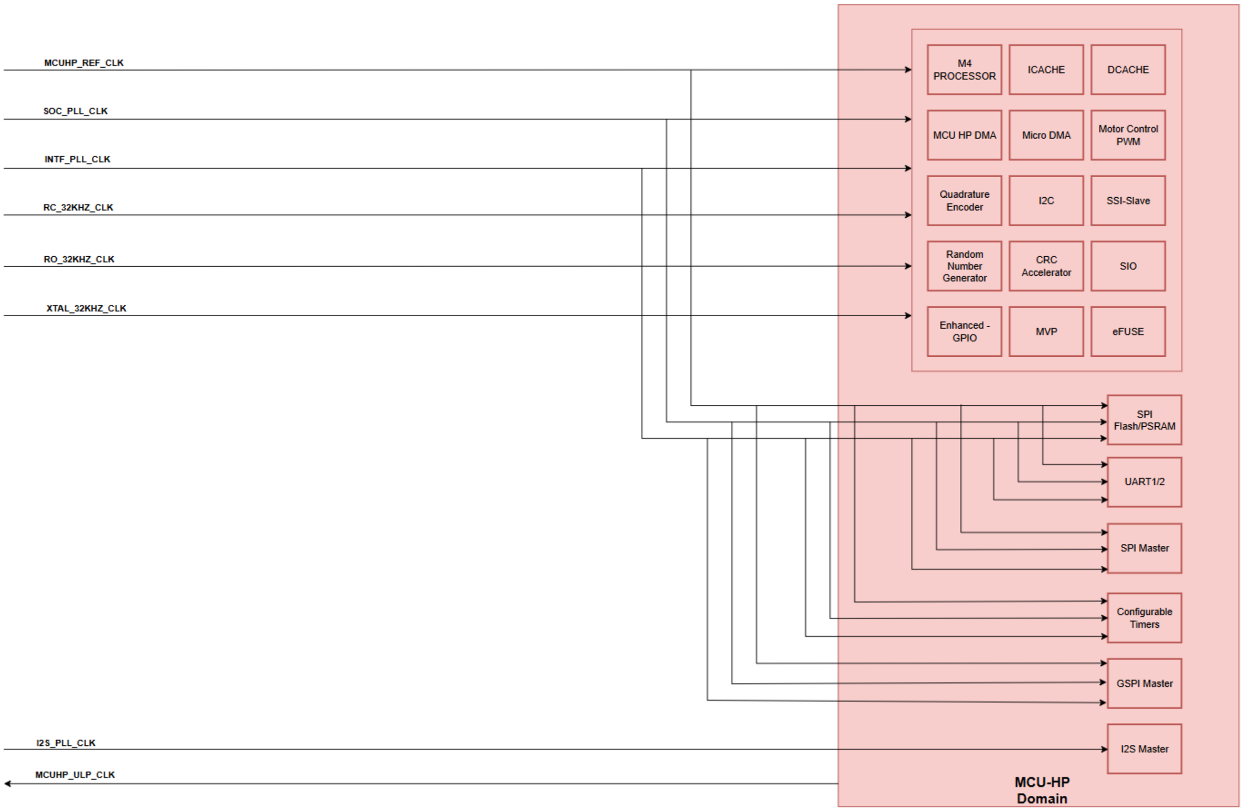


Figure 6.5. MCU HP Clocks

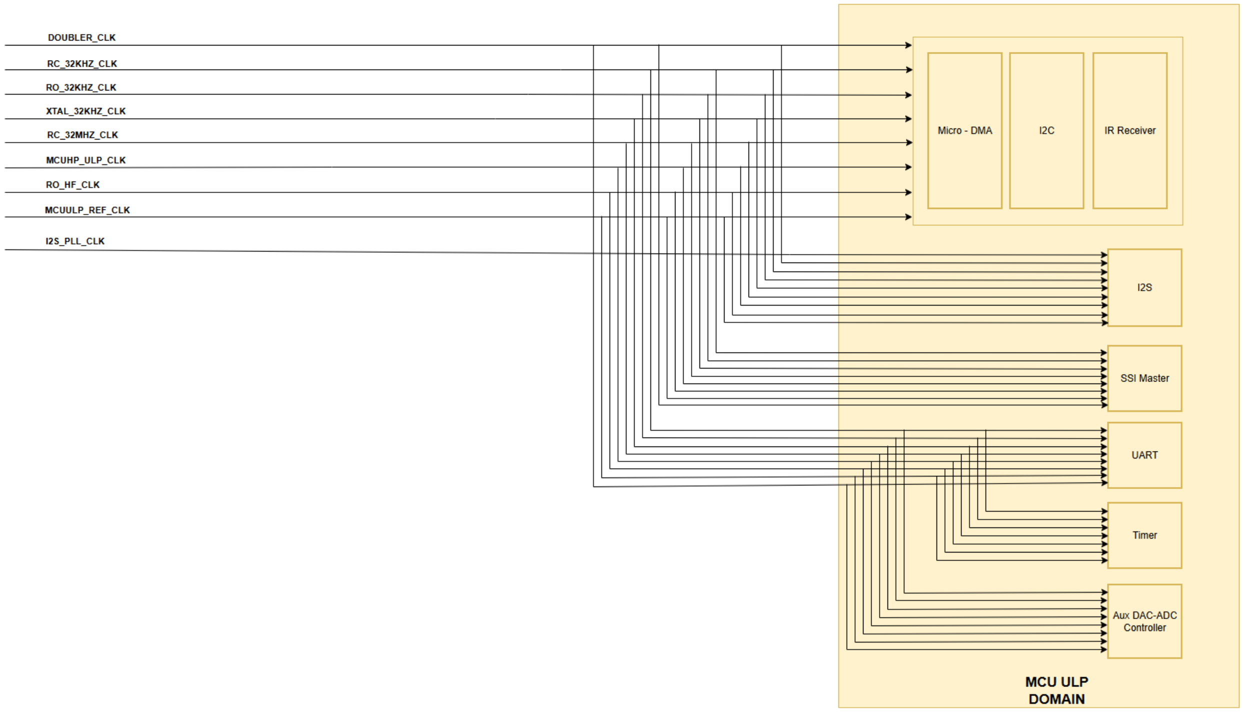


Figure 6.6. MCU ULP Clocks

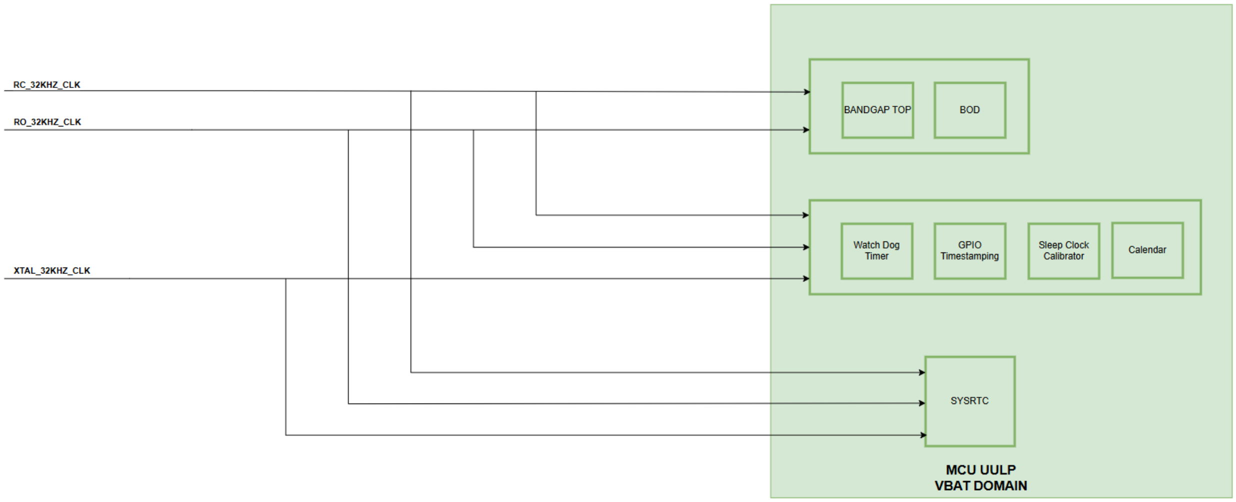


Figure 6.7. MCU UULP Clocks

Please refer to the following pages for more description regarding clock selection-:

Section 6.12 MCU HP Clock Architecture

Section 6.13 MCU ULP Clock Architecture

Section 6.14 MCU ULP VBAT Clock Architecture

6.11 High-Frequency PLL

6.11.1 General Description

The High-Frequency PLL is the source of the high frequency clocks used for Processor or the Peripherals.

6.11.2 Features

- High-Frequency Crystal clock (XTAL_CLOCK) or the Internal RC 32MHz clock (RC_32MHZ_CLK) can be used as a reference frequency for the PLLs.
- There are 3 independent PLLs present which generate the High Frequency clock sources.
  - SoC-PLL with frequency range of 1MHz - 180MHz
  - Interface-PLL with frequency range of 1MHz - 180MHz depending on the part number.
  - I2S-PLL with defined frequencies from 256KHz - 24.576MHz.
- PLL Lock time of 100µs
- Each of the PLL can be powered down independently for efficient power management.
- The Output clock from each PLL can be disabled independently.

6.11.3 Functional Description

6.11.3.1 Overview

The LDO's internal to the PLL acts as an voltage source for the PLL. The DCDC 1.35 and LDO SoC 1.1 (as described in Section 10. Power Management Unit) are used as the input source for these LDO's.

The Input voltage sources (DC-DC 1.35 and LDO SoC 1.1) as shown in the figure below needs to be enabled for the PLL to be active.

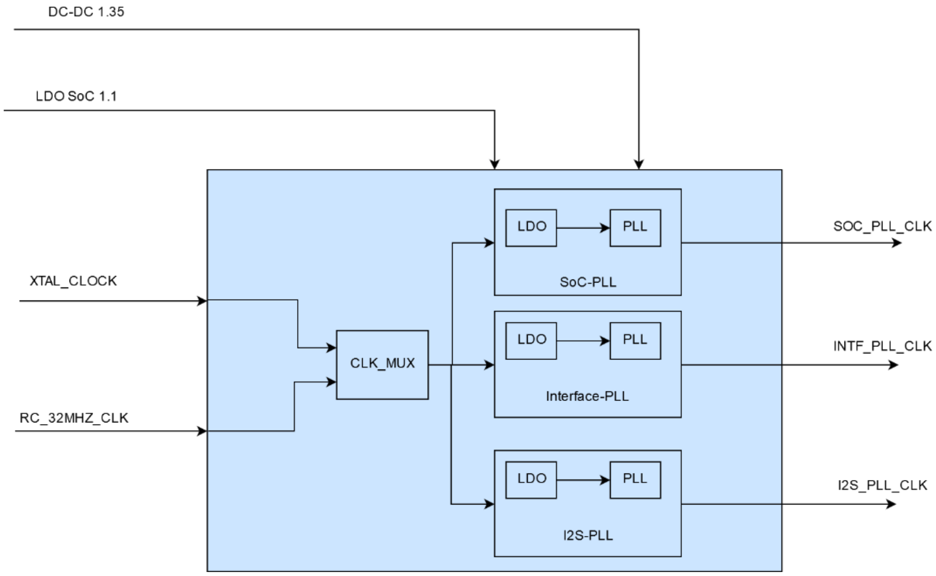


Figure 6.8. PLL Clocks

All PLL modules require certain locking period, during which the output may not be stable. As a result, before selecting a specific PLL output, the user program must check that the relevant lock bit is set before switching the clock.

The list below provides the names of the lock bits for SoC-PLL, Interface-PLL and I2S-PLL blocks.

- SOCPLL_LOCK in SOCPLL_STATUS_REG indicates the status for SoC-PLL
- INTFPLL_LOCK in INTFPLL_STATUS_REG indicates the status for Interface-PLL
- I2SPLL_LOCK in I2SPLL_STATUS_REG indicates the status for I2S-PLL

The following sections describe available configuration parameters for each PLL.

6.11.4 Input Reference Clock

There are two sources for the input reference clock to the PLLs. Either the High-Frequency Crystal Clock or the Internal RC-32MHz clock can be used as the reference clock.

The reference clock can be selected through REF_CLK_SEL in PLL_REF_CLK_CONFIG_REG Register.

6.11.5 SoC-PLL

6.11.5.1 Programming Sequence

The list below describes the programming sequence to be followed for achieving a particular output frequency.

1. Power-up the PLL through SOCPLL_PD in SOCPLL_CONFIG_REG1 Register.
2. Configure LDO Output Voltage through SOCPLL_LDO_PROG in PLL_LDO_CONFIG_REG register
   a. If the Output intended frequency is greater than 200 MHz it needs to be configured to 1.1 V, else it needs to be configured to 1.05 V.
3. Configure Frequency Range Selection through SOCPLL_RANGE_SEL in SOCPLL_CONFIG_REG1 Register.
4. Configure PLL Input Division Factor (NFAC) through SOCPLL_N in SOCPLL_CONFIG_REG2 Register.
   - NFAC should be chosen such that FIN (VCO Input Frequency) is in the range of 0.9 to 1.1 MHz
5. Configure PLL Output Division Factor (PFAC) through SOCPLL_P which is present in SOCPLL_CONFIG_REG2 Register.
   a. PFAC should be chosen from the following values - {1,3,7,15,31,127}.
   b. PFAC should be chosen such that VCO Output Frequency (VCOFREQ) is in the range of 127 to 180 MHz
6. Configure PLL MFAC (Multiplication Factor) through SOCPLL_M in SOCPLL_CONFIG_REG1 Register.
   a. MFAC is derived from VCOFREQ, FIN parmaeters obtained in the above steps.
7. Configure PLL Fractional Frequency Control Word (FCW) through SOCPLL_FCW which is present in SOCPLL_CONFIG_REG3 Register.
   - FCW is derived from MFAC, VCOFREQ, FIN parameters obtained in the above steps.
8. Enable the output clock through SOCPLL_CLK_EN in SOCPLL_CONFIG_REG1 Register.
9. Wait till the PLL output clock is stable by checking the SOCPLL_LOCK in SOCPLL_STATUS_REG Register.

6.11.6 Interface-PLL

6.11.6.1 Programming Sequence

The list below describes the programming sequence to be followed for achieving a particular output frequency.

1. Power-up the PLL through INTFPLL_PD in INTFPLL_CONFIG_REG1 Register.
2. Configure LDO Output Voltage through INTFPLL_LDO_PROG in PLL_LDO_CONFIG_REG register
   a. If the Output intended frequency is greater than 200 MHz it needs to be configured to 1.1 V, else it needs to be configured to 1.05 V.
3. Configure Frequency Range Selection through INTFPLL_RANGE_SEL in INTFPLL_CONFIG_REG1 Register.
4. Configure PLL Input Division Factor (NFAC) through INTFPLL_N in INTFPLL_CONFIG_REG2 Register.

   NFAC should be chosen such that FIN (VCO Input Frequency) is in the range of 0.9 to 1.1 MHz.
5. Configure PLL Output Division Factor (PFAC) through INTFPLL_P which is present in INTFPLL_CONFIG_REG2 Register.
   a. PFAC should be chosen from the following values - {1,3,7,15,31,127}.
   b. PFAC should be chosen such that VCO Output Frequency (VCOFREQ) is in the range of 127 to 180 MHz.
6. Configure PLL Multiplication Factor (MFAC) through INTFPLL_M in INTFPLL_CONFIG_REG1 Register.
   a. MFAC is derived from VCOFREQ, FIN parmaeters obtained in the above steps.
7. Configure PLL Fractional Frequency Control Word (FCW) through INTFPLL_FCW which is present in INTFPLL_CONFIG_REG3 Register.
   a. FCW is derived from MFAC, VCOFREQ, FIN parameters obtained in the above steps.
8. Enable the output clock through INTFPLL_CLK_EN in INTFPLL_CONFIG_REG1 Register.
9. Wait till the PLL output clock is stable by checking the INTFPLL_LOCK in INTFPLL_STATUS_REG Register.

6.11.7 I2S-PLL

The following Output frequencies can be generated using I2S-PLL

- 256 kHz
- 512 kHz
- 768 kHz
- 1.024 MHz
- 1.4112 MHz
- 2.048 MHz
- 2.8224 MHz
- 3.072 MHz
- 4.096 MHz
- 4.2336 MHz
- 4.608 MHz
- 5.6448 MHz
- 6.144 MHz
- 8.4672 MHz
- 9.216 MHz
- 11.2896 MHz
- 12.288 MHz
- 18.432 MHz
- 24.576 MHz

6.11.7.1 Programming Sequence

The list below describes the programming sequence to be followed for achieving a particular output frequency.

1. Power-up the PLL through I2SPLL_PD in I2SPLL_CONFIG_REG1 Register.
2. LDO Output Voltage has to be configured to 1.05V through I2SPLL_LDO_PROG in PLL_LDO_CONFIG_REG Register.
3. Configure PLL Input Division Factor (NFAC) through I2SPLL_N in I2SPLL_CONFIG_REG2 Register.
    a. NFAC is derived as per the table below. FREF is the Input reference clock frequency to the PLL.

Table 6.4. N_FAC Derivation for I2S-PLL

S.No	Fref (MHz)	NFAC	FIN (MHz)
1	9.6	10	0.96
2	13	13	1
3	16	16	1
4	19.2	20	0.96
5	26	26	1
6	32	32	1
7	38.4	40	0.96
8	40	40	1
9	52	52	1

1. Configure PLL Multiplication Factor (MFAC), PLL Output Division Factors (PFAC1, PFAC2) and PLL Fractional Control word (FCW). These parameters are derived as per the tables below based on the FIN derived in the above steps.
   a. MFAC can be configured through I2SPLL_M in I2SPLL_CONFIG_REG1 Register.
   b. PFAC1, PFAC2 can be configured through I2SPLL_P1, I2SPLL_P2 in I2SPLL_CONFIG_REG2 Register.
   c. FCW can be configured through I2SPLL_FCW which is present in I2SPLL_CONFIG_REG3 Register.

Table 6.5. MFAC, FCW, PFAC1, PFAC2 Derivation for I2S-PLL with FIN = 1MHz

FIN (Mhz)	MFAC	FCW	PFAC1	PFAC2	Output Freq (MHz)
1	73	11928	2	0	24.576
			3	0	18.432
			5	0	12.288
			7	0	9.216
			11	0	6.144
			15	0	4.608
			17	0	4.096
			23	0	3.072
			17	1	2.048
			23	1	1.536
			17	2	1.024
			11	3	0.768
			17	3	0.512
			17	4	0.256
	67	12085	5	0	11.2896
			7	0	8.4672
			11	0	5.6448
			15	0	4.2336
			23	0	2.8224
			23	1	1.4112

Table 6.6. MFAC, FCW, PFAC1, PFAC2 Derivation for I2S-PLL with FIN = 0.96MHz

Fin(Mhz)	M_Fac	FCW	P_Fac1	P_Fac2	Fout(Mhz)
0.96	76	13107	2	0	24.576
			3	0	18.432
			5	0	12.288
			7	0	9.216
			11	0	6.144
			15	0	4.608
			17	0	4.096
			17	1	2.048

2. Enable the output clock through I2SPLL_CLK_EN in I2SPLL_CONFIG_REG1 Register.
3. Wait till the PLL output clock is stable by checking the I2SPLL_LOCK in I2SPLL_STATUS_REG Register.

6.11.8 PLL Programming Baud Rate

The PLL programming buad rate has to be configured as described below before accessing the PLL Configuration Registers. This is derived from the Processor clock as described in PLL_PROG_CTRL_REG register.

1. The maximum programming Baud rate should be 50 MHz.

6.11.9 Register Summary

Base_address = 0x4618_0000

Register Name	Offset	Description
Section 6.11.10.1 PLL_REF_CLK_CONFIG_REG	0x04	Reference Clock Configuration Register
Section 6.11.10.2 PLL_LDO_CONFIG_REG	0x08	LDO Configuration Register
Section 6.11.10.3 SOCPLL_CONFIG_REG1	0x40	SoC-PLL Configuration Register1
Section 6.11.10.4 SOCPLL_CONFIG_REG2	0x44	SoC-PLL Configuration Register2
Section 6.11.10.5 SOCPLL_CONFIG_REG3	0x48	SoC-PLL Configuration Register3
Section 6.11.10.6 SOCPLL_STATUS_REG	0x70	SoC-PLL Status Register
Section 6.11.10.7 INTFPLL_CONFIG_REG1	0x80	Interface-PLL Configuration Register1
Section 6.11.10.8 INTFPLL_CONFIG_REG2	0x84	Interface-PLL Configuration Register2
Section 6.11.10.9 INTFPLL_CONFIG_REG3	0x88	Interface-PLL Configuration Register3
Section 6.11.10.10 INTFPLL_STATUS_REG	0xB0	Interface-PLL Status Register
Section 6.11.10.11 I2SPLL_CONFIG_REG1	0xC0	I2S-PLL Configuration Register1
Section 6.11.10.12 I2SPLL_CONFIG_REG2	0xC4	I2S-PLL Configuration Register2
Section 6.11.10.13 I2SPLL_CONFIG_REG3	0xC8	I2S-PLL Configuration Register3
Section 6.11.10.14 I2SPLL_STATUS_REG	0xF0	I2S-PLL Status Register

Base Address: 0x4008_0000

Table 6.7. Register Summary

Register Name	Offset	Description
Section 6.11.10.15 PLL_PROG_CTRL_REG	0x0	PLL Programming Control Register

6.11.10 Register Description

6.11.10.1 PLL_REF_CLK_CONFIG_REG

Table 6.8. Reference Clock Configuration Description

Bit	Access	Function	Reset Value	Description
31:16	-	Reserved	-	It is recommended to write these bits to 0.
15:14	R/W	REF_CLK_SEL	0	Specified the input reference clock for the PLL's 0 - XTAL_CLK 1 - Reserved 2 - RC_32MHZ_CLK 3 - Reserved
13:0	-	Reserved	-	It is recommended to write these bits to 0.

6.11.10.2 PLL_LDO_CONFIG_REG

Table 6.9. Common Control Register Description

Bit	Access	Function	Reset Value	Description
31:16	-	Reserved	-	It is recommended to write these bits to 0.
15:13	R/W	SOCPLL_LDO_PROG	4	Specified the configuration of SoC-PLL LDO output voltage 0-3 - Reserved 4 - 1.05V 5 - 1.1V 6,7 - Reserved
12:10	R/W	INTFPLL_LDO_PROG	4	Specified the configuration of Interface-PLL LDO output voltage 0-3 - Reserved 4 - 1.05V 5 - 1.1V 6,7 - Reserved
9:7	R/W	I2SPLL_LDO_PROG	4	Specified the configuration of I2S-PLL LDO output voltage 0-3 - Reserved 4 - 1.05V 5 - 1.1V 6,7 - Reserved

Bit	Access	Function	Reset Value	Description
6:0	-	Reserved	-	It is recommended to write these bits to 0.

6.11.10.3 SOCPLL_CONFIG_REG1

Table 6.10. SOCPLL_CONFIG_REG1 Description

Bit	Access	Function	Reset Value	Description
31:16	-	Reserved		It is recommended to write these bits to 0.
15:6	R/W	SOCPLL_M	179	Specifies the SoC-PLL Multiplication Factor.
5	-	Reserved	-	It is recommended to write these bits to 0.
4	R/W	SOCPLL_PD	0	Writing 1 to this disables power to the SoC-PLL. Writing 0 to this enables power to the SoC-PLL.
3	R/W	SOCPLL_CLK_EN	1	Writing 1 to this enables SoC-PLL Output clock. Writing 0 to this disables SoC-PLL Output clock.
2	-	Reserved	-	It is recommended to write these bits to 0.
1:0	R/W	SOCPLL_RANGE_SEL	1	Specifies the range for the Output frequency. 0 - Greater than 200MHz 1 - Less than 200MHz 2,3 - Reserved

6.11.10.4 SOCPLL_CONFIG_REG2

Table 6.11. SOCPLL_CONFIG_REG2 Description

Bit	Access	Function	Reset Value	Description
31:16	-	Reserved	-	It is recommended to write these bits to 0.
15:9	R/W	SOCPLL_P	0	Specifies the SoC-PLL Output Division Factor
8:3	R/W	SOCPLL_N	39	Specifies the SoC-PLL Input Division Factor
2:0	-	Reserved	-	It is recommended to write these bits to 0.

6.11.10.5 SOCPLL_CONFIG_REG3

Table 6.12. SOCPLL_CONFIG_REG3 Description

Bit	Access	Function	Reset Value	Description
31:16	-	Reserved	-	It is recommended to write these bits to 0.
15:2	R/W	SOCPLL_FCW	0	Specifies the SoC-PLL Fractional Frequency Control Word
1:0	-	Reserved	-	It is recommended to write these bits to 0.

6.11.10.6 SOCPLL_STATUS_REG

Table 6.13. SOCPLL_STATUS_REG Description

Bit	Access	Function	Reset Value	Description
31:16	-	Reserved	-	
15	R	SOCPLL_LOCK	0	Indicates the SoC-PLL Status 0 - Not Locked 1 - Locked
14:0	R/W	Reserved	-	

6.11.10.7 INTFPLL_CONFIG_REG1

Table 6.14. INTFPLL_CONFIG_REG1 Description

Bit	Access	Function	Reset Value	Description
31:16	-	Reserved		It is recommended to write these bits to 0.
15:6	R/W	INTFPLL_M	179	Specifies the Interface-PLL Multiplication Factor.
5	-	Reserved	-	It is recommended to write these bits to 0.
4	R/W	INTFPLL_PD	0	Writing 1 to this disables power to the Interface-PLL. Writing 0 to this enables power to the Interface-PLL.
3	R/W	INTFPLL_CLK_EN	1	Writing 1 to this enables Interface-PLL Output clock. Writing 0 to this disables Interface-PLL Output clock.
2	-	Reserved	-	It is recommended to write these bits to 0.
1:0	R/W	INTFPLL_RANGE_SEL	1	Specifies the range for the Output frequency. 0 - Greater than 200MHz 1 - Less than 200MHz 2,3 - Reserved

6.11.10.8 INTFPLL_CONFIG_REG2

Table 6.15. INTFPLL_CONFIG_REG2 Description

Bit	Access	Function	Reset Value	Description
31:16	-	Reserved	-	It is recommended to write these bits to 0.
15:9	R/W	INTFPLL_P	0	Specifies the Interface-PLL Output Division Factor
8:3	R/W	INTFPLL_N	39	Specifies the Interface-PLL Input Division Factor
2:0	-	Reserved	-	It is recommended to write these bits to 0.

6.11.10.9 INTFPLL_CONFIG_REG3

Table 6.16. INTFPLL_CONFIG_REG3 Description

Bit	Access	Function	Reset Value	Description
31:16	-	Reserved	-	It is recommended to write these bits to 0.
15:2	R/W	INTFPLL_FCW	0	Specifies the Interface-PLL Fractional Frequency Control Word
1:0	-	Reserved	-	It is recommended to write these bits to 0.

6.11.10.10 INTFPLL_STATUS_REG

Table 6.17. INTFPLL_STATUS_REG Description

Bit	Access	Function	Reset Value	Description
31:16	-	Reserved	-	
15	R	INTFPLL_LOCK	0	Indicates the Interface-PLL Status 0 - Not Locked 1 - Locked
14:0	R/W	Reserved	-	

6.11.10.11 I2SPLL_CONFIG_REG1

Table 6.18. I2SPLL_CONFIG_REG1 Description

Bit	Access	Function	Reset Value	Description
31:16	-	Reserved	-	It is recommended to write these bits to 0.
15:6	R/W	I2SPLL_M	73	Specifies the I2S-PLL Multiplication Factor
5	-	Reserved	-	It is recommended to write these bits to 0.
4	R/W	I2SPLL_PD	0	Writing 1 to this disables power to the I2S-PLL. Writing 0 to this enables power to the I2S-PLL.
3	-	Reserved	-	It is recommended to write these bits to 0.
2	R/W	I2SPLL_CLK_EN	1	Writing 1 to this enables I2S-PLL Output clock. Writing 0 to this disables I2S-PLL Output clock.
1:0	-	Reserved	-	It is recommended to write these bits to 0.

6.11.10.12 I2SPLL_CONFIG_REG2

Table 6.19. I2SPLL_CONFIG_REG2 Description

Bit	Access	Function	Reset Value	Description
31:16	-	Reserved	-	It is recommended to write these bits to 0.
15:11	R/W	I2SPLL_P1	11	Specifies the I2S-PLL Post Division factor1
10:8	R/W	I2SPLL_P2	0	Specifies the I2S-PLL Post Division factor2
7:1	R/W	I2SPLL_N	40	Specifies the I2S-PLL Input Division factor
0	-	Reserved	-	It is recommended to write these bits to 0.

6.11.10.13 I2SPLL_CONFIG_REG3

Table 6.20. I2SPLL_CONFIG_REG3 Description

Bit	Access	Function	Reset Value	Description
31:16	-	Reserved	-	It is recommended to write these bits to 0.
15:2	R/W	I2SPLL_FCW	14'd11928	Specifies the I2S-PLL Fractional frequency control word
1:0	-	Reserved	0	It is recommended to write these bits to 0.

6.11.10.14 I2SPLL_STATUS_REG

Table 6.21. I2SPLL_STATUS_REG Description

Bit	Access	Function	Reset Value	Description
31:16	-	Reserved	-	
15	R	I2SPLL_LOCK	0	Indicates the I2S-PLL Status 0 - Not Locked 1 - Locked
14:0	-	Reserved	-	

6.11.10.15 PLL_PROG_CTRL_REG

Table 6.22. PLL_PROG_CTRL_REG Description

Bit	Access	Function	Reset Value	Description
31:4	-	Reserved	-	It is recommended to write these bits to 0.
3:0	R/W	BAUD_RATE	4	Specifies the Programming Baud Rate w.r.t source clock Programming_Baud_Rate = Processor_Clock/((BAUD_RATE+1)*2)

6.12 MCU HP Clock Architecture

6.12.1 General Description

The MCU HP (High Performance) domain contains the Cortex-M4F Processor, FPU, Debugger, MCU High Speed Interfaces, MCU HP Peripherals, MCU HP DMA and MCU/NWP shareable Interfaces. This section describes the different clock sources possible for each interface/peripheral and the Processor.

6.12.2 Features

- The clock sources used for MCU HP domain includes RC, RO, XTAL clocks in addition to the PLL generated high frequency clocks.
- A dedicated PLL is present for High Speed Interfaces like UART, etc.
- A dedicated PLL is present for I^{2}S interface with pre-defined frequencies.
- The frequency and clock source for High Speed Interfaces and few Master Peripherals can be configured independently of the Processor clock.
- A clock synchronous to the processor clock is generated which can be used for MCU ULP AHB and Peripherals.
- The Processor, FPU, SRAM and MCU HP AHB Bridge operates on the same clock.

6.12.3 Functional Description

The sections below describes the clock architecture and the corresponding programming details. The following figure depicts the clock sources present in MCU HP domain.

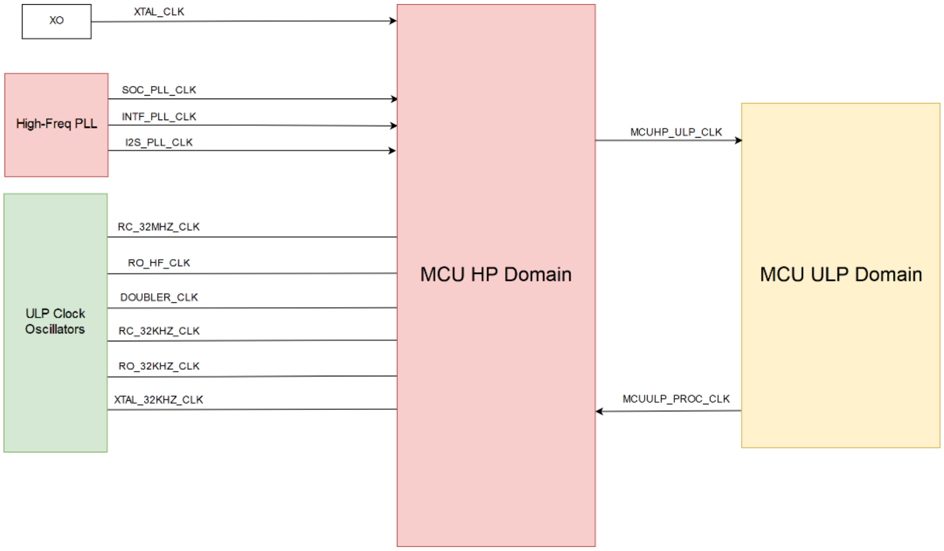


Figure 6.9. MCU HP Clocking Scheme Overview

The clocks to following blocks can be configured independently.

1. Processor
2. SPI PSRAM Controller
3. SPI Flash Controller
4. Low-Power Clock
5. UART0
6. UART1
7. MVP
8. SPI/SSI Master
9. I^{2}S in Master Mode
10. Configurable Timers
11. Generic-SPI Master
12. Clock for MCU ULP Domain
13. External Clock

The following blocks use the processor clock.

1. ICACHE
2. DCACHE
3. MCU HP DMA
4. UDMA
5. Motor-Control PWM
6. Quadrature Encoder
7. I^{2}C
8. SSI-Slave
9. Random-Number-Generator
10. CRC Accelerator
11. Enhanced-GPIO
12. eFUSE

The following sections describe the clock architecture for each of the functionality mentioned above. The reference clock generated for MCU-HP domain (MCUHP_REF_CLK) will be used in generation of the clocks for different peripherals/modules.

6.12.4 Low-Power Clock

The source for low power clock is configured through SLEEP_CLK_SEL in MCUHP_CLK_CONFIG_REG4 Register. The Clock switching status can be read from MCUHP_SLEEP_CLK_SWITCHED in MCUHP_PLL_STATUS_REG Register.

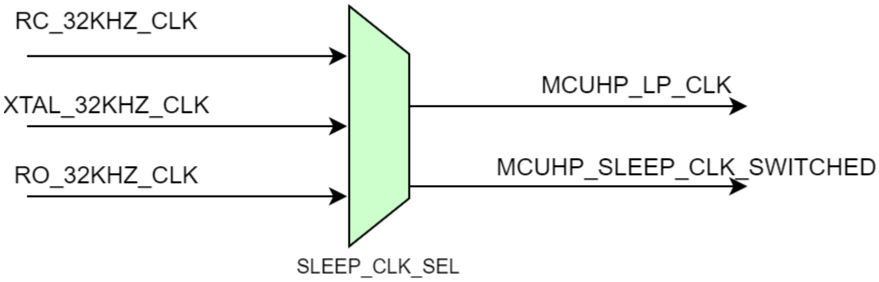


Figure 6.10. MCU-HP Low Power Clock Generation

6.12.5 Processor

The clock source and frequency for the Processor clock can be configured through MCUHP_M4_SOC_CLK_SEL and MCUHP_M4_SOC_CLK_DIV_FAC in MCUHP_CLK_CONFIG_REG5 Register. The Clock switching status can be read through MCUHP_M4_SOC_CLK_SWITCHED in MCUHP_PLL_STATUS_REG Register.

The 2nd stage of Clock mux is used for switching from ultra-low power state (PS2) and is configured through ULP_MODE_FUNC_SWITCH which is described in Section 9. Power Architecture. The MCUHP_M4_SOC_CLK_SWITCHED status is valid only in PS4 state.

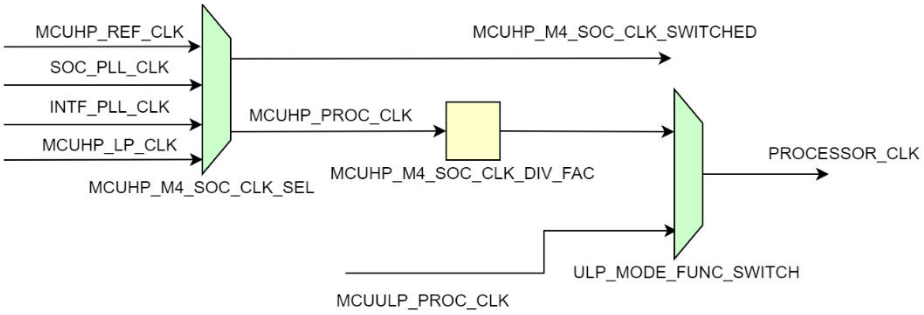


Figure 6.11. MCU-HP Processor Clock Generation

6.12.6 SPI Flash Controller

There are multiple modes of generating the clock for SPI Flash controller. It can be synchronous or independent of the Processor. The Clock switching status can be read through MCUHP_QSPI_CLK_SWITCHED in MCUHP_PLL_STATUS_REG Register. The status is valid only when Independent clock source is selected.

- Synchronous with Processor Clock. An undivided version of the Processor clock as shown in the Processor clock generation will be used in this mode.
  - Configure MCUHP_QSPI_CLK_SYNC_EN in MCUHP_CLKEN_SET_REG3/MCUHP_CLKEN_CLEAR_REG3 Register.
- Independent of Processor clock.
  - Clock source can be configured through MCUHP_QSPI_CLK_SEL in MCUHP_CLK_CONFIG_REG1 Register.
  - Division factor can be configured through MCUHP_QSPI_CLK_DIV_FAC in MCUHP_CLK_CONFIG_REG1 Register.
  - Output Clock with ODD/EVEN division factor can be selected through MCUHP_QSPI_CLK_ODD_SEL in MCUHP_CLK_CONFIG_REG2 Register.
- The clock to the controller can be disabled when not in use for efficient power consumption or before configuring the clock source and frequency. The controller clock and AHB clock can be controlled independently.
  - Configure MCUHP_QSPI_CLK_EN in MCUHP_CLKEN_SET_REG3/MCUHP_CLKEN_CLEAR_REG3 Register to enable/disable the controller clock.
  - Configure MCUHP_QSPI_AHB_CLK_EN in MCUHP_CLKEN_SET_REG2/MCUHP_CLKEN_CLEAR_REG2 Register to enable/disable the AHB Interface clock.
  - Configure MCUHP_QSPI_CLK_DIV_EN in MCUHP_CLKEN_SET_REG2/MCUHP_CLKEN_CLEAR_REG2 Register to enable/disable the QSPI Divider clocks.

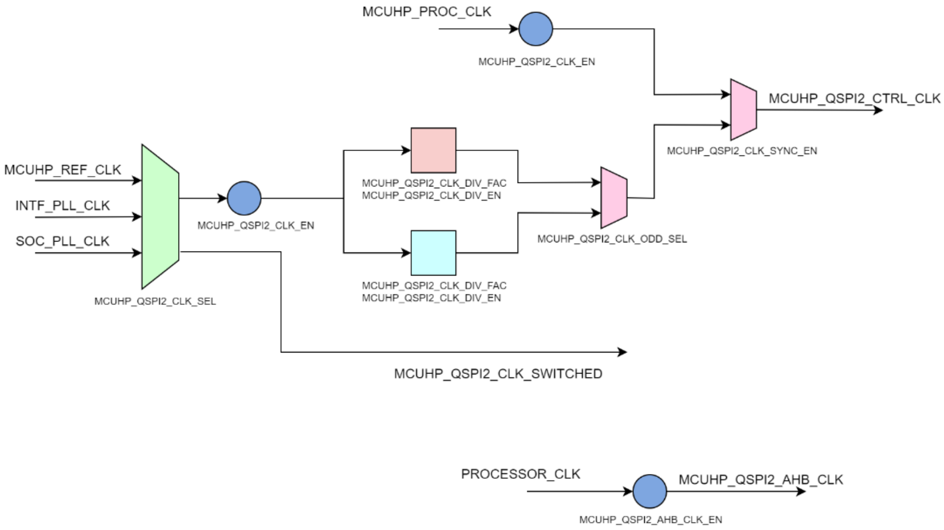


Figure 6.12. MCU-HP Flash Controller Clock Generation

6.12.7 SPI PSRAM Controller

There are multiple modes of generating the clock for SPI PSRAM controller. It can be synchronous or independent of the Processor. The Clock switching status can be read through MCUHP_QSPI2_CLK_SWITCHED in MCUHP_PLL_STATUS_REG Register. The status is valid only when Independent clock source is selected.

- Synchronous with Processor Clock. An undivided version of the Processor clock as shown in the Processor clock generation will be used in this mode.
  - Configure MCUHP_QSPI2_CLK_SYNC_EN in MCUHP_CLKEN_SET_REG1/MCUHP_CLKEN_CLEAR_REG1 Register.
- Independent of Processor clock.
  - Clock source can be configured through MCUHP_QSPI2_CLK_SEL in MCUHP_CLK_CONFIG_REG6 Register.
  - Division factor can be configured through MCUHP_QSPI2_CLK_DIV_FAC in MCUHP_CLK_CONFIG_REG6 Register.
  - Output Clock with ODD/EVEN division factor can be selected through MCUHP_QSPI2_CLK_ODD_SEL in MCUHP_CLK_CONFIG_REG6 Register.
- The clock to the controller can be disabled when not in use for efficient power consumption or before configuring the clock source and frequency. The controller clock and AHB clock can be controlled independently.
  - Configure MCUHP_QSPI2_CLK_EN in MCUHP_CLKEN_SET_REG1/MCUHP_CLKEN_CLEAR_REG1 Register to enable/disable the controller clock.
  - Configure MCUHP_QSPI2_AHB_CLK_EN in MCUHP_CLKEN_SET_REG1/MCUHP_CLKEN_CLEAR_REG1 Register to enable/disable the AHB Interface clock.
  - Configure MCUHP_QSPI2_CLK_DIV_EN in MCUHP_CLKEN_SET_REG1/MCUHP_CLKEN_CLEAR_REG1 Register to enable/disable the QSPI Divider clocks.

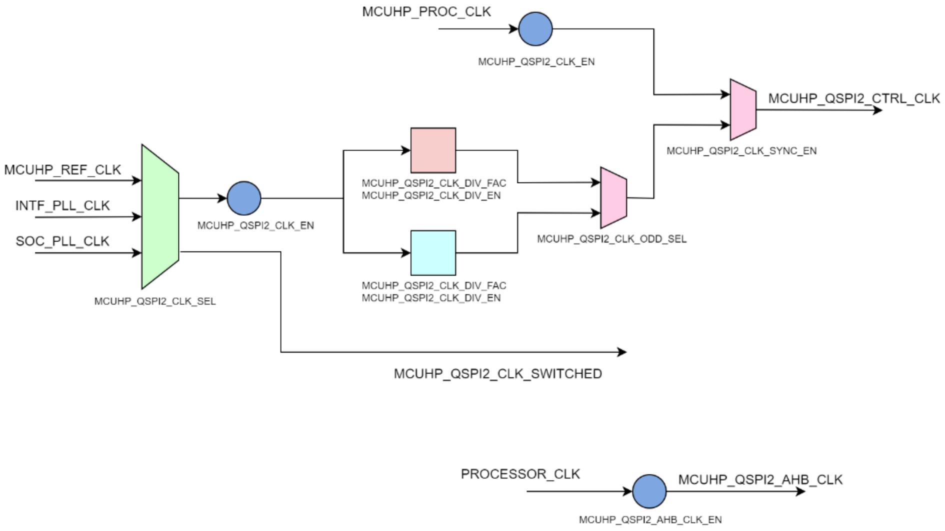


Figure 6.13. MCU-HP PSRAM Controller Clock Generation

6.12.8 UART0/UART1

The clocking scheme is similar for UART0 & UART1 controller except for the configuration registers. There are multiple modes of generating the clock for UART0/UART1 Controller. The Clock switching status can be read through MCUHP_UART0_CLK_SWITCHED, MCUHP_UART1_CLK_SWITCHED in MCUHP_CLK_STATUS_REG Register.

- Clock Generation
  - Clock source can be configured through MCUHP_UART0_CLK_SEL, MCUHP_UART1_CLK_SEL in MCUHP_CLK_CONFIG_REG2 Register.
  - Division factor can be configured through MCUHP_UART0_CLK_DIV_FAC, MCUHP_UART1_CLK_DIV_FAC in MCUHP_CLK_CONFIG_REG2 Register.
  - Divided clock from a Clock swallow or Fractional Divider can be selected through MCUHP_UART0_FRAC_CLK_SEL, MCUHP_UART1_FRAC_CLK_SEL in MCUHP_CLK_CONFIG_REG2 Register.
- The clocks to the controller can be disabled when not in use for efficient power consumption or before configuring the clock source and frequency. The controller clock and APB clock can be controlled independently.
  - Configure MCUHP_UART0_CLK_EN, MCUHP_UART1_CLK_EN in MCUHP_CLKEN_SET_REG1/ MCUHP_CLKEN_CLEAR_REG1 Register for enabling/disabling the Controller clock.
  - Configure MCUHP_UART0_APB_CLK_EN, MCUHP_UART1_APB_CLK_EN in MCUHP_CLKEN_SET_REG1/ MCUHP_CLKEN_CLEAR_REG1 Register for enabling/disabling the APB clock.

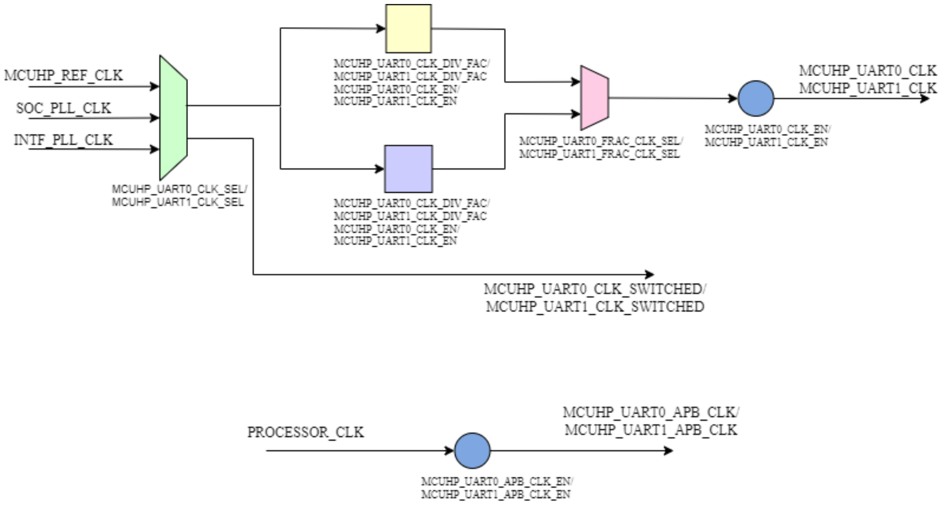


Figure 6.14. MCU-HP UART0/UART1 Clock Generation

6.12.9 SPI / SSI Primary

There are multiple clock sources for SPI/SSI Primary Controller. The Clock switching status can be read through MCUHP_SSI_MST_SCLK_SWITCHED in MCUHP_PLL_STATUS_REG Register.

- Clock Generation
  - Clock source can be configured through MCUHP_SSI_CLK_SEL in MCUHP_CLK_CONFIG_REG1 Register.
  - Division factor can be configured through MCUHP_SSI_CLK_DIV_FAC in MCUHP_CLK_CONFIG_REG1 Register.
- The clocks to the controller can be disabled when not in use for efficient power consumption or before configuring the clock source and frequency. The controller clock and APB clock can be controlled independently.
  - Configure MCUHP_SSI_CLK_EN in MCUHP_CLKEN_SET_REG2/MCUHP_CLKEN_CLEAR_REG2 Register for enabling/disabling the Controller clock.
  - Configure MCUHP_SSI_APB_CLK_EN in MCUHP_CLKEN_SET_REG2/MCUHP_CLKEN_CLEAR_REG2 Register for enabling/disabling the APB clock.

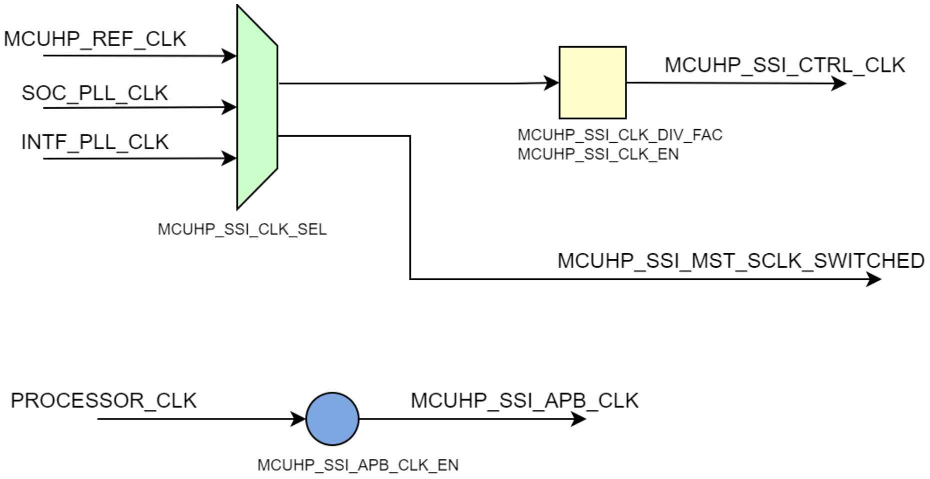


Figure 6.15. MCU-HP SPI/SSI Primary Clock Generation

6.12.10 I^{2}S Controller

There are multiple clock sources for I^{2}S Controller which is used in Primary Mode. The Clock switching status can be read through MCUHP_I2S_CLK_SWITCHED in MCUHP_PLL_STATUS_REG Register.

- Clock Generation
  - I^{2}S Secondary clock is derived from the external Primary Device through GPIO PAD's.
  - Clock source can be configured through MCUHP_I2S_CLK_SEL in MCUHP_CLK_CONFIG_REG5 Register.
  - Division factor can be configured through MCUHP_I2S_CLK_DIV_FAC in MCUHP_CLK_CONFIG_REG5 Register.
- The clocks to the controller can be disabled when not in use for efficient power consumption or before configuring the clock source and frequency. The controller clock and APB clock can be controlled independently.
  - Configure MCUHP_I2S_CLK_EN in MCUHP_CLKEN_SET_REG2/MCUHP_CLKEN_CLEAR_REG2 Register for enabling/disabling the Controller clock.
  - Configure MCUHP_I2S_APB_CLK_EN in MCUHP_CLKEN_SET_REG2/MCUHP_CLKEN_CLEAR_REG2 Register for enabling/disabling the APB Interface clock.
- In addition to the above, the I^{2}S Interface clock can be disabled.
  - Configure MCUHP_I2S_INTF_CLK_EN in MCUHP_CLKEN_SET_REG2/MCUHP_CLKEN_CLEAR_REG2 Register for enabling/disabling the Interface clock.
- The Primary/Secondary mode is configured through MCUHP_I2S_MASTER_SLAVE_MODE in MCUHP_MISC_CONFIG_3 Register.

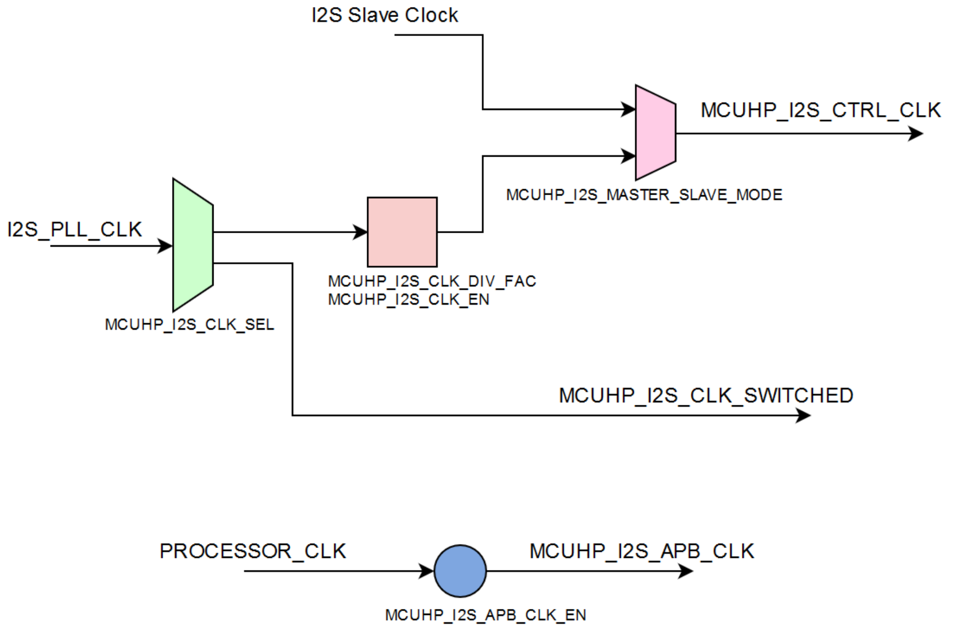


Figure 6.16. MCU-HP I2S Clock Generation

6.12.11 Configurable Timers

There are multiple clock sources for Configurable Timers. The Clock switching status can be read through MCUHP_CT_CLK_SWITCHED in MCUHP_PLL_STATUS_REG Register.

- Clock Generation
  - Clock source can be configured through MCUHP_CT_CLK_SEL in MCUHP_CLK_CONFIG_REG5 Register.
  - Division factor can be configured through MCUHP_CT_CLK_DIV_FAC in MCUHP_CLK_CONFIG_REG5 Register.
- The clocks to the controller can be disabled when not in use for efficient power consumption or before configuring the clock source and frequency.
  - Configure MCUHP_CT_CLK_EN in MCUHP_CLKEN_SET_REG1/MCUHP_CLKEN_CLEAR_REG1 Register for enabling/disabling the Controller clock.
  - Configure MCUHP_CT_PCLK_EN in MCUHP_CLKEN_SET_REG1/MCUHP_CLKEN_CLEAR_REG1 Register for enabling/disabling the Controller clock.

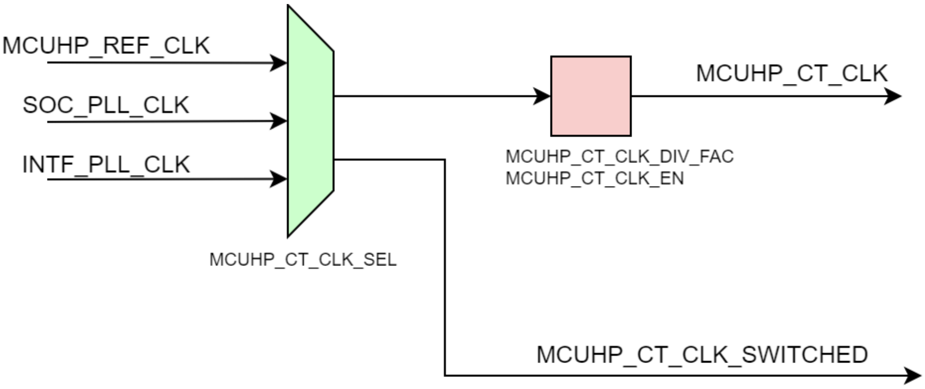


Figure 6.17. MCU-HP Configurable Timer Clock Generation

6.12.12 MVP

The MVP clock is generated using the Processor Clock.

- The clock to the controller can be disabled when not in use for efficient power consumption
  - Configure MCUHP_MVP_CLK_EN in MCUHP_CLKEN_SET_REG1/MCUHP_CLKEN_CLEAR_REG1 Register for enabling/disabling the Controller clock.

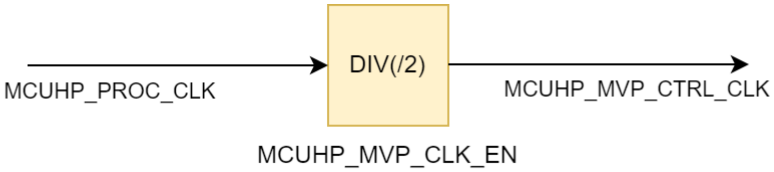


Figure 6.18. MCUHP MVP

6.12.13 Generic SPI Primary

There are multiple clock sources for Generic SPI Primary Controller. The Clock switching status can be read through MCUHP_GEN_SPI_MST1_SCLK_SWITCHED in MCUHP_PLL_STATUS_REG Register.

- Clock Generation
  - Clock source can be configured through MCUHP_GSPI_MST1_SCLK_SEL in MCUHP_CLK_CONFIG_REG1 Register.
- The clocks to the APB Interface can be disabled when not in use for efficient power consumption.
  - Configure MCUHP_GSPI_APB_CLK_EN in MCUHP_CLKEN_SET_REG2/MCUHP_CLKEN_CLEAR_REG2 Register for enabling/disabling the Controller clock.

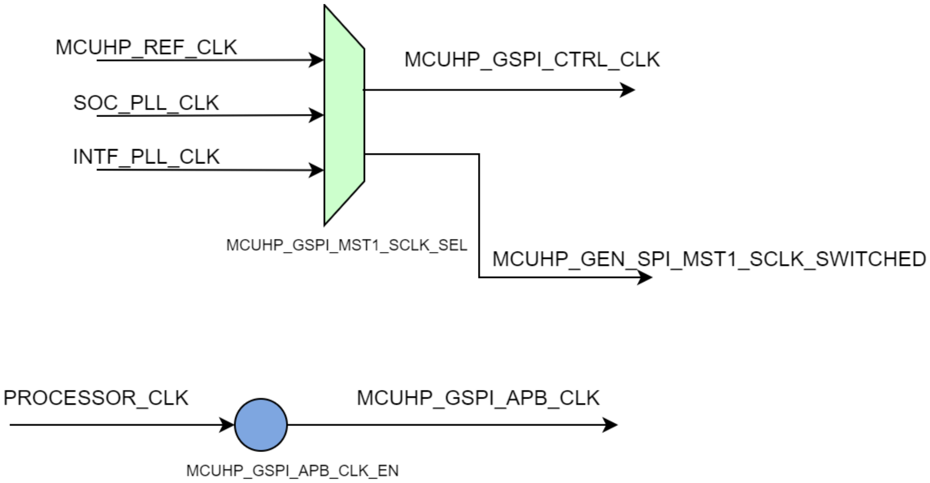


Figure 6.19. MCU-HP GSPI Clock Generation

6.12.14 MCU-ULP SoC Clock

There is only one clock sources for MCU ULP Clock from MCU HP domain. This can be used only in PS4 power state.

- Clock Generation
  - The Processor clock used in PS4 state as shown in the Processor clock generation will be used as the clock source.
  - Division factor can be configured through MCUHP_ULP_DIV_FAC in MCUHP_CLK_CONFIG_REG4 Register.
  - The divided clock can be selected through MCUHP_ULP_CLK_SEL in MCUHP_CLK_CONFIG_REG5 Register.
- The clocks to the controller can be disabled when not in use for efficient power consumption or before configuring the clock source and frequency.
  - Configure MCUHP_ULP_CLK_EN in MCUHP_CLK_EN_SET_REG1/MCUHP_CLK_EN_CLEAR_REG1 Register.

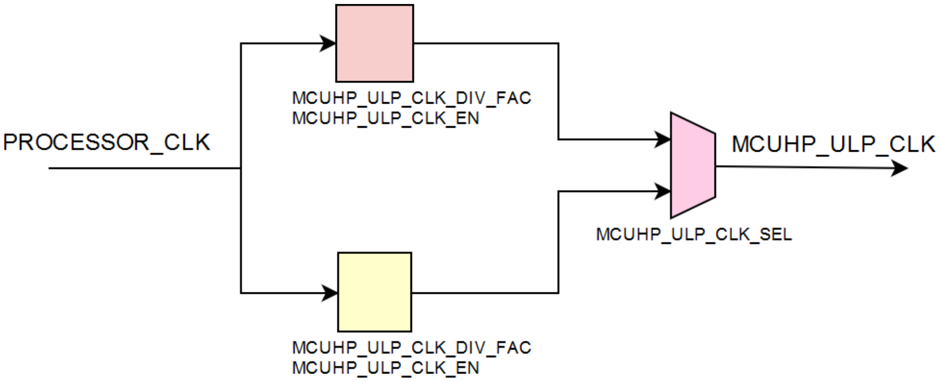


Figure 6.20. MCU-HP ULP Clock Generation

6.12.15 External Clock

There are multiple source for generating the clock for External components through GPIO PAD's. The Clock switching status can be read through MCUHP_EXT_CLK_SWITCHED in MCUHP_PLL_STATUS_REG Register.

- Clock Generation
  - Clock source can be configured through MCUHP_EXT_CLK_SEL in MCUHP_CLK_CONFIG_REG3 Register.
  - Division factor can be configured through MCUHP_EXT_CLK_DIV_FAC in MCUHP_CLK_CONFIG_REG3 Register.
- The clocks to the external components can be disabled when not in use for efficient power consumption or before configuring the clock source and frequency.
  - Configure MCUHP_EXT_CLK_EN in MCUHP_CLK_CONFIG_REG3 Register for enabling/disabling the clock.

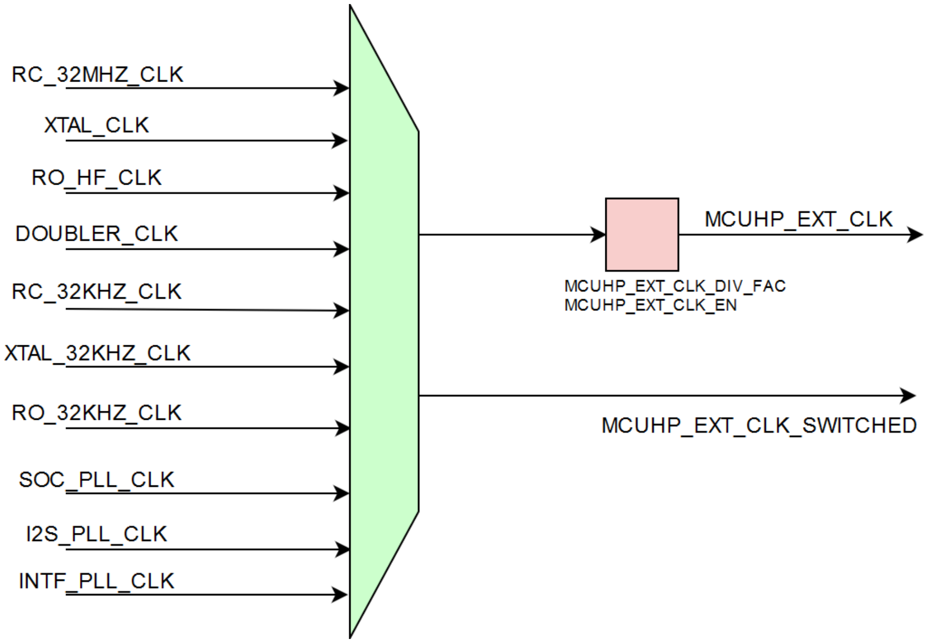


Figure 6.21. MCU-HP External Clock Generation

6.12.16 Static Clock Gated Domains

The clock to the domains which operate on the processor clock can be disabled when not in use for efficient power management. Below mentioned are the programming details for the same.

- **ICACHE:** Configure MCUHP_ICACHE_CLK_EN in MCUHP_CLKEN_SET_REG3/MCUHP_CLKEN_CLEAR_REG3 Register for enabling/disabling the clock to ICACHE module.
- **MCU HP DMA:** Configure MCUHP_DMA_CLK_EN in MCUHP_CLKEN_SET_REG1/MCUHP_CLKEN_CLEAR_REG1 Register for enabling/disabling the clock to DMA module.
- **Random-Number-Generator:** Configure MCUHP_RNG_CLK_EN in MCUHP_CLKEN_SET_REG1/ MCUHP_CLKEN_CLEAR_REG1 Register for enabling/disabling the clock to Random-Number-Generator module.
- **CRC Accelerator:** Configure MCUHP_CRC_CLK_EN in MCUHP_CLKEN_SET_REG1/MCUHP_CLKEN_CLEAR_REG1 Register for enabling/disabling the clock to CRC Accelerator module.
- **UDMA:** Configure MCUHP_UDMA_CLK_EN in MCUHP_CLKEN_SET_REG2/MCUHP_CLKEN_CLEAR_REG2 Register for enabling/disabling the clock to UDMA module.
- **Motor-Control PWM:** Configure MCUHP_MCPWM_CLK_EN in MCUHP_CLKEN_SET_REG2/MCUHP_CLKEN_CLEAR_REG2 Register for enabling/disabling the clock to Motor-Control PWM module.(whether to keep)
- **Quadrature Encoder:** Configure MCUHP_QE_CLK_EN in MCUHP_CLKEN_SET_REG2/MCUHP_CLKEN_CLEAR_REG2 Register for enabling/disabling the clock to Quadrature Encoder module.
- **I^{2}C - 2x:**
  - Configure MCUHP_I2C1_APB_CLK_EN, MCUHP_I2C0_APB_CLK_EN in MCUHP_CLKEN_SET_REG2/ MCUHP_CLKEN_CLEAR_REG2 Register for enabling/disabling the clock to I^{2}C APB Interface.
  - Configure MCUHP_I2C1_CLK_EN, MCUHP_I2C0_CLK_EN in MCUHP_CLKEN_SET_REG3/MCUHP_CLKEN_CLEAR_REG3 Register for enabling/disabling the clock to I^{2}C Controller.
- **SSI-SLV:**
  - Configure MCUHP_SSI_SLV_APB_CLK_EN in MCUHP_CLKEN_SET_REG2/MCUHP_CLKEN_CLEAR_REG2 Register for enabling/disabling the clock to I^{2}C APB Interface.
  - Configure MCUHP_SSI_SLV_CLK_EN in MCUHP_CLKEN_SET_REG2/MCUHP_CLKEN_CLEAR_REG2 Register for enabling/ disabling the clock to I^{2}C Controller.
- **Enhanced-GPIO:** Configure MCUHP_EGPIO_CLK_EN in MCUHP_CLKEN_SET_REG3/MCUHP_CLKEN_CLEAR_REG3 Register for enabling/disabling the clock to Enhanced-GPIO module.
- **eFUSE:** Configure MCUHP_EFUSE_CLK_EN in MCUHP_CLKEN_SET_REG3/MCUHP_CLKEN_CLEAR_REG3 Register for enabling/disabling the clock to eFUSE module.

6.12.17 Register Summary

Base Address: 0x4600_0000

Register Name	Offset	Description
Section 6.12.18.1 CLKEN_SET_REG1	0x00	Clock Enable Set Register 1
Section 6.12.18.2 CLKEN_CLEAR_REG1	0x04	Clock Enable Clear Register 1
Section 6.12.18.3 CLKEN_SET_REG2	0x08	Clock Enable Set Register 2
Section 6.12.18.4 CLKEN_CLEAR_REG2	0x0C	Clock Enable Clear Register 2
Section 6.12.18.5 CLKEN_SET_REG3	0x10	Clock Enable Set Register 3
Section 6.12.18.6 CLKEN_CLEAR_REG3	0x14	Clock Enable Clear Register 3
Section 6.12.18.7 CLK_CONFIG_REG1	0x18	Clock Configuration Register 1
Section 6.12.18.8 CLK_CONFIG_REG2	0x1C	Clock Configuration Register 2
Section 6.12.18.9 CLK_CONFIG_REG3	0x20	Clock Configuration Register 3
Section 6.12.18.10 CLK_CONFIG_REG4	0x24	Clock Configuration Register 4
Section 6.12.18.11 CLK_CONFIG_REG5	0x28	Clock Configuration Register 5
Section 6.12.18.15 DYN_CLK_GATE_DISABLE_REG	0x44	Dynamic Clock Gating Disable Register
Section 6.12.18.16 PLL_ENABLE_SET_REG	0x50	PLL Enable Set Register
Section 6.12.18.17 PLL_ENABLE_CLEAR_REG	0x54	PLL Enable Clear Register
Section 6.12.18.18 PLL_STAT_REG	0x58	PLL Enable status Register
Section 6.12.18.19 PLL_LOCK_INT_MASK_REG	0X5c	PLL Lock Interrupt Mask Register
Section 6.12.18.20 PLL_LOCK_INT_CLR_REG	0x60	PLL Lock Interrupt Clear Register
Section 6.12.18.21 PLL_LOCK_INT_DATA_REG	0x64	PLL Lock Interrupt Data Register
Section 6.12.18.22 SLEEP_CALIB_REG	0x68	Sleep Calibration Register
Section 6.12.18.23 CLK_CALIB_CTRL_REG1	0x6C	Clock Calibration Control Register 1
Section 6.12.18.24 CLK_CALIB_CTRL_REG2	0x70	Clock Calibration Control Register 2
Section 6.12.18.25 CLK_CALIB_STS_REG1	0x74	Clock Calibration Status Register 1
Section 6.12.18.26 CLK_CALIB_STS_REG2	0x78	Clock Calibration Status Register 2
Section 6.12.18.27 CLK_CONFIG_REG6	0x7C	Clock Configuration Register 6
Section 6.12.18.28 DYN_CLK_GATE_DISABLE_REG2	0x80	Dynamic Clock Gating Disable Register 2
Section 6.12.18.29 PLL_LOCK_INT_STATUS_REG	0x84	PLL Lock Interrupt Status Register
Section 6.12.18.30 MCUHP_SLEEP_CALIB_REG2	0x88	Read only Sleep Calibration Register

Base Address: 0x 4600_8000

Table 6.23. List of Registers

Register Name	Offset	Description
Section 6.12.18.12 MCUHP_MISC_CONFIG_1	0x14	Miscellaneous Clock Configuration Register1
Section 6.12.18.13 MCUHP_MISC_CONFIG_2	0x34	Miscellaneous Clock Configuration Register2
Section 6.12.18.14 MCUHP_MISC_CONFIG_3	0x44	Miscellaneous Clock Configuration Register3

6.12.18 Register Description

6.12.18.1 CLKEN_SET_REG1

Table 6.24. MCUHP_CLKEN_SET_REG1 Description

Bit	Access	Function	Default Value	Description
31	R/W	MCUHP_ULP_CLK_EN	0	Writing 1 to this enables clock to MCU-ULPDomain. Writing 0 to this has no effect.
30	R/W	MCUHP_MASK_HOST_CLK_AVAILABLE_FIX	0	This bit decides whether to consider negedge of host_clk_available in the generation of clock enable for host_clk gate in host mux 1'b1 => Don't consider 1'b0 => Invalid
29	R/W	Reserved	0	Reserved
28	R/W	MCUHP_MASK31_HOST_CLK_CNT	0	When MCUHP_MASK_HOST_CLK_WAIT_FIX is 1'b1, this bit decides whether to count for 32 or 16 XTAL clock cycles to come out of WAIT state in host mux FSM 1'b1 => Wait for 32 clock cycles 1'b0 => Invalid
27	R/W	MCUHP_MASK_HOST_CLK_WAIT_FIX	0	This bit decides whether to wait for a fixed number of xtal clock cycles(based on MCUHP_MASK31_HOST_CLK_CNT) or wait for a internally generated signal to come out of WAIT state in host mux FSM 1'b1 => Wait for fixed number of xtal clk cycles 1'b0 => Invalid
26	R/W	Reserved	0	Reserved
25	R/W	MCUHP_MVP_CLK_EN	0	Writing 1 to this enables clock to MVP. Writing 0 to this has no effect.
24	R/W	MCUHP_DCACHE_CLK_EN	1	Writing 1 to this enables clock to DCache. Writing 0 to this has no effect.
23	R/W	MCUHP_GNSS_MEM_CLK_ENABLE	0	Writing 1 to this enables clock to GNSS memory Writing 0 to this has no effect.
22	R/W	MCUHP_RNG_CLK_EN	0	Writing 1 to this enables clock to Random-Number-Generator. Writing 0 to this has no effect.
21	R/W	Reserved	0	Reserved
20	-	Reserved	-	It is recommended to write this bit to 0.
19	R/W	Reserved	0	Reserved
18	R/W	MCUHP_CRC_CLK_EN	0	Writing 1 to this enables clock to CRC Accelerator. Writing 0 to this has no effect.

Bit	Access	Function	Default Value	Description
17	R/W	MCUHP_SDIO_SYS_HCLK_ENABLE	0	Writing 1 to this enables clock to sdio AHB interface Writing 0 to this has no effect.
16	R/W	MCUHP_IID_CLK_ENABLE	0	Writing 1 to this enables clock to IID. Writing 0 to this has no effect.
15	-	Reserved	0	It is recommended to write this bit to 0.
14	R/W	MCUHP_SOC_PLL_SPI_CLK_ENABLE	0	Writing 1 to this enables clock for SOC PLL SPI Writing 0 to this has no effect.
13	R/W	MCUHP_DMA_CLK_EN	1	Writing 1 to this enables clock to DMA. Writing 0 to this has no effect.
12	R/W	MCUHP_ICACHE_CLK_2X_ENABLE	0	Writing 1 to this enables 2X clock for Icache Writing 0 to this has no effect.
11	R/W	MCUHP_ICACHE_CLK_ENABLE	0	Writing 1 to this enables clock for Icache Writing 0 to this has no effect.
10	R/W	MCUHP_CT_PCLK_EN	0	Writing 1 to this enables clock to Configurable Timers APB interface. Writing 0 to this has no effect.
9	R/W	MCUHP_CT_CLK_EN	0	Writing 1 to this enables clock to Configurable Timers. Writing 0 to this has no effect.
8	-	Reserved	-	It is recommended to write these bits to 0.
7	R/W	MCUHP_QSPI2_CLK_EN	1	Writing 1 to this enables clock to QSPI from dynamic mux Writing 0 to this has no effect.
6	R/W	MCUHP_QSPI2_CLK_SYNC_EN	0	Writing 1 to this enables SPI PSRM Controller clock in synchronous with Processor Clock. Writing 0 to this has no effect.
5	R/W	MCUHP_QSPI2_AHB_CLK_EN	1	Writing 1 to this enables clock to AHB Interface for SPI PSRAM Controller. Writing 0 to this has no effect.
4	R/W	MCUHP_QSPI2_CLK_DIV_EN	1	Writing 1 to this enables clock to Clock dividers for SPI PSRAM Controller. Writing 0 to this has no effect.
3	R/W	MCUHP_UART1_CLK_EN	0	Writing 1 to this enables clock to UART1 Controller. Writing 0 to this has no effect.
2	R/W	MCUHP_UART1_APB_CLK_EN	0	Writing 1 to this enables clock to UART1 APB Interface. Writing 0 to this has no effect.

Bit	Access	Function	Default Value	Description
1	R/W	MCUHP_USART0_CLK_EN	0	Writing 1 to this enables clock to USART0 Controller. Writing 0 to this has no effect.
0	R/W	MCUHP_USART0_APB_CLK_EN	0	Writing 1 to this enables clock to USART0 APB Interface. Writing 0 to this has no effect.

6.12.18.2 CLKEN_CLEAR_REG1

Table 6.25. MCUHP_CLKEN_CLEAR_REG1 Description

Bit	Access	Function	Default Value	Description
31	R/W	MCUHP_ULP_CLK_EN	0	Writing 1 to this disables clock to MCU-ULPDomain. Writing 0 to this has no effect.
30	R/W	MCUHP_MASK_HOST_CLK_AVAILABLE_FIX	0	Writing 1 to this bit clears the respective bit in MCUHP_CLKEN_SET_REG1 register Writing 0 to this has no effect.
29	R/W	Reserved	0	Reserved
28	R/W	MCUHP_MASK31_HOST_CLK_CNT	0	Writing 1 to this bit clears the respective bit in MCUHP_CLKEN_SET_REG1 register Writing 0 to this has no effect.
27	R/W	MCUHP_MASK_HOST_CLK_WAIT_FIX	0	Writing 1 to this bit clears the respective bit in MCUHP_CLKEN_SET_REG1 register Writing 0 to this has no effect.
26	R/W	Reserved	0	Reserved
25	R/W	MCUHP_MVP_CLK_EN	0	Writing 1 to this disables clock to MVP. Writing 0 to this has no effect.
24	R/W	MCUHP_DCACHE_CLK_EN	1	Writing 1 to this disables clock to DCache. Writing 0 to this has no effect.
23	R/W	MCUHP_GNSS_MEM_CLK_ENABLE	0	Writing 1 to this disables clock to GNSS memory Writing 0 to this has no effect.
22	R/W	MCUHP_RNG_CLK_EN	0	Writing 1 to this disables clock to Random-Number-Generator. Writing 0 to this has no effect.
21	R/W	Reserved	0	Reserved
20	-	Reserved	-	It is recommended to write these bits to 0.
19	R/W	Reserved	0	Reserved
18	R/W	MCUHP_CRC_CLK_EN	0	Writing 1 to this disables clock to CRC Accelerator. Writing 0 to this has no effect.
17	R/W	MCUHP_SDIO_SYS_HCLK_ENABLE	0	Writing 1 to this disables clock to sdio AHB interface Writing 0 to this has no effect.

Bit	Access	Function	Default Value	Description
16	R/W	MCUHP_IID_CLK_ENABLE	0	Writing 1 to this disables clock to IID. Writing 0 to this has no effect.
15	-	Reserved	0	It is recommended to write this bit to 0.
14	R/W	MCUHP_SOC_PLL_SPI_CLK_ENABLE	0	Writing 1 to this disables clock for SOC PLL SPI Writing 0 to this has no effect.
13	R/W	MCUHP_DMA_CLK_EN	1	Writing 1 to this disables clock to DMA. Writing 0 to this has no effect.
12	R/W	MCUHP_ICACHE_CLK_2X_ENABLE	0	Writing 1 to this disables 2X clock for Icache Writing 0 to this has no effect.
11	R/W	MCUHP_ICACHE_CLK_ENABLE	0	Writing 1 to this disables clock for Icache Writing 0 to this has no effect.
10	R/W	MCUHP_CT_PCLK_EN	0	Writing 1 to this disables clock to Configurable Timers APB interface. Writing 0 to this has no effect.
9	R/W	MCUHP_CT_CLK_EN	0	Writing 1 to this disables clock to Configurable Timers. Writing 0 to this has no effect.
8	-	Reserved	-	It is recommended to write these bits to 0.
7	R/W	MCUHP_QSPI2_CLK_SYNC_EN	0	Writing 1 to this enables SPI PSRM Controller clock in asynchronous with Processor Clock. Writing 0 to this has no effect.
6	R/W	MCUHP_QSPI2_CLK_EN	1	Writing 1 to this disables clock to QSPI from dynamic mux Writing 0 to this has no effect.
5	R/W	MCUHP_QSPI2_AHB_CLK_EN	1	Writing 1 to this disables clock to AHB Interface for SPI PSRAM Controller. Writing 0 to this has no effect.
4	R/W	MCUHP_QSPI2_CLK_DIV_EN	1	Writing 1 to this disables clock to Clock dividers for SPI PSRAM Controller. Writing 0 to this has no effect.
3	R/W	MCUHP_UART1_CLK_EN	0	Writing 1 to this disables clock to UART1 Controller. Writing 0 to this has no effect.
2	R/W	MCUHP_UART1_APB_CLK_EN	0	Writing 1 to this disables clock to UART1 APB Interface. Writing 0 to this has no effect.

Bit	Access	Function	Default Value	Description
1	R/W	MCUHP_UART0_CLK_EN	0	Writing 1 to this disables clock to UART0 Controller. Writing 0 to this has no effect.
0	R/W	MCUHP_UART0_APB_CLK_EN	0	Writing 1 to this disables clock to UART0 APB Interface. Writing 0 to this has no effect.

6.12.18.3 CLKEN_SET_REG2

Table 6.26. MCUHP_CLKEN_SET_REG2 Description

Bit	Access	Function	Default Value	Description
31	R/W	Reserved	0	Reserved
30	R/W	Reserved	0	Reserved
29	R/W	Reserved	0	Reserved
28	R/W	MCUHP_PLL_INTF_CLK_EN	0	Writing 1 to this enables clock to PLL interface. Writing 0 to this has no effect.
27	R/W	MCUHP_ROM_CLK_ENABLE	0	Writing 1 to this enables clock to ROM. Writing 0 to this has no effect.
26	R/W	MCUHP_MEM_CLK_ULP_ENABLE	0	Writing 1 to this enables clock to ULP memory Writing 0 to this has no effect.
25	R/W	Reserved	0	Reserved
24	R/W	MCUHP_SSI_CLK_EN	0	Writing 1 to this enables clock to SPI/SSI Primary Controller. Writing 0 to this has no effect.
23	R/W	MCUHP_SSI_APB_CLK_EN	0	Writing 1 to this enables clock to SPI/SSI Primary APB Interface. Writing 0 to this has no effect.
22	R/W	MCUHP_ARM_CLK_ENABLE	0	Writing 1 to this enables clock to arm Writing 0 to this has no effect.
21	R/W	MCUHP_EGPIO_PCLK_ENABLE	0	Writing 1 to this enables clock to EGPIO APB Interface. Writing 0 to this has no effect.
20	R/W	MCUHP_SGPIO_PCLK_ENABLE	0	Writing 1 to this enables clock to SGPIO APB Interface. Writing 0 to this has no effect.
19	-	Reserved	-	It is recommended to write this bit to 0.
18	R/W	MCUHP_MCPWM_CLK_EN	0	Writing 1 to this enables clock to Motor-Control PWM. Writing 0 to this has no effect.
17	R/W	MCUHP_QE_CLK_EN	0	Writing 1 to this enables clock to Quadrature Encoder APB interface. Writing 0 to this has no effect.
16	-	Reserved	-	It is recommended to write these bits to 0.
15	R/W	MCUHP_I2S_APB_CLK_EN	0	Writing 1 to this enables clock to I^{2}S APB Interface. Writing 0 to this has no effect.

Bit	Access	Function	Default Value	Description
14	R/W	MCUHP_I2S_INTF_CLK_EN	0	Writing 1 to this enables clock to I^{2}S Interface. Writing 0 to this has no effect.
13	R/W	MCUHP_I2S_CLK_EN	0	Writing 1 to this enables clock to I^{2}S Controller in Master Mode. Writing 0 to this has no effect.
12	R/W	MCUHP_QSPI_AHB_CLK_EN	1	Writing 1 to this enables clock to AHB Interface for SPI Flash Controller. Writing 0 to this has no effect.
11	R/W	MCUHP_QSPI_CLK_DIV_EN	1	Writing 1 to this enables clock to Clock dividers for SPI Flash Controller. Writing 0 to this has no effect.
10	R/W	MCUHP_SSI_SLV_CLK_EN	0	Writing 1 to this enables clock to SSI Secondary Controller. Writing 0 to this has no effect.
9	R/W	MCUHP_SSI_SLV_APB_CLK_EN	0	Writing 1 to this enables clock to SSI Secondary APB Interface. Writing 0 to this has no effect.
8	R/W	MCUHP_I2C1_APB_CLK_EN	0	Writing 1 to this enables clock to I^{2}C-1 . Writing 0 to this has no effect.
7	R/W	MCUHP_I2C_APB_CLK_EN	0	Writing 1 to this enables clock to I^{2}C-0. Writing 0 to this has no effect.
6	R/W	MCUHP_UDMA_CLK_EN	0	Writing 1 to this enables clock to Micro-DMA. Writing 0 to this has no effect.
5:4	-	Reserved	-	It is recommended to write these bits to 0.
3	R/W	Reserved	0	Reserved
2	R/W	Reserved	0	Reserved
1	R/W	Reserved	0	Reserved
0	R/W	MCUHP_GSPI_APB_CLK_EN	0	Writing 1 to this enables clock to Generic-SPI Primary APB Interface. Writing 0 to this has no effect.

6.12.18.4 CLKEN_CLEAR_REG2

Table 6.27. MCUHP_CLKEN_CLEAR_REG2 Description

Bit	Access	Function	Default Value	Description
31	R/W	Reserved	0	Reserved
30	R/W	Reserved	0	Reserved
29	R/W	Reserved	0	Reserved
28	R/W	MCUHP_PLL_INTF_CLK_EN	0	Writing 1 to this disables clock to PLL interface. Writing 0 to this has no effect.
27	R/W	MCUHP_ROM_CLK_ENABLE	0	Writing 1 to this disables clock to ROM. Writing 0 to this has no effect.
26	R/W	MCUHP_MEM_CLK_ULP_ENABLE	0	Writing 1 to this disables clock to ULP memory Writing 0 to this has no effect.
25	R/W	Reserved	0	Reserved
24	R/W	MCUHP_SSI_CLK_EN	0	Writing 1 to this disables clock to SPI/SSI Primary Controller. Writing 0 to this has no effect.
23	R/W	MCUHP_SSI_APB_CLK_EN	0	Writing 1 to this disables clock to SPI/SSI Primary APB Interface. Writing 0 to this has no effect.
22	R/W	MCUHP_ARM_CLK_ENABLE	0	Writing 1 to this disables clock to arm Writing 0 to this has no effect.
21	R/W	MCUHP_EGPIO_PCLK_ENABLE	0	Writing 1 to this disables clock to EGPIO APB Interface. Writing 0 to this has no effect.
20	R/W	MCUHP_SGPIO_PCLK_ENABLE	0	Writing 1 to this disables clock to SGPIO APB Interface. Writing 0 to this has no effect.
19	-	Reserved	-	It is recommended to write this bit to 0.
18	R/W	MCUHP_MCPWM_CLK_EN	0	Writing 1 to this disables clock to Motor-Control PWM. Writing 0 to this has no effect.
17	R/W	MCUHP_QE_CLK_EN	0	Writing 1 to this disables clock to Quadrature Encoder. Writing 0 to this has no effect.
16	-	Reserved	-	It is recommended to write these bits to 0.
15	R/W	MCUHP_I2S_APB_CLK_EN	0	Writing 1 to this disables clock to I^{2}S APB Interface. Writing 0 to this has no effect.

Bit	Access	Function	Default Value	Description
14	R/W	MCUHP_I2S_INTF_CLK_EN	0	Writing 1 to this disables clock to I^{2}S Interface. Writing 0 to this has no effect.
13	R/W	MCUHP_I2S_CLK_EN	0	Writing 1 to this disables clock to I^{2}S Controller in Master Mode. Writing 0 to this has no effect.
12	R/W	MCUHP_QSPI_AHB_CLK_EN	1	Writing 1 to this disables clock to AHB Interface for SPI Flash Controller. Writing 0 to this has no effect.
11	R/W	MCUHP_QSPI_CLK_DIV_EN	1	Writing 1 to this disables clock to Clock dividers for SPI Flash Controller. Writing 0 to this has no effect.
10	R/W	MCUHP_SSI_SLV_CLK_EN	0	Writing 1 to this disables clock to SSI Secondary Controller. Writing 0 to this has no effect.
9	R/W	MCUHP_SSI_SLV_APB_CLK_EN	0	Writing 1 to this disables clock to SSI Secondary APB Interface. Writing 0 to this has no effect.
8	R/W	MCUHP_I2C1_APB_CLK_EN	0	Writing 1 to this disables clock to I^{2}C-1 APB Interface. Writing 0 to this has no effect.
7	R/W	MCUHP_I2C_APB_CLK_EN	0	Writing 1 to this disables clock to I^{2}C-0 APB Interface. Writing 0 to this has no effect.
6	R/W	MCUHP_UDMA_CLK_EN	0	Writing 1 to this disables clock to Micro-DMA. Writing 0 to this has no effect.
5:4	-	Reserved	-	It is recommended to write these bits to 0.
3	R/W	Reserved	0	Reserved
2	R/W	Reserved	0	Reserved
1	R/W	Reserved	0	Reserved
0	R/W	MCUHP_GSPI_APB_CLK_EN	0	Writing 1 to this disables clock to Generic-SPI Primary APB Interface. Writing 0 to this has no effect.

6.12.18.5 CLKEN_SET_REG3

Table 6.28. MCUHP_CLKEN_SET_REG3 Description

Bit	Access	Function	Default Value	Description
31:28	-	Reserved	-	It is recommended to write these bits to 0.
27	R/W	MCUHP_ICACHE_ENABLE	1	Writing 1 to this enables clock to icache. Writing 0 to this has no effect.
26	R/W	MCUHP_M4_SOC_CLK_FOR_OTHER_ENABLE	0	Writing 1 to this enables M4-SOC Other Clock. Writing 0 to this has no effect.
25	R/W	MCUHP_ROM_MISC_STATIC_ENABLE	0	Writing 1 to this enables clock to ROM. Writing 0 to this has no effect.
24	R/W	Reserved	1	Reserved
23	R/W	MCUHP_TASS_M4SS_SDIO_SWITCH_CLK_ENABLE	1	UNUSED
22	R/W	MCUHP_TASS_M4SS_128K_SWITCH_CLK_ENABLE	1	UNUSED
21	R/W	MCUHP_TASS_M4SS_64K_SWITCH_CLK_ENABLE	1	UNUSED
20	R/W	Reserved	0	Reserved
19	R/W	MCUHP_EFUSE_PCLK_ENABLE	0	Writing 1 to this enables clock to efuse controller APB interface. Writing 0 to this has no effect.
18	R/W	MCUHP_I2C1_CLK_EN	0	Writing 1 to this enables clock to I^{2}C-1 Controller. Writing 0 to this has no effect.
17	R/W	MCUHP_I2C0_CLK_EN	0	Writing 1 to this enables clock to I^{2}C-0 Controller. Writing 0 to this has no effect.

Bit	Access	Function	Default Value	Description
16	R/W	MCUHP_EGPIO_CLK_EN	0	Writing 1 to this enables clock to Enhanced-GPIO Controller. Writing 0 to this has no effect.
15	-	Reserved	-	It is recommended to write these bits to 0.
14	R/W	MCUHP_QSPI_CLK_SYNC_EN	0	Writing 1 to this enables SPI Flash Controller clock in synchronous with Processor Clock. Writing 0 to this has no effect.
13	R/W	MCUHP_QSPI_CLK_EN	1	Writing 1 to this enables clock to QSPI from dynamic mux Writing 0 to this has no effect.
12	R/W	Reserved	1	Reserved
11:9	R/W	Reserved	-	It is recommended to write these bits to 0.
8	R/W	Reserved	0	Reserved
7	R/W	Reserved	0	Reserved
6	R/W	MCUHP_ICM_CLK_ENABLE	0	Writing 1 to this enables clock to ICM. Writing 0 to this has no effect.
5	R/W	MCUHP_EFUSE_CLK_EN	1	Writing 1 to this enables clock to eFUSE Controller. Writing 0 to this has no effect.
4	R/W	MCUHP_MISC_CONFIG_PCLK_ENABLE	0	Writing 1 to this enables clock to misc config registers APB interface. Writing 0 to this has no effect.
3	-	Reserved	-	It is recommended to write this bit to 0.

Bit	Access	Function	Default Value	Description
2	R/W	MCUHP_CM_BUS_CLK_ENABLE	0	Writing 1 to this enables clock to cm bus. Writing 0 to this has no effect.
1	R/W	MCUHP_M4_CORE_CLK_ENABLE	1	Writing 1 to this enables clock to M4 core Writing 0 to this has no effect.
0	R/W	MCUHP_BUS_CLK_ENABLE	0	Writing 1 to this enables bus_clk Writing 0 to this has no effect.

6.12.18.6 CLKEN_CLEAR_REG3

Table 6.29. MCUHP_CLKEN_CLEAR_REG3 Description

Bit	Access	Function	Default Value	Description
31:28	-	Reserved	-	It is recommended to write these bits to 0.
27	R/W	MCUHP_ICACHE_ENABLE	1	Writing 1 to this disables clock to icache. Writing 0 to this has no effect.
26	R/W	MCUHP_M4_SOC_CLK_FOR_OTHER_ENABLE	0	Writing 1 to this disables M4-SOC Other Clock. Writing 0 to this has no effect.
25	R/W	MCUHP_ROM_MISC_STATIC_ENABLE	0	Writing 1 to this disables clock to ROM. Writing 0 to this has no effect.
24	R/W	Reserved	0	Reserved
23	R/W	MCUHP_TASS_M4SS_SDIO_SWITCH_CLK_ENABLE	1	UNUSED
22	R/W	MCUHP_TASS_M4SS_128K_SWITCH_CLK_ENABLE	1	UNUSED
21	R/W	MCUHP_TASS_M4SS_64K_SWITCH_CLK_ENABLE	1	UNUSED
20	R/W	Reserved	0	Reserved
19	R/W	MCUHP_EFUSE_PCLK_ENABLE	0	Writing 1 to this disables clock to efuse controller APB interface. Writing 0 to this has no effect.
18	R/W	MCUHP_I2C1_CLK_EN	0	Writing 1 to this disables clock to I^{2}C-1 Controller. Writing 0 to this has no effect.
17	R/W	MCUHP_I2C0_CLK_EN	0	Writing 1 to this disables clock to I^{2}C-0 Controller. Writing 0 to this has no effect.

Bit	Access	Function	Default Value	Description
16	R/W	MCUHP_EGPIO_CLK_EN	0	Writing 1 to this disables clock to Enhanced-GPIO Controller. Writing 0 to this has no effect.
15	-	Reserved	-	It is recommended to write these bits to 0.
14	R/W	MCUHP_QSPI_CLK_SYNC_EN	0	Writing 1 to this disables SPI Flash Controller clock in synchronous with Processor Clock. Writing 0 to this has no effect.
13	R/W	MCUHP_QSPI_CLK_EN	1	Writing 1 to this disables clock to SPI Flash Controller. Writing 0 to this has no effect.
12	R/W	Reserved	0	Reserved
11:9	R/W	Reserved	-	It is recommended to write these bits to 0.
8	R/W	Reserved	0	Reserved
7	R/W	Reserved	0	Reserved
6	R/W	MCUHP_ICM_CLK_ENABLE	0	Writing 1 to this disables clock to ICM. Writing 0 to this has no effect.
5	R/W	MCUHP_EFUSE_CLK_EN	1	Writing 1 to this disables clock to eFUSE Controller. Writing 0 to this has no effect.
4	R/W	MCUHP_MISC_CONFIG_PCLK_ENABLE	0	Writing 1 to this disables clock to misc config registers APB interface. Writing 0 to this has no effect.
3	-	Reserved	-	It is recommended to write this bit to 0.

Bit	Access	Function	Default Value	Description
2	R/W	MCUHP_CM_BUS_CLK_ENABLE	0	Writing 1 to this disables clock to cm bus. Writing 0 to this has no effect.
1	R/W	MCUHP_M4_CORE_CLK_ENABLE	1	Writing 1 to this disables clock to M4 core Writing 0 to this has no effect.
0	R/W	MCUHP_BUS_CLK_ENABLE	0	Writing 1 to this disables bus_clk Writing 0 to this has no effect.

6.12.18.7 CLK_CONFIG_REG1

Table 6.30. MCUHP_CLK_CONFIG_REG1 Description

Bit	Access	Function	Default Value	Description
31:27	-	Reserved	-	It is recommended to write these bits to 0.
26:24	R/W	MCUHP_GSPI_SCLK_SEL	7	Specifies the clock source for GSPI. 0 - Reserved 1 - MCU-HP Reference Clock 2 - SoC-PLL Clock 3 - Reserved 4 - Interface-PLL Clock 5,6,7 - Output clock is gated
23	R/W	MCUHP_PLL_INTF_CLK_SWALLOW_SEL	0	Selects the Divider type for PLL INTF Clk. 0 - Clock divided by 2 with 50% duty cycle 1 - Swallowed clock is selected with division factor MCUHP_PLL_INTF_CLK_DIV_FAC
22:19	R/W	MCUHP_PLL_INTF_CLK_DIV_FAC	2	Specifies the clock division factor for PLL interface clock. If pll_intf_clk_enable is 1'b0 clock is gated. Else, – when pll_intf_clk_swallow_sel is 1'b1, output clock is a swallowed clock with the following frequency. 4'h0,4'h1 => clk_out = clk_in >64h1 => clk_out = clk_in/ pll_intf_clk_div_fac when pll_intf_clk_swallow_sel is 1'b0 output clock is a 50% duty cycle clock with the following frequency. 4'h0 => clk_out = clk_in/2 >4'h0 => clk_out = clk_in/ (2*pll_intf_clk_div_fac)
18	R/W	MCUHP_PLL_INTF_CLK_SEL	0	Specifies the clock source for PLL interface clock 0 - Interface-PLL Clock 1 - SoC-PLL Clock

Bit	Access	Function	Default Value	Description
17:15	R/W	MCUHP_SSI_CLK_SEL	7	Specifies the clock source for SPI/SSI Primary. 0 - MCU-HP Reference Clock 1 - SoC-PLL Clock 2 - Reserved 3 - Interface-PLL Clock 4,5 - Reserved 6,7 - Output Clock is gated
14:11	R/W	MCUHP_SSI_CLK_DIV_FAC	1	Specifies the clock division factor for for SPI/SSI Primary. If ssi_mst_sclk_enable is 1'b0 clock is gated. Else output clock is a swallowed clock with the following frequency. 4'h0,4'h1 => Divider is bypassed >4'h1 => clk_out = clk_in/ ssi_mst_sclk_div_fac
10	R/W	MCUHP_SLP_RF_CLK_SEL	0	Specifies clock for m4_soc_rf_ref_clk 0 - m4_soc_clk 1 - rf_ref_clk
9	R/W	MCUHP_QSPI_CLK_SWALLOW_SEL	0	Specifies divider type for QSPI 0 - Clock divider is selected with 50% duty cycle. Division factor is MCUHP_QSPI_CLK_DIV_FAC 1 - Swallowed clock is selected with MCUHP_QSPI_CLK_DIV_FAC Before Changing this ensure that the input clocks are gated

Bit	Access	Function	Default Value	Description
8:3	R/W	MCUHP_QSPI_CLK_DIV_FAC	1	Specifies the clock division factor for SPI Flash Controller. If qspi_clk_enable is 1'b0 clock is gated. Else 1) when qspi_clk_swallow_sel is 1'b1 and qspi_odd_div_sel is 1'b0 output clock is a swallowed clock with the following frequency. 6'h0,6'h1 => clk_out = clk_in >6'h1 => clk_out = clk_in/ qspi_clk_div_fac 2) 2) when qspi_clk_swallow_sel is 1'b0 and qspi_odd_div_sel is 1'b0 output clock is a 50% duty cycle clock with the following frequency. 6'h0 => clk_out = clk_in/2 >6'h0 => clk_out = clk_in/ (2*qspi_clk_div_fac) 3) When qspi_odd_div_sel is 1'b1, output clock is a 50% duty cycle clock with the following frequency. clk_out = clk_in/qspi_clk_div_fac when qspi_clk_div_fac is an odd number >=3, else output clock is gated
2:0	R/W	MCUHP_QSPI_CLK_SEL	0	Specifies the clock source for SPI Flash controller when independent clock source w.r.t Processor is selected. 0 - MCU-HP Reference Clock PLL_STATUS_REG 1 - Interface-PLL Clock 2 - Reserved 3 - SoC-PLL Clock 4,5,6,7 - Output Clock is gated

6.12.18.8 CLK_CONFIG_REG2

Table 6.31. MCUHP_CLK_CONFIG_REG2 Description

Bit	Access	Function	Default Value	Description
31	R/W	Reserved	-	Reserved
30	R/W	MCUHP_UART1_FRAC_CLK_SEL	0	Selects the Divider type for UART1 Controller. 0 - Clock Swallow output is selected 1 - Fractional Clock Divider output is selected
29	R/W	MCUHP_UART0_FRAC_CLK_SEL	0	Selects the Divider type for UART0 Controller. 0 - Clock Swallow output is selected 1 - Fractional Clock Divider output is selected
28	R/W	MCUHP_QSPI_CLK_ODD_SEL	0	Selects the Divider type for SPI Flash Controller when independent clock source w.r.t Processor is selected. 0 - EVEN Clock Divider output is selected 1 - ODD Clock Divider output is selected ODD division factor supported values are 3,5,7 and so on till 63 . Division factor value 1 is not supported
27:24	R/W	Reserved	1	Reserved
23:14	-	Reserved	-	It is recommended to write these bits to 0.
13:10	R/W	MCUHP_UART1_CLK_DIV_FAC	1	Specifies the clock division factor for UART1 Controller. If usart1_sclk_enable is 1'b0 clock is gated. Else output clock is a swallowed clock with the following frequency. **when usart1_sclk_frac_sel is 1'b0** 4'h0,4'h1 => Divider is bypassed. >4'h1 => clk_out = clk_in/ usart1_sclk_div_fac **when usart1_sclk_frac_sel is 1'b1** Only 3 bits,[2:0] are valid 3'h0 => Clock is gated 3'h1 to 3''h7 => clk_out = clk_in/ (usart1_sclk_div_fac+0.5)

Bit	Access	Function	Default Value	Description
9:7	R/W	MCUHP_UART1_CLK_SEL	7	Specifies the clock source to be used for UART1 Controller. 0 - MCU-HP Reference Clock 1 - SoC-PLL Clock 2 - Reserved 3 - Interface-PLL Clock 4 - Reserved 5,6,7 - Clock is gated
6:3	R/W	MCUHP_UART0_CLK_DIV_FAC	1	Specifies the clock division factor for UART0 Controller. If usart0_sclk_enable is 1'b0 clock is gated. Else output clock is a swallowed clock with the following frequency. **when usart0_sclk_frac_sel is 1'b0** 4'h0,4'h1 => Divider is bypassed >4'h1 => clk_out = clk_in/ usart0_sclk_div_fac **when usart0_sclk_frac_sel is 1'b1** Only 3 bits,[2:0] are valid 3'h0 => Clock is gated 3'h1 to 3''h7 => clk_out = clk_in/ (usart0_sclk_div_fac+0.5)
2:0	R/W	MCUHP_UART0_CLK_SEL	7	Specifies the clock source to be used for UART0 Controller. 0 - MCU-HP Reference Clock 1 - SoC-PLL Clock 2 - Reserved 3 - Interface-PLL Clock 4 - Reserved 5,6,7 - Output Clock is gated.

6.12.18.9 CLK_CONFIG_REG3

Table 6.32. MCUHP_CLK_CONFIG_REG3 Description

Bit	Access	Function	Default Value	Description
31:19	-	Reserved	-	It is recommended to write these bits to 0.
18	R/W	MCUHP_EXT_CLK_EN	0	Writing 1 to this enables the External clock on GPIO PAD. Writing 0 to this disables the External clock on GPIO PAD.
17:12	R/W	MCUHP_EXT_CLK_DIV_FAC	0	Specifies the clock division factor for External Clock. 6'h0 => Divider is bypassed. >6'h0 => clk_out = clk_in/ (mcu_clkout_div_fac * 2)
11:8	R/W	MCUHP_EXT_CLK_SEL	0	Specifies the clock source to be used for External Clock. 0 - Output Clock is Gated 1 - High Freq RC Clock source 2 - XTAL Clock source 3 - Reserved 4 - High Freq RO Clock source 5 - Doubler Clock 6 - Reserved 7 - Low Freq RC Clock source 8 - Low Freq XTAL Clock source 9 - Low Freq RO Clock source 10 - Interface-PLL Clock 11,12 - Reserved 13 - SoC-PLL Clock 14 - I2S-PLL Clock 15 - Reserved
7:0	R/W	Reserved.	1	Reserved.

6.12.18.10 CLK_CONFIG_REG4

Table 6.33. MCUHP_CLK_CONFIG_REG4 Description

Bit	Access	Function	Default Value	Description
31	R/W	Reserved.	0	Reserved.
30:25	R/W	MCUHP_ULP_DIV_FAC	4	Specifies the clock division factor for MCU ULP Domain source. If ulpss_clk_enable is 1'b0 clock is gated. Else output clock is a divided clock with the following frequency. 6'h0 => Divider is bypassed. >6'h0 => clk_out = clk_in/ 2* ulpss_clk_div_fac
24	R/W	Reserved.	1	
23	R/W	Reserved.	0	
22:21	R/W	SLEEP_CLK_SEL	2	Specifies clock source for sleep clock 0 -32KHz RC clock 1 -32KHz crystal clock 2 - Output Clock is gated 3 - 32KHz RO clock Note: Use RO & RC clock only if accuracy is not crucial. Otherwise use crystal clock
20	R/W	BYPASS_INTF_PLL_CLK	0	Specifies Interface PLL clock select 0 -PLL clock 1 - Bypass clock selected by MCUHP_INTF_PLL_CLK_BYP_SEL
19	R/W	BYPASS_MODEM_PLL_CLK2	0	Specifies MODEM PLL clock select 0 -PLL clock 1 - Bypass clock selected by MCUHP_MODEM_PLL_CLK_BYP_SEL
18	R/W	BYPASS_MODEM_PLL_CLK1	0	Specifies MODEM PLL clock select 0 -PLL clock 1 - Bypass clock selected by MCUHP_MODEM_PLL_CLK_BYP_SEL
17	R/W	BYPASS_I2S_PLL_CLK	0	Specifies I2S PLL clock select 0 -PLL clock 1 - Bypass clock selected by MCUHP_I2S_PLL_CLK_BYP_SEL

Bit	Access	Function	Default Value	Description
16	R/W	BYPASS_SOC_PLL_CLK	0	Specifies SOC PLL clock select 0 -PLL clock 1 - Bypass clock selected by MCUHP_SOC_PLL_CLK_BYP_SEL
15	R/W	MODEM_PLL_BYPCLK_CLKCLNR_OFF	1	
14	R/W	MODEM_PLL_BYPCLK_CLKCLNR_ON	0	
13	R/W	I2S_PLL_BYPCLK_CLKCLNR_OFF	1	
12	R/W	I2S_PLL_BYPCLK_CLKCLNR_ON	0	
11:10	-	Reserved	2	It is recommended to retain the contents by using read/modify write to this register.
9	R/W	MCUHP_SOC_INTF_PLL_BYPCLK_CLKCLNR_OFF	1	
8	R/W	MCUHP_SOC_INTF_PLL_BYPCLK_CLKCLNR_ON	0	
7:6	R/W	MCUHP_INTF_PLL_CLK_BYP_SEL	0	Specifies bypass clock source for Intercace PLL 0 - Interface PLL bypass clock 1 - Modem PLL bypass clock 2 - Reference bypass clock 3 - I2S PLL bypass clock
5:4	R/W	MCUHP_MODEM_PLL_CLK_BYP_SEL	1	Specifies bypass clock source for MODEM PLL 0 - Interface PLL bypass clock 1 - Modem PLL bypass clock 2 - Reference bypass clock 3 - I2S PLL bypass clock
3:2	R/W	MCUHP_I2S_PLL_CLK_BYP_SEL	3	Specifies bypass clock source for I2S PLL 0 - Interface PLL bypass clock 1 - Modem PLL bypass clock 2 - Reference bypass clock 3 - I2S PLL bypass clock
1:0	R/W	MCUHP_SOC_PLL_CLK_BYP_SEL	0	Specifies bypass clock source for SOC PLL 0 - Interface PLL bypass clock 1 - Modem PLL bypass clock 2 - Reference bypass clock 3 - I2S PLL bypass clock

6.12.18.11 CLK_CONFIG_REG5

Table 6.34. MCU_CLK_CONFIG_REG5 Description

Bit	Access	Function	Default Value	Description
30:29	-	Reserved	-	It is recommended to retain the contents by using read/modify write to this register.
28	R/W	MCUHP_ULP_CLK_SEL	0	Specifies the divider type to be selected to MCU ULP Domain source. 0 - Clock Divider output is selected 1 - Odd Clock Divider output is selected
27	-	Reserved	-	It is recommended to write this bit to 0.
26	R/W	MCUHP_M4_SOC_HOST_CLK_SEL	0	
25:20	R/W	MCUHP_CT_CLK_DIV_FAC	1	Specifies the clock division factor for Configurable Timers. If ct_clk_enable is 1'b0 clock is gated. Else output clock is a 50% divided clock with the following frequency. 6'h0 => Divider is bypassed. >6'h0 => clk_out = clk_in/ 2*ct_clk_div_fac
19:17	R/W	MCUHP_CT_CLK_SEL	7	Specifies the clock source to be used for Configurable Timers. 0 - MCU-HP Reference Clock 1 - Interface-PLL Clock 2 - SoC-PLL Clock 3 - Reserved 4,5,6,7 - Output Clock is gated
16:11	R/W	MCUHP_I2S_CLK_DIV_FAC	1	Specifies the clock division factor for I^{2}S Controller in Master Mode. Output clock is a 50% divided clock with the following frequency. 6'h0 => Divider is bypassed. >6'h0 => clk_out = clk_in/ 2*i2s_clk_div_fac

Bit	Access	Function	Default Value	Description
10	R/W	MCUHP_I2S_CLK_SEL	0	Specifies the clock source to be used for I^{2}S Controller in Master Mode. 0 - I2S-PLL Clock 1 - Reserved
9:4	R/W	MCUHP_M4_SOC_CLK_DIV_FAC	1	Specifies the clock division factor for Processor Clock. Output clock is a swallowed clock with the following frequency. 6'h0,6'h1 => Divider is bypassed. >6'h1 => clk_out = clk_in/ m4_soc_clk_div_fac
3:0	R/W	MCUHP_M4_SOC_CLK_SEL	0	Specifies the clock source to be used for Processor. 0 - MCU-HP Reference Clock 1 - Reserved 2 - SoC-PLL Clock 3 - Reserved 4 - Interface-PLL Clock 5 - Low-Power Clock 6,7 - Reserved

6.12.18.12 MCUHP_MISC_CONFIG_1

Table 6.35. MCUHP_MISC_CONFIG_1 Description

Bit	Access	Function	Default Value	Description
31:17	-	Reserved	-	It is recommended to retain the contents by using read/modify write to this register.
16	R/W	Reserved	0	Reserved
15:0	-	Reserved	-	It is recommended to retain the contents by using read/modify write to this register.

6.12.18.13 MCUHP_MISC_CONFIG_2

Table 6.36. MCUHP_MISC_CONFIG_2 Description

Bit	Access	Function	Default Value	Description
31:6	R/W	Reserved		
5:5	R/W	Reserved	0	Reserved
4:4	R/W	Reserved	0	Reserved
4:2	R/W	PCM_BIT_RES	0	The bit-resolution of the data on PCM. 3'b000 - 8-bit 3'b001 - 12-bit 3'b010 - 16-bit, 3'b011 - 24-bit 3'b1xx - 32-bit
1:1	R/W	PCM_FSYNC_START	0	This bit has to be programmed according to when the MS bit of the PCM data is driven w.r.t. the FSYNC signal of PCM. '1' - The MS bit of data is driven in the same clock cycle as FSYNC going high. '0' - The MS bit of data is driven one clock cycle after FSYNC goes high.
0:0	R/W	PCM_ENABLE	0	Enable/disable PCM mode of I2S interface. When PCM is enabled, I2S is disabled and vice versa '1' - PCM mode is enabled and I2S mode is disabled. This programming is valid only when the GPIO signals are programmed for I2S mode. '0' - PCM mode is disabled and I2S mode is enabled. This programming is in addition to the other GPIO level programming to enable I2S mode.

6.12.18.14 MCUHP_MISC_CONFIG_3

Table 6.37. MCUHP_MISC_CONFIG_3 Description

Bit	Access	Function	Default Value	Description
31:24	-	Reserved	-	It is recommended to retain the contents by using read/modify write to this register.
23	R/W	MCUHP_I2S_PRIMARY_SECONDARY_MODE	0	Writing 1 to this configures I^{2}S controller to Primary Mode. Writing 0 to this configures I^{2}S controller to Secondary Mode.
22:0	-	Reserved	-	It is recommended to retain the contents by using read/modify write to this register.

6.12.18.15 DYN_CLK_GATE_DISABLE_REG

Table 6.38. MCUHP_DYN_CLK_GATE_DISABLE_REG

Bit	Access	Function	Default Value	Description
31	R/W	MCUHP_MISC_CONFIG_PCLK_DYN_CTRL_DISABLE	1'b0	Writing 1 to this disables dynamic clock gating. Writing 0 to this enables dynamic clock gating.
30	R/W	MCUHP_DCACHE_DYN_GATING_DISABLE	1'b0	Writing 1 to this disables dynamic clock gating. Writing 0 to this enables dynamic clock gating.
29	R/W	MCUHP_ICACHE_DYN_GATING_DISABLE	1'b0	Writing 1 to this disables dynamic clock gating. Writing 0 to this enables dynamic clock gating.
28	R/W	MCUHP_SSI_MST_PCLK_DYN_CTRL_DISABLE	1'b0	Writing 1 to this disables dynamic clock gating. Writing 0 to this enables dynamic clock gating.
27	R/W	Reserved.	1'b0	Reserved.
26	R/W	Reserved.	1'b0	Reserved.
25	R/W	MCUHP_MVP_DYN_GATING_DISABLE	1'b0	Writing 1 to this disables dynamic clock gating. Writing 0 to this enables dynamic clock gating.
24	R/W	MCUHP_MEM_CLK_ULP_DYN_CTRL_DISABLE	1'b0	Writing 1 to this disables dynamic clock gating. Writing 0 to this enables dynamic clock gating.
23	R/W	Reserved.	1'b0	Reserved.
22	R/W	Reserved.	1'b0	Reserved.

Bit	Access	Function	Default Value	Description
21	R/W	MCUHP_SSI_MST_SCLK_DYN_CTRL_DISABLE	1'b0	Writing 1 to this disables dynamic clock gating. Writing 0 to this enables dynamic clock gating.
20	R/W	MCUHP_ARM_CLK_DYN_CTRL_DISABLE	1'b0	Writing 1 to this disables dynamic clock gating. Writing 0 to this enables dynamic clock gating.
19	R/W	MCUHP_SEMAPHORE_CLK_DYN_CTRL_DISABLE	1'b0	Writing 1 to this disables dynamic clock gating. Writing 0 to this enables dynamic clock gating.
18	R/W	Reserved.	1'b0	Reserved.
17	R/W	RESERVED	1'b0	Reserved
16	R/W	MCUHP_SSI_SLV_PCLK_DYN_CTRL_DISABLE	1'b 0	Writing 1 to this disables dynamic clock gating. Writing 0 to this enables dynamic clock gating.
15	R/W	MCUHP_SSI_SLV_SCLK_DYN_CTRL_DISABLE	1'b 0	Writing 1 to this disables dynamic clock gating. Writing 0 to this enables dynamic clock gating.
14	R/W	Reserved.	1'b 1	Reserved.
13	R/W	Reserved.	1'b 1	Reserved.
12	R/W	MCUHP_UART1_PCLK_DYN_CTRL_DISABLE	1'b 0	Writing 1 to this disables dynamic clock gating. Writing 0 to this enables dynamic clock gating.
11	R/W	MCUHP_UART1_SCLK_DYN_CTRL_DISABLE	1'b 0	Writing 1 to this disables dynamic clock gating. Writing 0 to this enables dynamic clock gating.

Bit	Access	Function	Default Value	Description
10	R/W	MCUHP_USART0_PCLK_DYN_CTRL_DISABLE	1'b 0	Writing 1 to this disables dynamic clock gating. Writing 0 to this enables dynamic clock gating.
9	R/W	MCUHP_USART0_SCLK_DYN_CTRL_DISABLE	1'b 0	Writing 1 to this disables dynamic clock gating. Writing 0 to this enables dynamic clock gating.
8	R/W	Reserved.	1'b 0	Reserved.
7	R/W	MCUHP_TOT_CLK_DYN_CTRL_DISABLE	1'b 0	Writing 1 to this disables dynamic clock gating. Writing 0 to this enables dynamic clock gating.
6	R/W	MCUHP_SGPIO_PCLK_DYN_CTRL_DISABLE	1'b 0	Writing 1 to this disables dynamic clock gating. Writing 0 to this enables dynamic clock gating.
5	R/W	MCUHP_EGPIO_PCLK_DYN_CTRL_DISABLE	1'b 0	Writing 1 to this disables dynamic clock gating. Writing 0 to this enables dynamic clock gating.
4	R/W	MCUHP_GPDMA_HCLK_DYN_CTRL_DISABLE	1'b 0	Writing 1 to this disables dynamic clock gating. Writing 0 to this enables dynamic clock gating.
3	R/W	Reserved.	1'b 0	Reserved.
2	R/W	Reserved.	1'b 0	Reserved.
1	R/W	MCUHP_BUS_CLK_DYN_CTRL_DISABLE	1'b 0	Writing 1 to this disables dynamic clock gating. Writing 0 to this enables dynamic clock gating.

Bit	Access	Function	Default Value	Description
0	R/W	MCUHP_SDIO_SYS_HCLK_DYN_CTRL_DISABLE	1'b 0	Writing 1 to this disables dynamic clock gating. Writing 0 to this enables dynamic clock gating.

6.12.18.16 PLL_ENABLE_SET_REG

Table 6.39. MCUHP_PLL_ENABLE_SET_REG

Bit	Access	Function	Default Value	Description
0	R/W	MCUHP_SOCPLL_SPI_SW_RESET	1'b 0	Writing 1 to this enables SPI soft reset for SoC PLL Writing 0 is Invalid

6.12.18.17 PLL_ENABLE_CLEAR_REG

Table 6.40. MCUHP_PLL_ENABLE_CLEAR_REG

Bit	Access	Function	Default Value	Description
0	R/W	MCUHP_SOCPLL_SPI_SW_RESET	1'b 0	Writing 1 to this disables SPI soft reset for SoC PLL Writing 0 is Invalid

6.12.18.18 PLL_STAT_REG

Table 6.41. MCUHP_PLL_STATUS_REG

Bit	Access	Function	Default Value	Description
31	R	MCUHP_ULP_REF_CLK_SWITCHED	1'b1	Status of Dynamic Clock Mux in Reference Clock Generation 1 indicates Clock got switched and output clock can be used 0 indicates Clock switching is in progress
30	R	MCUHP_CLK_FREE_OR_SLP_SWITCHED	1'b1	Status of Dynamic Clock Mux in Reference Clock Generation 1 indicates Clock got switched and output clock can be used 0 indicates Clock switching is in progress
29	R	MCUHP_TASS_M4SS_64K0_CLK_SWITCHED	1'b1	Status of Dynamic Clock Mux in Reference Clock Generation 1 indicates Clock got switched and output clock can be used 0 indicates Clock switching is in progress
28	R	MCUHP_UART1_CLK_SWITCHED	1'b1	Status of Dynamic Clock Mux in Reference Clock Generation 1 indicates Clock got switched and output clock can be used 0 indicates Clock switching is in progress
27	R	MCUHP_USART0_CLK_SWITCHED	1'b1	Status of Dynamic Clock Mux in Reference Clock Generation 1 indicates Clock got switched and output clock can be used 0 indicates Clock switching is in progress
26	R	MCUHP_TASS_M4SS_192K_CLK_SWITCHED	1'b1	Status of Dynamic Clock Mux in Reference Clock Generation 1 indicates Clock got switched and output clock can be used 0 indicates Clock switching is in progress
25	R	MCUHP_CC_CLOCK_MUX_SWITCHED	1'b1	Status of Dynamic Clock Mux in Reference Clock Generation 1 indicates Clock got switched and output clock can be used 0 indicates Clock switching is in progress
24	R	MCUHP_TASSITCHED	1'b1	Status of Dynamic Clock Mux in Reference Clock Generation 1 indicates Clock got switched and output clock can be used 0 indicates Clock switching is in progress
23	R	MCUHP_QSPI2_CLK_SWITCHED	1'b1	Status of Dynamic Clock Mux in Reference Clock Generation 1 indicates Clock got switched and output clock can be used 0 indicates Clock switching is in progress
22	R	MCUHP_EXT_CLK_SWITCHED	1'b1	Status of Dynamic Clock Mux in Reference Clock Generation 1 indicates Clock got switched and output clock can be used 0 indicates Clock switching is in progress

Bit	Access	Function	Default Value	Description
21	R	MCUHP_SLEEP_CLK_SWITCHED	1'b1	Status of Dynamic Clock Mux in Reference Clock Generation 1 indicates Clock got switched and output clock can be used 0 indicates Clock switching is in progress
20	R	Reserved.	NA	Reserved.
19	R	Reserved.	NA	Reserved.
18	R	MCUHP_PLL_INTF_CLK_SWITCHED	1'b1	Status of Dynamic Clock Mux in Reference Clock Generation 1 indicates Clock got switched and output clock can be used 0 indicates Clock switching is in progress
17	R	MCUHP_I2S_CLK_SWITCHED	1'b1	Status of Dynamic Clock Mux in Reference Clock Generation 1 indicates Clock got switched and output clock can be used 0 indicates Clock switching is in progress
16	R	MCUHP_M4_TA_SOC_CLK_SWITCHED_SDIO	1'b1	Status of Dynamic Clock Mux in Reference Clock Generation 1 indicates Clock got switched and output clock can be used 0 indicates Clock switching is in progress
15	R	MCUHP_SCT_CLK_SWITCHED	1'b1	Status of Dynamic Clock Mux in Reference Clock Generation 1 indicates Clock got switched and output clock can be used 0 indicates Clock switching is in progress
14	R	Reserved.	1'b1	Reserved.
13	R	MCUHP_SSI_MST_SCLK_SWITCHED	1'b1	Status of Dynamic Clock Mux in Reference Clock Generation 1 indicates Clock got switched and output clock can be used 0 indicates Clock switching is in progress
12	R	MCUHP_GSPI_SCLK_SWITCHED	1'b1	Status of Dynamic Clock Mux in Reference Clock Generation 1 indicates Clock got switched and output clock can be used 0 indicates Clock switching is in progress
11	R	MCUHP_UART1_SCLK_SWITCHED	1'b1	Status of Dynamic Clock Mux in Reference Clock Generation 1 indicates Clock got switched and output clock can be used 0 indicates Clock switching is in progress
10	R	MCUHP_USART0_SCLK_SWITCHED	1'b1	Status of Dynamic Clock Mux in Reference Clock Generation 1 indicates Clock got switched and output clock can be used 0 indicates Clock switching is in progress
9	R	MCUHP_QSPI_CLK_SWITCHED	1'b1	Status of Dynamic Clock Mux in Reference Clock Generation 1 indicates Clock got switched and output clock can be used 0 indicates Clock switching is in progress

Bit	Access	Function	Default Value	Description
8	R	MCUHP_M4_SOC_CLK_SWITCHED	1'b1	Status of Dynamic Clock Mux in Reference Clock Generation 1 indicates Clock got switched and output clock can be used 0 indicates Clock switching is in progress
7	R	MCUHP_PLL_LOCK_DATA_TRIG	1'b0	1 indicates PLL Lock Statuses are equal to MCUHP_PLL_LOCK_INT_DATA_REG
6	R	MCUHP_MODEMPLL_LOCK	1'b1	Indicates PLL lock status
5	R	MCUHP_SOCPLL_LOCK	1'b1	Indicates PLL lock status
4	R	MCUHP_I2SPLL_LOCK	1'b1	Indicates PLL lock status
3	R	MCUHP_INTFPLL_LOCK	1'b1	Indicates PLL lock status
2	R	MCUHP_APPLL_LOCK	1'b0	Indicates PLL lock status
1	R	MCUHP_DDRPLL_LOCK	1'b0	Indicates PLL lock status
0	R	MCUHP_LCDPLL_LOCK	1'b0	Indicates PLL lock status

6.12.18.19 PLL_LOCK_INT_MASK_REG

Table 6.42. MCUHP_PLL_LOCK_INT_MASK_REG

Bit	Access	Function	Default Value	Description
15	R/W	MCUHP_PLL_LOCK_DATA_TRIG_MASK_FE	1'b1	Writing 1 to this masks the Falling edge PLL Lock Interrupt. Writing 0 to this unmasks the Falling edge PLL Lock Interrupt.
14	R/W	MCUHP_MODEM_PLL_LOCK_MASK_FE	1'b1	Writing 1 to this masks the Falling edge PLL Lock Interrupt. Writing 0 to this unmasks the Falling edge PLL Lock Interrupt.
13	R/W	MCUHP_SOC_PLL_LOCK_MASK_FE	1'b1	Writing 1 to this masks the Falling edge PLL Lock Interrupt. Writing 0 to this unmasks the Falling edge PLL Lock Interrupt.
12	R/W	MCUHP_I2S_PLL_LOCK_MASK_FE	1'b1	Writing 1 to this masks the Falling edge PLL Lock Interrupt. Writing 0 to this unmasks the Falling edge PLL Lock Interrupt.
11	R/W	MCUHP_INTF_PLL_LOCK_MASK_FE	1'b1	Writing 1 to this masks the Falling edge PLL Lock Interrupt. Writing 0 to this unmasks the Falling edge PLL Lock Interrupt.
10	R/W	MCUHP_AP_PLL_LOCK_MASK_FE	1'b1	Writing 1 to this masks the Falling edge PLL Lock Interrupt. Writing 0 to this unmasks the Falling edge PLL Lock Interrupt.
9	R/W	MCUHP_DDR_PLL_LOCK_MASK_FE	1'b1	Writing 1 to this masks the Falling edge PLL Lock Interrupt. Writing 0 to this unmasks the Falling edge PLL Lock Interrupt.

Bit	Access	Function	Default Value	Description
8	R/W	MCUHP_LCD_PLL_LOCK_MASK_FE	1'b1	Writing 1 to this masks the Falling edge PLL Lock Interrupt. Writing 0 to this unmasks the Falling edge PLL Lock Interrupt.
7	R/W	MCUHP_PLL_LOCK_DATA_TRIG_MASK_RE	1'b1	Writing 1 to this masks the Rising edge PLL Lock Interrupt. . Writing 0 to this unmasks the Rising edge PLL Lock Interrupt.
6	R/W	MCUHP_MODEM_PLL_LOCK_MASK_RE	1'b1	Writing 1 to this masks the Rising edge PLL Lock Interrupt. . Writing 0 to this unmasks the Rising edge PLL Lock Interrupt.
5	R/W	MCUHP_SOC_PLL_LOCK_MASK_RE	1'b1	Writing 1 to this masks the Rising edge PLL Lock Interrupt. . Writing 0 to this unmasks the Rising edge PLL Lock Interrupt.
4	R/W	MCUHP_I2S_PLL_LOCK_MASK_RE	1'b1	Writing 1 to this masks the Rising edge PLL Lock Interrupt. . Writing 0 to this unmasks the Rising edge PLL Lock Interrupt.
3	R/W	MCUHP_INTF_PLL_LOCK_MASK_RE	1'b1	Writing 1 to this masks the Rising edge PLL Lock Interrupt. . Writing 0 to this unmasks the Rising edge PLL Lock Interrupt.
2	R/W	MCUHP_AP_PLL_LOCK_MASK_RE	1'b1	Writing 1 to this masks the Rising edge PLL Lock Interrupt. . Writing 0 to this unmasks the Rising edge PLL Lock Interrupt.
1	R/W	MCUHP_DDR_PLL_LOCK_MASK_RE	1'b1	Writing 1 to this masks the Rising edge PLL Lock Interrupt. . Writing 0 to this unmasks the Rising edge PLL Lock Interrupt.

Bit	Access	Function	Default Value	Description
0	R/W	MCUHP_LCD_PLL_LOCK_MASK_RE	1'b1	Writing 1 to this masks the Rising edge PLL Lock Interrupt. . Writing 0 to this unmasks the Rising edge PLL Lock Interrupt.

6.12.18.20 PLL_LOCK_INT_CLR_REG

Table 6.43. MCUHP_PLL_LOCK_INT_CLR_REG

Bit	Access	Function	Default Value	Description
15	R/W	MCUHP_PLL_LOCK_DATA_TRIG_CLEAR_FE	1'b0	Writing 1 to this clears the Falling edge PLL Lock Interrupt. Writing 0 to this does not clear the Falling edge PLL Lock Interrupt.
14	R/W	MCUHP_MODEM_PLL_LOCK_CLEAR_FE	1'b0	Writing 1 to this clears the Falling edge PLL Lock Interrupt. Writing 0 to this does not clear the Falling edge PLL Lock Interrupt.
13	R/W	MCUHP_SOC_PLL_LOCK_CLEAR_FE	1'b0	Writing 1 to this clears the Falling edge PLL Lock Interrupt. Writing 0 to this does not clear the Falling edge PLL Lock Interrupt.
12	R/W	MCUHP_I2S_PLL_LOCK_CLEAR_FE	1'b0	Writing 1 to this clears the Falling edge PLL Lock Interrupt. Writing 0 to this does not clear the Falling edge PLL Lock Interrupt.
11	R/W	MCUHP_INTF_PLL_LOCK_CLEAR_FE	1'b0	Writing 1 to this clears the Falling edge PLL Lock Interrupt. Writing 0 to this does not clear the Falling edge PLL Lock Interrupt.
10	R/W	MCUHP_AP_PLL_LOCK_CLEAR_FE	1'b0	Writing 1 to this clears the Falling edge PLL Lock Interrupt. Writing 0 to this does not clear the Falling edge PLL Lock Interrupt.
9	R/W	MCUHP_DDR_PLL_LOCK_CLEAR_FE	1'b0	Writing 1 to this clears the Falling edge PLL Lock Interrupt. Writing 0 to this does not clear the Falling edge PLL Lock Interrupt.
8	R/W	MCUHP_LCD_PLL_LOCK_CLEAR_FE	1'b0	Writing 1 to this clears the Falling edge PLL Lock Interrupt. Writing 0 to this does not clear the Falling edge PLL Lock Interrupt.

Bit	Access	Function	Default Value	Description
7	R/W	MCUHP_PLL_LOCK_DATA_TRIG_CLEAR_RE	1'b0	Writing 1 to this clears the Rising edge PLL Lock Interrupt. Writing 0 to this does not clear the Rising edge PLL Lock Interrupt.
6	R/W	MCUHP_MODEM_PLL_LOCK_CLEAR_RE	1'b0	Writing 1 to this clears the Rising edge PLL Lock Interrupt. Writing 0 to this does not clear the Rising edge PLL Lock Interrupt.
5	R/W	MCUHP_SOC_PLL_LOCK_CLEAR_RE	1'b0	Writing 1 to this clears the Rising edge PLL Lock Interrupt. Writing 0 to this does not clear the Rising edge PLL Lock Interrupt.
4	R/W	MCUHP_I2S_PLL_LOCK_CLEAR_RE	1'b0	Writing 1 to this clears the Rising edge PLL Lock Interrupt. Writing 0 to this does not clear the Rising edge PLL Lock Interrupt.
3	R/W	MCUHP_INTF_PLL_LOCK_CLEAR_RE	1'b0	Writing 1 to this clears the Rising edge PLL Lock Interrupt. Writing 0 to this does not clear the Rising edge PLL Lock Interrupt.
2	R/W	MCUHP_AP_PLL_LOCK_CLEAR_RE	1'b0	Writing 1 to this clears the Rising edge PLL Lock Interrupt. Writing 0 to this does not clear the Rising edge PLL Lock Interrupt.
1	R/W	MCUHP_DDR_PLL_LOCK_CLEAR_RE	1'b0	Writing 1 to this clears the Rising edge PLL Lock Interrupt. Writing 0 to this does not clear the Rising edge PLL Lock Interrupt.
0	R/W	MCUHP_LCD_PLL_LOCK_CLEAR_RE	1'b0	Writing 1 to this clears the Rising edge PLL Lock Interrupt. Writing 0 to this does not clear the Rising edge PLL Lock Interrupt.

6.12.18.21 PLL_LOCK_INT_DATA_REG

Table 6.44. PLL_LOCK_INT_DATA_REG

Bit	Access	Function	Default Value	Description
6	R/W	MCUHP_MODEM_PLL_LOCK	1'b0	1 indicates Modem PLL Lock is used as trigger. 0 indicates Modem PLL Lock is not used as trigger.
5	R/W	MCUHP_SOC_PLL_LOCK	1'b0	1 indicates SoC PLL Lock is used as trigger. 0 indicates SoC PLL Lock is not used as trigger
4	R/W	MCUHP_I2S_PLL_LOCK	1'b0	1 indicates I2S PLL Lock is used as trigger. 0 indicates I2S PLL Lock is not used as trigger
3	R/W	MCUHP_INTF_PLL_LOCK	1'b0	1 indicates INTF PLL Lock is used as trigger. 0 indicates INTF PLL Lock is not used as trigger
2	R/W	MCUHP_AP_PLL_LOCK	1'b0	1 indicates AP PLL Lock is used as trigger. 0 indicates Ap PLL Lock is not used as trigger
1	R/W	MCUHP_DDR_PLL_LOCK	1'b0	1 indicates DDR PLL Lock is used as trigger. 0 indicates DDR PLL Lock is not used as trigger
0	R/W	MCUHP_LCD_PLL_LOCK	1'b0	1 indicates LCD PLL Lock is used as trigger. 0 indicates LCD PLL Lock is not used as trigger

6.12.18.22 SLEEP_CALIB_REG

Table 6.45. MCUHP_SLEEP_CALIB_REG

Bit	Access	Function	Default Value	Description
19	R	MCUHP_SLP_CALIB_DONE	NA	Indicates the end of calibration
18:3	R	MCUHP_SLP_CALIB_DURATION	NA	Specifies the number of processor clock cycles present in one sleep clock cycle.
2:1	R/W	MCUHP_SLP_CALIB_CYCLES	2'd0	These bits are used to program the number of clock cycles over which clock calibration is to be done. By writing 0 to this calibration is done over 1 clock cycle. By writing 1 to this calibration is done over 2 clock cycles. By writing 2 to this calibration is done over 3 clock cycles. By writing 3 to this calibration is done over 4 clock cycles.
0	R/W	MCUHP_SLP_CALIB_START	1'b 0	This bit is used to start the calibration. This bit is self-clearing. Writing 1 to this initiates calibration. When read, if high indicates the completion of calibration process.

6.12.18.23 CLK_CALIB_CTRL_REG1

Table 6.46. MCUHP_CLK_CALIB_CTRL_REG1

Bit	Access	Function	Default Value	Description
6:3	R/W	MCUHP_CC_CLKIN_SEL	4'd0	Writing a value to this selects the corresponding clock to be calibrated 0 indicates ulp_ref_clk is selected 1 indicates mems_ref_clk is selected 2 indicates ulp_20mhz_ringosc_clk is selected 3 indicates modem_pll_clk1 is selected 4 indicates modem_pll_clk2 is selected 5 indicates intf_pll_clk is selected 6 indicates soc_pll_clk is selected 7 indicates i2s_pll_clk is selected 8 indicates sleep_clk is selected 9 indicates bus_clkby2_apss2m4ss_sram is selected 10 indicates ipmu_testpin_in is selected Values 11 to 14 are Invalid. 15 indicates the clock is Gated.
2	R/W	MCUHP_CC_CHANGE_TEST_CLK	1'b0	Writing 1 to this changes the test clock. Writing 0 to this does not change the test clock
1	R/W	MCUHP_CC_START	1'b0	Writing 1 to this initiates the clock calibration
0	R/W	MCUHP_CC_SOFT_RST	1'b0	Soft Reset for clock calibrator. 1 indicates soft reset is enabled. 0 indicates soft reset is disabled.

6.12.18.24 CLK_CALIB_CTRL_REG2

Table 6.47. MCUHP_CLK_CALIB_CTRL_REG2

Bit	Access	Function	Default Value	Description
31:0	R/W	MCUHP_CC_NUM_REF_CLKS	32'b 0	Specifies numbers of clocks to be considered for calibration

6.12.18.25 CLK_CALIB_STS_REG1

Table 6.48. MCUHP_CLK_CALIB_STS_REG1

Bit	Access	Function	Default Value	Description
1	R	MCUHP_CC_ERROR	1'b 0	1 indicates that there is an error in Clock Calibration 0 indicates that there is no error in Clock Calibration
0	R	MCUHP_CC_DONE	1'b 0	1 indicates that Clock Calibration is done. 0 indicates that Clock Calibration is not done.

6.12.18.26 CLK_CALIB_STS_REG2

Table 6.49. MCUHP_CLK_CALIB_STS_REG2

Bit	Access	Function	Default Value	Description
31:0	R	MCUHP_CC_NUM_TEST_CLKS	32'd0	specifies number of test clock cycles present in specified number of ref_clk cycles

6.12.18.27 CLK_CONFIG_REG6

Table 6.50. MCUHP_CLK_CONFIG_REG6

Bit	Access	Function	Default Value	Description
19	R/W	MCUHP_QSPI2_ODD_DIV_SEL	0	Clock Select for clock swallow 0 - 50% odd clock divider output is selected with division factor MCUHP_QSPI2_CLK_DIV_FAC. 1 - 50% even clock divider output or swallowed is selected with division with division factor MCUHP_QSPI2_CLK_DIV_FAC based on MCUHP_QSPI2_CLK_SWALLOW_SEL
18	R	Reserved	0	Reserved
17:12	R/W	MCUHP_QSPI2_CLK_DIV_FAC	1	Specifies the clock division factor for SPI PSRAM Controller. 1) when qspi_clk_swallow_sel is 1'b1 and qspi_odd_div_sel is 1'b0 output clock is a swallowed clock with the following frequency. 6'h0,6'h1 => clk_out = clk_in. >6'h1 => clk_out = clk_in/ qspi_clk_div_fac 2) when qspi_clk_swallow_sel is 1'b0 and qspi_odd_div_sel is 1'b0, output clock is a 50% duty cycle clock with the following frequency. 6'h0 => clk_out = clk_in/2. >6'h0 => clk_out = clk_in/ (2*qspi_clk_div_fac) 3) When qspi_odd_div_sel is 1'b1, output clock is a 50% duty cycle clock with the following frequency. when qspi_clk_div_fac is an odd number >=3, clk_out = clk_in/qspi_clk_div_fac. else output clock is gated

Bit	Access	Function	Default Value	Description
11:9	R/W	MCUHP_QSPI2_CLK_SEL	0	Specifies the clock source for SPI PSRAM controller when independent clock source w.r.t Processor is selected. 0 - MCU-HP Reference Clock (See Table 6.41 MCUHP_PLL_STATUS_REG on page 98.) 1 - Interface-PLL Clock 2 - Reserved 3 - SoC-PLL Clock 4, 5, 6, 7 - Output Clock is gated
8:5	R/W	MCUHP_PADCFG_PCLK_DIV_FAC	4'd4	Specifies Clock division factor for pclk_pad_config_m4ss. output clock is a swallowed clock with the following frequency. 6'h0,6'h1 => clk_out = clk_in. >6'h1 => clk_out = clk_in/ padcfg_pclk_div_fac
4:3	R/W	Reserved.	2'd1	Reserved.
2:0	R/W	MCUHP_IID_KH_CLK_DIV_FAC	3'd0	Specifies Clock division factor for iid_clk. output clock is a swallowed clock with the following frequency. 6'h0,6'h1 => clk_out = clk_in. >6'h1 => clk_out = clk_in/iid_kh_clk_div_fac

6.12.18.28 DYN_CLK_GATE_DISABLE_REG2

Table 6.51. MCUHP_DYN_CLK_GATE_DISABLE_REG2

Bit	Access	Function	Default Value	Description
8	R/W	Reserved.	1'b1	Reserved.
7	R/W	MCUHP_EFUSE_PCLK_DYN_CTRL_DISABLE	1'b0	Dynamic clock gate disable control efuse pclk Writing 1 to this disables dynamic clock gating. Writing 0 to this enables dynamic clock gating.
6	R/W	MCUHP_EFUSE_CLK_DYN_CTRL_DISABLE	1'b0	Dynamic clock gate disable control efuse clk Writing 1 to this disables dynamic clock gating. Writing 0 to this enables dynamic clock gating.
5	R/W	MCUHP_I2SM_PCLK_DYN_CTRL_DISABLE	1'b0	Dynamic clock gate disable control I2S Master pclk Writing 1 to this disables dynamic clock gating. Writing 0 to this enables dynamic clock gating.
4	R/W	Reserved.	1'b0	Reserved.
3	R/W	MCUHP_SCT_PCLK_DYN_CTRL_DISABLE	1'b0	Dynamic clock gate disable control SCT pclk Writing 1 to this disables dynamic clock gating. Writing 0 to this enables dynamic clock gating.

Bit	Access	Function	Default Value	Description
2	R/W	MCUHP_I2C_1_BUS_CLK_DYN_CTRL_DISBALE	1'b0	Dynamic clock gate disable control I2C_2 bus clk Writing 1 to this disables dynamic clock gating. Writing 0 to this enables dynamic clock gating.
1	R/W	MCUHP_I2C_BUS_CLK_DYN_CTRL_DISABLE	1'b0	Dynamic clock gate disable control I2C bus clk Writing 1 to this disables dynamic clock gating. Writing 0 to this enables dynamic clock gating.
0	R/W	MCUHP_SOC_PLL_SPI_CLK_DYN_CTRL_DISABLE	1'b0	Dynamic clock gate disable control soc pll spi clk Writing 1 to this disables dynamic clock gating. Writing 0 to this enables dynamic clock gating.

6.12.18.29 PLL_LOCK_INT_STATUS_REG

Table 6.52. MCUHP_PLL_LOCK_INT_STATUS_REG

Bit	Access	Function	Default Value	Description
15	R/W	MCUHP_PLL_LOCK_DATA_TRIG_INTR_FE	1'b0	1 indicates Interrupt is encountered. 0 indicates Interrupt is not encountered.
14	R/W	MCUHP_MODEM_PLL_LOCK_FE	1'b0	1 indicates Interrupt is encountered. 0 indicates Interrupt is not encountered.
13	R/W	MCUHP_SOC_PLL_LOCK_FE	1'b0	1 indicates Interrupt is encountered. 0 indicates Interrupt is not encountered.
12	R/W	MCUHP_I2S_PLL_LOCK_FE	1'b0	1 indicates Interrupt is encountered. 0 indicates Interrupt is not encountered.
11	R/W	MCUHP_INTF_PLL_LOCK_FE	1'b0	1 indicates Interrupt is encountered. 0 indicates Interrupt is not encountered.
10	R/W	MCUHP_AP_PLL_LOCK_FE	1'b0	1 indicates Interrupt is encountered. 0 indicates Interrupt is not encountered.
9	R/W	MCUHP_DDR_PLL_LOCK_FE	1'b0	1 indicates Interrupt is encountered. 0 indicates Interrupt is not encountered.
8	R/W	MCUHP_LCD_PLL_LOCK_FE	1'b0	1 indicates Interrupt is encountered. 0 indicates Interrupt is not encountered.
7	R/W	MCUHP_PLL_LOCK_DATA_TRIG_INTR_RE	1'b0	1 indicates Interrupt is encountered. 0 indicates Interrupt is not encountered.
6	R/W	MCUHP_MODEM_PLL_LOCK_RE	1'b0	1 indicates Interrupt is encountered. 0 indicates Interrupt is not encountered.

Bit	Access	Function	Default Value	Description
5	R/W	MCUHP_SOC_PLL_LOCK_RE	1'b0	1 indicates Interrupt is encountered. 0 indicates Interrupt is not encountered.
4	R/W	MCUHP_I2S_PLL_LOCK_RE	1'b0	1 indicates Interrupt is encountered. 0 indicates Interrupt is not encountered.
3	R/W	MCUHP_INTF_PLL_LOCK_RE	1'b0	1 indicates Interrupt is encountered. 0 indicates Interrupt is not encountered.
2	R/W	MCUHP_AP_PLL_LOCK_RE	1'b0	1 indicates Interrupt is encountered. 0 indicates Interrupt is not encountered.
1	R/W	MCUHP_DDR_PLL_LOCK_RE	1'b0	1 indicates Interrupt is encountered. 0 indicates Interrupt is not encountered.
0	R/W	MCUHP_LCD_PLL_LOCK_RE	1'b0	1 indicates Interrupt is encountered. 0 indicates Interrupt is not encountered.

6.12.18.30 MCUHP_SLEEP_CALIB_REG2

Table 6.53. MCUHP_SLEEP_CALIB_REG2

Bit	Access	Function	Default Value	Description
31 : 16	R	Reserved	NA	-
15 : 0	R/W	MCUHP_SLEEP_PULSE_DURATION	16'b0	The duration of sleep clock pulse in terms of reference clock

6.13 MCU ULP Clock Architecture

6.13.1 General Description

The MCU ULP (Ultra Low Power) domain contains the MCU ULP AHB Inter-Connect-Matrix, MCU ULP Peripherals and the direct AHB Interface with the Processor. This section describes the different clock sources possible for each peripheral and the AHB/APB Interfaces.

6.13.2 Features

- The clock sources used for MCU ULP domain includes RC, RO, XTAL clocks in addition to the PLL generated high frequency clocks.
- The SoC-PLL can be used as a clock source for MCU ULP Processing Interface.

  The I2S-PLL can be used for ULP-I^{2}S peripheral.
- The clocks from High-Frequency PLL's are valid only in PS4/PS3 state which are described in Power Architecture section.
- The XTAL Clock is valid only in PS4/PS3 state which are described in Power Architecture section.
- The frequency and clock source for Peripherals can be configured independently of the MCU ULP AHB clock.
- A clock source from MCU-HP Domain which is a divided version of Processor clock can be used for MCU ULP AHB Clock and Peripherals.
- The UULP Vbat Peripherals are configured by the processor through the MCU ULP APB Interface.

6.13.3 Functional Description

The following figure depicts the clock sources used for the functionality present in MCU ULP domain.

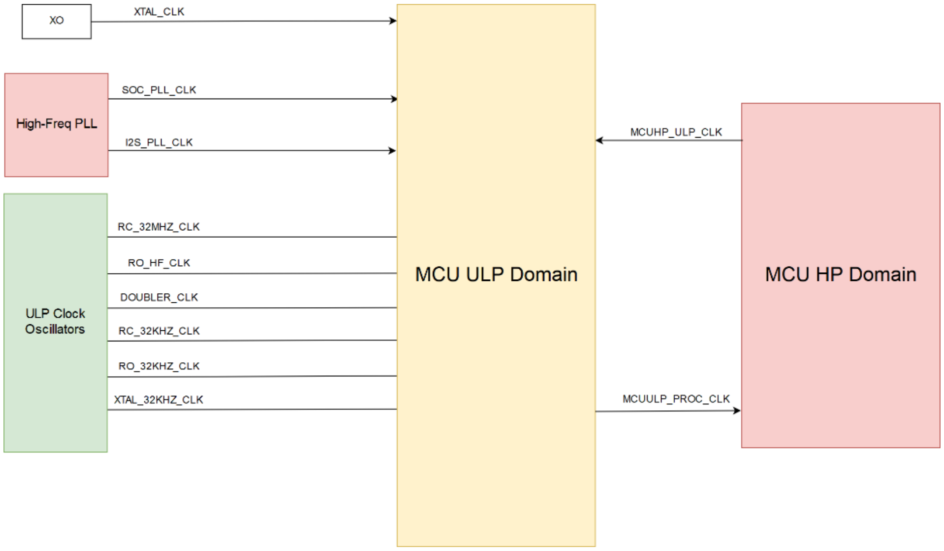


Figure 6.22. MCU-ULP Clocking Scheme Overview

The clock to the following blocks can be configured independently.

1. MCU-ULP AHB Clock
2. I^{2}S in Master Mode
3. SPI/SSI Master
4. UART
5. Timer
6. Aux-ADC/DAC Controller
7. UULP APB Clock

The following blocks operate using the MCU ULP AHB clock.

- UDMA
- I^{2}C

The following sections describe the clock architecture for each of the functionality mentioned above. The reference clock generated for MCU-ULP domain (MCUULP_REF_CLK) is used in generation of the clocks for different peripherals/modules.

6.13.4 AHB Interface Clock

The MCU ULP AHB ICM, MCU ULP AHB Bridge and the MCU ULP APB Interfaces operate using this clock. The clock from MCU-HP Domain (MCUHP_ULP_CLK) is used in generation of this clock. The clock source and frequency for this clock can be configured through MCUULP_PROC_CLK_SEL and MCUULP_PROC_CLK_DIV_DAC in MCUULP_PROC_CLK_CONFIG Register. The Clock switching status can be read through MCUULP_PROC_CLK_SWITCHED in MCUULP_CLK_STATUS_REG Register.

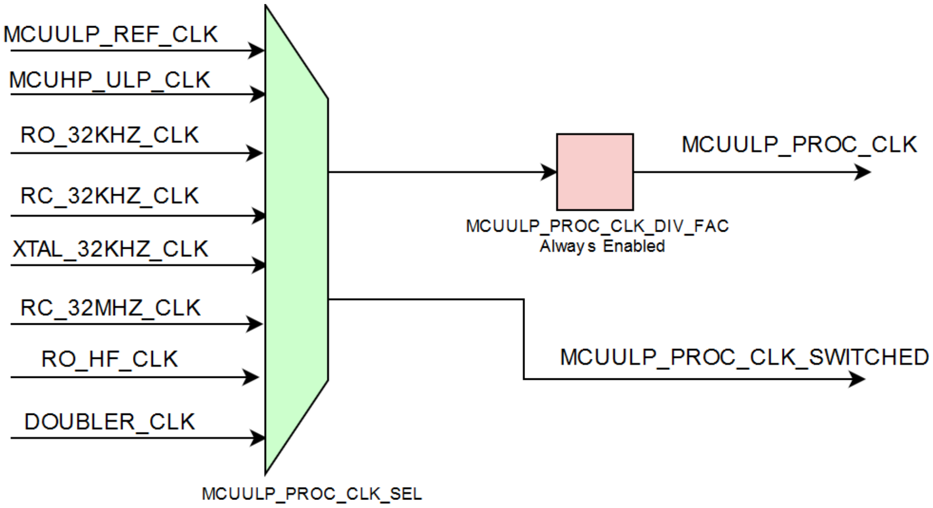


Figure 6.23. MCU-ULP AHB Interface Clock Generation

6.13.5 UART

There are multiple modes of generating the clock for UART Controller. The Clock switching status can be read through MCUHP_UART_CLK_SWITCHED in MCUULP_CLK_STATUS_REG Register.

- Clock Generation
  - Clock source can be configured through MCUULP_UART_CLK_SEL in MCUULP_UART_CLK_CONFIG Register.
  - Clock Division factor can be configured through MCUULP_UART_CLK_DIV_FAC in MCUULP_UART_CLK_CONFIG Register.
  - Divided clock from a Clock swallow or Fractional Divider can be selected through MCUULP_UART_FRAC_CLK_SEL in MCUULP_UART_CLK_CONFIG Register.
- The clocks to the controller can be disabled when not in use for efficient power consumption or before configuring the clock source and frequency. The controller clock and APB clock can be controlled independently.
  - Configure MCUULP_UART_CLK_EN in MCUULP_CLK_EN_REG1 Register for enabling/disabling the Controller clock.
  - Configure MCUULP_UART_APB_CLK_EN in MCUULP_CLK_EN_REG1 Register for enabling/disabling the APB clock.

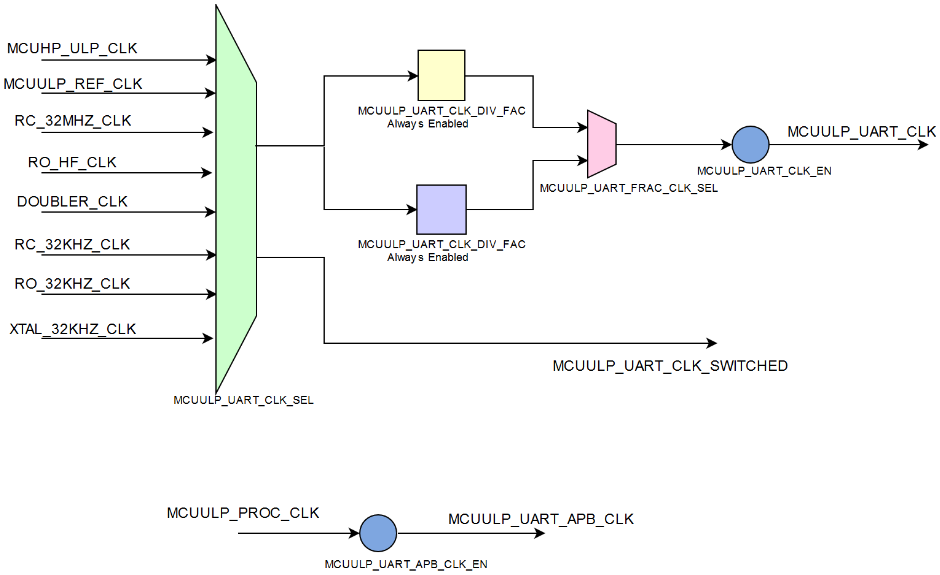


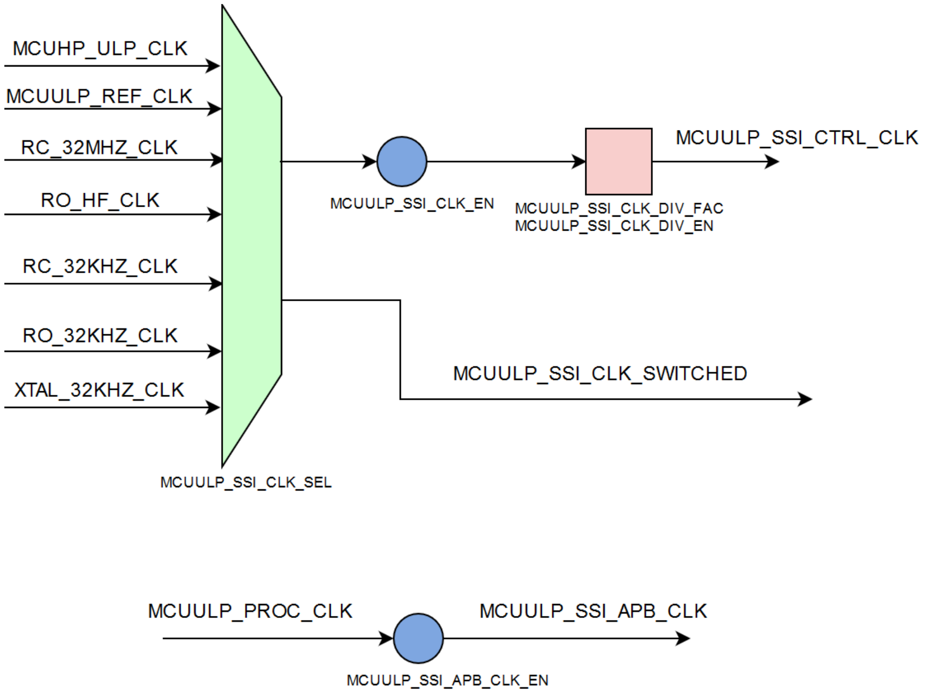
Figure 6.24. MCU-ULP UART Clock Generation

6.13.6 SPI / SSI Primary

There are multiple clock sources for SPI/SSI Primary Controller. The Clock switching status can be read through MCUULP_SSI_CLK_SWITCHED in MCUULP_CLK_STATUS_REG Register.

- Clock Generation
  - Clock source can be configured through MCUULP_SSI_CLK_SEL in MCUULP_I2C_SSI_CLK_CONFIG Register.
  - Clock Division factor can be configured through MCUULP_SSI_CLK_DIV_FAC in MCUULP_I2C_SSI_CLK_CONFIG Register.
  - Clock Divider can be disabled by configuring MCUULP_SSI_CLK_DIV_EN in MCUULP_I2C_SSI_CLK_CONFIG Register.
- The clocks to the controller can be disabled when not in use for efficient power consumption or before configuring the clock source and frequency. The controller clock and APB clock can be controlled independently.
  - Configure MCUULP_SSI_CLK_EN in MCUULP_CLK_EN_REG1 Register for enabling/disabling the Controller clock.
  - Configure MCUULP_SSI_APB_CLK_EN in MCUULP_CLK_EN_REG1 Register for enabling/disabling the APB clock.

Figure 6.25. MCU-ULP SSI Clock Generation



6.13.7 I^{2}S Controller

There are multiple clock sources for I^{2}S Controller which is used in Primary Mode. The Clock switching status can be read through MCUULP_I2S_CLK_SWITCHED in MCUULP_CLK_STATUS_REG Register.

- Clock Generation
  - I^{2}S Secondary clock is derived from the external Primary Device through GPIO PAD's.
  - Clock source can be configured through MCUULP_I2S_CLK_SEL in MCUULP_I2S_CLK_CONFIG Register.
  - Clock Division factor can be configured through MCUULP_I2S_CLK_DIV_FAC in MCUULP_I2S_CLK_CONFIG Register.
  - Clock Divider can be disabled by configuring MCUULP_I2S_CLK_DIV_EN in MCUULP_I2S_CLK_CONFIG Register.
- The clocks to the controller can be disabled when not in use for efficient power consumption or before configuring the clock source and frequency. The controller clock and AHB clock can be controlled independently.
  - Configure MCUULP_I2S_CLK_EN in MCUULP_CLK_EN_REG1 Register for enabling/disabling the Controller clock.
  - Configure MCUULP_I2S_APB_CLK_EN in MCUULP_I2S_CLK_CONFIG Register for enabling/disabling the APB Interface clock.
- The Primary/Secondary mode is configured through MCUULP_I2S_MASTER_SLAVE_MODE in MCUULP_I2S_CLK_CONFIG Register.

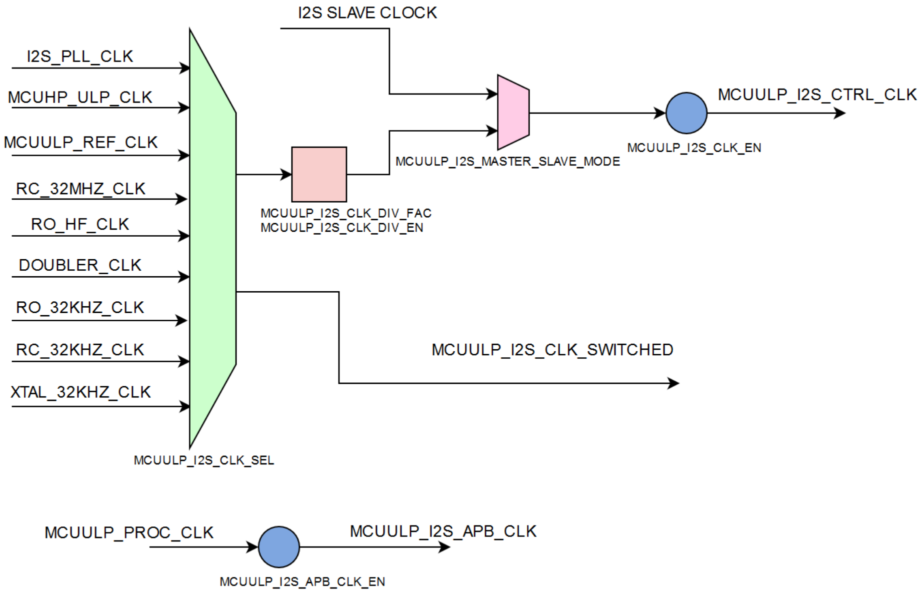


Figure 6.26. I2S Controller Clock Generation

6.13.8 Timer

There are multiple clock sources for Timers. The Clock switching status can be read through MCUULP_TIMER_CLK_SWITCHED in MCUULP_CLK_STATUS_REG Register.

- Clock Generation
  - Clock source can be configured through MCUULP_TIMER_CLK_SEL in MCUULP_TIMER_CLK_CONFIG Register.

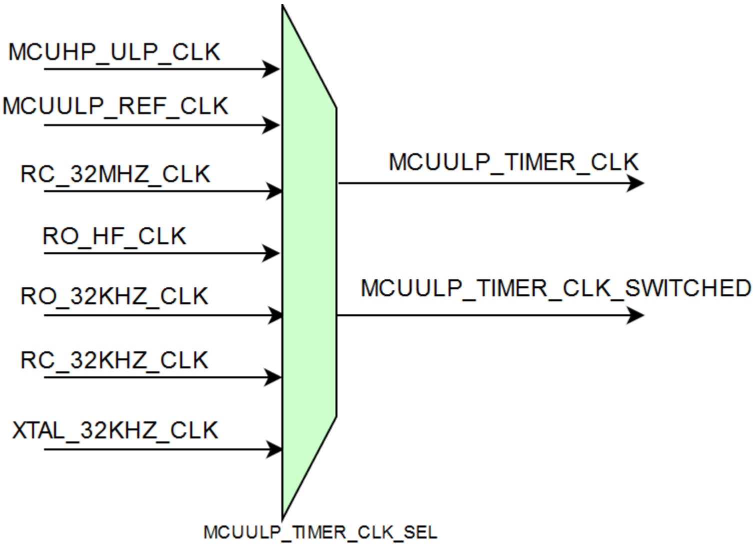


Figure 6.27. MCU-ULP Timer Clock Generation

6.13.9 Aux-ADC/DAC Controller

There are multiple clock sources for Aux-ADC/DAC Controller. The Clock switching status can be read through MCUULP_ADCDAC_CLK_SWITCHED in MCUULP_CLK_STATUS_REG Register.

- Clock Generation
  - Clock source can be configured through MCUULP_ADCDAC_CLK_SEL in MCUULP_ADCDAC_CLK_CONFIG Register.
- The clocks to the controller can be disabled when not in use for efficient power consumption.
  - Configure MCUULP_ADCDAC_CLK_EN in MCUULP_CLK_EN_REG2 Register for enabling/disabling the Controller clock.

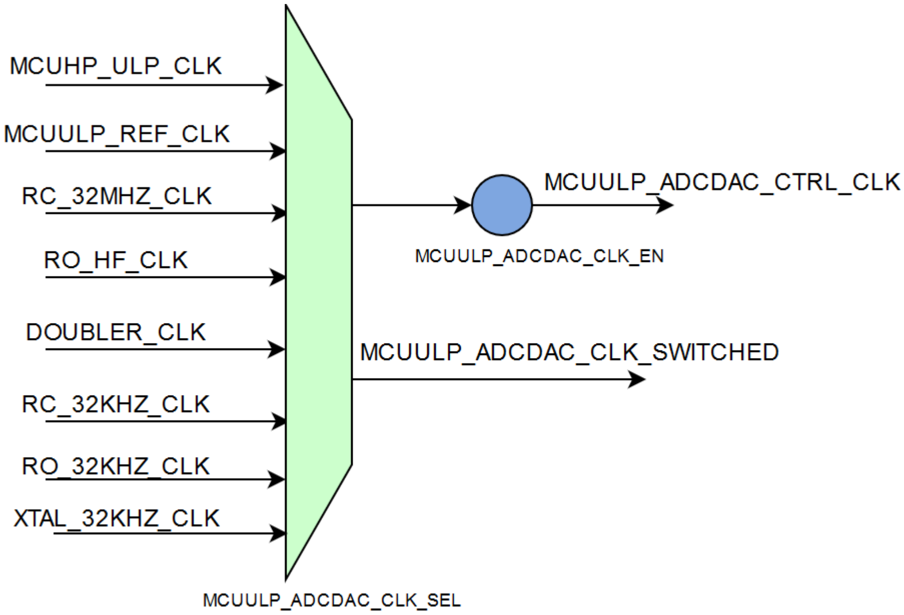


Figure 6.28. MCU-ULP Aux-ADC/DAC Clock Generation

6.13.10 UULP APB Clock

The APB Clock needs to be configured such that the max frequency is 20MHz. This clock is derived from the MCU ULP AHB clock.

- The Clock Division factor can be configured through MCUULP_APB_CLK_DIV_FAC in MCUULP_UULP_CLK_CONFIG Register.

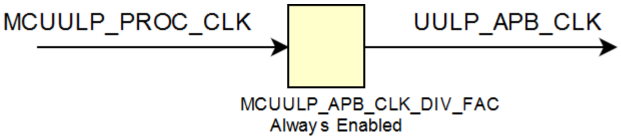


Figure 6.29. MCU-ULP UULP APB Clock Generation

6.13.11 Static Clock Gated Domains

The clock to the domains which operate on the AHB Interface clock can be disabled when not in use for efficient power management. Below mentioned are the programming details for the same.

- **UDMA:** Configure MCUULP_UDMA_CLK_EN in MCUULP_CLK_EN_REG2 Register for enabling/disabling the clock to Micro-DMA module.
- **I^{2}C:**
    - Configure MCUULP_I2C_APB_CLK_EN in MCUULP_CLK_EN_REG1 Register for enabling/disabling the clock to I^{2}C APB Interface.
    - Configure MCUULP_I2C_CLK_EN in MCUULP_I2C_SSI_CLK_CONFIG Register for enabling/disabling the clock to I^{2}C Controller.
- **Enhanced-GPIO:** Configure MCUULP_EGPIO_CLK_EN in MCUULP_CLK_EN_REG1 Register for enabling/disabling the clock to Enhanced-GPIO module.

6.13.12 Register Summary

Base_address = 0x2404_1400

Table 6.54. List of Registers

Register Name	Offset	Description
Section 6.13.13.1 MCUULP_CLK_EN_REG1	0x00	MCU-ULP Clock Enable Register 1
Section 6.13.13.2 ULP_TA_PERI_ISO_REG	0x04	
Section 6.13.13.3 ULP_TA_PERI_RESET_REG	0x08	
Section 6.13.13.4 ULPSS_SPARE_REG	0x0C	
Reserved	0x10	
Section 6.13.13.5 MCUULP_PROC_CLK_CONFIG	0x14	MCU-ULP AHB Clock Configuration Register
Section 6.13.13.6 MCUULP_I2C_SSI_CLK_CONFIG	0x18	MCU-ULP SSI Master and I2C Clock Configuration Register
Section 6.13.13.7 MCUULP_I2S_CLK_CONFIG	0x1C	MCU-ULP I^{2}S Clock Configuration Register
Section 6.13.13.8 MCUULP_UART_CLK_CONFIG	0x20	MCU-ULP UART Clock Configuration Register
Section 6.13.13.9 M4LP_CTRL_REG	0x24	
Section 6.13.13.10 MCUULP_CLK_STATUS_REG	0x28	MCU-ULP Clock Status Register
Reserved	0x2C	
Section 6.13.13.11 MCUULP_TIMER_CLK_CONFIG	0x30	MCU-ULP Timer Clock Configuration Register
Section 6.13.13.12 MCUULP_ADCDAC_CLK_CONFIG	0x34	MCU-ULP Aux-ADC/DAC Configuration Register
Section 6.13.13.13 BYPASS_I2S_CLK_REG	0x3C	
Reserved	0x40	
Section 6.13.13.14 ULP_RM_RME_REG	0x44	
Section 6.13.13.15 ULP_CLK_ENABLE_REG	0x48	
Reserved	0x4C	
Section 6.13.13.16 SYSTICK_CLK_GEN_REG	0x50	
Reserved	0x54	
Reserved	0x58	
Reserved	0x5C	
Section 6.13.13.18 MCUULP_CLK_EN_REG2	0xA0	MCU-ULP Clock Enable Register 2
Section 6.13.13.18 MCUULP_CLK_EN_REG2	0xA4	UULP APB Clock Configuration Register

6.13.13 Register Description

6.13.13.1 MCUULP_CLK_EN_REG1

Table 6.55. MCUULP_CLK_EN_REG1 Description

Bit	Access	Function	Reset Value	Description
31	R	Reserved	-	Reserved
30	R/W	Reserved	0	
29	R/W	BIT_RES[2]	0	Used in PCM
28	R/W	Reserved	-	Reserved
27	R/W	ULPSS_TASS_QUASI_SYNC	0	
26	R/W	ULPSS_M4SS_SLV_SEL	0	
25	R/W	AUX_SOC_EXT_TRIG_2_SEL	0	AUX ADC trigger 2 selection 0 - soc aux ext trigger2 1- Timer interrupt 3
24	R/W	AUX_SOC_EXT_TRIG_1_SEL	0	AUX ADC trigger 1 selection 0 - soc aux ext trigger1 1 - Timer interrupt 2
23	R/W	AUX_ULP_EXT_TRIG_2_SEL	0	AUX ADC ULP trigger 2 selection 0 - ulp gpio aux ext trigger2 1- Timer interrupt 1
22	R/W	AUX_ULP_EXT_TRIG_1_SEL	0	AUX ADC ULP trigger 1 selection 0- ulp gpio aux ext trigger1 1 - Timer interrupt 0
21	R/W	MCUULP_TIMER_PCLK_EN	0	This bit is used to enable static clock to Timer APB Interface.
20	R/W	EGPIO_PCLK_ENABLE	0	Writing 1 to this enables clock to EGPIO APB Interface. Writing 0 to this disables clock to EGPIO APB Interface.
19	R/W	EGPIO_PCLK_DYN_CTRL_DISABLE	0	Writing 1 to this disables dynamic clock gating to EGPIO APB Interface. Writing 0 to this enables dynamic clock gating to EGPIO APB Interface.
18	R/W	CLK_ENABLE_ULP_MEMORIES	0	Writing 1 to this enables clock to ULP memory Writing 0 to this disables clock to ULP memory
17	R/W	Reserved.	0	Reserved.
16	R/W	Reserved	0	

Bit	Access	Function	Reset Value	Description
15	R/W	REG_ACCESS_SPI_CLK_EN		Writing 1 to this enables clock to reg access spi Writing 0 to this disables clock to reg access spi
14	R/W	MCUULP_EGPIO_CLK_EN	0	Writing 1 to this enables clock to Enhanced-GPIO. Writing 0 to this disables clock to Enhanced-GPIO.
13	R/W	MCUULP_TIMER_CLK_EN	0	Writing 1 to this enables clock to Timers. Writing 0 to this disables clock to Timers.
12	R/W	Reserved.	0	Reserved.
11	R/W	Reserved.	0	Reserved.
10	R/W	MCUULP_UART_CLK_EN	0	Writing 1 to this enables clock to UART Controller.(Asynchronous serial clock) Writing 0 to this disables clock to UART Controller.
9	R/W	MCUULP_UART_APB_CLK_EN	0	Writing 1 to this enables clock to UART APB Interface.(Pheripheral bus clock) Writing 0 to this disables clock to UART APB Interface.
8	R/W	MCUULP_SSI_CLK_EN	0	Writing 1 to this enables clock to SPI/SSI Master. Writing 0 to this disables clock to SPI/SSI Master.
7	R/W	MCUULP_SSI_APB_CLK_EN	0	Writing 1 to this enables clock to SPI/SSI Master APB Interface. Writing 0 to this disables clock to SPI/SSI Master APB Interface.
6	R/W	MCUULP_I2S_CLK_EN	0	Writing 1 to this enables clock to I^{2}S Controller. Writing 0 to this disables clock to I^{2}S Controller.
5	R/W	MCUULP_I2C_APB_CLK_EN	0	Writing 1 to this enables clock to I^{2}C APB Interface. Writing 0 to this disables clock to I^{2}C APB Interface.
4	R	Reserved	-	
3:2	R/W	BIT_RES	0	Specifies bit resolution for PCM
1	R/W	PCM_FSYNC_START	0	Writing 1 to this enables PCM fsync Writing 0 to this disables PCM fsync
0	R/W	PCM_ENABLE	0	Writing 1 to this enables PCM Writing 0 to this disables PCM

6.13.13.2 ULP_TA_PERI_ISO_REG

Table 6.56. ULP_TA_PERI_ISO_REG Description

Bit	Access	Function	Reset Value	Description
31:23	NA	Reserved	-	Reserved
22	R/W	Reserved.	0	Reserved.
21	R/W	Reserved.	0	Reserved.
20	R/W	Reserved.	0	Reserved.
19	R/W	Reserved.	0	Reserved.
18	R/W	mem_2k_4_iso_cntrl	0	2k SRAM memory isolation enable 1: enable 0: disable
17	R/W	mem_2k_3_iso_cntrl	0	2k SRAM memory isolation enable 1: enable 0: disable
16	R/W	mem_2k_2_iso_cntrl	0	2k SRAM memory isolation enable 1: enable 0: disable
15	R/W	mem_2k_1_iso_cntrl	0	2k SRAM memory isolation enable 1: enable 0: disable
14	R/W	Reserved	0	
13:10	R/W	Reserved	-	Reserved
9	R/W	proc_misc_iso_cntrl	0	mis top(TOT, semaphore, interrupt cntrl, Timer) module isolation enable 1: enable 0: disable
8	R/W	Reserved	0	Reserved
7	R/W	Reserved.	0	Reserved.
6	R/W	aux_a2d_iso_cntrl	0	AUX a2d module isolation enable 1: enable 0: disable
5	R/W	uart_iso_cntrl	0	UART module isolation enable 1: enable 0: disable
4	R/W	ssi_iso_cntrl	0	SSI module isolation enable 1: enable 0: disable
3	R/W	i2s_iso_cntrl	0	I2S module isolation enable 1: enable 0: disable
2	R/W	i2c_iso_cntrl	0	I2C module isolation enable 1: enable 0: disable
1	R	Reserved	-	

Bit	Access	Function	Reset Value	Description
0	R/W	udma_iso_cntrl	0	UDMA module isolation enable 1: enable 0: disable

6.13.13.3 ULP_TA_PERI_RESET_REG

Table 6.57. ULP_TA_PERI_RESET_REG Description

Bit	Access	Function	Reset Value	Description
31:15	NA	Reserved	-	Reserved
14	R/W	Reserved	0	
13:12	R/W	Reserved	-	Reserved
11	R/W	compator2_interrupt_unmask	0	This is ULP comparator2 interrupt unmasking signal. 0 means comparator2 interrupt is masked and 1 means unmasking. It is masked at power-on time.
10	R/W	compator1_interrupt_unmask	0	This is ULP comparator1 interrupt unmasking signal. 0 means comparator1 interrupt is masked and 1 means unmasking. It is masked at power-on time.
9	R/W	proc_misc_soft_reset_cntrl	0	mis top(TOT, semaphore, interrupt cntrl, Timer) module soft reset enable 1: out of soft reset 0: in reset
8	R/W	Reserved	0	Reserved
7	R/W	Reserved.	0	Reserved.
6	R/W	aux_a2d_soft_reset_cntrl	0	AUX a2d module soft reset enable 1: out of soft reset 0: in reset
5	R/W	uart_soft_reset_cntrl	0	UART module soft reset enable 1: out of soft reset 0: in reset
4	R/W	ssi_soft_reset_cntrl	0	SSI module soft reset enable 1: out of soft reset 0: in reset
3	R/W	i2s_soft_reset_cntrl	0	I2S module soft reset enable 1: out of soft reset 0: in reset
2	R/W	i2c_soft_reset_cntrl	0	I2C module soft reset enable 1: out of soft reset 0: in reset
1	R	Reserved	-	
0	R/W	udma_soft_reset_cntrl	0	UDMA module soft reset enable 1: out of soft reset 0: in reset

6.13.13.4 ULPSS_SPARE_REG

Table 6.58. ULPSS_SPARE_REG Description

Bit	Access	Function	Reset Value	Description
[31:16]	NA	Reserved	-	Reserved
[15:0]	R/W	Reserved	16'd0	ULPSS Spare register

6.13.13.5 MCUULP_PROC_CLK_CONFIG

Table 6.59. MCUULP_PROC_CLK_CONFIG Description

Bit	Access	Function	Reset Value	Description
31:13	-	Reserved	0	It is recommended to write these bits to 0.
12:5	R/W	MCUULP_PROC_CLK_DIV_DAC	0	Specifies the clock division factor for AHB Interface Clock.
4:1	R/W	MCUULP_PROC_CLK_SEL	0	Specifies the clock source to be used for AHB Interface. 0 - MCU-ULP Reference Clock 1 - ro_32khz_clk 2 - rc_32khz_clk 3 - xtal_32khz_clk 4 - rc_32mhz_clk 5 - ro_hf_clk 6 - MCU-HP ULP Clock 7 - doubler_clk 8-15 - Output clock is gated
0	R/W	MCUULP_BRIDGE_CLK_EN	1	Controls the clock used for ULP-PERIPHERAL accesses in PS4/PS3(described in Section 9. Power Architecture) power states 0 - Clock is disabled 1 - Clock is enabled

6.13.13.6 MCUULP_I2C_SSI_CLK_CONFIG

Table 6.60. MCUULP_I2C_SSI_CLK_CONFIG Description

Bit	Access	Function	Reset Value	Description
31:28	R/W	MCUULP_SSI_CLK_SEL	15	Specifies the clock source to be used for SPI/SSI Primary 0 - MCU-ULP Reference Clock 1 - ro_32khz_clk 2 - rc_32khz_clk 3 - xtal_32khz_clk 4 - rc_32mhz_clk 5 - ro_hf_clk 6 - MCU-HP ULP Clock 7-14 - Reserved 15 - Output clock is gated
27:24	-	Reserved	-	It is recommended to write these bits to 0.
23:17	R/W	MCUULP_SSI_CLK_DIV_FAC	0	Specifies the clock division factor for SPI/SSI Primary Clock.
16	R/W	MCUULP_SSI_CLK_DIV_EN	0	Writing 1 to this enables clock to SPI/SSI Primary Clock Dividers. Writing 0 to this disables clock to SPI/SSI Primary Clock Dividers.
15:1	-	Reserved	-	It is recommended to write these bits to 0.
0	R/W	MCUULP_I2C_CLK_EN	0	Writing 1 to this enables clock to I^{2}C Controller. Writing 0 to this disables clock to I^{2}C Controller.

6.13.13.7 MCUULP_I2S_CLK_CONFIG

Table 6.61. MCUULP_I2S_CLK_CONFIG Description

Bit	Access	Function	Reset Value	Description
31:19	-	Reserved	-	It is recommended to write these bits to 0.
18	R/W	MCUULP_I2S_APB_CLK_EN	0	Writing 1 to this enables clock to I^2S APB Interface. Writing 0 to this disables clock to I^2S APB Interface.
17	R/W	ULP_I2S_PCLK_DYN_CTRL_DISABLE	0	Writing 1 to this disables dynamic clock gating to I^2S APB Interface. Writing 0 to this enables dynamic clock gating to I^2S APB Interface.
16	R/W	ULP_I2S_LOOP_BACK_MODE	0	Writing 1 to this enables I^2S loop-back mode. Writing 0 to this disables I^2S loop-back mode.
15	-	Reserved	-	Reserved
14	R/W	ULP_I2S_SCLK_DYN_CTRL_DISABLE	0	Writing 1 to this disables dynamic clock gating to I^2S serial clk. Writing 0 to this enables dynamic clock gating to I^2S serial clk.
13	R/W	MCUULP_I2S_MASTER_SLAVE_MODE	0	Writing 1 to this configures I^2S to Primary Mode. Writing 0 to this configures I^2S to Secondary Mode.
12:5	R/W	MCUULP_I2S_CLK_DIV_FAC	0	Specifies the clock division factor for I^2S Primary Clock.

Bit	Access	Function	Reset Value	Description
4:1	R/W	MCUULP_I2S_CLK_SEL	15	Specifies the clock source to be used for I^{2}S Primary 0 - MCU-ULP Reference Clock 1 - ro_32khz_clk 2 - rc_32khz_clk 3 - xtal_32khz_clk 4 - rc_32mhz_clk 5 - ro_hf_clk 6 - MCU-HP ULP Clock 7 - doubler_clk 8 - i2s_pll_clk 9-14 - Reserved 15 - Output clock is gated
0	R/W	MCUULP_I2S_CLK_DIV_EN	0	Writing 1 to this enables clock to I^{2}S Clock Dividers. Writing 0 to this disables clock to I^{2}S Clock Dividers.

6.13.13.8 MCUULP_UART_CLK_CONFIG

Table 6.62. MCUULP_UART_CLK_CONFIG Description

Bit	Access	Function	Default Value	Description
31:18	-	Reserved	-	It is recommended to write these bits to 0.
17	R/W	MCUULP_INTF_PLL_BYPCLK_CLKCLNR_OFF	1	Clock cleaner OFF Control for Interface PLL Bypass Clock
16	R/W	MCUULP_INTF_PLL_BYPCLK_CLKCLNR_ON	0	Clock cleaner ON Control for Interface PLL Bypass Clock
15	R/W	MCUULP_BYPASS_INTF_PLL_CLK_SEL	0	Select to choose bypass clock or PLL clock 1'b0 => intf_pll_clk 1'b1 => One of the bypass clocks based on MCUULP_INTF_PLL_CLK_BYP_SEL
14:13	R/W	MCUULP_INTF_PLL_CLK_BYP_SEL	0	Selects one of the bypass clock for Interface PLL Clocks 00 => ap_ddr_soc_intf_pll_byp_clk 01 => Not Valid 10 => i2s_pll_byp_clk 11 => ref_byp_clk
12:5	R/W	MCUULP_UART_CLK_DIV_FAC	0	Specifies the clock division factor for UART.
4:1	R/W	MCUULP_UART_CLK_SEL	15	Specifies the clock source to be used for UART 0 - MCU-ULP Reference Clock 1 - ro_32khz_clk 2 - rc_32khz_clk 3 - xtal_32khz_clk 4 - rc_32mhz_clk 5 - ro_hf_clk 6 - MCU-HP ULP Clock 7 - doubler_clk 8. intf_pll_clk Give values 9-15 for clock gating of dynamic MUX
0	R/W	MCUULP_UART_FRAC_CLK_SEL	0	Selects the Divider type for UART Controller. 0 - Clock Swallow output is selected 1 - Fractional Clock Divider output is selected

6.13.13.9 M4LP_CTRL_REG

Table 6.63. M4LP_CTRL_REG Register

Bit	Access	Function	Default Value	Description
[31:5]	R/W	Reserved	0x0	Reserved
4	R/W	ulp_mem_clk_ulp_dyn_ctrl_disable	0x0	Disable the dynamic clock gating for M4 memories in ULP mode 1: Dynamic control disabled 0: Dynamic control enabled
3	R/W	ulp_mem_clk_ulp_enable	0x0	Static clock enable for M4 memories in ULP mode 1: clock enabled 0: Dynamic control
2	R/W	ulp_m4_core_clk_enable	1	Static clock enable m4 core in ULP mode 1:Clk enabled 0:clk diabled
1:0	R/W	Reserved	0	Reserved

6.13.13.10 MCUULP_CLK_STATUS_REG

Table 6.64. MCUULP_CLK_STATUS_REG Register

Bit	Access	Function	Reset Value	Description
31:13	-	Reserved	-	Reserved
12	R	CLOCK_SWITCHED_SYSTICK	0	Status of Dynamic Clock Mux in systick Clock Generation 1 : Clock got switched and output clock can be used 0 : Clock switching is in progress
11	R	Reserved.	0	Reserved.
10	R	Reserved.	0	Reserved.
9	R	Reserved	0	Reserved
8	R	MCUULP_TIMER_CLK_SWITCHED	0	Status of Dynamic Clock Mux in Timer Clock Generation 1 : Clock got switched and output clock can be used 0 : Clock switching is in progress
7	R	MCUULP_ADCDAC_CLK_SWITCHED	0	Status of Dynamic Clock Mux in Aux-ADC/DAC Clock Generation 1 : Clock got switched and output clock can be used 0 : Clock switching is in progress
6	R	Reserved.	0	Reserved.
5	R	MCUULP_SSI_CLK_SWITCHED	0	Status of Dynamic Clock Mux in SPI/SSI Primary Clock Generation 1 : Clock got switched and output clock can be used 0 : Clock switching is in progress
4	R	CLOCK_SWITCHED_I2C_b	0	Status of Dynamic Clock Mux in I2C Clock Generation 1 : Clock got switched and output clock can be used 0 : Clock switching is in progress
3	R	MCUULP_PROC_CLK_SWITCHED	0	Status of Dynamic Clock Mux in AHB Interface Clock Generation 1 : Clock got switched and output clock can be used 0 : Clock switching is in progress

Bit	Access	Function	Reset Value	Description
2	R	CLOCK_SWITCHED_CORTEX_SLEEP_CLK	0	Status of Dynamic Clock Mux in M4 sleep Clock Generation 1 : Clock got switched and output clock can be used 0 : Clock switching is in progress
1	R	MCUULP_I2S_CLK_SWITCHED	0	Status of Dynamic Clock Mux in I^{2}S Controller Clock Generation 1 : Clock got switched and output clock can be used 0 : Clock switching is in progress
0	R	MCUHP_UART_CLK_SWITCHED	0	Status of Dynamic Clock Mux in UART Clock Generation 1 : Clock got switched and output clock can be used 0 : Clock switching is in progress

6.13.13.11 MCUULP_TIMER_CLK_CONFIG

Table 6.65. MCUULP_TIMER_CLK_CONFIG Description

Bit	Access	Function	Reset Value	Description
13	-	ULP_TIMER_IN_SYNC	-	It is recommended to write these bits to 0.
12:5		Reserved		Reserved
4:1	R/W	MCUULP_TIMER_CLK_SEL	15	Specifies the clock source to be used for Timer 0 - MCU-ULP Reference Clock 1 - ro_32khz_clk 2 - rc_32khz_clk 3 - xtal_32khz_clk 4 - rc_32mhz_clk 5 - ro_hf_clk 6 - MCU-HP ULP Clock 7-14 - Reserved 15 - Output clock is gated
0	-	Reserved	-	It is recommended to write these bits to 0.

6.13.13.12 MCUULP_ADCDAC_CLK_CONFIG

Table 6.66. MCUULP_ADCDAC_CLK_CONFIG Description

Bit	Access	Function	Reset Value	Description
12:5	-	Reserved	-	It is recommended to write these bits to 0.
4:1	R/W	MCUULP_ADCDAC_CLK_SEL	15	Specifies the clock source to be used for Aux-ADC/DAC Controller 0 - MCU-ULP Reference Clock 1 - ro_32khz_clk 2 - rc_32khz_clk 3 - xtal_32khz_clk 4 - rc_32mhz_clk 5 - ro_hf_clk 6 - MCU-HP ULP Clock 7 - doubler_clk 8 - i2s_pll_clk 8-14 - Reserved 15 - Output clock is gated
0	R/W	ULP_AUX_CLK_EN	0	Writing 1 to this enables clock to AUX ADC DAC. Writing 0 to this disables clock to AUX ADC DAC.

6.13.13.13 BYPASS_I2S_CLK_REG

Table 6.67. BYPASS_I2S_CLK_REG Description

Bit	Access	Function	Reset Value	Description
2	R/W	bypass_i2s_pll_clk_cln_off	0x1	I2S PLL Bypass clock cleaner OFF
1	R/W	bypass_i2s_pll_clk_cln_on	0x0	I2S PLL Bypass clock cleaner ON
0	R/W	bypass_i2s_pll_sel	0x0	Bypass_I2S PLL clock 1: Bypass clock is used 0: I2S Clock is used

6.13.13.14 ULP_RM_RME_REG

Table 6.68. ULP_RM_RME_REG Description

Bit	Access	Function	Reset Value	Description
[6:5]	R/W	ulp_mem_rm_sram	0x1	Read Margin (RM) ports for sram memories. This needs to be programmed when the SRAM is not active
4	R/W	ulp_mem_rme_sram	0x0	RM enable signal for sram memories. This needs to be programmed when the SRAM is not active
3	R	Reserved	0x0	Reserved
[2:1]	R/W	ulp_mem_rm	0x1	RM ports for memories internal to peripheral. This needs to be programmed when the peripheral memories are not active
0	R/W	ulp_mem_rme	0x0	RM enable signal for memories internal tp peripherals. This needs to be programmed when the peripheral memories are not active

6.13.13.15 ULP_CLK_ENABLE_REG

Table 6.69. ULP_CLK_ENABLE_REG Description

Bit	Access	Function	Reset Value	Description
31:10	NA	Reserved	-	Reserved
9	R/W	intf_pll_clk_en_prog	0x0	Static Clock enable to iPMU for INTF-PLL Clock 1: enable 0: disable
8	R/W	ref_clk_en_ips_prog	0	Static Clock enable to iPMU for REF Clock 1: enable 0: disable
7	R/W	i2s_pllclk_en_prog	0	Static Clock enable to iPMU for I2S-PLL Clock 1: enable 0: disable
6	R/W	soc_clk_en_prog	0	Static Clock enable to iPMU for PLL-500 Clock 1: enable 0: disable
5	R/W	ulp_32mhz_rc_clk_en_prog	0	Static Clock enable to iPMU for 32MHz RC Clock 1: enable 0: disable
4	R/W	ulp_20mhz_ro_clk_en_prog	0	Static Clock enable to iPMU for 20MHz RO Clock 1: enable 0: disable
3	R/W	ulp_doubler_clk_en_prog	0	Static Clock enable to iPMU for Doubler Clock 1: enable 0: disable
2	R/W	ulp_32khz_xtal_clk_en_prog	0	Static Clock enable to iPMU for 32KHz XTAL Clock 1: enable 0: disable
1	R/W	ulp_32khz_rc_clk_en_prog	0	Static Clock enable to iPMU for 32KHz RC Clock 1: enable 0: disable
0	R/W	ulp_32khz_ro_clk_en_prog	0	Static Clock enable to iPMU for 32KHz RO Clock 1: enable 0: disable

6.13.13.16 SYSTICK_CLK_GEN_REG

Table 6.70. SYSTICK_CLK_GEN_REG Description

Bit	Access	Function	Reset Value	Description
[31:13]	NA	Reserved	0x0	Reserved
[12:5]	R/W	systick _clkdiv_factor	0x0	systick clock division factor
[4:1]	R/W	systick _clk_sel	0xF	systick clock select 0: ref_clk (output of dynamic clock mux for different possible ref_clk sources) 1: systick_1s 2: systick_1ms 3: ulp_32khz_rc_clk 4: ulp_32khz_xtal_clk 5: ulp_32mhz_rc_clk
0	R/W	systick_clk_en	0x0	systick clk enable 1: enable 0: disable

6.13.13.17 ULP_SOC_GPIO_n_MODE_REG

Table 6.71. ULP_SOC_GPIO_n_MODE_REG Description

Bit	Access	Function	Reset Value	Description
[31:3]	NA	Reserved	-	Reserved
[2:0]	R/W	ulp_socgpio_n_mode	0x0	mode bits for SoC GPIO n

6.13.13.18 MCUULP_CLK_EN_REG2

Table 6.72. MCUULP_CLK_EN_REG2 Description

Bit	Access	Function	Reset Value	Description
31:18	-	Reserved	-	It is recommended to write these bits to 0.
17	R/W	MCUULP_UDMA_CLK_EN	0	Writing 1 to this enables clock to UDMA. Writing 0 to this disables clock to UDMA.
16	R/W	AUX_CLK_MEM_DYN_CTRL_DISABLE	0	Writing 1 to this disables dynamic clock gating for AUX ADC-DAC MEMORY Writing 0 to this enables dynamic clock gating for AUX ADC-DAC MEMORY
15	R/W	AUX_CLK_DYN_CTRL_DISABLE	0	Writing 1 to this disables dynamic clock gating for AUX ADC-DAC Writing 0 to this enables dynamic clock gating for AUX ADC-DAC
14	R/W	AUX_PCLK_DYN_CTRL_DISABLE	0	Writing 1 to this disables dynamic clock gating for AUX ADC-DAC APB interface Writing 0 to this enables dynamic clock gating for AUX ADC-DAC APB interface
13	R/W	AUX_MEM_EN	0	Writing 1 to this enables AUX ADC-DAC memory Writing 0 to this disables AUX ADC-DAC memory
12	R/W	MCUULP_ADCDAC_CLK_EN	0	Writing 1 to this enables clock to AUX ADC-DAC Writing 0 to this disables clock to AUX ADC-DAC
11	R/W	AUX_PCLK_EN	0	Writing 1 to this enables clock to AUX ADC-DAC APB interface Writing 0 to this disables clock to AUX ADC-DAC APB interface
10	R/W	Reserved.	0	Reserved.
9	R/W	Reserved	0	Reserved.

Bit	Access	Function	Reset Value	Description
8	R/W	REG_ACCESS_SPI_CLK_DYN_CTRL_DISABLE	0	Writing 1 to this disables dynamic clock gating for reg access SPI Writing 0 to this enables dynamic clock gating for reg access SPI
7	R/W	TIMER_SCLK_DYN_CTRL_DISABLE	0	Writing 1 to this disables dynamic clock gating for timer clock Writing 0 to this enables dynamic clock gating for timer clock
6	R/W	TIMER_PCLK_DYN_CTRL_DISABLE	0	Writing 1 to this disables dynamic clock gating for timer APB interface Writing 0 to this enables dynamic clock gating for timer APB interface
5	R/W	UART_SCLK_DYN_CTRL_DISABLE	1	Writing 1 to this disables dynamic clock gating for UART serial clock Writing 0 to this enables dynamic clock gating for UART serial clock
4	R/W	UART_CLK_DYN_CTRL_DISABLE	1	Writing 1 to this disables dynamic clock gating for UART clock Writing 0 to this enables dynamic clock gating for UART clock
3	R/W	SSI_MST_SCLK_DYN_CTRL_DISABLE	1	Writing 1 to this disables dynamic clock gating for SSI primary serial clock Writing 0 to this enables dynamic clock gating for SSI primary serial clock
2	R/W	SSI_MST_PCLK_DYN_CTRL_DISABLE	1	Writing 1 to this disables dynamic clock gating for SSI primary APB interface Writing 0 to this enables dynamic clock gating for SSI primary APB interface
1	R/W	I2S_CLK_DYN_CTRL_DISABLE	0	Writing 1 to this disables dynamic clock gating for I2S Writing 0 to this enables dynamic clock gating for I2S

Bit	Access	Function	Reset Value	Description
0	R/W	I2C_PCLK_DYN_CTRL_DISABLE	0	Writing 1 to this disables dynamic clock gating for I2S APB interface Writing 0 to this enables dynamic clock gating for I2S APB interface

6.13.13.19 MCUULP_UULP_CLK_CONFIG

Table 6.73. MCUULP_UULP_CLK_CONFIG Description

Bit	Access	Function	Reset Value	Description
31:9	-	Reserved	-	It is recommended to write these bits to 0.
8	R/W	ENABLE	1	Writing 1 to this enable SLPSS APB interface Writing 0 to this disables able SLPSS APB interface
7:0	R/W	MCUULP_APB_CLK_DIV_FAC	2	Specifies the clock division factor for UULP APB Interface

6.14 MCU ULP VBAT Clock Architecture

6.14.1 General Description

The MCU ULP VBAT domain contains the MCU ULP VBAT Peripherals. This section describes the different clock sources possible for each peripheral.

6.14.2 Features

- The clock sources used for MCU ULP VBAT domain includes RC, RO, XTAL clocks from the ULP Clock Oscillators.
- The Peripherals are configured by the processor through the APB Interface for which the clock is fed from MCU ULP Domain.

6.14.3 Functional Description

The following figure depicts the clock sources used for the functionality present in MCU ULP VBAT domain.

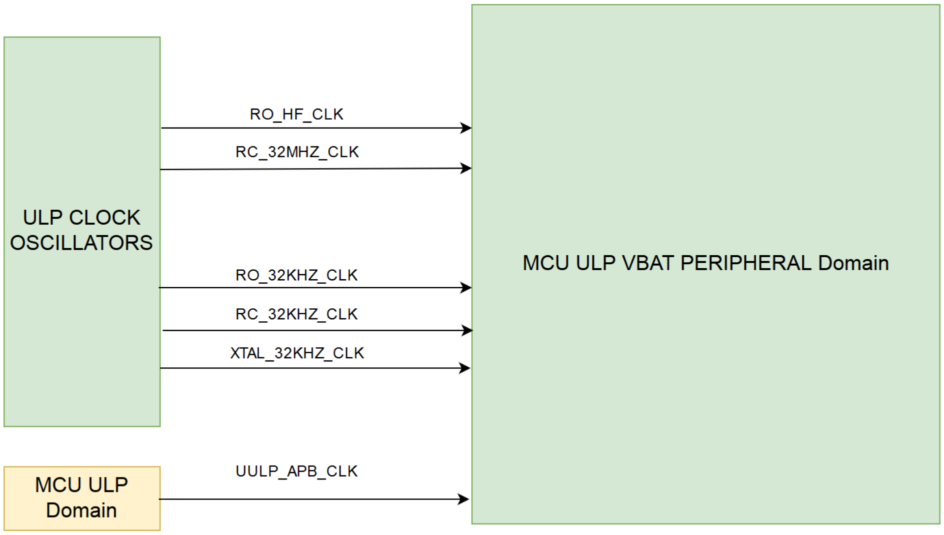


Figure 6.30. Clock Generation

The list below contains the different peripherals for which clock can be configured independently.

1. HIGH-FREQ CLOCK
   a. This clock is used for Low-Power State machines.
   b. This can be configured using MCUULP_VBAT_HF_CLK_SEL in Section 6.14.5.2 MCUULP_VBAT_HFCLK_REG Register.
   c. The clock switching status can be read through MCUULP_VBAT_HF_CLK_SWITCHED in Section 6.14.5.2 MCUULP_VBAT_HFCLK_REG Register
2. LOW-FREQ CLOCK
   a. This clock is used for the following for the following UULP-PERIPHERALS
      i. WatchDog Timer
      ii. Deep-Sleep Timer
      iii. GPIO Timestamping
   b. This can be configured using MCUULP_VBAT_LF_CLK_SEL in Section 6.14.5.1 MCUULP_VBAT_LFCLK_REG Register.
   c. The clock switching status can be read through MCUULP_VBAT_LF_CLK_SWITCHED in 6.14.5.1 MCUULP_VBAT_LFCLK_REG Register
3. SYSRTC

6.14.4 Register Summary

Base_address = 0x2404_8000

Table 6.74. Register Summary

Register Name	Offset	Description
Section 6.14.5.1 MCUULP_VBAT_LFCLK_REG	0x020	Low Frequency Clock Select Register
Section 6.14.5.2 MCUULP_VBAT_HFCLK_REG	0x118	High Frequency Clock Select Register

6.14.5 Register Description

6.14.5.1 MCUULP_VBAT_LFCLK_REG

Table 6.75. MCUULP_VBAT_LFCLK_REG Description

Bit	Access	Function	POR Value	Description
31:24	-	Reserved	-	-
23:18	R/W	MCUULP_VBAT_SYS_RTC_CLK_DIV_FAC	6'b0	Division factor for RC_32_MHZ_CLK 6'b000001 - Divide by 2 6'b010000 - Divide by 32
17:15	-	Reserved	-	-
14	R/W	MCUULP_VBAT_SYS_RTC_CLK_EN	1'b0	Writing 1 to this enables clock to SYSRTC from dynamic mux Writing 0 to this has no effect.
13	R	MCUULP_VBAT_SYS_RTC_CLK_SWITCHED	1'b1	Status of Dynamic Clock Mux in Reference Clock Generation 1 indicates Clock got switched and output clock can be used 0 indicates Clock switching is in progress
12:9	R/W	MCUULP_VBAT_SYS_RTC_CLK_SEL	4'b0	0010 - 1 KHZ clock, MCUULP_VBAT_SYS_RTC_CLK_DIV_FAC must be set to 6'b010000. 0010 - RO_32KHZ_CLK 0100 - RC_32KHZ_CLK 1000 - XTAL_32KHZ_CLK
8:4	R	Reserved	-	-
3	R	MCUULP_VBAT_LF_CLK_SWITCHED	1'b1	Status of MCUULP Low Frequency Clock Dynamic Clock Mux 1 : Clock got switched and output clock can be used 0 : Clock switching is in progress
2:0	R/W	MCUULP_VBAT_LF_CLK_SEL	2'b00	001 : ro_32k_clk 010 : rc_32k_clk 100 : xtal_32k_clk After programming, wait for MCUULP_LF_CLK_SWITCHED to be 1'b1

6.14.5.2 MCUULP_VBAT_HFCLK_REG

Table 6.76. NPSS High Frequency Clock Select Register

Bit	Access	Function	Reset Value	Description
30:16	R	Reserved	NA	Reserved
15	R	MCUULP_VBAT_HF_CLK_SWITCHED	1'b1	Status of MCUULP High Frequency Clock Dynamic Clock Mux 1 : Clock got switched and output clock can be used 0 : Clock switching is in progress
14:5	R	Reserved	NA	Reserved
4:2	R/W	MCUULP_VBAT_HF_CLK_SEL	2'd0	0 : Clock is Gated 1 : ro_20m_clk 2 : rc_32m_clk After programming, wait for MCUULP_VBAT_HF_CLK_SWITCHED to be 1'b1
1:0	R	Reserved	NA	Reserved

6.15 ULP Clock Oscillators

6.15.1 General Description

ULP Clock Oscillators is the source of the Low frequency RC/RO and High frequency RC/RO clocks which are used by MCU HP, MCU ULP and MCU ULP VBat domains as one of the clock sources for multiple peripherals.

6.15.2 Features

1. Low-Power Clock Oscillators for generation of RC-Based and RO-Based clocks.
2. Fast Start-up times. The table below shows the start-up times for each clock

S.No	Clock Source	Start-Up time
1	RC_32MHZ_CLK	10 Micro-seconds
2	RO_HF_CLK	30 Micro-seconds
3	DOUBLER_CLK	30 Micro-seconds
4	RC_32KHZ_CLK	500 Micro-seconds
5	RO_32KHZ_CLK	500 Micro-seconds
6	XTAL_32KHZ_CLK	2.5 seconds.

6.15.3 Functional Description

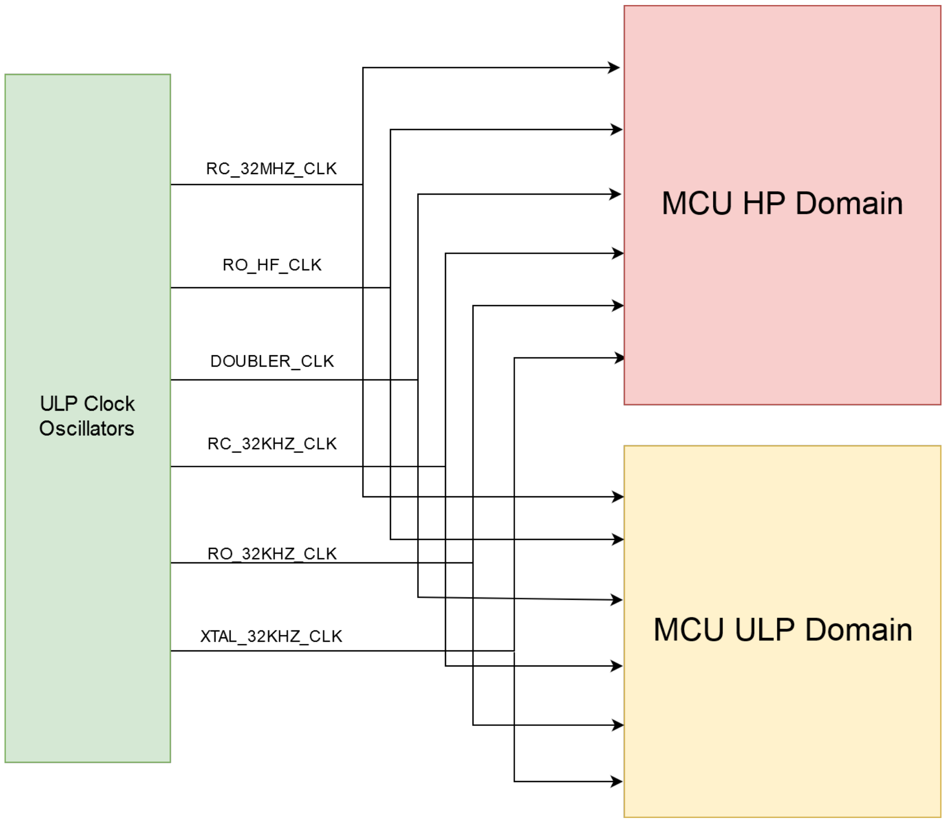


Figure 6.31. Clock Generation

There are different Oscillators to generate the Low-Frequency and High-Frequency clocks. The clock oscillators can be controlled individually through Register programming which are described below

6.15.4 Register Summary

Base Address: 0x2404_8100

Table 6.77. Register summary

Register Name	Offset	Description
Section 6.15.5.1 ULP_CLKOSC_CTRL_REG	0x20	ULP Clock Oscillators Control Register

6.15.5 Register Description

6.15.5.1 ULP_CLKOSC_CTRL_REG

Table 6.78. ULP_CLKOSC_CTRL_REG

Bit	Access	Function	Reset Value	Description
31:22	-	Reserved	-	It is recommended to write these bits to 0.
22	R/W	XTAL_40MHZ_CLK_EN	1	Writing 1 to this enables the XTAL-40MHz Clock Writing 0 to this disables the XTAL-40MHz Clock
21	R/W	DOUBLER_CLK_EN	0	Writing 1 to this enables the Doubler Clock Writing 0 to this disables the Doubler Clock
20	R/W	RO_HF_CLK_EN	0	Writing 1 to this enables the RO High-Frequency Clock Writing 0 to this disables the RO High-Frequency Clock
19	R/W	RC_32MHZ_CLK_EN	1	Writing 1 to this enables the RC 32MHz Clock Writing 0 to this disables the RC 32MHz Clock
18	R/W	XTAL_32KHZ_CLK_EN	0	Writing 1 to this enables the XTAL 32KHz Clock Writing 0 to this disables the XTAL 32KHz Clock
17	R/W	RO_32KHZ_CLK_EN	1	Writing 1 to this enables the RO 32KHz Clock Writing 0 to this disables the RO 32KHz Clock
16	R/W	RC_32KHZ_CLK_EN	1	Writing 1 to this enables the RC 32KHz Clock Writing 0 to this disables the RC 32KHz Clock
15:0	-	Reserved	-	

7. Resets

7.1 General Description

The device is brought into initial known state by applying the reset. The main resets in the chip are described in this section.

7.2 Features

1. Resets the design and set it to an initial state according to the reset source
2. Reset Resources
    a. Primary reset , RESET_N_PAD
    b. Power on reset, POC_IN
    c. Black out Monitor
    d. Watch Dog Reset
    e. Reset request from host or Processor

7.3 Functional Description

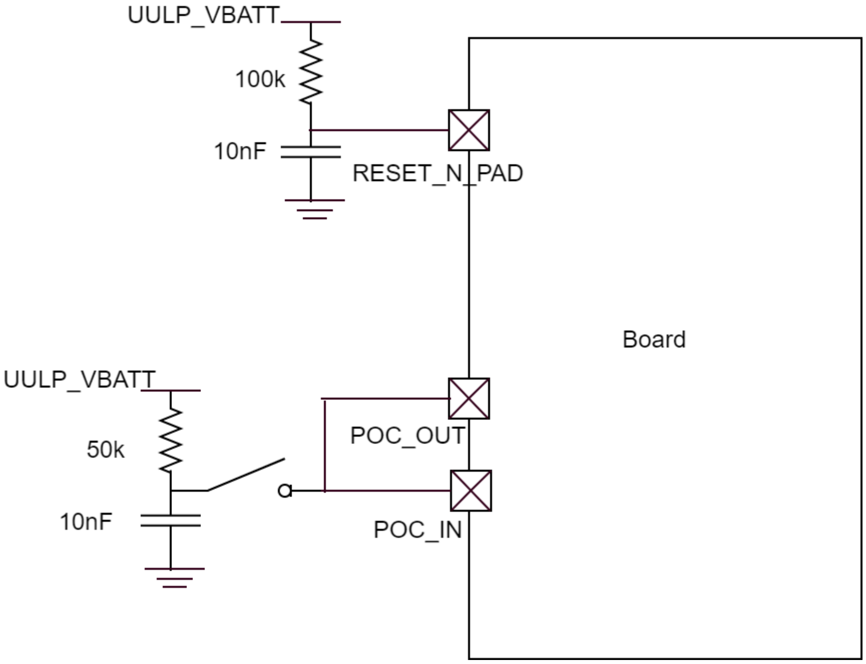


Figure 7.1. External Connections on Board for Reset

The diagram above shows the required off chip components and the external board connections required for reset.

7.3.1 RESET_N_PAD

This is the primary reset to the chip. The external resistor and capacitor components are as shown in the diagram above. The complete design can be reset using this reset. This will be forced low when POC_IN is low. When POC_IN is high, the external resistor will pull up RESET_N_PAD. If the REST_N_PAD is also controlled by host IC, the host IC's GPIO should be in open-drain mode (strong pull down). Refer to Chip Reset Generation reference schematic for more information on RESET circuit.

7.3.2 POC_IN

There is an internally generated Power On Reset in the design which is generated once VBATT supply is >1.7V and core supply is stable by blackout monitor. That reset comes out as POC_OUT. This has to be looped back to POC_IN on board (recommended option).

POC_IN can be also be externally controlled instead of connecting to POC_OUT. A resistor and capacitor have to be connected on board as shown in the diagram above. Refer to Chip POC Generation reference schematic for more information on POC circuit.

7.3.3 Black Out Monitor

By default Blackout Monitor is disabled. It has to be enabled by enabling BLACKOUT_EN.

Blackout occurs when UULP_VBATT is lower than 1.7V, this will make POC_OUT low. Upon blackout event, RESET_N_PAD is autonomously pulled low if POC_OUT is connected to POC_IN. When the voltage on UULP_VBATT becomes greater than 1.7V, RESET_N_PAD charges again and the design will be out of reset.

7.3.4 WatchDog Reset

Timer value is programmed in the watch dog timer. Once the timeout happens it generates an interrupt to the processor. If the processor does not service the interrupt, the watch dog reset is generated and the entire design is reset. The detailed description is present in WatchDog Timer block description.

Note:

For more elaboration on Reset, refer to the Reset Pin table in the datasheet.

7.4 Reset request from host or Processor

Cortex M4 can request for reset on SYSRESETREQ. There can be reset requests from the host SDIO or SPI. When there are any such requests from Cortex M4 or host, it will reset the digital blocks.

7.5 Register Summary

Base Address : 0x2405_A000

Table 7.1. Register Summary

Register Name	Offset	Description
Section 7.6.1 BLACK_OUT_MON_EN_REG	0x4AC	Black Out Monitor Enable Register

7.6 Register Description

7.6.1 BLACK_OUT_MON_EN_REG

Table 7.2. Black Out Monitor Enable Register Description

Bit	Access	Function	Reset Value	Description
5	R/W	BLACKOUT_EN	1'b1	0: Black Out Monitor is Disabled 1: Black Out Monitor is Enabled

8. Interrupts

8.1 General Description

MCU uses nested vectored interrupt controller (NVIC) for interrupts handling. It is programmable and its registers are located in the M4 System Control Space (SCS) of the memory map. The NVIC handles the exceptions and interrupt configurations, prioritization, and interrupt masking.

8.2 Features

The NVIC has the following features:

- Supports 99 interrupts
- Flexible exception and interrupt management
- Nested exception/interrupt support
- Vectored exception/interrupt entry
- Interrupt masking

8.3 Functional Description

8.3.1 Flexible Exception and Interrupt Management

Each interrupt (apart from the NMI) can be enabled or disabled and can have its pending status set or cleared by software. The NVIC can handle various types of interrupt sources:

- Pulsed interrupt request e the interrupt request is at least one clock cycle long. When the NVIC receives a pulse at its interrupt input, the pending status is set and held until the interrupt gets serviced.
- Level triggered interrupt request e the interrupt source holds the request high until the interrupt is serviced.

  The signal level at the NVIC input is active high. However, the actual external interrupt input on the microcontroller could be designed differently and is converted to an active high signal level by on-chip logic.

8.3.2 Nested Exception/Interrupt Support

Each exception has a priority level. Some exceptions, such as interrupts, have programmable priority levels and some others (e.g., NMI) have a fixed priority level. When an exception occurs, the NVIC will compare the priority level of this exception to the current level. If the new exception has a higher priority, the current running task will be suspended. Some of the registers will be stored on the stack memory, and the processor will start executing the exception handler of the new exception. This process is called “preemption.” When the higher priority exception handler is complete, it is terminated with an exception return operation and the processor automatically restores the registers from stack and resumes the task that was running previously. This mechanism allows nesting of exception services without any software overhead.

8.3.3 Vectored Exception/Interrupt Entry

When an exception occurs, the processor will need to locate the starting point of the corresponding exception handler. Traditionally software handles this step. The Cortex-M4 processor automatically locate the starting point of the exception handler from a vector table in the memory. As a result, the delays from the start of the exception to the execution of the exception handlers are reduced.

8.3.4 Interrupt Masking

The NVIC in processor provide several interrupt masking registers such as the PRIMASK special register. Using the PRIMASK register, all exceptions excluding HardFault and NMI are disabled. This masking is useful for operations that should not be interrupted, like time critical control tasks or real-time multimedia codecs. Alternatively we can also use the BASEPRI register to select mask exceptions or interrupts which are below a certain priority level. The flexibility and capability of the NVIC also make the Cortex-M4 processors very easy to use, and provide better a system response by reducing the software overhead in interrupt processing, which also leads to smaller code size.

8.3.5 Vector Table

When an exception event takes place and is accepted by the processor core, the corresponding exception handler is executed. To determine the starting address of the exception handler, a vector table mechanism is used. The vector table is an array of word data inside the system memory, each representing the starting address of one exception type. The vector table is relocatable and the relocation is controlled by a programmable register in the NVIC called the Vector Table Offset Register (VTOR). After reset, the VTOR is reset to 0; therefore, the vector table is located at address 0x0 after reset. The beginning of the memory space contains the vector table, and the first two words in the vector table are the initial value for the Main Stack Pointer (MSP), and the reset vector, which is the starting address of the reset handler. After these two words are read by the processor, the processor then sets up the MSP and the Program Counter (PC) with these values.

For example, if the reset is exception type 1, the address of the reset vector is 1 times 4 (each word is 4 bytes), which equals 0x00000004, and the NMI vector (type 2) is located at 2 x 4 = 0x00000008. The address 0x00000000 is used to store the starting value of the Main Stack Pointer.

8.3.6 Vectored Interrupt Table (VIT)

99 interrupts are mapped on NVIC. MCU HP peripheral interrupts, MCU ULP peripheral interrupts, MCU UULP peripheral interrupts and NWP peripheral interrupts are mapped into following Vectored interrupt table.

Interrupt Number	Interrupt
19 : 0	MCU ULP peripheral Interrupts
29: 20	MCU UULP peripheral Interrupts
74:30	MCU HP peripheral interrupts
98:75	NWP peripheral interrupts

MCU HP Peripheral Interrupts

There are 45 MCU HP peripheral interrupts in the SiWx917 chip. The following table provides the list of MCU HP peripheral interrupts and their interrupt number in vector interrupt table.

Table 8.1. Register Summary

Interrupt Number in VIT	MCU HP Peripheral interrupt
30	Reserved
31	GPDMA interrupt
32	Reserved
33	MCU HP UDMA interrupt
34	SCT interrupt
35	HIF Interrupt 1
36	HIF Interrupt 2
37	Reserved
38	USART 0 Interrupt
39	UART 1 Interrupt
40	Reserved
41	EGPIO wakeup interrupts
42	I2C0 Interrupt
43	Reserved
44	SSI Secondary Interrupt
45	Reserved
46	GSPI Primary Interrupt
47	SSI Primary Interrupt
48	MCPWM Interrupt
49	QEI Interrupt
51 : 50	GPIO Group Interrupt
59 : 52	GPIO Pin Interrupt
60	QSPI Interrupt (Flash Controller)
61	I2C1 Interrupt
62	MVP Interrupt
63	MVP Wake up Interrupt
64	I2S master Interrupt
65	Reserved
66	Dcache Secure Interrupt
67	Dcache Non-Secure Interrupt
68	Reserved
69	PLL clock ind Interrupt
70	Reserved

Interrupt Number in VIT	MCU HP Peripheral interrupt
71	QSPI PSRAM Controller Interrupt
72	Reserved
73	Reserved
74	NWP P2P interrupt

Base Address: 0x4611_0000

Table 8.2. MCU Multi-Channel Interrupt Selection Registers

Register Name	Offset	Description
RESERVED	0x00	Reserved
M4SS_GPDMA_INTR_SEL	0x04	MCU GPDMA interrupt selection register
RESERVED	0x08	Reserved
M4SS_UDMA_INTR_SEL	0x0C	MCU HP uDMA interrupt selection register
M4SS_SCT_INTR_SEL	0x10	SCT interrupt selection register
M4SS_GPDMA_INTR_SEL_TASS	0x2C	The is used to select the GPDMA interrupts (8) that to be passed to NWP. Selected interrupts are ored mapped on to gpdma bit in below MCU to NWP Interrupts
M4SS_UDMA_INTR_SEL_TASS	0x34	The is used to select the UDMA interrupts (32) that to be passed to NWP. Selected interrupts are ored mapped on to UDMA bit in below MCU to NWP Interrupts
M4SS_SCT_INTR_SEL_TASS	0x38	The is used to select the SCT interrupts (32) that to be passed to NWP. Selected interrupts are ored mapped on to SCT bit in below MCU to NWP Interrupts

Register Description

M4SS_GPDMA_INTR_SEL

Table 8.3. M4SS_GDMA_INTR_SEL Register Description

Bit	Access	Function	Reset Value	Description
31:8	R	Reserved	0	Reserved
7:0	R/W	dma_m_interrupt_sel	0	This bit unmasks m th the GPDMA channel interrupt '1' – Unmasked '0' – Masked Upon read, If '0' seen upon reading this bit, this indicates that the interrupt is masked If '1' is read, this indicates interrupt is not masked.

M4SS_UDMA_INTR_SEL

Table 8.4. M4SS_UDMA_INTR_SEL Register Description

Bit	Access	Function	Reset Value	Description
31:0	R/W	udma_m_interrupt_sel	0	This bit unmasks the m th UDMA channel interrupt '1' – Unmasked '0' – Masked Upon read, If '0' seen upon reading this bit, this indicates that the interrupt is masked If '1' is read, this indicates interrupt is not masked.

M4SS_SCT_INTR_SEL

Table 8.5. M4SS_SCT_INTR_SEL Register Description

Bit	Access	Function	Reset Value	Description
31:0	R/W	sct_m_interrupt_sel	0	This bit unmasks the m th SCT channel interrupt '1' – Unmasked '0' – Masked Upon read, If '0' seen upon reading this bit, this indicates that the interrupt is masked If '1' is read, this indicates interrupt is not masked.

The group of UULP Peripheral interrupts will be converted to 10 UULP interrupt and mapped to 29-20 interrupt in MCU.

Features

- Supports 19 UULP peripheral interrupts and mapped them into 10 MCU interrupts.
- UULP GPIO interrupts have additional functionality of rise edge, fall edge and level detection.
- UULP interrupts other than GPIO interrupts will have only rise edge detection.
- Each interrupts can be masked or unmasked based on the requirement.
- UULP GPIO pins unmasked status can be read from register.

NPSS Interrupt Numbers

19 NPSS interrupts mapping and the description of the interrupts are given below table. The priority of the interrupt decreases as the interrupt number increases.

NPSS interrupt number	Interrupt Number in VIT	NPSS Interrupt
0	20	uulp_wdt_interrupt
1-5	21	uulp_gpio_interrupt
6-9	22	uulp_cmp_interrupt
10	22	uulp_sysrtc_interrupt
11	23	uulp_bod_interrupt
12	24	uulp_button_interrupt
13	25	uulp_sdc_interrupt
14	26	uulp_wireless_interrupt
15	27	uulp_wakeup_interrupt
16	28	uulp_alarm_interrupt
17	29	uulp_sec_interrupt
18	29	uulp_msec_interrupt

Programming Sequence

- All UULP interrupts to the Processor are masked by default. To unmask any interrupt, set the corresponding bit in the UULP_INTR_MASK_CLR_REG register.
- To mask the interrupts, set the corresponding bit in the UULP_INTR_MASK_SET_REG register.
- When the interrupt is raised, check UULP_INTR_STATUS_REG to find out which interrupt among the mapped intr is raised.
- Clear it by setting the corresponding bit in the UULP_INTR_CLEAR_REG register so that the interrupt will not be seen again.
- UULP GPIO interrupts will have additional functionality of fall edge detection and level detection. GPIO interrupts will be rise detection by default.
- Set the required interrupt detection mode for each GPIO in UULP_GPIO_CONFIG_REG.
- To clear fall/rise edge interrupt UULP_INTR_CLEAR_REG[5:0] can be used. But to clear level interrupts, level0/level1 enable bits in UULP_GPIO_CONFIG_REG as to be reset.

Register Summary

All the below registers are 16 bit and 32 bit accessible

Base Address: 0x1208_0000

Table 8.6. UULP Peripheral Interrupt controller Register Table

Register Name	Offset	Description
• UULP_INTR_MASK_SET_REG on page 156	0x0	UULP interrupt mask set register
• UULP_INTR_MASK_CLR_REG on page 157	0x4	UULP interrupt mask clear register
• UULP_INTR_CLEAR_REG on page 157	0x8	UULP interrupt clear register
• UULP_INTR_STATUS_REG on page 157	0xC	UULP interrupt status register
• UULP_GPIO_CONFIG_REG on page 158	0x10	UULP GPIO configuration register
• UULP_GPIO_STATUS_REG on page 159	0x14	UULP GPIO status register
• M4_WIC_CLEAR_REG on page 159	0x18	MCU WIC clear register
• M4_ULP_SLP_STATUS_REG on page 160	0x1C	MCU ULP sleep status register
• M4ULP_ISO_ENABLE_REG on page 161	0x20	MCU ULP isolation enable register
• M4ULP_RST_CTRL_REG on page 162	0x24	MCU ULP reset control register
• M4ULP_A2A_BRIDG_CTRL_REG on page 163	0x28	MCU ULP AHB2AHB Bridge Control Register
• M4ULP_A2A_BRIDG_STAT_REG on page 163	0x2C	MCU ULP AHB2AHB Bridge Status Register
• M4ULP_STCALIB_REG on page 164	0x30	MCU ULP Systick Calibration Register
• M4ULP_SYSTICK_CLK_ENABLE_REG on page 165	0x34	MCU ULP Systick Clock Enable Register
• M4ULP_SPARE_REG on page 165	0x38	MCU ULP Spare Register

Register Description

UULP_INTR_MASK_SET_REG

Table 8.7. UULP_INTR_MASK_SET_REG Description

Bit	Access	Function	Reset Value	Description
31:19	R	Reserved	1b0	Reserved for future use.
n(18:0)	R/W	UULP_INTR_n_MASK	1b1	This bit is used to mask UULP interrupt 'n' For write operation, '1'- Mask Interrupt '0'- Writing a zero into this has no effect. For read operation, '1' – Interrupt masked '0' – Not masked

UULP_INTR_MASK_CLR_REG

Table 8.8. UULP_INTR_MASK_CLR_REG Description

Bit	Access	Function	Reset Value	Description
[31:19]	R	Reserved	1b0	Reserved for future use.
n (0-18)	R/W	UULP_INTR_n_MASK	1b1	This bit is used to mask UULP interrupt 'n' For write operation, '1'- Unmask Interrupt '0'- Writing a zero into this has no effect. For read operation, '1' – Interrupt masked '0' – Not masked

UULP_INTR_CLEAR_REG

Table 8.9. UULP_INTR_CLEAR_REG Description

Bit	Access	Function	Reset Value	Description
[31:19]	R	Reserved	1b0	Reserved for future use.
n (0-18)	WO	UULP_INTR_n_CLEAR	1b1	This bit is used to clear UULP interrupt 'n' Main interrupt has to cleared at the source For write operation, '1'- Clears the Interrupt '0'- Writing a zero into this has no effect.

UULP_INTR_STATUS_REG

Table 8.10. UULP_INTR_STATUS_REG Description

Bit	Access	Function	Reset Value	Description
[31:19]	R	Reserved	1b0	Reserved for future use.
n (0-18)	RO	UULP_INTR_n_STATUS	1b0	This bit is used to read the masked UULP interrupt 'n' status For Read operation, '1'- indicates that the 'n'th UULP interrupt has been raised '0'- indicates that the interrupt is masked or not been raised

Please refer to ● NPSS Interrupt Numbers on page 155 for the respective interrupt mapping details

UULP_GPIO_CONFIG_REG

Table 8.11. UULP_GPIO_CONFIG Reg Description

Bit	Access	Function	Reset Value	Description
31:29	R	Reserved	5b0	Reserved for future use.
28	W/R	level high enable 4	1'b0	'1'- Enables level high interrupt detection for UULP_VBAT_GPIO_4 '0'- Disables level 1 interrupt detection for UULP_VBAT_GPIO_4
27	W/R	level high enable 3	1'b0	'1'- Enables level high interrupt detection for UULP_VBAT_GPIO_3 '0'- Disables level 1 interrupt detection for UULP_VBAT_GPIO_3
26	W/R	level high enable 2	1'b0	'1'- Enables level high interrupt detection for UULP_VBAT_GPIO_2 '0'- Disables level 1 interrupt detection for UULP_VBAT_GPIO_2
25	W/R	level high enable 1	1'b0	'1'- Enables level high interrupt detection for UULP_VBAT_GPIO_1 '0'- Disables level 1 interrupt detection for UULP_VBAT_GPIO_1
24	W/R	level high enable 0	1'b0	'1'- Enables level high interrupt detection for UULP_VBAT_GPIO_0 '0'- Disables level 1 interrupt detection for UULP_VBAT_GPIO_0
23:21	R	Reserved	5b0	Reserved for future use.
20	W/R	level low enable 4	1'b0	'1'- Enables level low interrupt detection for UULP_VBAT_GPIO_4 '0'- Disables level 0 interrupt detection for UULP_VBAT_GPIO_4
19	W/R	level low enable 3	1'b0	'1'- Enables level low interrupt detection for UULP_VBAT_GPIO_3 '0'- Disables level 0 interrupt detection for UULP_VBAT_GPIO_3
18	W/R	level low enable 2	1'b0	'1'- Enables level low interrupt detection for UULP_VBAT_GPIO_2 '0'- Disables level 0 interrupt detection for UULP_VBAT_GPIO_2
17	W/R	level low enable 1	1'b0	'1'- Enables level low interrupt detection for UULP_VBAT_GPIO_1 '0'- Disables level 0 interrupt detection for UULP_VBAT_GPIO_1
16	W/R	level low enable 0	1'b0	'1'- Enables level low interrupt detection for UULP_VBAT_GPIO_0 '0'- Disables level 0 interrupt detection for UULP_VBAT_GPIO_0
15:13	R	Reserved	5b0	Reserved for future use.
12	W/R	Fall edge enable 4	1'b0	'1'- Enables fall edge interrupt detection for UULP_VBAT_GPIO_4 '0'- Disables fall edge interrupt detection for UULP_VBAT_GPIO_4
11	W/R	Fall edge enable 3	1'b0	'1'- Enables fall edge interrupt detection for UULP_VBAT_GPIO_3 '0'- Disables fall edge interrupt detection for UULP_VBAT_GPIO_3
10	W/R	Fall edge enable 2	1'b0	'1'- Enables fall edge interrupt detection for UULP_VBAT_GPIO_2 '0'- Disables fall edge interrupt detection for UULP_VBAT_GPIO_2
9	W/R	Fall edge enable 1	1'b0	'1'- Enables fall edge interrupt detection for UULP_VBAT_GPIO_1 '0'- Disables fall edge interrupt detection for UULP_VBAT_GPIO_1
8	W/R	Fall edge enable 0	1'b0	'1'- Enables fall edge interrupt detection for UULP_VBAT_GPIO_0 '0'- Disables fall edge interrupt detection for UULP_VBAT_GPIO_0
7:5	R	Reserved	5b0	Reserved for future use.
4	W/R	Rise edge enable 4	1'b1	'1'- Enables rise edge interrupt detection for UULP_VBAT_GPIO_4 '0'- Disables rise edge interrupt detection for UULP_VBAT_GPIO_4
3	W/R	Rise edge enable 3	1'b1	'1'- Enables rise edge interrupt detection for UULP_VBAT_GPIO_3 '0'- Disables rise edge interrupt detection for UULP_VBAT_GPIO_3
2	W/R	Rise edge enable 2	1'b1	'1'- Enables rise edge interrupt detection for UULP_VBAT_GPIO_2 '0'- Disables rise edge interrupt detection for UULP_VBAT_GPIO_2
1	W/R	Rise edge enable 1	1'b1	'1'- Enables rise edge interrupt detection for UULP_VBAT_GPIO_1 '0'- Disables rise edge interrupt detection for UULP_VBAT_GPIO_1

Bit	Access	Function	Reset Value	Description
0	W/R	Rise edge enable 0	1'b1	'1'- Enables rise edge interrupt detection for UULP_VBAT_GPIO_0 '0'- Disables rise edge interrupt detection for UULP_VBAT_GPIO_0

UULP_GPIO_STATUS_REG

Table 8.12. UULP_GPIO_STATUS_REG Description

Bit	Access	Function	Reset Value	Description
31:5	R	Reserved	27'b0	Reserved for future use.
4	R	GPIO4 status	1'b0	Gives the pin status for UULP_VBAT_GPIO_4
3	R	GPIO3 status	1'b0	Gives the pin status for UULP_VBAT_GPIO_3
2	R	GPIO2 status	1'b0	Gives the pin status for UULP_VBAT_GPIO_2
1	R	GPIO1 status	1'b0	Gives the pin status for UULP_VBAT_GPIO_1
0	R	GPIO0 status	1'b0	Gives the pin status for UULP_VBAT_GPIO_0

M4_WIC_CLEAR_REG

Table 8.13. M4_WIC_CLEAR_REG Description

Bit	Access	Function	Reset Value	Description
31:2	R	Reserved	29'd0	
1	R/W	enable_negedge_ulp	1'b1	Enables the negedge path in ULP Mode. This needs to be programmed before switching to ULP mode. This mode has to be enabled only if we intend to do supply switching for processor in ULP mode. When supply is switched level shifters will be enabled and we must use negedge path to save level snifters power due to combi toggles 0 - Posedge path 1 - Negedge path
0	R/W	Reserved	1'b0	-

M4_ULP_SLP_STATUS_REG

Table 8.14. M4_ULP_SLP_STATUS_REG Description

Bit	Access	Function	Reset Value	Description
31:5	R	Reserved	28'd0	
4	R	ulp_wakeup_por	1'b9	Indicates POR status.
3	R	ulp_mode_switched_npss	1'b0	Indicates the status of Physical switching to ULP Mode operation 0 - PS4 State 1 - PS2 State
2	R	ulp_mode_aftr_clk_sw	1'b0	Indicates the status of functional switching to ULP Mode operation 0 - PS4 state 1 - PS2 state
1	R	RAM RETENTION STATUS	1'b0	Indicates the status of Ram retention on Wakeup 0 - RAM not retained 1 - RAM retained
0	R	ULP WAKEUP	1'b0	Status Indication for Wakeup mode 0 - First Bootup 1 - ULP Wakeup

M4ULP_ISO_ENABLE_REG

Table 8.15. M4ULP_ISO_ENABLE_REG Description

Bit	Access	Function	Reset Value	Description
31:23	R	Reserved	9'd0	
22:16	R/W	ISO_ENABLE_REG_SRAM	7'd0	Enables isolation on outputs of M4 ULP-SRAM Power Domain. This is used on power domain controls bypass mode 0 - Disables isolation 1 - Enables isolation
15:9	R	Reserved	7'd0	
8	R/W	ISO_ENABLE_REG_M4_ROM	1'b0	Enables isolation on outputs of M4 ROM Power Domain. This is used on power domain controls bypass mode 0 - Disables isolation 1 - Enables isolation
7:3	R	Reserved	5'd0	
2	R/W	Reserved	1'b0	Reserved
1	R/W	ISO_ENABLE_REG_M4_DEBUG_FPU	1'b0	Enables isolation on outputs of M4 Debug_FPU Power Domain. This is used on power domain controls bypass mode 0 - Disables isolation 1 - Enables isolation
0	R	Reserved	1'b0	

M4ULP_RST_CTRL_REG

Table 8.16. M4ULP_RST_CTRL_REG Descrption

Bit	Access	Function	Reset Value	Description
31:9	R	Reserved	23'd0	
8	R/W	RS_CTRL_REG_M4_ROM	1'b0	Enables Reset for M4 ROM Power Domain. This is used on power domain controls bypass mode 0 - Out of Reset 1 - In Reset
7:3	R	Reserved	5'd0	
2	R/W	Reserved	1'b0	Reserved
1	R/W	RST_CTRL_REG_M4_DEBUG_FPU	1'b0	Enables Reset for M4 DEBUG_FPU Power Domain. This is used on power domain controls bypass mode 0 - Out of Reset 1 - In Reset
0	R	Reserved	1'b0	

M4ULP_A2A_BRIDG_CTRL_REG

Table 8.17. M4ULP_A2A_BRIDG_CTRL_REG Descrption

Bit	Access	Function	Reset Value	Description
31:4	R	Reserved	28'd0	
3	R/W	mode_change_req_bridge2	1'b0	Mode change Request bit for AHB2AHB bridge 2 Setting of this bit is required along with changing of mode[1:0] bit. 0 - not requesting mode change 1 - requesting mode change
2	R/W	mode_change_req_bridge1	1'b0	Mode change Request bit for AHB2AHB bridge 1 Setting of this bit is required along with changing of mode[1:0] bit. 0 - not requesting mode change 1 - requesting mode change
1:0	R/W	mode	2'd2	Enables mode of operation 2'd1 - synchronous 2'd2 - asynchronous 2'd0,2'd3 -Invalid

M4ULP_A2A_BRIDG_STAT_REG

Table 8.18. M4ULP_A2A_BRIDG_STAT_REG Descrption

Bit	Access	Function	Reset Value	Description
31:2	R	Reserved	30'd0	
1	R	mode_change_ack_bridge2	1'b0	Indicates the status of mode change of ahb2ahb bridge2 0 - inactive 1 - mode change is done
0	R	mode_change_ack_bridge1	1'b0	Indicates the status of mode change of ahb2ahb bridge1 0 - inactive 1 - mode change is done

M4ULP_STCALIB_REG

Table 8.19. M4ULP_STCALIB_REG Descrption

Bit	Access	Function	Reset Value	Description
31:26	R	Reserved	6'd0	Reserved
25	R/W	NOREF	1'd0	STCALIB[25] of M4 Indicates that no alternative reference clock source has been integrated. Tie HIGH if STCLK has been tied off.
24	R/W	SKEW	1'b0	STCALIB[24] of M4 Tie this LOW if the system timer clock, the external reference clock, or FCLK as indicated by STCALIB[25], can guarantee an exact multiple of 10ms. Otherwise, tie this signal HIGH.
23:0	R/W	TENMS	24'h4E200	STCALIB[23:0] of M4. Default value kept for 32Mhz FCLK for 10ms Provides an integer value to compute a 10ms (100Hz) delay from either the reference clock, or FCLK if the reference clock is not implemented. For example, apply the value 0x07A11F if no reference is implemented, and FCLK is 50MHz.

M4ULP_SYSTICK_CLK_ENABLE_REG

Table 8.20. M4ULP_SYSTICK_CLK_ENABLE_REG Descrption

Bit	Access	Function	Reset Value	Description
31:4	R	Reserved	29'd0	Reserved
3	R/W	tsclkchange	1'b0	Timestamp clock ratio change. When M4 core clock frequency is changed, SW has to set this bit to 1'b1 when TRACE is enabled. It will be cleared by hardware after 1 clock cycle.
2	R/W	fclk_systick_clk_sel	1'b0	clock mux to select either fclk or systick clock generated from ULPSS 1: systick clock generated from ULPSS will be connected M4 Systick clock(STCLK) 0: fclk will be connected M4 Systick clock(STCLK) → wont work Note: Program 1 always for external STCLK. Connecting FCLK option via external STCLK, wont work functionally as ARM Systick is running on FCLK always.
1	R/W	m4systick_clk_enable_systikclk_ulpss	1'b0	clock gate enable for systick clock coming from ULPSS 1: systick clock coming from ULPSS is enabled for M4 systick clock(STCLK) 0: systick clock coming from ULPSS is not enabled for M4 systick clock(STCLK) By default this clock is gated.
0	R/W	m4systick_clk_enable_fclk	1'd0	Reserved

M4ULP_SPARE_REG

Table 8.21. M4ULP_SPARE_REG Descrption

Bit	Access	Function	Reset Value	Description
31:16	R	Reserved	29'd0	Reserved
15:0	R/W	Reserved	16'b0	Spare register

MCU ULP Peripheral Interrupts

There are 20 MCU ULP peripheral interrupts are mapped on MCU NVIC. Following table provides the list of MCU ULP peripheral interrupts and their interrupt number in vector interrupt table.

Interrupt Number in VIT	MCu ULP Peripheral interrupt
19	egpio_group interrupt
18	egpio_pin interrupt (OR'ed egpio_pin interrupt)
17	Reserved
16	ssi_mst_interrupt
15	Reserved
14	i2s_interrupt
13	i2c_interrupt
12	uart_interrupt
11	aux_adc_dac_interrupt
10	udma_interrupt
9	ulp_gpio_wakeup_interrupt
8:7	comp_intr (comparator Interrupt)
6	cap_sense_interrupt (Cap_sense_wake_up or Cap_sense_intrerrupt)
5:2	ulp_timer_interrupt_status (Timer0, Timer 1, Timer2, Timer3)
1:0	Reserved

9. Power Architecture

9.1 General Description

SiWx917 achieves ultra low power without compromising on features that have been traditionally considered "power-hungry". Hierarchical partitioning and numerous system and circuit level innovations have been used to achieve ultra low power while retaining high performance capability.

The Power Control Hardware implements the control sequences for transitioning between different power states (Active/Standby/Sleep/Shutdown) and the power control for different Group of Peripherals. In-addition, wakeup from any of the Standby/Sleep/Shutdown states based on hardware events or peripheral interrupts is supported.

The Standby/Sleep/Shutdown states can be reached from Active mode only and through a WFI instruction. Wakeup from Standby/Sleep/Shutdown states is through a hardware event or interrupt (Peripheral or External). The different wakeup interrupts are listed in Section 8. Interrupts

Different SRAM sizes and Peripherals are available in each Active/Sleep states which is described in the functional description.

9.2 Features

- Two integrated buck switching regulators (High performance and ULP) to enable efficient Dynamic Voltage Scaling across wide operating mode currents ranging from <1uA to 300mA
- High performance and ultra-low-power MCU peripheral subsystems and buses.
- Multiple voltage domains with Independent voltage scaling of each domain.
- Fine grained power-gating including peripherals, buses and pads, thereby reducing power consumption when the peripheral/buses/pads are inactive.
- Multiple Active states using "gear-shifting" approach based on processing requirements, thereby reducing power consumption for low-power applications.
- Flexible switching between different Active states with controls from Software.
- Hardware based wakeup from Standby/Sleep/Shutdown states.
- All the peripherals are clock gated by default thereby reducing the power consumption in inactive state.
- Wakeup times are configurable by Software before going into sleep.

9.3 Functional Description

9.3.1 Power Domains

All the Applications, High Speed Interfaces and Peripherals are segregated into multiple power domains to achieve lower current consumption when they are inactive. At reset, all the domains are powered ON.

The programming for power control for PLL Core is described in Clocking section.

The table below describes the different group of peripherals for which power is controlled through software

Table 9.1. List of Power Domains

S.No	Section	Domain Name	Functionality of the Power Domain
1	APPLICATIONS	DEBUG	Debug Functionality for Cortex-M4F-M4F, Floating Point Unit for Cortex-M4F
2		ROM	ROM Core/Interface
3		SRAM	SRAM Banks
4	HIGH SPEED INTERFACE	QSPI_ICACHE	Quad SPI SDR/DDR Flash Interface and ICache for the Cortex-M4F Processor, QSPI2, DCACHE
5	HP-PERIPHERALS	PERI_EFUSE	SPI/SSI Primary, I2C, USART, UDMA Controller, UART, SPI/SSI Secondary, Generic-SPI Primary, Config Timer, Random-Number Generator, CRC Accelerator, I2C, I2S Primary/Secondary, QEI, MCPWM and EFUSE for configuration information, MVP
6			
7		DMA	General Purpose DMA Controller
8		SDIO-SPI	SDIO 2.0 Secondary, SPI Secondary.
9	HIGH SPEED FLASH MEMORY	FLASH-LDO	LDO-FL 1.8 for Flash Memory
10	HIGH-FREQ-PLL	PLL-REGISTERS	PLL Programming Registers for High frequency clocks.
11	ULP-PERIPHERALS	DMA	UDMA Controller
13		Aux ADC-DAC	Aux ADC and DAC Controller
14		I2C	I2C Primary/Secondary
15		SSI	SPI/SSI Primary
16		UART	UART
17		TIMER	Timers

S.No	Section	Domain Name	Functionality of the Power Domain
18	UULP-PERIPHERALS	WDT	Window-Watch Dog Timer
19		TS	Temperature Sensor
20		PS	Process Sensor
21		RTC	Real-Time Clock/ MCU SYSRTC
22		STORAGE-DOMAIN1	Storage Flops - Set1. Contains 8bytes
23		STORAGE-DOMAIN2	Storage Flops - Set2. Contains 8bytes
24		STORAGE-DOMAIN3	Storage Flops - Set3. Contains 16bytes
25		SLEEP-FSM	FSM for Sleep/Wakeup
26		CLOCK-CALIB	Calibration block for Sleep Clock.
27		BBFFS	Programming Registers which can be retained during sleep.
28		DS-TIMER	DEEP SLEEP Timer.
29		TIMESTAMP	Timestamping Controller.
30		LP-FSM	Low-Power FSM
31		CLNDR	Calender
32		RETEN	Retention Flops which can be retained during sleep.
33	Analog-PERIPHERALS	Aux-ADC	Auxillary ADC
34			
35			
36		Aux-DAC	Auxillary DAC
37		BOD-CORR	Brown-Out Detector

The SRAM is also segregated into multiple power domains to achieve lower current consumption as per the Memory requirement. The power for the SRAM domains in active states can be controlled in the following manners

- **Shut-down mode/Deepsleep without retention mode:** SRAM Domains as described in the table below can be powered down for unused SRAM sections. The RAM contents are not retained in this mode
- **Deep-Sleep (Lower power consumption) mode:** The RAM contents are retained in this mode. The SRAM is not accessible in this state. This is configurable on a Bank granularity.

The table below describes the segregation of power domains for SRAM (328KB). The addressing for these banks are described in Section 5. Memory Architecture .

Table 9.2. List of SRAM Power Domains

S.No	Section	Domain Name	Functionality of the Power Domain
1	LP-SRAM	LP-SRAM-1	4KB of SRAM (1x Banks)
2		LP-SRAM-2	4KB of SRAM (1x Banks)
3		LP-SRAM-3	4KB of SRAM (1x Banks)
4		LP-SRAM-4	4KB of SRAM (1x Banks)
5		LP-SRAM-5	16KB of SRAM (1x Banks)
6		LP-SRAM-6	32KB of SRAM (2x Banks)
7		LP-SRAM-7	64KB of SRAM (4x Banks)
8		LP-SRAM-8	64KB of SRAM (4x Banks)
9		LP-SRAM-9	64KB of SRAM (4x Banks)
10		LP-SRAM-10	64KB of SRAM (4x Banks)
11	ULP-SRAM	ULP-SRAM-1	2KB of SRAM (1x Banks)
12		ULP-SRAM-2	2KB of SRAM (1x Banks)
13		ULP-SRAM-3	2KB of SRAM (1x Banks)
14		ULP-SRAM-4	2KB of SRAM (1x Banks)

Also, 352KB SRAM is split into 22 banks of 16K each. Each bank has its own clock gating input to reduce power consumption. The table below describes the segregation of power domains for SRAM (352KB).

S.No	Domain Name	Functionality of the Power Domain
1	LP-SRAM-1	64KB of SRAM (4x Banks)
2	LP-SRAM-2	32KB of SRAM (2x Banks)
3	LP-SRAM-3	32KB of SRAM (2x Banks)
4	LP-SRAM-4	32KB of SRAM (2x Banks)
5	LP-SRAM-5	32KB of SRAM (2x Banks)
6	LP-SRAM-6	64KB of SRAM (4x Banks)
7	LP-SRAM-7	64KB of SRAM (4x Banks)
8	LP-SRAM-8	32KB of SRAM (2x Banks)

9.3.2 Programming Sequence

The power for the above domains except for SRAM can be controlled as described below

- APPLICATIONS, HIGH-SPEED INTERFACES and HP-PERIPHERALS
  - Program the particular bit as shown in Section 9.9.3 M4SS_PWRCTRL_SET_REG and Section 9.9.4 M4SS_PWRCTRL_CLEAR_REG Register.
- HIGH SPEED FLASH MEMORY

  Program the particular bit as shown in Section 9.9.15 MCU_PMU_LDO_CTRL_SET and Section 9.9.16 MCU_PMU_LDO_CTRL_CLEAR Register
- HIGH-FREQ-PLL
  - Program the particular bit as shown in Section 9.9.5 M4SS_PLL_PWRCTRL_REG Register.
- ULP-PERIPHERALS
  - Program the particular bit as shown in Section 9.9.17 ULPSS_PWRCTRL_SET_REG and 9.9.18 ULPSS_PWRCTRL_CLEAR_REG Register.
- UULP-PERIPHERALS
  - Program the particular bit as shown in Section 9.9.25 UULP_PWRCTRL_SET and 9.9.26 UULP_PWRCTRL_CLEAR Register.
  - Program the particular bit as shown in Section 9.9.42 MCU_FSM_CRTL_PDM_AND_ENABLES Register.
- Analog-PERIPHERALS
  - Program the particular bit as shown in Section 9.9.55 Analog_Power_Control Register.

The programming for the power controls of the SRAM domains are described below

- SRAM Domain Power Up
  - For LP-SRAM, configure the particular bit as shown in Section 9.9.6 M4_SRAM_PWRCTRL_SET_REG1 and 9.9.8 M4_SRAM_PWRCTRL_SET_REG2 Register.
  - For ULP-SRAM, configure the particular bit as shown in Section 9.9.19 ULPSS_RAM_PWRCTRL_REG1_SET and 9.9.23 ULPSS_RAM_PWRCTRL_REG3_SET Register.
  - Both the Registers need to be configured for achieving the functionality.
- SRAM Domain Power Down
  - For LP-SRAM, configure the particular bit as shown in Section 9.9.7 M4_SRAM_PWRCTRL_CLEAR_REG1 and Section 9.9.9 M4_SRAM_PWRCTRL_CLEAR_REG2 Register.
  - For ULP-SRAM, configure the particular bit as shown in Section 9.9.20 ULPSS_RAM_PWRCTRL_REG1_CLEAR and 9.9.24 ULPSS_RAM_PWRCTRL_REG3_CLEARRegister.
  - Both the Registers need to be configured for achieving the functionality.
- SRAM Standby
  - For LP-SRAM, configure the particular bit as shown in Section 9.9.12 M4_SRAM_PWRCTRL_SET_REG4 Register. This has to be cleared for SRAM accesses by configuring the particular bit as shown in Section 9.9.13 M4_SRAM_PWRCTRL_CLEAR_REG4 Register.
  - For ULP-SRAM, configure the particular bit in 9.9.21 ULPSS_RAM_PWRCTRL_REG2_SET Register. This has to be cleared for SRAM accesses by configuring the particular bit in 9.9.22 ULPSS_RAM_PWRCTRL_REG2_CLEAR Register.

The details of the above Registers are provided in the Register Description Section below.

9.4 Voltage Domains

All the Applications, High Speed Interfaces and Peripherals are segregated into multiple voltage domains to configure the operating voltages in different power states. This section describes the voltage domains and voltage source options available for each domain. These are configured based on the Power state which the device is operating in. The voltage for each domain can be shut-off during sleep by configuring the source to LDO SoC 1.15 (This supply is turned OFF during Sleep).

The table below lists down the different voltage sources and the possible output voltages of each source at different Power states. The voltage sources are described in detail in the Power Management Section.

Table 9.3. List of Voltage Sources

S.No	Voltage Source	Possible O/P Voltage
1	LDO SoC 1.15	1.15V 1.05V
2	SC-DC 1.05	1.05V
3	LDO 0.75V	0.75V

The table below lists down the different voltage domains and the possible voltage sources for each domain. The voltage source for each domain in different power-states are defined in Section 9.5 Power States section below.

Table 9.4. List of Voltage Domains

S.No	Voltage Domain	Functionality	LDO SoC 1.15	SC-DC 1.05	LDO 0.75V
1	PROC-DOMAIN	Processor, DEBUG	Yes	Yes	Yes
2	HIGH-VOLTAGE-DOMAIN	ICACHE, HIGH-SPEED-INTERFACES, HP-PERIPHERALS, DCACHE	Yes	No	No
3	LOW-VOLTAGE-LPRAM-16KB	LP-SRAM-1, LP-SRAM-2, LP-SRAM-3, LP-SRAM-4,	Yes	Yes	No
4	LOW-VOLTAGE-LPRAM	ROM LP-SRAM-5, LP-SRAM-6, LP-SRAM-7, LP-SRAM-8, LP-SRAM-9, LP-SRAM-10	Yes	Yes	No
5	LOW-VOLTAGE-ULPPERIPH	ULP-PERIPHERALS	Yes	Yes	No
6	LOW-VOLTAGE-ULPRAM	ULP-SRAM	Yes	Yes	No
7	LOW-VOLTAGE-UULPPERIPH	UULP-PERIPHERALS	No	Yes	No

9.5 Power States

The power states available in different power modes (PS0, PS1*, PS2, PS3, PS4) are listed below

- Reset State
- Active States
    - Power State1 (PS1*)
    - Power State2 (PS2)
    - Power State3 (PS3)
    - Power State4 (PS4)
- Standby States
    - PS2-STANDBY
    - PS3-STANDBY
    - PS4-STANDBY
- Sleep States
    - PS2-SLEEP
    - PS3-SLEEP
    - PS4-SLEEP
- Shutdown States (Deep Sleep state)
    - Power State0 (PS0)

After reset, the processor starts in PS4 state which is the highest activity state where the full functionality is available. The other Active states (PS2/PS3) will have limited functionality or Processing power.

A transition from Active states (PS2/PS3/PS4) to any other state can only be triggered by software.

A transition from Standby/Sleep/Shutdown states can be triggered by an enabled interrupt as configured by software before entering these states.

A transition from Standby/Sleep to Active state is possible from where these states are entered.

The figure below shows the transitions between different power states. The programming sequence for each transition is mentioned in the Transition sections below.

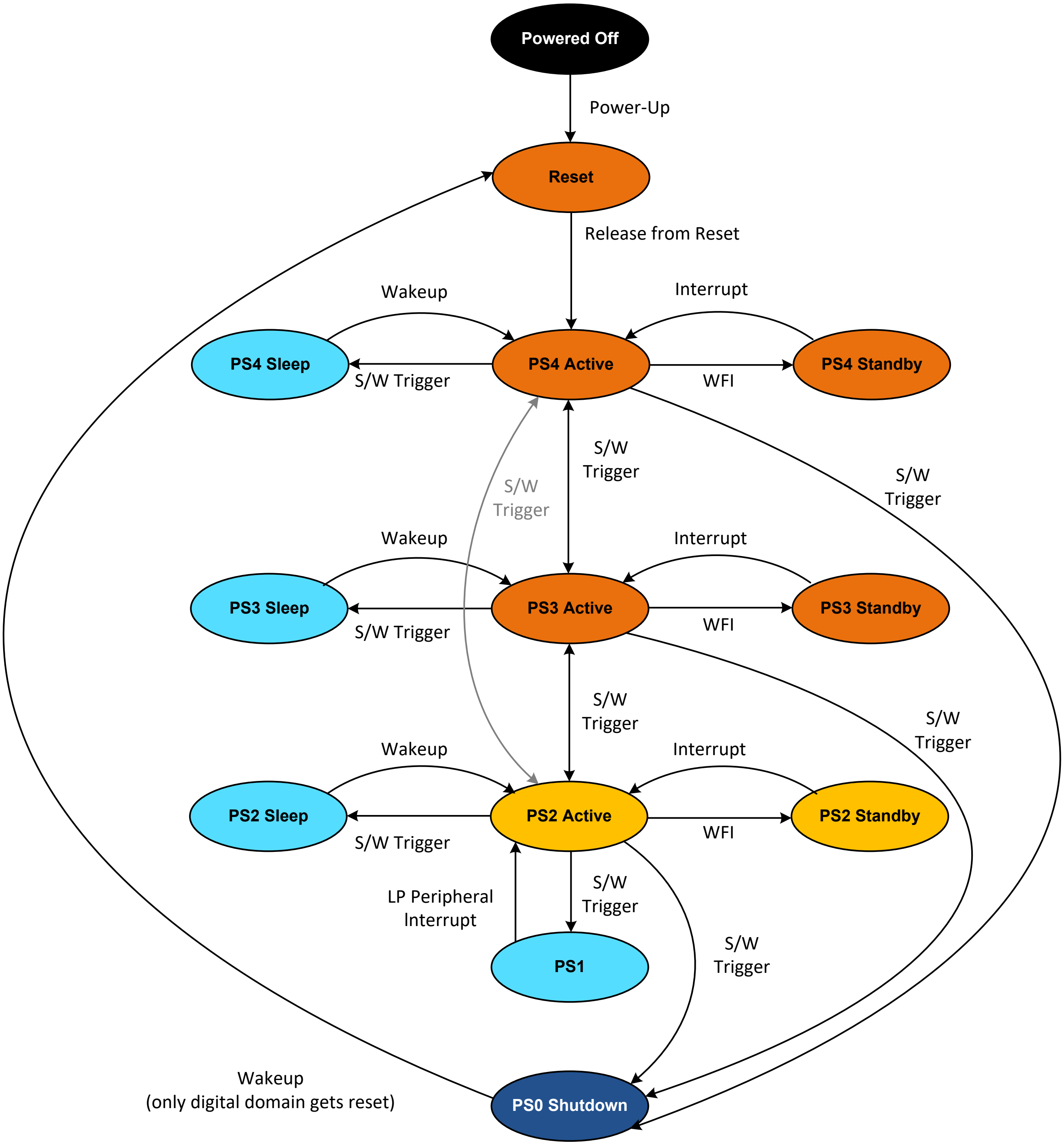


Figure 9.1. Power State Transitions

Note: Wakeup event from PS0, doesn't reset the complete system, instead only the digital domain undergoes a reset.

9.5.1 PS4

This is an Active state where the complete functionality is available. The CPU, Peripherals and SRAM operate on the LDO SoC 1.15V Supply at voltage of 1.15V.

The functionality available in this state are mentioned below:

- Maximum CPU Operating frequency of 180MHz. The CPU can operate on the HIGH-FREQ-PLL output clocks.
- APPLICATIONS - DEBUG, FPU, ICACHE and ROM.
- HIGH SPEED INTERFACE - as listed in Table 9.1 List of Power Domains on page 168.
- HIGH-FREQ-PLL - as listed in the Table 9.1 List of Power Domains on page 168.
- All the Peripherals consisting of HP-PERIPHERALS, ULP-PERIPHERALS, UULP-PERIPHERALS and Analog-PERIPHERALS - as listed in Section 9.3.1 Power Domains .
- All the GPIOs - 30 (GPIO) + 11 (ULP-GPIO) + 4 (UULP Vbat GPIO)
- Complete SRAM of 328KB (LP-SRAM and ULP-SRAM).

9.5.2 PS3

This is an Active state where the complete functionality is available similar to PS4 state and operates at a lower voltage thereby reducing current consumption. The CPU, Peripherals and SRAM operate on the LDO SoC 1.15 Supply with output voltage of 1.05V. The Maximum CPU frequency is limited to 90MHz in this state.

9.5.3 PS2

This is an Active state where a limited set of functionality is available and operates a much lower voltage compared to PS3/PS4 thereby achieving lower current consumption. The CPU, Peripherals and SRAM can operate at different voltages and are configurable by software before entering this state.

The functionality available in this state are mentioned below:

- CPU Operating frequency depends on the voltage source selected for PS2 state. The CPU operates on the ULP-Peripheral AHB Interface clock which is described in Section 6.13 MCU ULP Clock Architecture.
  - If LDO 0.75V is used, Maximum frequency is 20MHz.
  - If SC-DC 1.05 V is used, Maximum frequency is 32MHz.
- APPLICATIONS - DEBUG, FPU and ROM.
- Limited peripherals consisting of ULP-PERIPHERALS, UULP-PERIPHERALS and Analog-PERIPHERALS - as listed in the Table 9.1 List of Power Domains on page 168.
- 15 GPIOs are available - 11 (ULP-GPIO) + 4 (UULP Vbat GPIO)
- Total SRAM of 328KB (LP-SRAM and ULP-SRAM).

9.5.4 PS1

This state can be entered from PS2 only through a Software Instruction. The CPU is power-gated and a limited set of peripherals are active. The peripheral interrupts are used as wakeup source or to trigger sleep once the peripheral functionality is complete. The Peripherals and SRAM operate at the same voltage as PS2 state. The peripherals need to be configured by the Software for the defined functionality in the PS2 state before entering this state.

The functionality available in this state are mentioned below:

- Limited peripherals consisting of ULP-PERIPHERALS, UULP-PERIPHERALS and Analog-PERIPHERALS - as listed in the Table 9.1 List of Power Domains on page 168.
- 15 GPIO's are available - 11 (ULP-GPIO) + 4 (UULP Vbat GPIO)
- SRAM of 320KB (LP-SRAM) can be retained in this state.
- SRAM of 8KB (ULP-SRAM) is active for Peripheral functionality.

9.5.5 STANDBY

This includes multiple states like PS4-STANDBY, PS3-STANDBY and PS2-STANDBY. These are Standby states entered from PS4/PS3/PS2 state through a WFI instruction. CPU is clock gated in this state.

All the Interrupts in the NVIC table as described in Section 8. Interrupts will act as a wakeup source in PS4-STANDBY and PS3-STANDBY state.

9.5.6 SLEEP

This includes multiple states like PS4-SLEEP PS3-SLEEP and PS2-SLEEP which can be entered from PS4, PS3 and PS2 state respectively through a software instruction. The CPU is power-gated and a much lower set of peripherals are available.

The status of resources in this state are:

- UULP-PERIPHERALS and BOD are available and are configured before entering this state.
- 4 UULP Vbat GPIO
- SRAM can be retained.

9.5.7 PS0

This is a Shutdown state (Deep Sleep state) entered from PS4 state through a Software instruction. The CPU is power-gated and a much smaller set of peripherals are available.

The status of resources in this state are

- UULP-PERIPHERALS and Analog-PERIPHERALS are available and are configured before entering this state.
- 4 UULP Vbat GPIO
- SRAM can not be retained.

9.5.8 Programming Sequence for Transitions

PS4 -> PS3

The programming sequence for this transitions is described below

1. Configure the LDO SoC 1.15 output voltage to a lower voltage of 1.05V.
   a. Program corresponding bit in 9.9.16 MCU_PMU_LDO_CTRL_CLEAR Register.
2. Configure the DC-DC 1.45 output voltage to a lower voltage of 1.25V
   a. Program corresponding bit in 9.9.16 MCU_PMU_LDO_CTRL_CLEAR Register.

PS3 -> PS4

The programming sequence for this transitions is described below

1. Configure the LDO SoC 1.15 output voltage to a lower voltage of 1.15V.
   a. Program corresponding bit in 9.9.15 MCU_PMU_LDO_CTRL_SET Register.
2. Configure the DC-DC 1.45 output voltage to a lower voltage of 1.45V
   a. Program corresponding bit in 9.9.15 MCU_PMU_LDO_CTRL_SET Register.

PS4/PS3 -> PS2

This includes the following transitions and the programming sequence is same for all these transitions.

- PS4 -> PS2
- PS3 -> PS2

The programming sequence for these transitions is described below

1. Switch the Processor clock to MCUHP_REF_CLK as described in Section 6.12 MCU HP Clock Architecture.
2. The following settings need to be configured for PS4-PS2 transition. The switching times between PS4<->PS2 is determined by the state of the LDO SoC 1.15 and DC-DC 1.45. The OFF delays for these are described in the Power Management section
   a. Configure the voltage source for the domains listed in Section 9.4 Voltage Domains above as per the requirement in PS2 state.
      i. The voltage sources need to be selected as per the CPU performance required.
      ii. These can be configured through VOLTAGE_SEL_ULP_SRAM,VOLTAGE_SEL_LP_SRAM, VOLTAGE_SEL_LP_SRAM_16KB, VOLTAGE_SEL_PROC and VOLTAGE_SEL_ULP_PERIPH parameters in 9.9.35 MCU_FSM_POWER_CTRL_AND_DELAY Register.
   b. The LDO SoC 1.15 and DC-DC 1.45 state during PS2 state.
      i. These supplies needs to be configured to ON state during PS2 to achieve lower switching time from PS2 to PS4/PS3 states
      ii. These can be configured through DCDC_EN and LDoSoC_EN parameters in 9.9.35 MCU_FSM_POWER_CTRL_AND_DELAY Register.
3. Configure the ULP AHB Interface clock source and output frequency as described in Section 6.13 MCU ULP Clock Architecture.
4. Dummy read for flushing the pending transactions of ULP-PERIPHERAL accesses.
5. If the required SRAM is less than 32KB, then a lower current consumption can be achieved by configuring the ULP MODE (Refer to 9.9.34 MCU_FSM_PERI_CONFIG_REG Register).
6. Enable functional switching to PS2 state (Refer to 9.9.34 MCU_FSM_PERI_CONFIG_REG Register).
7. Poll for the status bit for the functional switching done above (Refer M4_ULP_SLP_STATUS_REG Register described in Section 8. Interrupts section).
8. Disable the clock used for ULP-PERIPHERAL accesses in PS4 state
   a. Program MCUULP_BRIDGE_CLK_EN in MCUULP_PROC_CLK_CONFIG Register described in Section 6.13 MCU ULP Clock Architecture.
9. Enable Isolation for all the interfaces with HIGH-VOLTAGE-DOMAIN which are not operational in PS2 state.
10. Enable physical switching for PS2 state (Refer to 9.9.34 MCU_FSM_PERI_CONFIG_REG Register described below).
11. Poll for the status bit for the voltage switching done above (Refer M4_ULP_SLP_STATUS_REG Register described in Section 8. Interrupts).

CPU will be operating on MCU-ULP AHB Interface Clock as described in Section 6.12 MCU HP Clock Architecture once we switch to PS2 state.

PS2 -> PS4/PS3

This includes the following transitions and the programming sequence is same for all these transitions.

- PS4 -> PS2

- PS3 -> PS2

The programming sequence for these transitions is described below

1. Disable the clock used for ULP-PERIPHERAL accesses in PS4 state
   a. Program MCUULP_BRIDGE_CLK_EN in MCUULP_PROC_CLK_CONFIG Register described in Section 6.13 MCU ULP Clock Architecture .
2. The following settings need to be configured for PS2-PS4 transition (Refer 9.9.35 MCU_FSM_POWER_CTRL_AND_DELAY Register described below). The switching times from PS2 to PS4 is determined by the state of the LDO SoC 1.15 and DC-DC 1.45 configured when entering the PS2 state. The ON delays for these are described in the Power Management section
   a. Configure the voltage source for the domains listed in Section 9.4 Voltage Domains above to LDO-SOC 1.15.
      i. These can be configured through VOLTAGE_SEL_ULP_SRAM, VOLTAGE_SEL_LP_SRAM, VOLTAGE_SEL_LP_SRAM_16KB, VOLTAGE_SEL_PROC and VOLTAGE_SEL_ULP_PERIPH parameters in 9.9.35 MCU_FSM_POWER_CTRL_AND_DELAY Register.
   b. The ON times for DC-DC 1.45 and LDO-SoC 1.15 needs to be configured based on their state before entering PS2 state.
      i. These can be configured to lowest possible value if there are maintained in ON state during PS2.
      ii. These can be configured through DCDC_ON_TIME and LDoSoC_ON_TIME parameters in 9.9.35 MCU_FSM_POWER_CTRL_AND_DELAY Register.
3. Switch the CPU clock to Reference clock as described in Section 6.12 MCU HP Clock Architecture.
4. Enable voltage switching for PS4 state (Refer to 9.9.34 MCU_FSM_PERI_CONFIG_REG Register).
5. Poll for the status bit for the voltage switching done above (Refer M4_ULP_SLP_STATUS_REG Register described in Section 8. Interrupts).
6. Disable Isolation for all the interfaces with HIGH-VOLTAGE Domains.
7. Disable the clock used for ULP-PERIPHERAL accesses in PS4 state
   a. Program MCUULP_BRIDGE_CLK_EN in MCUULP_PROC_CLK_CONFIG Register described in Section 6.13 MCU ULP Clock Architecture.
8. Enable functional switching to PS4 state (Refer to 9.9.34 MCU_FSM_PERI_CONFIG_REG Register described below).
9. Poll for the status bit for the functional switching done above (Refer M4_ULP_SLP_STATUS_REG Register described in Section 8. Interrupts).

CPU will be operating on Reference Clock as described in Section6.12 MCU HP Clock Architecture once we switch PS3/PS4 state.

PS2 -> PS1*

The programming sequence for this transition is described below. One of the ULP-Peripheral or Sensor-Data-Collector needs to configured for the required activity before initiating the transition to PS1.

1. Configure any of the ULP-Peripheral or Sensor-Data-Collector as wakeup source for transition from PS2 to PS1
   a. This can be configured through BITS(29:16) in 9.9.33 MCU_FSM_SLEEP_CTRLS_AND_WAKEUP_MODE Register described below.
2. Configure the SRAM domains to be retained during PS1 state
   a. This can be configured through BITS[7:3] in 9.9.33 MCU_FSM_SLEEP_CTRLS_AND_WAKEUP_MODE Register described below.
3. Enter Sleep state. Please refer to the "Powersave" section in the "siwx91x-software-reference-manual" present at https://github.com/SiliconLabs/wiseconnect/blob/master/docs/software-reference/manuals/siwx91x-software-reference-manual.md for more details.

ACTIVE -> SLEEP

This includes the following transitions and the programming sequence is same for all these transitions. The state of the LDO SoC 1.15, LDO FL 1.8 and DC-DC 1.45 during SLEEP state can be configured for achieving lower wakeup time from Sleep to Active state. The Power Domains will remain in the same state (Power-Up/Power-Down) upon Wake-up from Sleep.

- PS4 -> PS4-SLEEP (Switches to PS4 upon wakeup)
- PS3 -> PS3-SLEEP (Switches to PS3 upon wakeup)
- PS2 -> PS2-SLEEP (Switches to PS2 upon wakeup)

The programming sequence for these transitions is described below

1. Configure the state of LDO SoC 1.15, LDO FL 1.8 and DC-DC 1.45 during Sleep state
   a. This can be configured through BITS[11:8] in 9.9.33 MCU_FSM_SLEEP_CTRLS_AND_WAKEUP_MODE Register described below.

2. Configure the SRAM domains to be retained during Sleep state
   a. This can be configured through BITS[7:3] in 9.9.33 MCU_FSM_SLEEP_CTRLS_AND_WAKEUP_MODE Register described below.
3. Configure the wakeup source for transition from SLEEP state (PS4-SLEEP, PS3-SLEEP, PS2-SLEEP) to ACTIVE state (PS4, PS3, PS2)
   a. This can be configured through BITS[29:16] in 9.9.33 MCU_FSM_SLEEP_CTRLS_AND_WAKEUP_MODE Register described below.
4. Configure the ON times for DC-DC 1.45 and LDO SoC 1.15 if they are configured to OFF state during Sleep.
   a. This can be configured through PMU_POWERGOOD_TIME in 9.9.38 MCU_FSM_XTAL_AND_PMU_GOOD_COUNT_REG Register described below.
   b. The amount of ON time for DC-DC 1.45 and LDO SoC 1.15 are described in Section 10. Power Management Unit
5. Configure the ON time for HF-Crystal clock
   a. This can be configured through HF_CRYSTAL_SETTLING_TIME in 9.9.38 MCU_FSM_XTAL_AND_PMU_GOOD_COUNT_REG Register described below.
6. The ON time for HF-Crystal clock can be skipped to reduce the wakeup time if the MCUHP_REF_CLK is not configured to XTAL_CLK before entering Sleep state as described in Section 6. Clock Architecture.
   a. This can be configure through SKIP_XTAL_WAIT_TIME in 9.9.33 MCU_FSM_SLEEP_CTRLS_AND_WAKEUP_MODE Register described below.
7. Enter Sleep state. Please refer to the "Powersave" section in the "siwx91x-software-reference-manual" present at https://github.com/SiliconLabs/wiseconnect/blob/master/docs/software-reference/manuals/siwx91x-software-reference-manual.md for more details.

PS4/PS3 -> PS0

This includes the following transitions and the programming sequence is same for all these transitions. The state of the LDO SoC 1.15, LDO FL 1.8 and DC-DC 1.45 during SLEEP state can be configured for achieving lower wakeup time from Sleep to Active state. The Power Domains will remain in the same state (Power-Up/Power-Down) upon Wake-up from Sleep.

- PS4 -> PS0 (Switches to PS4 upon wakeup)
- PS3 -> PS0 (Switches to PS3 upon wakeup)

The programming sequence for this transition is described below

1. Reset the state of LDO SoC 1.15, LDO FL 1.8 and DC-DC 1.45 during Sleep state
   a. This can be configured through BITS[11:8] in 9.9.33 MCU_FSM_SLEEP_CTRLS_AND_WAKEUP_MODE Register described below.
2. Reset the SRAM domains to be retained during Sleep state
   a. This can be configured through BITS[7:3] in 9.9.33 MCU_FSM_SLEEP_CTRLS_AND_WAKEUP_MODE Register described below.
3. Configure the wakeup source for transition from SLEEP state (PS4-SLEEP, PS3-SLEEP, PS2-SLEEP) to ACTIVE state (PS4, PS3, PS2)
   a. This can be configured through BITS[29:16] in 9.9.33 MCU_FSM_SLEEP_CTRLS_AND_WAKEUP_MODE Register described below.
4. Configure the ON times for DC-DC 1.45 and LDO SoC 1.15 if they are configured to OFF state during Sleep.
   a. This can be configured through PMU_POWERGOOD_TIME in 9.9.38 MCU_FSM_XTAL_AND_PMU_GOOD_COUNT_REG Register described below.
   b. The amount of ON time for DC-DC 1.45 and LDO SoC 1.15 are described in Section 10. Power Management Unit
5. Configure the ON time for HF-Crystal clock
   a. This can be configured through HF_CRYSTAL_SETTLING_TIME in 9.9.38 MCU_FSM_XTAL_AND_PMU_GOOD_COUNT_REG Register described below.
6. The ON time for HF-Crystal clock can be skipped to reduce the wakeup time if the MCUHP_REF_CLK is not configured to XTAL_CLK before entering Sleep state as described in Section 6. Clock Architecture.
   a. This can be configure through SKIP_XTAL_WAIT_TIME in 9.9.33 MCU_FSM_SLEEP_CTRLS_AND_WAKEUP_MODE Register described below.
7. Enter Sleep state. Please refer to the "Powersave" section in the "siwx91x-software-reference-manual" present at https://github.com/SiliconLabs/wiseconnect/blob/master/docs/software-reference/manuals/siwx91x-software-reference-manual.md for more details.

PS2 -> PS0

The Power Domains will be switched to reset state upon Wakeup from Sleep.

The programming sequence for this transition is described below

1. Reset the SRAM Voltage domains retention state
   a. This can be configured through BITS[7:3] in 9.9.33 MCU_FSM_SLEEP_CTRLS_AND_WAKEUP_MODE Register described below.
2. Configure the wakeup source for transition from PS0 to PS4/PS3
   a. This can be configured through BITS[29:16] in 9.9.33 MCU_FSM_SLEEP_CTRLS_AND_WAKEUP_MODE Register described below.
3. Enable Reset of Power Domain Control Battery FFs
   a. This can be configured through RESET_BFF_EN in 9.9.42 MCU_FSM_CRTL_PDM_AND_ENABLES Register described below.
4. Configure the ON times for DC-DC 1.45 and LDO SoC 1.15 if they are configured to OFF state during Sleep.
   a. This can be configured through PMU_POWERGOOD_TIME in 9.9.38 MCU_FSM_XTAL_AND_PMU_GOOD_COUNT_REG Register described below.
   b. The amount of ON time for DC-DC 1.45 and LDO SoC 1.15 are described in Section 10. Power Management Unit
5. Configure the ON time for HF-Crystal clock
   a. This can be configured through HF_CRYSTAL_SETTLING_TIME in 9.9.38 MCU_FSM_XTAL_AND_PMU_GOOD_COUNT_REG Register described below.
6. The ON time for HF-Crystal clock can be skipped to reduce the wakeup time if the MCUHP_REF_CLK is not configured to XTAL_CLK before entering Sleep state as described in Section 6. Clock Architecture.
   a. This can be configure through SKIP_XTAL_WAIT_TIME in 9.9.33 MCU_FSM_SLEEP_CTRLS_AND_WAKEUP_MODE Register described below.
7. Enter Sleep state. Please refer to the "Powersave" section in the "siwx91x-software-reference-manual" present at https://github.com/SiliconLabs/wiseconnect/blob/master/docs/software-reference/manuals/siwx91x-software-reference-manual.md for more details.

9.6 Blocks Availability in Different Power States

The table below indicates the Peripherals, SRAM banks available in each power states.

Table 9.5. Peripherals and SRAM in Different States

S.No	Block	PS4	PS3	PS2	PS1	PS0	PS4 Sleep	PS3 Sleep	PS2 Sleep	PS4 Standby	PS3 Standby	PS2 Standby
1	HP-Peripherals	ON	ON	OFF	OFF	OFF	OFF	OFF	OFF	ON	ON	OFF
2	ULP-Peripherals	ON	ON	ON	OFF	OFF	OFF	OFF	OFF	ON	ON	ON
3	UULP-Peripherals	ON	ON	ON	ON	ON	ON	ON	ON	ON	ON	ON
4	Analog-Peripherals	ON	ON	ON	ON	ON	ON	ON	ON	ON	ON	ON
5	SoC GPIOs	ON	ON	OFF	OFF	OFF	OFF	OFF	OFF	ON	ON	OFF
6	ULP-GPIOs	ON	ON	ON	ON	OFF	OFF	OFF	OFF	ON	ON	ON
7	UULP Vbat GPIOs	ON	ON	ON	ON	ON	ON	ON	ON	ON	ON	ON
8	LP_SRAM (320k)	ON	ON	ON	RET	OFF	RET	RET	RET	ON	ON	ON
9	ULP-RAMS(8k)	ON	ON	ON	ON	OFF	RET	RET	RET	ON	ON	ON

9.7 Memory Retention in Sleep / Shutdown states

The table below indicates the SRAM banks and Backup Register Array which can be retained in each Sleep/Shutdown state.

Table 9.6. SRAM in Different SStates

S.No	Power State	LP-SRAM (320 KB)	ULP-SRAM (8 KB)	Backup Register Array (32 bytes)
1	PS4-SLEEP	Yes	Yes	Yes
2	PS3-SLEEP	Yes	Yes	Yes
3	PS2-SLEEP	Yes	Yes	Yes
4	PS1	Yes	Yes	Yes
5	PS0	No	No	Yes

9.8 Register Summary

9.8.1 High-Performance Power Domains

This includes the APPLICATIONS, HIGH-SPEED-INTERFACES, HP-PERIPHERALS, HIGH-FREQ-PLL, DDR-FLASH-DLL, HP-SRAM and LP-SRAM.

Base Address: 0x2404_8400

Table 9.7. High-Power Domains Control Registers

Register Name	Offset	Description
Section 9.9.1 M4SS_BYPASS_PWRCTRL_REG1	0x00	Enables software based isolation and reset control for M4SS core, memories, peripherals,
Section 9.9.2 M4SS_BYPASS_PWRCTRL_REG2	0x04	Enables software based isolation and reset control for M4SS SRAM1 and SRAM2
Section 9.9.3 M4SS_PWRCTRL_SET_REG	0x08	Enables power for APPLICATIONS, HIGH SPEED INTERFACES, HP-PERIPHERALS domains
Section 9.9.4 M4SS_PWRCTRL_CLEAR_REG	0x0C	Disables power for APPLICATIONS, HIGH SPEED INTERFACES, HP-PERIPHERALS domains
Section 9.9.6 M4_SRAM_PWRCTRL_SET_REG1	0x10	Enables power for HP-SRAM1, HP-SRAM2 and LP-SRAM domains
Section 9.9.7 M4_SRAM_PWRCTRL_CLEAR_REG1	0x14	Disables power for HP-SRAM, HP-SRAM2 and LP-SRAM domains
Section 9.9.8 M4_SRAM_PWRCTRL_SET_REG2	0x18	Enables power for HP-SRAM, HP-SRAM2 and LP-SRAM domains
Section 9.9.9 M4_SRAM_PWRCTRL_CLEAR_REG2	0x1C	Disables power for HP-SRAM, HP-SRAM2 and LP-SRAM domains
Section 9.9.10 M4_SRAM_PWRCTRL_SET_REG3	0x20	Disables isolation on HP-SRAM, HP-SRAM2 and LP-SRAM input signals
Section 9.9.11 M4_SRAM_PWRCTRL_CLEAR_REG3	0x24	Enables isolation on HP-SRAM, HP-SRAM2 and LP-SRAM input signals
Section 9.9.12 M4_SRAM_PWRCTRL_SET_REG4	0x28	Enables Deep-Sleep for HP-SRAM, HP-SRAM2 and LP-SRAM domains
Section 9.9.13 M4_SRAM_PWRCTRL_CLEAR_REG4	0x2C	Disables Deep-Sleep for HP-SRAM, HP-SRAM2 and LP-SRAM domains
Section 9.9.14 MCU_FSM_CTRL_BYPASS	0x64	Control Register for bypassing Sleep-FSM controls
Section 9.9.15 MCU_PMU_LDO_CTRL_SET	0x68	Control Register for PMU Supply Voltages
Section 9.9.16 MCU_PMU_LDO_CTRL_CLEAR	0x6C	Control Register for PMU Supply Voltages
Section 9.9.5 M4SS_PLL_PWRCTRL_REG	0x80	Controls power for HIGH-FREQ-PLL domains

9.8.2 Low-Power Domains

This includes the ULP-PERIPHERALS and ULP-SRAM.

Base Address: 0x2404_8400

Table 9.8. Low-Power Domains Control Registers

Register Name	Offset	Description
Section 9.9.17 ULPSS_PWRCTRL_SET_REG	0x44	Enables power for ULP-PERIPHERALS domains
Section 9.9.18 ULPSS_PWRCTRL_CLEAR_REG	0x48	Disables power for ULP-PERIPHERALS domains
Section 9.9.19 ULPSS_RAM_PWRCTRL_REG1_SET	0x4C	Enables power for ULPSS SRAM domains
Section 9.9.20 ULPSS_RAM_PWRCTRL_REG1_CLEAR	0x50	Disables power for ULPSS SRAM domains
Section 9.9.21 ULPSS_RAM_PWRCTRL_REG2_SET	0x54	Enables Deep-Sleep and input isolation for ULP-TASS SRAM domains
Section 9.9.22 ULPSS_RAM_PWRCTRL_REG2_CLEAR	0x58	Disables Deep-Sleep and input isolation for ULP-TASS SRAM domains
Section 9.9.23 ULPSS_RAM_PWRCTRL_REG3_SET	0x5C	Enables power for ULPTASS SRAM Dual Rail pins
Section 9.9.24 ULPSS_RAM_PWRCTRL_REG3_CLEAR	0x60	Disables power for ULPTASS SRAM Dual Rail pins

9.8.3 Ultra Low-Power Domains

This includes UULP-PERIPHERALS.

Base Address: 0x2404_8000

Table 9.9. Ultra Low-Power Domains Control Registers

Register Name	Offset	Description
Section 9.9.25 UULP_PWRCTRL_SET	0x00	Enables power for UULP-PERIPHERALS domains
Section 9.9.26 UULP_PWRCTRL_CLEAR	0x04	Disables power for UULP-PERIPHERALS domains
RESERVED	0x08	--
Section 9.9.27 MCUAON_IPMU_RESET_CTRL	0x0C	Resets IPMU SPI and ULP Analog SPI
Section 9.9.28 MCUAON_SHELF_MODE	0x10	Configures Shelf mode and wake up of Chip
Section 9.9.29 MCUAON_GEN_CTRLS	0x14	Gives NPSS power supply, ULP analog wake up access and NPSS GPIO clock
Section 9.9.30 MCUAON_PDO_CTRLS	0x18	Turns OFF IO supply for SDIO, QSPI and SOC domains
Section 9.9.31 MCUAON_WDT_CHIP_RST	0x1C	Gices Power ON Reset and NON Power On Reset
Section 9.9.32 MCUAON_KHZ_CLK_SEL_POR_RESET_STATUS	0x20	Selects NPSS AON KHz clk and gives Power ON Reset Status

9.8.4 Analog Domains

This includes ANALOG-PERIPHERALS.

Base Address: 0x2404_A508

Table 9.10. Analog Power Domains Control Registers

Register Name	Offset	Description
Section 9.9.55 Analog_Power_Control	0x00	Controls the power for Analog Power domains

9.8.5 SLEEP FSM

Base Address: 0x2404_8100

Table 9.11. SLEEP FSM Registers

Register Name	Offset	Description
Section 9.9.33 MCU_FSM_SLEEP_CTRLS_AND_WAKEUP_MODE	0x00	Sleep Control Signals and Wakeup source selection
Section 9.9.34 MCU_FSM_PERI_CONFIG_REG	0x04	Configuration for Ultra Low-Power Mode of the processor (PS2 State)
Section 9.9.36 GPIO_WAKEUP_REGISTER	0x08	GPIO based wake up controls
Section 9.9.37 MCU_FSM_DEEP_SLEEP_DURATION_LSB_REG	0x0C	Configuration for LSB bits of deep sleep duration counter
Section 9.9.38 MCU_FSM_XTAL_AND_PMU_GOOD_COUNT_REG	0x10	Configuration Register for PMU and HF-Crystal ON time
Section 9.9.35 MCU_FSM_POWER_CTRL_AND_DELAY	0x14	Power Control and Delay Configuration for Ultra Low-Power Mode of the processor (PS2 State)
Section 9.9.39 MCU_FSM_CLKS_REG	0x18	Configuration register for high frequency fsm clock enable and select
Section 9.9.40 MCU_FSM_REF_CLK_REG	0x1C	Configuration register for reference clock select
Section 9.9.41 MCU_FSM_CLK_ENS_AND_FIRST_BOOTUP	0x20	Configuration register for FSM clock enables, first bootup and chip mode valid
Section 9.9.42 MCU_FSM_CRTL_PDM_AND_ENABLES	0x24	Power Domains Controlled by Sleep FSM
Section 9.9.43 MCU_GPIO_TIMESTAMPING_CONFIG	0x28	Enables GPIO time stamping Feature on GPIOs
Section 9.9.44 MCU_GPIO_TIMESTAMP_READ	0x2C	Configuration register for GPIO event count full and partial
Section 9.9.45 MCU_SLEEPHOLD_REQ	0x30	Enables SLEEPHOLD req and ack and selects the FSM mode
RESERVED	0x34	--
Section 9.9.46 MCU_FSM_WAKEUP_STATUS_REG	0x38	Configuration register to know the wake-up status register
Section 9.9.47 MCU_FSM_WAKEUP_STATUS_CLEAR	0x3C	Configuration register to clear the wake-up status register
Section 9.9.48 MCU_FSM_PMU_STATUS_REG	0x40	Configuration Register for PMU status
Section 9.9.49 MCU_FSM_PMUX_CTRLS_RET	0x44	Controls for RAM power muxes
Section 9.9.50 MCU_FSM_TOGGLE_COUNT	0x48	Configuration register for toggle count and toggle data

9.8.6 ULP Configuration

Base Address: 0x2404_8400

Table 9.12. ULP Configuration Register

Register Name	Offset	Description
Section 9.9.51 M4SS_TASS_CTRL_SET_REG	0x34	M4SS control of Turn ON power supply for NWP
Section 9.9.52 M4SS_TASS_CTRL_CLEAR_REG	0x38	M4SS control of Turn OFF power supply for NWP
Section 9.9.53 M4_ULP_MODE_CONFIG	0x3C	Isolation Configuration for Ultra Low-Power Mode of the processor (PS2 State)
Section 9.9.54 ULPSS_BYPASS_PWRCTRL_REG	0x40	Enables software based control of output isolation for ULPTASS SRAM, ULPSDCSS AON, ULPTASS AON, ULP MISC and ULP Peripherals

9.8.7 MCU Retention

Base Address: 0x2404_8600

Table 9.13. MCU Retention

Register Name	offset	Description
Section 9.9.56 MCURET_QSPI_WR_OP_DIS	0x00	Disables write operation to Flash
Section 9.9.57 MCURET_BOOTSTATUS	0x04	Gives Boost Status information to the MCU
RESERVED	0x08	---
RESERVED	0x0C	---
Section 9.9.58 MCUAON_CTRL_REG4	0x10	Enable and Selects for NPSS Test modes
Section 9.9.59 NPSS_GPIO_0_CTRLS	0x1C	Control signals to NPSS GPIO-0
Section 9.9.60 NPSS_GPIO_1_CTRLS	0x20	Control signals to NPSS GPIO-1
Section 9.9.61 NPSS_GPIO_2_CTRLS	0x24	Control signals to NPSS GPIO-2
Section 9.9.62 NPSS_GPIO_3_CTRLS	0x28	Control signals to NPSS GPIO-3
Section	0x2C	Control signals to NPSS GPIO-4

9.9 Register Description

9.9.1 M4SS_BYPASS_PWRCTRL_REG1

Table 9.14. M4SS_BYPASS_PWRCTRL_REG1

Bit	Access	Function	Reset Value	Description
31:23	--	--	--	--
22	R/W	BYPASS_M4SS_PWRCTL_ULP_ROM_b	0	Writing 1 to this enables software based control of isolation as well as reset for M4SS ROM. Writing 0 to this disables software based control.
21:20	--	--	--	--
19	R/W	BYPASS_M4SS_PWRCTL_ULP_AON_b	0	Writing 1 to this enables software based control of isolation as well as reset for ULP AON Writing 0 to this disables software based control.
18	R/W	BYPASS_M4SS_PWRCTL_ULP_M4_CORE_b	0	Writing 1 to this enables software based control of isolation as well as reset for M4SS CORE Writing 0 to this disables software based control.
17	R/W	BYPASS_M4SS_PWRCTL_ULP_M4_DEBUG_b	0	Writing 1 to this enables software based control of isolation as well as reset for M4SS DEBUG and FPU. Writing 0 to this disables software based control.
16	R	Reserved	0	Reserved
15	R/W	Reserved	0	Reserved
14	R/W	BYPASS_M4SS_PWRCTL_ULP_IID_b	0	Writing 1 to this enables software based control of isolation as well as reset for ULP IID Writing 0 to this disables software based control.
13	R/W	BYPASS_M4SS_PWRCTL_ULP_QSPI_ICACHE_b	0	Writing 1 to this enables software based control of isolation as well as reset for ULP quad SPI and ICACHE Writing 0 to this disables software based control.
12	R/W	Reserved	0	Reserved

Bit	Access	Function	Reset Value	Description
11	R/W	BYPASS_M4SS_PWRCTL_ULP_HIF_SDIO_SPI_b	0	Writing 1 to this enables software based control of isolation as well as reset for HIF SDIO SPI Writing 0 to this disables software based control.
10	R/W	Reserved	0	Reserved
9	R/W	BYPASS_M4SS_PWRCTL_ULP_GPDMA_b	0	Writing 1 to this enables software based control of isolation as well as reset for GPDMA Writing 0 to this disables software based control.
8	R/W	Reserved	0	Reserved
7	R/W	Reserved	0	Reserved
6	R/W	Reserved	0	Reserved
5	R/W	Reserved	0	Reserved
4	R/W	BYPASS_M4SS_PWRCTL_ULP_EFUSE_b	0	Writing 1 to this enables software based control of isolation as well as reset for ULP EFUSE and PERI domains. Writing 0 to this disables software based control.
3	R/W	BYPASS_M4SS_PWRCTL_ULP_M4_ULP_AON_b	0	Writing 1 to this enables software based control of isolation as well as reset for ULP AON M4SS Writing 0 to this disables software based control.
2	R	Reserved	-	Reserved
1:0	R	Reserved	--	Reserved

9.9.2 M4SS_BYPASS_PWRCTRL_REG2

Table 9.15. M4SS_BYPASS_PWRCTRL_REG2

Bit	Access	Function	Reset Value	Description
31:10	R	Reserved	-	Reserved
9:0	R/W		0	Writing 1 to this enables software based control of isolation as well as reset for M4SS SRAM 1 Writing 0 to this disables software based control.

9.9.3 M4SS_PWRCTRL_SET_REG

Table 9.16. M4SS_PWRCTRL_SET_REG

Bit	Access	Function	Reset Value	Description
31:23	-	Reserved	-	It is recommended to write these bits to 0.
22	RW	M4SS_EXT_PWRGATE_EN_N_ULP_ROM_b	1	Writing 1 to this enables power to the ROM. Writing 0 to this has no effect.
21:19	-	Reserved	-	It is recommended to write these bits to 0.
18	RW	M4SS_PWRGATE_EN_N_ULP_M4_CORE_b	1	Writing 1 to this enables power to the M4 CORE. Writing 0 to this has no effect.
17	RW	M4SS_PWRGATE_EN_N_ULP_M4_DEBUG_b	1	Writing 1 to this enables power to the DEBUG and FPU. Writing 0 to this has no effect.
16:15	-	Reserved	-	Reserved
14	RW	M4SS_PWRGATE_EN_N_ULP_IID_b	1	Writing 1 to this enables power to the IID. Writing 0 to this has no effect.
13	RW	M4SS_PWRGATE_EN_N_ULP_QSPI_ICACHE_b	1	Writing 1 to this enables power to the QSPI and ICACHE. Writing 0 to this has no effect.
12	-	Reserved	-	Reserved
11	RW	M4SS_PWRGATE_EN_N_ULP_HIF_SDIO_SPI_b	1	Writing 1 to this enables power to the SDIO-SPI Slave. Writing 0 to this has no effect.
10	-	Reserved	-	Reserved
9	RW	M4SS_PWRGATE_EN_N_ULP_GPDMA_b	1	Writing 1 to this enables power to the DMA. Writing 0 to this has no effect.
8:5	-	Reserved	-	Reserved
4	RW	M4SS_PWRGATE_EN_N_ULP_EFUSE_b	1	Writing 1 to this enables power to the EFUSE and PERI. Writing 0 to this has no effect.
3:0	-	Reserved	-	It is recommended to write these bits to 0.

9.9.4 M4SS_PWRCTRL_CLEAR_REG

Table 9.17. M4SS_PWRCTRL_CLEAR_REG

Bit	Access	Function	Reset Value	Description
31:23	-	Reserved	-	It is recommended to write these bits to 0.
22	RW	M4SS_EXT_PWRGATE_EN_N_ULP_ROM_b	1	Writing 1 to this disables power to the ROM. Writing 0 to this has no effect.
21:19	-	Reserved	-	It is recommended to write these bits to 0.
18	RW	M4SS_PWRGATE_EN_N_ULP_M4_CORE_b	1	Writing 1 to this disables power to the M4 CORE. Writing 0 to this has no effect.
17	RW	M4SS_PWRGATE_EN_N_ULP_M4_DEBUG_b	1	Writing 1 to this disables power to the DEBUG and FPU. Writing 0 to this has no effect.
16:15	-	Reserved	-	Reserved
14	RW	M4SS_PWRGATE_EN_N_ULP_IID_b	1	Writing 1 to this disables power to the IID Domain. Writing 0 to this has no effect.
13	RW	M4SS_PWRGATE_EN_N_ULP_QSPI_ICACHE_b	1	Writing 1 to this disables power to the QSPI and ICACHE. Writing 0 to this has no effect.
12	-	Reserved	-	Reserved
11	RW	M4SS_PWRGATE_EN_N_ULP_HIF_SDIO_SPI_b	1	Writing 1 to this disables power to the SDIO-SPI Slave. Writing 0 to this has no effect.
10	-	Reserved	-	Reserved
9	RW	M4SS_PWRGATE_EN_N_ULP_GPDMA_b	1	Writing 1 to this disables power to the DMA. Writing 0 to this has no effect.
8:5	-	Reserved	-	Reserved
4	RW	M4SS_PWRGATE_EN_N_ULP_EFUSE_b	1	Writing 1 to this disables power to the EFUSE and PERI domains. Writing 0 to this has no effect.
3::0	-	Reserved	-	It is recommended to write these bits to 0.

9.9.5 M4SS_PLL_PWRCTRL_REG

Table 9.18. PLL Power Control Register

Bit	Access	Function	Reset Value	Description
31:8	-	Reserved	-	It is recommended to write these bits to 0.
7	RW	SOCPLL_VDD13_ISO_EN	1	This is used for isolation of signals from DC-DC 1.45 to the VBATT supply in SoC-PLL to avoid leakage. Writing 1 to this disables SOC PLL Macro Writing 0 to this enables SOC PLL Macro
6	RW	SOCPLL_SPI_PG_EN	0	Writing 0 to this enables power to the SOCPLL SPI. Writing 1 to this disables power to the SOCPLL SPI.
5:0	-	Reserved	-	It is recommended to write these bits to 0.

9.9.6 M4_SRAM_PWRCTRL_SET_REG1

Table 9.19. M4_SRAM_PWRCTRL_REG1_SET

Bit	Access	Function	Reset Value	Description
31:10	-	Reserved	-	It is recommended to write these bits to 0.

9:0	R/W	M4SS_EXT_PWRGATE_EN_N_ULP_SRAM_1	1023	Functional Control signal for M4SS SRAM
				If set, functional mode is Enabled.
				Clearing this bit has no effect
				BIT(0) - 4KB (Bank1 of first 192k chunk)
				BIT(1) - 4KB (Bank2 of first 192k chunk)
				BIT(2) - 4KB (Bank3 of first 192k chunk)
				BIT(3) - 4KB (Bank4 of first 192k chunk)
				BIT(4) - 16KB (Bank 5 of first 192k chunk)
				BIT(5) - 32KB (Bank 6-7 of first 192k chunk)
				BIT(6) - 64KB (Bank 8-11 of first 192k chunk)
				BIT(7) - 64KB (Bank 12-15 of first 192k chunk)
				BIT(8) - 64KB (Bank 1-4 of second 64k chunk)
				BIT(9) - 64KB (Bank 1-4 of third 64k chunk)

9.9.7 M4_SRAM_PWRCTRL_CLEAR_REG1

Table 9.20. M4_SRAM_PWRCTRL_CLEAR_REG1

Bit	Access	Function	Reset Value	Description
31:10	-	Reserved	-	It is recommended to write these bits to 0.

9:0	R/W	M4SS_EXT_PWRGATE_EN_N_ULP_SRAM_1	1023	Functional Control signal for M4SS SRAM
				If set, functional mode is disabled.
				Clearing this bit has no effect
				BIT(0) - 4KB (Bank1 of first 192k chunk)
				BIT(1) - 4KB (Bank2 of first 192k chunk)
				BIT(2) - 4KB (Bank3 of first 192k chunk)
				BIT(3) - 4KB (Bank4 of first 192k chunk)
				BIT(4) - 16KB (Bank 5 of first 192k chunk)
				BIT(5) - 32KB (Bank 6-7 of first 192k chunk)
				BIT(6) - 64KB (Bank 8-11 of first 192k chunk)
				BIT(7) - 64KB (Bank 12-15 of first 192k chunk)
				BIT(8) - 64KB (Bank 1-4 of second 64k chunk)
				BIT(9) - 64KB (Bank 1-4 of third 64k chunk)

9.9.8 M4_SRAM_PWRCTRL_SET_REG2

Table 9.21. M4_SRAM_PWRCTRL_SET_REG2

Bit	Access	Function	Reset Value	Description
31:10	-	Reserved	-	It is recommended to write these bits to 0.

9:0	R/W	M4SS_EXT_PWRGATE_EN_N_ULP_SRAM_PERI_1	1023	Functional Control signal for M4SS SRAM
				If set, functional mode is Enabled.
				Clearing this bit has no effect
				BIT(0) - 4KB (Bank1 of first 192k chunk)
				BIT(1) - 4KB (Bank2 of first 192k chunk)
				BIT(2) - 4KB (Bank3 of first 192k chunk)
				BIT(3) - 4KB (Bank4 of first 192k chunk)
				BIT(4) - 16KB (Bank 5 of first 192k chunk)
				BIT(5) - 32KB (Bank 6-7 of first 192k chunk)
				BIT(6) - 64KB (Bank 8-11 of first 192k chunk)
				BIT(7) - 64KB (Bank 12-15 of first 192k chunk)
				BIT(8) - 64KB (Bank 1-4 of second 64k chunk)
				BIT(9) - 64KB (Bank 1-4 of third 64k chunk)

9.9.9 M4_SRAM_PWRCTRL_CLEAR_REG2

Table 9.22. M4_SRAM_PWRCTRL_CLEAR_REG2

Bit	Access	Function	Reset Value	Description
31:10	-	Reserved	-	It is recommended to write these bits to 0.

9:0	R/W	M4SS_EXT_PWRGATE_EN_N_ULP_SRAM_PERI_1	1023	Functional Control signal for M4SS SRAM
				If set, functional mode is disabled.
				Clearing this bit has no effect
				BIT(0) - 4KB (Bank1 of first 192k chunk)
				BIT(1) - 4KB (Bank2 of first 192k chunk)
				BIT(2) - 4KB (Bank3 of first 192k chunk)
				BIT(3) - 4KB (Bank4 of first 192k chunk)
				BIT(4) - 16KB (Bank 5 of first 192k chunk)
				BIT(5) - 32KB (Bank 6-7 of first 192k chunk)
				BIT(6) - 64KB (Bank 8-11 of first 192k chunk)
				BIT(7) - 64KB (Bank 12-15 of first 192k chunk)
				BIT(8) - 64KB (Bank 1-4 of second 64k chunk)
				BIT(9) - 64KB (Bank 1-4 of third 64k chunk)

9.9.10 M4_SRAM_PWRCTRL_SET_REG3

Table 9.23. M4_SRAM_PWRCTRL_SET_REG3

Bit	Access	Function	Reset Value	Description
31:10	-	Reserved	-	It is recommended to write these bits to 0.

9:0	RW	M4SS_SRAM_INPUT_DISABLE_ISOLATION_ULP_1	1023	Functional Control signal for M4SS SRAM
				Writing 1 to particular bit, disables isolation for the respective mentioned memory bank.
				Clearing any of the bits has no effect
				BIT(0) - 4KB (Bank1 of first 192k chunk)
				BIT(1) - 4KB (Bank2 of first 192k chunk)
				BIT(2) - 4KB (Bank3 of first 192k chunk)
				BIT(3) - 4KB (Bank4 of first 192k chunk)
				BIT(4) - 16KB (Bank 5 of first 192k chunk)
				BIT(5) - 32KB (Bank 6-7 of first 192k chunk)
				BIT(6) - 64KB (Bank 8-11 of first 192k chunk)
				BIT(7) - 64KB (Bank 12-15 of first 192k chunk)
				BIT(8) - 64KB (Bank 1-4 of second 64k chunk)
				BIT(9) - 64KB (Bank 1-4 of third 64k chunk)

9.9.11 M4_SRAM_PWRCTRL_CLEAR_REG3

Table 9.24. M4_SRAM_PWRCTRL_CLEAR_REG3

Bit	Access	Function	Reset Value	Description
31:10	-	Reserved	-	It is recommended to write these bits to 0.

9:0	RW	M4SS_SRAM_INPUT_DISABLE_ISOLATION_ULP_1	1023	Functional Control signal for M4SS SRAM
				Writing 1 to particular bit, enables isolation for the respective mentioned memory bank.
				Clearing any of the bits has no effect
				BIT(0) - 4KB (Bank1 of first 192k chunk)
				BIT(1) - 4KB (Bank2 of first 192k chunk)
				BIT(2) - 4KB (Bank3 of first 192k chunk)
				BIT(3) - 4KB (Bank4 of first 192k chunk)
				BIT(4) - 16KB (Bank 5 of first 192k chunk)
				BIT(5) - 32KB (Bank 6-7 of first 192k chunk)
				BIT(6) - 64KB (Bank 8-11 of first 192k chunk)
				BIT(7) - 64KB (Bank 12-15 of first 192k chunk)
				BIT(8) - 64KB (Bank 1-4 of second 64k chunk)
				BIT(9) - 64KB (Bank 1-4 of third 64k chunk)

9.9.12 M4_SRAM_PWRCTRL_SET_REG4

Table 9.25. M4_SRAM_PWRCTRL_SET_REG4

Bit	Access	Function	Reset Value	Description
31:23	-	Reserved	-	Reserved
22:0	RW	M4SS_SRAM_DS_1	0	Deep-Sleep control for M4SS SRAM If set, Deep-sleep mode in RAM is enabled. Clearing this bit has no effect

9.9.13 M4_SRAM_PWRCTRL_CLEAR_REG4

Table 9.26. M4_SRAM_PWRCTRL_CLEAR_REG4

Bit	Access	Function	Reset Value	Description
31:23	-	Reserved	-	Reserved
22:0	RW	M4SS_SRAM_DS_1	0	Deep-Sleep control for M4SS SRAM If set, Deep-sleep mode in RAM is disabled. Clearing this bit has no effect

9.9.14 MCU_FSM_CTRL_BYPASS

Table 9.27. MCU_FSM_CTRL_BYPASS

Bit	Access	Function	Reset Value	Description
31:6	-	Reserved	-	It is recommended to write these bits to 0.
5	RW	MCU_BUCK_BOOST_ENABLE_BYPASS	0	Writing 1 to this enables the Buck-boost when Sleep-FSM is bypassed Writing 0 to this disables the Buck-boost when Sleep-FSM is bypassed
4	RW	MCU_BUCK_BOOST_ENABLE_BYPASS_CTRL	1	Writing 1 to this uses the Sleep-FSM for controlling the Buck-boost Writing 0 to this bypasses the Sleep-FSM for controlling the Buck-boost
3	RW	MCU_PMU_SHUT_DOWN_BYPASS	1	Writing 1 to this puts the PMU to be in Shutdown mode when Sleep-FSM is bypassed Writing 0 to this puts the PMU to be in Active mode when Sleep-FSM is bypassed
2	RW	MCU_PMU_SHUT_DOWN_BYPASS_CTRL	1	Writing 1 to this uses the Sleep-FSM for controlling the PMU mode Writing 0 to this bypasses the Sleep-FSM for controlling the PMU mode
1	RW	MCU_XTAL_EN_40MHZ_BYPASS	0	Writing 1 to this enables the 40MHz Crystal clock when Sleep-FSM is bypassed Writing 0 to this disables the 40MHz Crystal clock when Sleep-FSM is bypassed
0	RW	MCU_XTAL_EN_40MHZ_BYPASS_CTRL	1	Writing 1 to this uses the Sleep-FSM for controlling the 40MHz Crystal clock Writing 0 to this bypasses the Sleep-FSM for controlling the 40MHz Crystal clock

9.9.15 MCU_PMU_LDO_CTRL_SET

Table 9.28. MCU PMU LDO Control SET Register

Bit	Access	Function	Reset Value	Description
31:19	-	Reserved	-	It is recommended to write these bits to 0.
18	RW	MCU_DCDC_LVL	1	Writing 1 to this configures DC-DC 1.45 Supply voltage to 1.45V. Writing 0 to this has no effect.
17	RW	MCU_SOC_LDO_LVL	1	Writing 1 to this configures LDO-SoC 1.15 Supply voltage to 1.15V. Writing 0 to this has no effect.
16:4	-	Reserved	-	It is recommended to write these bits to 0.
3	RW	m4_ignore_pmudcdc_en_in_pfmmode	0	If this bit is set, it will Ignore PMU dcdc enable in PFM mode for M4
2	RW	MCU_DCDC_EN	1	Writing 1 to this enables PMU DCDC for providing supply to M4SS. Writing 0 to this has no effect
1	RW	MCU_SCO_LDO_EN	1	Writing 1 to this enables SoC LDO for providing supply to M4SS. Writing 0 to this has no effect
0	RW	MCU_FLASH_LDO_EN	1	Writing 1 to this enables LDO-FL 1.8 Output voltage. Writing 0 to this has no effect.

9.9.16 MCU_PMU_LDO_CTRL_CLEAR

Table 9.29. PMU LDO Control CLEAR Register

Bit	Access	Function	Reset Value	Description
31:19	-	Reserved	-	It is recommended to write these bits to 0.
18	RW	MCU_DCDC_LVL	1	Writing 1 to this configures DC-DC 1.45 Supply voltage to 1.25V. Writing 0 to this has no effect.
17	RW	MCU_SOC_LDO_LVL	1	Writing 1 to this configures LDO-SoC 1.15 Supply voltage to 1.05V. Writing 0 to this has no effect.
16:4	-	Reserved	-	It is recommended to write these bits to 0.
3	R/W	m4_ignore_pmudcdc_en_in_pfmmode	0	If this bit is set, it will consider PMU dcdc enable in PFM mode for M4
2	RW	MCU_DCDC_EN	1	Writing 1 to this disables PMU DCDC. Writing 0 to this has no effect
1	RW	MCU_SCO_LDO_EN	1	Writing 1 to this disables SoC LDO. Writing 0 to this has no effect
0	RW	MCU_FLASH_LDO_EN	1	Writing 1 to this disables LDO-FL 1.8 Output voltage. Writing 0 to this has no effect.

9.9.17 ULPSS_PWRCTRL_SET_REG

Table 9.30. ULPSS Power Control SET Register

Bit	Access	Function	Reset Value	Description
31:28	-	Reserved	-	It is recommended to write these bits to 0.
27	RW	PWRCTRL_DMA	1	Writing 1 to this enables power to the DMA. Writing 0 to this has no effect.
26	R	Reserved	-	Reserved
25	RW	PWRCTRL_ADC_DAC	1	Writing 1 to this enables power to the Aux ADC/ DAC. Writing 0 to this has no effect.
24	RW	PWRCTRL_I2C	1	Writing 1 to this enables power to the I2C. Writing 0 to this has no effect.
23	RW	PWRCTRL_I2S	1	Writing 1 to this enables power to the I2S. Writing 0 to this has no effect.
22	RW	PWRCTRL_SSI	1	Writing 1 to this enables power to the SPI/SSI. Writing 0 to this has no effect.
21	RW	PWRCTRL_UART	1	Writing 1 to this enables power to the UART. Writing 0 to this has no effect.
20	--	--	--	Reserved
19	RW	PWRGATE_EN_N_ULP_CAP	1	Writing 1 to this enables power to the ULPSS CAP. Writing 0 to this has no effect.
18	RW	PWRGATE_EN_N_ULP_MISC	1	Writing 1 to this enables power to the MISC. Writing 0 to this has no effect.
17:0	-	Reserved	-	It is recommended to write these bits to 0.

9.9.18 ULPSS_PWRCTRL_CLEAR_REG

Table 9.31. ULPSS Power Control CLEAR Register

Bit	Access	Function	Reset Value	Description
31:28	-	Reserved	-	It is recommended to write these bits to 0.
27	RW	PWRCTRL_DMA	1	Writing 1 to this disables power to the DMA. Writing 0 to this has no effect.
26	R	Reserved	-	-
25	RW	PWRCTRL_ADC_DAC	1	Writing 1 to this disables power to the Aux ADC/DAC. Writing 0 to this has no effect.
24	RW	PWRCTRL_I2C	1	Writing 1 to this disables power to the I2C. Writing 0 to this has no effect.
23	RW	PWRCTRL_I2S	1	Writing 1 to this disables power to the I2S. Writing 0 to this has no effect.
22	RW	PWRCTRL_SSI	1	Writing 1 to this disables power to the SPI/SSI. Writing 0 to this has no effect.
21	RW	PWRCTRL_UART	1	Writing 1 to this disables power to the UART. Writing 0 to this has no effect.
20	-	--	-	Reserved
19	RW	PWRGATE_EN_N_ULP_CAP	1	Writing 1 to this disables power to the ULPSS CAP. Writing 0 to this has no effect.
18	RW	PWRGATE_EN_N_ULP_MISC	1	Writing 1 to this disables power to the MISC. Writing 0 to this has no effect.
17:0	-	Reserved	-	It is recommended to write these bits to 0.

9.9.19 ULPSS_RAM_PWRCTRL_REG1_SET

Table 9.32. ULP SRAM Power Control Register1 SET

Bit	Access	Function	Reset Value	Description
31:4	-	Reserved	-	It is recommended to write these bits to 0.
3:0	RW	PWRCTRL1_ULP_SRAM	15	Writing 1 to this enables power to ULP-SRAM. Writing 0 to this has no effect.

9.9.20 ULPSS_RAM_PWRCTRL_REG1_CLEAR

Table 9.33. ULP SRAM Power Control Register1 CLEAR

Bit	Access	Function	Reset Value	Description
31:4	-	Reserved	-	It is recommended to write these bits to 0.
3:0	RW	PWRCTRL1_ULP_SRAM	15	Writing 1 to this disables power to ULP-SRAM. Writing 0 to this has no effect.

9.9.21 ULPSS_RAM_PWRCTRL_REG2_SET

Table 9.34. ULP SRAM Power Control Register2 SET

Bit	Access	Function	Reset Value	Description
31:20	-	Reserved	-	It is recommended to write these bits to 0.
19:16	RW	DS_ULPSRAM__PROC_1	0	Writing 1 to this enables deep sleep mode ULPTASS SRAM. Writing 0 to this has no effect.
15:4	-	Reserved	-	It is recommended to write these bits to 0.
3:0	RW	INP_ISO_SRAM	15	Writing 1 to this disables isolation to ULPTASS SRAM. Writing 0 to this has no effect.

9.9.22 ULPSS_RAM_PWRCTRL_REG2_CLEAR

Table 9.35. ULP SRAM Power Control Register2 CLEAR

Bit	Access	Function	Reset Value	Description
31:20	-	Reserved	-	It is recommended to write these bits to 0.
19:16	RW	DS_ULPSRAM__PROC_1	0	Writing 1 to this disables deep sleep mode ULPTASS SRAM. Writing 0 to this has no effect.
15:4	-	Reserved	-	It is recommended to write these bits to 0.
3:0	RW	INP_ISO_SRAM	15	Writing 1 to this enables isolation to ULPTASS SRAM. Writing 0 to this has no effect.

9.9.23 ULPSS_RAM_PWRCTRL_REG3_SET

Table 9.36. ULP SRAM Power Control Register3 SET

Bit	Access	Function	Reset Value	Description
31:4	-	Reserved	-	It is recommended to write these bits to 0.
3:0	RW	PWRCTRL_ULPTASS_SRAM_PERI_1	15	Writing 1 to this enables power to ULPTASS SRAM dual rail pins. Writing 0 to this has no effect.

9.9.24 ULPSS_RAM_PWRCTRL_REG3_CLEAR

Table 9.37. ULP SRAM Power Control Register3 CLEAR

Bit	Access	Function	Reset Value	Description
31:4	-	Reserved	-	It is recommended to write these bits to 0.
3:0	RW	PWRCTRL_ULPTASS_SRAM_PERI_1	15	Writing 1 to this disables power to ULPTASS SRAM dual rail pins. Writing 0 to this has no effect.

9.9.25 UULP_PWRCTRL_SET

Table 9.38. UULP Peripheral Power Control SET Register

Bit	Access	Function	Reset Value	Description
31:17	-	Reserved	-	It is recommended to write these bits to 0.
16		SLPSS_PWRGATE_EN_N_ULP_NWPAPB_MCU_CTRL	0	
15:11	-	Reserved	-	
10	RW	SLPSS_PWRGATE_EN_N_ULP_TIMEPERIOD	1	
9	RW	SLPSS_PWRGATE_EN_N_ULP_MCUSTORE3	1	Writing 1 to this enables power to the STORAGE-DOMAIN3. Writing 0 to this has no effect.
8	RW	SLPSS_PWRGATE_EN_N_ULP_MCUSTORE2	1	Writing 1 to this enables power to the STORAGE-DOMAIN2. Writing 0 to this has no effect.
7	RW	SLPSS_PWRGATE_EN_N_ULP_MCUSTORE1	1	Writing 1 to this enables power to the STORAGE-DOMAIN1. Writing 0 to this has no effect.
6	RW	SLPSS_PWRGATE_EN_N_ULP_MCUTS	1	Writing 1 to this enables power to the TS. Writing 0 to this has no effect.
5	RW	SLPSS_PWRGATE_EN_N_ULP_MCUPS	1	Writing 1 to this enables power to the PS. Writing 0 to this has no effect.
4	RW	SLPSS_PWRGATE_EN_N_ULP_MCUWDT	1	Writing 1 to this enables power to the WDT. Writing 0 to this has no effect.
3	RW	SLPSS_PWRGATE_EN_N_ULP_MCURTC	1	Writing 1 to this enables power to the RTC. Writing 0 to this has no effect.

Bit	Access	Function	Reset Value	Description
2	RW	SLPSS_PWRGATE_EN_N_ULP_MCUFSM	1	Writing 1 to this enables power to the FSM. Writing 0 to this has no effect.
1	RW	SLPSS_PWRGATE_EN_N_ULP_MCUBFFS	1	Writing 1 to this enables power to the BBFFS. Writing 0 to this has no effect.
0	-	Reserved	-	It is recommended to write these bits to 0.

9.9.26 UULP_PWRCTRL_CLEAR

Table 9.39. UULP Peripheral Power Control CLEAR Register

Bit	Access	Function	Reset Value	Description
31:17	-	Reserved	-	It is recommended to write these bits to 0.
16	RW	SLPSS_PWRGATE_EN_N_ULP_NWPAPB_MCU_CTRL	0	
15:11	-	Reserved	-	
10	RW	SLPSS_PWRGATE_EN_N_ULP_TIMEPERIOD	1	
9	RW	SLPSS_PWRGATE_EN_N_ULP_MCUSTORE3	1	Writing 1 to this disables power to the STORAGE-DOMAIN3. Writing 0 to this has no effect.
8	RW	SLPSS_PWRGATE_EN_N_ULP_MCUSTORE2	1	Writing 1 to this disables power to the STORAGE-DOMAIN2. Writing 0 to this has no effect.
7	RW	SLPSS_PWRGATE_EN_N_ULP_MCUSTORE1	1	Writing 1 to this disables power to the STORAGE-DOMAIN1. Writing 0 to this has no effect.
6	RW	SLPSS_PWRGATE_EN_N_ULP_MCUTS	1	Writing 1 to this disables power to the TS. Writing 0 to this has no effect.
5	RW	SLPSS_PWRGATE_EN_N_ULP_MCUPS	1	Writing 1 to this disables power to the PS. Writing 0 to this has no effect.
4	RW	SLPSS_PWRGATE_EN_N_ULP_MCUWDT	1	Writing 1 to this disables power to the WDT. Writing 0 to this has no effect.
3	RW	SLPSS_PWRGATE_EN_N_ULP_MCURTC	1	Writing 1 to this disables power to the RTC. Writing 0 to this has no effect.

Bit	Access	Function	Reset Value	Description
2	RW	SLPSS_PWRGATE_EN_N_ULP_MCUFSM	1	Writing 1 to this disables power to the FSM. Writing 0 to this has no effect.
1	RW	SLPSS_PWRGATE_EN_N_ULP_MCUBFFS	1	Writing 1 to this disables power to the BBFFS. Writing 0 to this has no effect.
0	-	Reserved	-	It is recommended to write these bits to 0.

9.9.27 MCUAON_IPMU_RESET_CTRL

Table 9.40. MCUAON_IPMU_RESET_CTRL

Bit	Access	Function	Default Value	Description
31:2	–	–	0	–
1	R/W	IPMU_SPI_RESET_N	1	Writing 0 to this Resets IPMU SPI Writing 1 has no effect
0	R/W	ULP_ANALOG_SPI_RESET_N	1	Writing 0 to this Resets ULP Analog SPI Writing 1 has no effect

9.9.28 MCUAON_SHELF_MODE

Table 9.41. MCUAON_SHELF_MODE Register

Bit	Access	Function	Default Value	Description
31:22	–	–	–	–
21:19	R/W	SHELF_MODE_WAKEUP_DELAY	5	Writing a value to this Programs the delay for resetting Chip during exit phase of shelf mode. 0 - 1 clock cycle delay 1 - 2 clock cycles delay 2 - 4 clock cycles delay 3 - 8 clock cycles delay 4 - 16 clock cycles delay 5 - 32 clock cycles delay 6 - 64 clock cycles delay 7 - 128 clock cycles delay
18	R/W	SHELF_MODE_GPIOBASED	0	GPIO based Shelf mode Entering. when written '1' by processor, On Falling edge of GPIO (Based on the option used in "shutdown_wakeup_mode" register) chip will enter Shelf mode. When written 0 it has no effect.
17:16	R/W	SHUTDOWN_WAKEUP_MODE	0	GPIO based wakeup mode configuration. When written 00 - NPSS GPIO 2 Based wakeup. NPSS GPIO2 Should go high to wakeup Chip from Shelf mode. When written 01 - NPSS GPIO 3 Based wakeup. NPSS GPIO3 Should go high to wakeup Chip from Shelf mode. When written 10 - NPSS GPIO 3 & NPSS GPIO 2 Based wakeup. Both NPSS GPIO3 & NPSS GPIO2 Should go high to wakeup Chip from Shelf mode When written 11 - NPSS GPIO 3 Or NPSS GPIO 2 Based wakeup. If , NPSS GPIO3 or NPSS GPIO2 go high will wakeup Chip from Shelf mode.
15:0	W	ENTER_SHELF_MODE	0	when 0xAAAA is written to this the chip enters Shelf mode. when any other value is written, it has no effect.

9.9.29 MCUAON_GEN_CTRLS

Table 9.42. MCUAON_GEN_CTRLS Register

Bit	Access	Function	Default Value	Description
31:20	–	–	–	–
19	R/W	mask_glitch_32mhz_rc_clk_en	0	Before going to power save, there is glitch on 32mhz_rc clock enable from NPSS due to difference in isolation value and reset value 1 - mask the 32mhz_rc clock enable glitch in MCU mode 0 - unmask the 32mhz_rc clock enable glitch in MCU mode
18	R/W	allow_m4sys_rst_in_ps2	0	Bit is used to un-mask generation of chip reset when host reset/ M4-system reset is asserted 0 - Chip reset is generated when host/M4 system reset is asserted and M4 is in PS4 state, both M4 & NWP are active. 1 - Chip reset is generated when host/M4 system reset is asserted irrespective of PS2/PS4 states
17	R/W	NPSS_SUPPLY_0P9	1	When 0 is written to this, Npss supply will switch from 0.6V to 0.9V based on high frequency enables. If high frequency enables are preset NPSS supply will be 1.05V from SCDCDC Main Supply. If high frequency enables are not-preset, NPSS supply will be 0.75V from SCDCDC Retention Supply. When 1 is written to this, Npss supply always at 0.9V
16	R/W	ENABLE_PDO	0	Enable Turning Off POD power domain when SOC_LDO EN is low. When 1 is written to this, Up on SoC LDO Enable going low, IO supply (3.3v) to SOC Pads will be tuned-off. Writing 0 has no effect
15:2	--	--	--	--
1	R/W	ULP_ANALOG_WAKEUP_ACCESS	0	ULP Analog Wakeup Source Access Writing 1 to this - ULP Analog Wakeup Access is mapped to NWP Writing 0 to this - ULP Analog Wakeup Access is mapped to M4SS Wakeup source from IPMU such as comperator output for BOD, Button will be mapped to NWP or MCU

Bit	Access	Function	Default Value	Description
0	R/W	XTAL_CLK_FROM_GPIO	0	Writing 1 to this selects XTAL clock from GPIO Pins. Please refer to NPSS GPIO Pin Muxing for configuration. Writing 0 to this selects XTAL Clock from IPMU clock sources.

9.9.30 MCUAON_PDO_CTRLS

Table 9.43. MCUAON_PDO_CTRLS Register

Bit	Access	Function	Default Value	Description
31:5	–	–	–	–
4	R/W	SDIO_IO_DOMAIN_EN_B	0	Writing 1 to this Turns-Off IO supply of SDIO domain. Writing 0 to this Turns-On IO supply of SDIO domain.
3	R/W	QSPI_IO_DOMAIN_EN_B	0	Writing 1 to this Turns-Off IO supply of QSPI domain. Writing 0 to this Turns-On IO supply of QSPI domain.
2	R/W	SOC_T_IO_DOMAIN_EN_B	0	Writing 1 to this Turns-Off IO supply of SOC domain on Top Side. Writing 0 to this Turns-On IO supply of SOC domain on Top Side.
1	R/W	SOC_L_IO_DOMAIN_EN_B	0	Writing 1 to this Turns-Off IO supply of SOC domain on Left Side. Writing 0 to this Turns-On IO supply of SOC domain on Left Side.
0	R/W	SOC_B_IO_DOMAIN_EN_B	0	Writing 1 to this Turns-Off IO supply of SOC domain on Bottom Side. Writing 0 to this Turns-On IO supply of SOC domain on Bottom Side.

9.9.31 MCUAON_WDT_CHIP_RST

Table 9.44. MCUAON_WDT_CHIP_RST Register

Bit	Access	Function	Default Value	Description
31:1	–	–	–	–
0	R/W	MCU_WDT_BASED_CHIP_RESET	1	When 0 is written to this, Power-On Reset (POR) will be generated. When 1 is written to this, NON Power-On Reset (NON - POR) will be generated.

9.9.32 MCUAON_KHZ_CLK_SEL_POR_RESET_STATUS

Table 9.45. MCUAON_KHZ_CLK_SEL_POR_RESET_STATUS Register

Bit	Access	Function	Default Value	Description
31:24	–	–	–	–
23:18	R/W	MCUULP_VBAT_SYS_RTC_CLK_DIV_FAC	6'b000000	Clock division factor for 32Khz clk (Used in SYSRTC and MCU WWD) Note: Need to set 6'b010000 to generate 1 KHz clock If 0th bit is set → it divides by 2 Similarly, if 4th bit is set → it divides by 32
17	R/W	MCU_FIRST_POWERUP_RESET_N	0	Write 1 to this upon power_up. It will become 0 when reset pin is pulled low
16	R/W	MCU_FIRST_POWERUP_POR	0	Write 1 to this upon power_up. It will become 0 when Vbatt power is removed.
15	R	Reserved	-	Reserved
14	R/W	MCUULP_VBAT_SYS_RTC_CLK_EN	0	Static Clock gating Enable for SYSRTC 1'b1 => Clock is enabled 1'b0 => Invalid
13	R	MCUULP_VBAT_SYS_RTC_CLK_SWITCHED	1	If Khz clock mux select for sysrtc is modified. Please poll this bit and wait till it becomes one (wait for almost half cycle).

Bit	Access	Function	Default Value	Description
[12:9]	R/W	MCUULP_VBAT_SYS_RTC_CLK_SEL	0	NPSS SySRTC KHz clock selection. 0001 - 1Khz clk sel 0010 - 32Khz RO clk sel 0100 - 32Khz RC clk sel 1000 - 32Khz XTAL clk sel Note: Use RO & RC clock only if accuracy is not crucial. Otherwise use crystal clock
8:4	–	–	–	–
3	R	AON_KHZ_CLK_SEL_CLOCK_SWITCHED	1	If Khz clock mux select is modified, please poll this bit and wait till it becomes 1. Read 1 indicates clock is switched Read 0 indicates clock is not switched
2:0	R/W	AON_KHZ_CLK_SEL	0	NPSS AON KHz clock selection. 001 - 32Khz RO clk is selected 010 - 32Khz RC clk is selected 100 - 32Khz Xtal clk is selected Note: Use RO & RC clock only if accuracy is not crucial. Otherwise use crystal clock

9.9.33 MCU_FSM_SLEEP_CTRLS_AND_WAKEUP_MODE

Table 9.46. FSM SLEEP CTRLS and WAKEUP MODE Register

Bit	Access	Function	Reset Value	Description
31	R/W	SDCSS_BASED_SLEEP		Writing 1 to this enables Sensor Data Collector Interrupt as a Sleep source Writing 0 to this disables Sensor Data Collector Interrupt as a Sleep source
30	RW	ULPSS_BASED_SLEEP	0	Writing 1 to this enables ULP Peripheral Interrupt as a Sleep source Writing 0 to this disables ULP Peripheral Interrupt as a Sleep source
29	RW	WDT_INTR_BASED_WAKEUP_b	0	Writing 1 to this enables WDT Interrupt as a Wakeup source Writing 0 to this disables WDT Interrupt as a Wakeup source Note: Set "EN_WDT_SLEEP" in 9.9.42 MCU_FSM_CRTL_PDM_AND_ENABLES register.
28	RW	MSEC_BASED_WAKEUP_b	0	Writing 1 to this enables Milli-Second Interrupt as a Wakeup source Writing 0 to this disables Milli-Second Interrupt as a Wakeup source
27	RW	SEC_BASED_WAKEUP_b	0	Writing 1 to this enables Second Interrupt as a Wakeup source Writing 0 to this disables Second Interrupt as a Wakeup source
26	RW	ALARM_BASED_WAKEUP_b	0	Writing 1 to this enables ALARM Interrupt as a Wakeup source Writing 0 to this disables ALARM Interrupt as a Wakeup source
25	RW	SDCSS_BASED_WAKEUP_b	0	Writing 1 to this enables Sensor Data Collector Interrupt as a Wakeup source Writing 0 to this disables Sensor Data Collector Interrupt as a Wakeup source
24	RW	ULPSS_BASED_WAKEUP_b	0	Writing 1 to this enables ULP Peripheral Interrupt as a Wakeup source Writing 0 to this disables ULP Peripheral Interrupt as a Wakeup source
23	R/W	WIC_BASED_WAKEUP_b	1'b0	WIC based wakeup mask
22	R/W	SYSRTC_BASED_WAKEUP	1'b0	System RTC Based Wakeup
21	RW	COMPR_BASED_WAKEUP_b	0	Writing 1 to this enables 4x-Comparator/BOD/BUTTON Interrupt as a Wakeup source Writing 0 to this disables 4x-Comparator/BOD/BUTTON Interrupt as a Wakeup source

Bit	Access	Function	Reset Value	Description
20	RW	GPIO_BASED_WAKEUP_b	0	Writing 1 to this enables UULP Vbat GPIO Interrupt as a Wakeup source Writing 0 to this disables UULP Vbat GPIO Interrupt as a Wakeup source The Selection of UULP Vbat GPIO's for wake-up is described "GPIO_WAKEUP_CONFIG" Register (Refer GPIO Configuration Section).
19	RW	M4_PROC_BASED_WAKEUP_b	0	Wakeup based on M4 processor enable
18	RW	WIRELESS_BASED_WAKEUP_b	0	Writing 1 to this enables NWP Interrupt as a Wakeup source Writing 0 to this disables NWP Interrupt as a Wakeup source
17	RW	HOST_BASED_WAKEUP_b	0	
16	RW	TIMER_BASED_WAKEUP_b	0	Writing 1 to this enables Deep-Sleep Timer Interrupt as a Wakeup source Writing 0 to this disables Deep-Sleep Timer Interrupt as a Wakeup source
15	RW	SLEEP_WAKEUP	0	Wakeup indication from Processor
14	RW	MCUFSM_WAKEUP_NWPFSM	0	When Set, mcufsm wakeup enable will wakeup both NWP FSM and MCU FSM. Clear this BIT if this feature is not required.
13:12	RW	MCU_DELAY_BW_AON_CORE_ON	0	Programmable delay for Aon and Processor Power-up during wakeup: 00: 1 cycle delay 01: 2 cycles delay 10: 3 cycles delay 11: 4 cycles delay
11	RW	SKIP_XTAL_WAIT_TIME	0	Writing 1 to this skips the settling time for High Frequency XTAL during wakeup. Writing 0 to this includes the settling time for High Frequency XTAL during wakeup.
10	RW	PMU_DCDC_ON_b	0	Writing 1 to this maintains DC-DC 1.45 in ON state during Sleep. Writing 0 to this maintains DC-DC 1.45 in OFF state during Sleep.
9	RW	LDO_FLASH_ON_b	0	Writing 1 to this maintains LDO FL 1.8 in ON state during Sleep. Writing 0 to this maintains LDO FL 1.8 in OFF state during Sleep.
8	RW	LDO_SOC_ON_b	0	Writing 1 to this maintains LDO SoC 1.15 in ON state during Sleep. Writing 0 to this maintains LDO SoC 1.15 in OFF state during Sleep.

Bit	Access	Function	Reset Value	Description
7	RW	M4ULP_RAM16K_RETENTION_MODE_EN	0	SRAM retention Control for 16KB of LP-SRAM (LP-SRAM-1, LP-SRAM-2, LP-SRAM-3, LP-SRAM-4) Writing 1 to this enables Retention during sleep Writing 0 to this disables Retention during sleep
6	RW	ULPSS_RAM_RETENTION_MODE_EN	0	SRAM retention Control for 16KB of ULP-SRAM Writing 1 to this enables Retention during sleep Writing 0 to this disables Retention during sleep
5	RW	TA_RAM_RETENTION_MODE_EN	0	SRAM retention Control for NWP-SRAM Writing 1 to this enables Retention during sleep Writing 0 to this disables Retention during sleep
4	RW	M4ULP_RAM_RETENTION_MODE_EN_b	0	SRAM retention Control for 320KB of LP-SRAM Writing 1 to this enables Retention during sleep Writing 0 to this disables Retention during sleep
3	R	Reserved	0	Reserved
2	RW	LP_SLEEP_MODE_b		Writing 1 to this enables sleep mode Writing 0 to this disables sleep mode
1	-	Reserved	-	It is recommended to write these bits to 0.
0	RW	MCUFSM_SHUTDOWN_ENABLE	0	shutdown enable pulse

9.9.34 MCU_FSM_PERI_CONFIG_REG

Bit	Access	Function	Reset Value	Description
31:17	-	Reserved	-	It is recommended to write these bits to 0.
16	RW	BGPMU_SAMPLING_EN_R	0	Controls the mode of Band-Gap for DC-DC 1.45 during PS2 state. Writing 1 to this enables sampling mode of Band-Gap. This is described in Power Management Section. Writing 0 to this disables sampling mode of Band-Gap. This is described in Power Management Section.

Bit	Access	Function	Reset Value	Description
15:4	-	Reserved	-	It is recommended to write these bits to 0.
3	RW	WICENREQ	0	Enable Request to WIC module 1 - If WIC wake up required. 0 - If WIC wake up is not required.
2:1	RW	M4SS_CONTEXT_SWITCH_TOP_ULP_MODE	0	00, 10 - HP-MCU/LP-MCU Mode 01 - ULP-MCU Mode 11 - UULP-MCU Mode (ULP-MCU mode with UM bypassed)
0	RW	ULP_MCU_MODE_EN	0	0 - Voltages are as per HP-MCU/LP-MCU Mode 1 - Voltages are as per ULP-MCU Mode

9.9.35 MCU_FSM_POWER_CTRL_AND_DELAY

Table 9.47. FSM_POWER_CTRL_DELAY Register

Bit	Access	Function	Reset Value	Description
31:28	-	Reserved	-	It is recommended to write these bits to 0.
27:26	RW	POWER_MUX_SEL_ULPSS_RAM	3	Configures the Voltage source to be used for LOW-VOLTAGE-ULPRAM Domain in PS2 state 3 – LDO SoC 1.15 1 – SC-DC 1.05 0 – Reserved
25:24	RW	POWER_MUX_SEL_M4_ULP_RAM	3	Configures the Voltage source to be used for LOW-VOLTAGE-LPRAM Domain in PS2 state 3 – LDO SoC 1.15 1 – SC-DC 1.05 0 – Reserved

Bit	Access	Function	Reset Value	Description
23:22	RW	POWER_MUX_SEL_M4_ULP_RAM_16KB	3	Configures the Voltage source to be used for LOW-VOLTAGE-LPRAM-16KB Domain in PS2 state 3 – LDO SoC 1.15 1 – SC-DC 1.05 0 – Reserved
21:20	RW	POWER_MUX_SEL_M4_ULP	3	Configures the Voltage source to be used for PROC-DOMAIN Domain in PS2 state 3 – LDO SoC 1.15 1 – SC-DC 1.05 0 – LDO 0.75V
19	RW	POWER_MUX_SEL_ULPSS	1	Configures the Voltage source to be used for LOW-VOLTAGE-ULPPERIPH Domain in PS2 state 1 – LDO SoC 1.15 0 – SC-DC 1.05
18	-	Reserved	0	Reserved
17	RW	DCDC_EN	0	Writing 1 to this configures DC-DC 1.45 to ON state during PS2 Writing 0 to this configures DC-DC 1.45 in OFF state during PS2
16	RW	LDoSoC_EN	0	Writing 1 to this configures LDO SoC 1.15 to ON state during PS2 Writing 0 to this configures LDO SoC 1.15 in OFF state during PS2
15:12	RW	PG4_BUCK_ON_DELAY	0	Configures the time for switching ON the DC-DC 1.45 during transition from PS2 to PS4 state. 0 - 50us 1 - 100us 2 - 200us 3 - 300us .. 15 - 1500us

Bit	Access	Function	Reset Value	Description
11:8	RW	PS4_SOCLDO_ON_DELAY	0	Configures the time for switching ON the LDO SoC 1.15 during transition from PS2 to PS4 state. 0 - 50us 1 - 100us 2 - 200us 3 - 300us .. 15 - 1500us
7:6		Reserved		
5:0	RW	PS2_PMU_LDO_OFF_DELAY	0	

9.9.36 GPIO_WAKEUP_REGISTER

Table 9.48. GPIO_WAKEUP_REGISTER Register

Bit	Access	Function	Default Value	Description
31:19	--	--	--	--
18	R/W	DS_TIMER_SOFT_RESET	0	Soft reset Deep sleep time block
17	R/W	CONTINIOUS_TIMER_ENABLE	0	Enable Deep sleep timer mode continuous
16	R/W	CONTINIOUS_START	0	Trigger Deep sleep timer to start counting.
15:5	--	--	--	--
4	R/W	GPIO_4_WAKEUP	0	Enable gpio 4 based wakeup
3	R/W	GPIO_3_WAKEUP	0	Enable gpio 3 based wakeup
2	R/W	GPIO_2_WAKEUP	0	Enable gpio 2 based wakeup
1	R/W	GPIO_1_WAKEUP	0	Enable gpio 1 based wakeup
0	R/W	GPIO_0_WAKEUP	0	Enable gpio 0 based wakeup

9.9.37 MCU_FSM_DEEP_SLEEP_DURATION_LSB_REG

Table 9.49. MCU_FSM_DEEP_SLEEP_DURATION_LSB_REG Register

Bit	Access	Function	Default Value	Description
31:0	R/W	MCUFSM_DEEPSLEEP_DURATION_COUNT	32'd0	LSB bits of deep sleep duration counter after which system wakes up is timeout wakeup is enabled If 1000µs is required. We need to program = 1000

9.9.38 MCU_FSM_XTAL_AND_PMU_GOOD_COUNT_REG

Table 9.50. FSM ON-Time Configuration Register

Bit	Access	Function	Reset Value	Description
31:23	-	Reserved	-	It is recommended to write these bits to 0.
22:16	RW	MCUFSM_XTAL_GOODTIME_DURATION_COUNT	15	Programmable duration for XTAL good time (22:21) 00 - same value to be assigned 01 - left shift by 1 bit (multiply by 2) and then assign to counters 10 - left shift by 2 bits (multiply by 4) and then assign to counters 11- left shift by 3 bits (multiply by 8) and then assign to counters (Max delay is 5.8ms) Specifies the combined ON Time for DC-DC 1.45 and LDO-SoC 1.15 0 - 10us 1 - 20us 2 - 25us 3 - 50us 4 - 100us 5 - 150us ..... 31 - 1450us
15:7	-	Reserved	-	It is recommended to write these bits to 0.

Bit	Access	Function	Reset Value	Description
6:0	RW	MCUFSM_PMU_POWERGOOD_DURATION_COUNT	15	Programmable duration for PMU power good time (6:5) 00 - same value to be assigned 01 - left shift by 1 bit (multiply by 2) and then assign to counters 10 - left shift by 2 bits (multiply by 4) and then assign to counters 11- left shift by 3 bits (multiply by 8) and then assign to counters (Max delay is 5.8ms) Specifies the Settling Time for HF-Crystal Clock 0 - 10us 1 - 20us 2 - 25us 3 - 50us 4 - 100us 5 - 150us ..... 31 - 1450us

9.9.39 MCU_FSM_CLKS_REG

Table 9.51. MCU_FSM_CLKS_REG Register

Bit	Access	Function	Default Value	Description
30:26	R	Reserved	--	Reserved
25	R/W	HF_FSM_CLK_EN	1	high frequency mcu fsm clock enable
24	R/W	HF_FSM_GEN_2MHZ	0	Enable 2Mhz clock for FSM 1 -Enable 2Mhz option 0- Enable 4MHz option
23:22	R/W	US_DIV_COUNT	3	One Micro second division factor. Program value to 3. If "hf_fsm_gen_2mhz" is '0 Program value to 1. If "hf_fsm_gen_2mhz" is '1'
21:16	R/W	HF_FSM_CLK_FREQ	32	High Frequency Source Clock value in MHz ex:Program 32 if 32MHz clock is used.
15	R	HF_FSM_CLK_SWITCHED_SYNC	1	If high freq fsm clock select is modified. Please poll this bit and wait till it becomes one.
14:5	--	--	--	--
4:2	R/W	HF_FSM_CLK_SELECT	0	0: No Clock 1: 20MHz RO 2: 32MHz RC 4: Not Valid
1:0	--	--	--	--

9.9.40 MCU_FSM_REF_CLK_REG

Table 9.52. MCU_FSM_REF_CLK_REG Register

Bit	Access	Function	Default Value	Description
31	R/W	SDCSS_STATIC_CLK_EN_b	1'b0	To enable static clk for sensor data collector subsystem
30	R/W	SDCSS_CLK_EN_b	1'b0	To enable dynamic clock for sdcss
29:28	R/W	SDCSS_CLK_SEL_b	2'd0	select between RC / RO 32KHz clk in sdcss 01 - 32MHz RC Clock 10- 20MHz RO Clock
27:25	--	--	0	--
24	R/W	ULPSS_REF_CLK_CLEANER_ON_b	1'b1	Clock cleaner On signal for ulpss ref clock
23	R/W	ULPSS_REF_CLK_CLEANER_OFF_b	1'b0	Clock cleaner Off signal for ulpss ref clock
22:19	--	--	--	--
18:16	R/W	ULPSS_REF_CLK_SEL_b	3'd1	Dynamic Reference Clock Mux select of ULPSS 0 : Clock will be gated at dynamic mux output in ULPSS 1 : ref_byp_clk to NWP 2 : ulp_32mhz_rc_clk 3 : rf_ref_clk 4 : mems_ref_clk 5 : ulp_20mhz_ringosc_clk 6 : ulp_doubler_clk 7: Clock will be gated at dynamic mux output in ULPSS
15	--	--	--	--
14:12	R/W	TASS_REF_CLK_SEL	3'd1	Dynamic Reference Clock Mux select of NWP controlled by M4. 0 : Clock will be gated at dynamic mux output of NWP 1 : ulp_32mhz_rc_byp_clk 2 : ulp_32mhz_rc_clk 3 : rf_ref_clk 4 : mems_ref_clk 5 : ulp_20mhz_ringosc_clk 6 : ref_byp_clk to NWP
11:9	--	--	0	--
8	R/W	M4SS_REF_CLK_CLEANER_ON_b	1'b1	Enable clk cleaner for m4ss reference clock
7	R/W	M4SS_REF_CLK_CLEANER_OFF_b	1'b0	Disable signal for m4ss reference clock
6:3	--	--	--	--

Bit	Access	Function	Default Value	Description
2:0	R/W	M4SS_REF_CLK_SEL	3'd1	Dynamic Reference Clock Mux select of M4SS 0 : Clock will be gated at dynamic mux output of M4SS 1 : ulp_32mhz_rc_byp_clk 2 : ulp_32mhz_rc_clk 3 : rf_ref_clk 4 : mems_ref_clk 5 : ulp_20mhz_ringosc_clk 6 : ulp_doubler_clk 7 : ref_byp_clk to NWP

9.9.41 MCU_FSM_CLK_ENS_AND_FIRST_BOOTUP

Table 9.53. MCU_FSM_CLK_ENS_AND_FIRST_BOOTUP Register

Bit	Access	Function	Default Value	Description
31:23	--	--	0	--
22	R/W	MCU_ULP_40MHZ_CLK_EN_b	1	Enables 40MHz XTAL clock
21	R/W	MCU_ULP_DOUBLER_CLK_EN_b	0	Enables ULP Doubler Clock
20	R/W	MCU_ULP_20MHZ_RING_OSC_CLK_EN_b	0	Enables ULP 20mhz RO Clock
19	R/W	MCU_ULP_32MHZ_RC_CLK_EN_b	1	Enables ULP 32MHz RC Clock
18	R/W	MCU_ULP_32KHZ_XTAL_CLK_EN_b	0	Enables ULP 32KHz Xtal Clock
17	R/W	MCU_ULP_32KHZ_RO_CLK_EN_b	1	Enables ULP 32KHz RO Clock
16	R/W	MCU_ULP_32KHZ_RC_CLK_EN_b	1	Enables ULP 32KHz Rc Clock
15	R	MCU_FSM_RESET_N_SYNC_b	0	Indicated MCU FSM is out of reset. 1 : Indicated MCU FSM is out of reset 0 : Indicated MCU FSM is in reset
14	R/W	MCU_ULP_1KHZ_RC_CLK_EN_b	1'b1	Enables ULP 1KHz Rc Clock (For MCU SYSRTC)
13:5	--	--	0	--
4	R	STORAGE_DOMAIN_ON_b	1	Indicates to S/W that MCU Data Storage 1 domain is ON. 1 - Domain is ON. 0 - Domain is OFF.
3	R	CHIP_MODE_VALID_b	0	Indicates to S/W that ChipMode programming are valid and need not read EFUSE. 1 - ChipMode are Valid. 0 - ChipModes are invalid.
2	R	RETENTION_DOMAIN_ON_b	1	Indicates to S/W that Retention domain is ON. 1 - Domain is ON. 0 - Domain is OFF.
1	R	RAM_RETENTION_STATUS_M4SS_b	0	Indicates to S/W that RAM's were in retention mode during Sleep time. 1 - RAM's are in retention mode during sleep. 0 - RAM's are not in retention mode during sleep.Domain is OFF.
0	R/W	FIRST_BOOTUP_MCU_N_b	0	Indication for S/W to distinguish b/w First Power or ULP wakeup. S/W need to set this Bit after first power .

9.9.42 MCU_FSM_CRTL_PDM_AND_ENABLES

Table 9.54. FSM Controlled POWER Domains Register

Bit	Access	Function	Reset Value	Description
31:20	-	Reserved	-	It is recommended to write these bits to 0.
19	RW	POWER_ENABLE_RETENTION_DM_b	1	Writing 1 to this enables Power to Retention Flops. These Flops are used for storing Chip Configuration. Writing 0 to this disables Power to Retention Flops. These Flops are used for storing Chip Configuration.
18	RW	POWER_ENABLE_DEEPSLEEP_TIMER_b	1	Writing 1 to this enables Power to DEEP SLEEP Timer. Writing 0 to this disables Power to DEEP SLEEP Timer.
17	RW	POWER_ENABLE_TIMESTAMPING_b	1	Writing 1 to this enables Power to TIMESTAMP. Writing 0 to this disables Power to TIMESTAMP.
16	RW	POWER_ENABLE_FSM_PERI_b	1	Writing 1 to this enables Power to Low-Power FSM. Writing 0 to this disables Power to Low-Power FSM.
15:5	-	Reserved	-	It is recommended to write these bits to 0.
4	RW	ENABLE_SRAM_DS_CRTL_b	1	
3	RW	DISABLE_TURNOFF_SRAM_PERI_b	1	
2	RW	RESET_MCU_BBF_DM_EN_b	0	Writing 1 to this enables reset of Power Domain Control Battery FF's on wakeup Writing 0 to this disables reset of Power Domain Control Battery FF's on wakeup
1	R	Reserved	0	Reserved
0	RW	ENABLE_WDT_IN_SLEEP_b	0	Writing 1 to this enables WDT during Sleep/ Shutdown states. Writing 0 to this disables WDT during Sleep/ Shutdown states.

9.9.43 MCU_GPIO_TIMESTAMPING_CONFIG

Table 9.55. MCU_GPIO_TIMESTAMPING_CONFIG Register

Bit	Access	Function	Default Value	Description
31:6	--		--	--
5	R/W	TIMESTAMPING_ON_GPIO4_b	0	Enable GPIO time stamping on GPIO4
4	R/W	TIMESTAMPING_ON_GPIO3_b	0	Enable GPIO time stamping on GPIO3
3	R/W	TIMESTAMPING_ON_GPIO2_b	0	Enable GPIO time stamping on GPIO2
2	R/W	TIMESTAMPING_ON_GPIO1_b	0	Enable GPIO time stamping on GPIO1
1	R/W	TIMESTAMPING_ON_GPIO0_b	0	Enable GPIO time stamping on GPIO0
0	R/W	ENABLE_GPIO_TIMESTAMPING_b	0	Enable GPIO time stamping Feature. This will enable measurement of GPIO high duration from SLEEP to wakeup

9.9.44 MCU_GPIO_TIMESTAMP_READ

Table 9.56. MCU_GPIO_TIMESTAMP_READ Register

Bit	Access	Function	Default Value	Description
31:27	--	--	--	--
26:16	R	GPIO_EVENT_COUNT_FULL	0	Counter value indicating number for 32MHz clock present in 1 Sleep clock (MCU FSM Clock)
15:11	--	--	--	--
10:0	R	GPIO_EVENT_COUNT_PARTIAL	0	Counter value indicating the duration from GPIO going high to first Sleep clock(MCU FSM Clock) posedge from GPIO going high with respect to 32MHz clock.

9.9.45 MCU_SLEEPHOLD_REQ

Table 9.57. MCU_SLEEPHOLD_REQ Register

Bit	Access	Function	Default Value	Description
31:17	--	--	--	--
16	W/R	SELECT_FSM_MODE	0	Enable for selecting secondary FSM. 1 - Select Secondary FSM 0 - Select Primary FSM
15:2	--	--	--	--
1	R	SLEEPHOLDACKn	1	SLEEPHOLDACK response to SLEEPHOLDREQ.
0	W/R	SLEEPHOLDREQn	1	Sleepholdreq when enable will gate the clock to M4. 1 - Sleepholdreq is Disabled. 0 - Sleepholdreq is Enabled

9.9.46 MCU_FSM_WAKEUP_STATUS_REG

Table 9.58. MCU_FSM_WAKEUP_STATUS_REG Register

Bit	Access	Function	Default Value	Description
31:18	--	--	--	--
17	R	MCU_FIRST_POWERUP_RESET_N	0	Indication to Processor that system came out of Reset.
16	R	MCU_FIRST_POWERUP_POR	0	Indication to Processor that system came out first power up.
15:11	--	--	--	--
10:0	R	WAKEUP_STATUS	0	To know the wakeup source. bit [10] - Host reset request bit [9] - nwp_wwd_window_reset bit [8] - nwp_wwd_reset bit [6] - mcu_wwd_window_reset bit [5] - mcu_wwd_reset bit [4] - mcu_processor_wake_stat bit [3] - cdbg power up request wakeup bit [2] - host based wakeup bit [1] - timeout wakeup bit [0] - wakeup indication

9.9.47 MCU_FSM_WAKEUP_STATUS_CLEAR

Table 9.59. MCU_FSM_WAKEUP_STATUS_CLEAR Register

Bit	Access	Function	Default Value	Description
31:11	--	--	--	--
10:0	R/W	wakeup_status_clear	0	To Clear Wakeup status register. Set Bits[10] - To Clear Sysrtc status indication. Set Bits[9] - To Clear Button-wake status indication. Set Bits[8] - To Clear BOD Wakeup status indication. Set Bits[7] - To Clear Comp4 wakeup (Bandgap En - VBatt Scale) status indication Set Bits[6] - To Clear comp3 wakeup (Analog IP1 & VBatt Scale) status indication. Set Bits[5] - To Clear comp2 wakeup (Analog IP1 & BandGap Scale) status indication. Set Bits[4] - To Clear comp1 wakeup (Analog IP1 & Analog IP2) status indication. Set Bits[3] - To Clear RTC Alarm wakeup status indicaition. Set Bits[2] - To Clear Second Tick wakeup status indication. Set Bits[1] - To Clear Milli-Second Wakeup status indication. Set Bits[0] - To Clear WatchDog Interrupt status indication.

9.9.48 MCU_FSM_PMU_STATUS_REG

Table 9.60. MCU_FSM_PMU_STATUS_REG Register

Bit	Access	Function	Reset Value	Description
30:23	--	Reserved	0	Reserved
22	R	POWERGOOD_DC1P3	0	Powergood signal read for DC 1.3V
21	R	POWERGOOD_LDORF	0	Powergood signal from LDORF
20	R	POWERGOOD_LDOSOC	0	Powergood signal from ldosoc
19	R/W	STANDBY_DC1P3_R	0	Standby state for DC1p3
18	R/W	STANDBY_LDOSOC_R	0	Standby state for LDO soc
17	R/W	STANDBY_LDORF_R	0	Standby state for LDO RF
16:8	--	--	--	--
7:4	R/W	socldo_powergood_delay_count	8	Delayed counter for powergood signal Minimum value that needs to be programmed for this bit is 4
3:2	R/W	pmu_timer_powergood_based_en	0	Option for deciding on the PMU DCDC/SoC-LDO power good. 00 - Based on timer 01 - Based on powergood signal 1X - Based on both timer and PMU powergood
1	R/W	BGPMU_SLEEP_EN_R_b	0	Sleep en for BG PMU
0	R/W	SCDCDC_LP_MODE_EN	0	SCDC in LP mode

9.9.49 MCU_FSM_PMUX_CTRLS_RET

Table 9.61. MCU_FSM_PMUX_CTRLS_RET Register

Bit	Access	Function	Reset Value	Description
31:8	--	--	--	--
7:6	R	POWER_SW_CTRL_ULPSS_RAM_IN_RETAIN	0	Select value for ULPSS RAM Power Mux In Retention mode 3 – SOC LDO 1 – SCDCDC 0.9 0 – SCDCDC 0.6
5:4	R/W	POWER_SW_CTRL_M4ULP_RAM16K_IN_RETAIN	0	Select value for M4ULP 16K RAM Power Mux In Retention mode 3 – SOC LDO 1 – SCDCDC 0.9 0 – SCDCDC 0.6
3:2	R/W	POWER_SW_CTRL_M4ULP_RAM_IN_RETAIN	0	Select value for M4ULP RAM Power Mux In Retention mode 3 – SOC LDO 1 – SCDCDC 0.9 0 – SCDCDC 0.6
1	R/W	Reserved	0	Reserved
0	R/W	POWER_SW_CTRL_TASS_RAM_IN_RETAIN	0	Select value for NWP RAM Power Mux In Retention mode 1 – SOC LDO 0 – SCDCDC 0.6

9.9.50 MCU_FSM_TOGGLE_COUNT

Table 9.62. MCU_FSM_TOGGLE_COUNT Register

Bit	Access	Function	Reset Value	Description
31	R	TOGGLE_DATA_READY		Toggle data Ready
30:28	--	--	0	--
27:16	R	GPIO_TOGGLE_COUNT	0	GPIO toogle data count
15	W	LATCH_TOGGLE_DATA	0	Trigger indication to read GPIO toggle data.
14:1	--	--	0	--
0	W	TOGGLE_COUNT_RSTART	0	Start counting GIPO Toggle's

9.9.51 M4SS_TASS_CTRL_SET_REG

Table 9.63. M4SS_TASS_CTRL_SET_REG

Bit	Access	Function	Reset Value	Description
31:3	R	Reserved	--	Reserved
2	R/W	M4SS_CTRL_TASS_AON_PWR_DMN_RST_BYPASS	1	Writing 1 to this , M4SS can Turn-ON NWP AON domain's reset pin in bypass mode. Writing 0 to this , M4SS can't Turn-ON NWP AON domain's reset pin in bypass mode.
1	R/W	M4SS_CTRL_TASS_AON_DISABLE_ISOLATION_BYPASS	0	Writing 1 to this , M4SS can Turn-ON NWP AON domain's isolation enable in bypass mode. Writing 0 to this , M4SS can't Turn-ON NWP AON domain's isolation enable in bypass mode.
0	R/W	M4SS_CTRL_TASS_AON_PWRGATE_EN	0	Writing 1 to this , M4SS can Turn-ON NWP AON domain. Writing 0 to this , M4SS can't Turn-ON NWP AON domain.

9.9.52 M4SS_TASS_CTRL_CLEAR_REG

Table 9.64. M4SS_TASS_CTRL_CLEAR_REG

Bit	Access	Function	Reset Value	Description
31:3	R	Reserved	--	Reserved
2	R/W	M4SS_CTRL_TASS_AON_PWR_DMN_RST_BYPASS	1	Writing 1 to this , M4SS can Turn-OFF NWP AON domain's reset pin in bypass mode. Writing 0 to this , M4SS can't Turn-OFF NWP AON domain's reset pin in bypass mode.
1	R/W	M4SS_CTRL_TASS_AON_DISABLE_ISOLATION_BYPASS	0	Writing 1 to this , M4SS can Turn-OFF NWP AON domain's isolation enable in bypass mode. Writing 0 to this , M4SS can't Turn-OFF NWP AON domain's isolation enable in bypass mode.
0	R/W	M4SS_CTRL_TASS_AON_PWRGATE_EN	0	Writing 1 to this , M4SS can Turn-OFF NWP AON domain. Writing 0 to this , M4SS can't Turn-OFF NWP AON domain.

9.9.53 M4_ULP_MODE_CONFIG

Table 9.65. M4_ULP_MODE_CONFIG

Bit	Access	Function	Reset Value	Description
31:6	--	--	--	--
5	R/W	ULPMODE_ISOLATION_CTRL_M4_ROM	0	Writing 1 to this enables ULP-Mode non-functional paths for ROM Writing 0 to this disables ULP-Mode non-functional paths for ROM
4	R/W	ULPMODE_ISOLATION_CTRL_M4_DEBUG_FPU	0	Writing 1 to this enables ULP-Mode non-functional paths for M4_DEBUG and FPU. Writing 0 to this disables ULP-Mode non-functional paths for M4_DEBUG and FPU.
3	R/W	ULPMODE_ISOLATION_CTRL_M4_CORE	0	Writing 1 to this enables ULP-Mode non-functional paths for M4_CORE Writing 0 to this disables ULP-Mode non-functional paths for M4_CORE
2	R/W	ULPMODE_ISOLATION_CTRL_M4_ULP	0	Writing 1 to this enables ULP-Mode non-functional paths for M4ULP_AON Writing 0 to this disables ULP-Mode non-functional paths for M4ULP_AON
1	R/W	ULPMODE_ISOLATION_CTRL_M4SS_AON	0	Writing 1 to this enables ULP-Mode non-functional paths for M4SS-AON Writing 0 to this disables ULP-Mode non-functional paths for M4SS-AON
0	R/W	ULPMODE_ISOLATION_CTRL_ULPSS	0	Writing 1 to this enables ULP-Mode non-functional paths for ULPSS Writing 0 to this disables ULP-Mode non-functional paths for ULPSS

9.9.54 ULPSS_BYPASS_PWRCTRL_REG

Table 9.66. ULPSS_BYPASS_PWRCTRL_REG

Bit	Access	Function	Reset Value	Description
31:23	--	--	0	--
22:19	R/W	BYPASS_ULPTASS_PWRCTL_ULP_SRAM	0	Writing 1 to this Enables software based control of output isolation for ULPTASS SRAM Writing 0 to this disables software based control of output isolation for ULPTASS SRAM
18:15	--	--	0	--
14	R/W	BYPASS_ULPTASS_PWRCTL_ULP_UDMA	0	Writing 1 to this Enables software based control of output isolation for ULP UDMA Writing 0 to this disables software based control of output isolation for ULP UDMA
13	R	Reserved	-	
12	R/W	BYPASS_ULPTASS_PWRCTL_ULP_AUX	0	Writing 1 to this Enables software based control of output isolation for ULP AUX Writing 0 to this disables software based control of output isolation for ULP AUX
11	R/W	BYPASS_ULPTASS_PWRCTL_ULP_I2C	0	Writing 1 to this Enables software based control of output isolation for ULP I2C Writing 0 to this disables software based control of output isolation for ULP I2C
10	R/W	BYPASS_ULPTASS_PWRCTL_ULP_I2S	0	Writing 1 to this Enables software based control of output isolation for ULP I2S Writing 0 to this disables software based control of output isolation for ULP I2S
9	R/W	BYPASS_ULPTASS_PWRCTL_ULP_SSI	0	Writing 1 to this Enables software based control of output isolation for ULP SSI Writing 0 to this disables software based control of output isolation for ULP SSI
8	R/W	BYPASS_ULPTASS_PWRCTL_ULP_UART	0	Writing 1 to this Enables software based control of output isolation for ULP UART Writing 0 to this disables software based control of output isolation for ULP UART

Bit	Access	Function	Reset Value	Description
7	R/W	Reserved	0	Reserved
6	R/W	BYPASS_ULPTASS_PWRCTL_ULP_CAP	0	Writing 1 to this Enables software based control of output isolation for ULP CAP Writing 0 to this disables software based control of output isolation for ULP CAP
5	R/W	BYPASS_ULPTASS_PWRCTL_ULP_MISC	0	Writing 1 to this Enables software based control of output isolation for ULP MISC Writing 0 to this disables software based control of output isolation for ULP MISC
4	--	--	--	--
3	R/W	BYPASS_ULPSDCSS_PWRCTRL_ULP_AON	0	Writing 1 to this Enables software based control of output isolation for ULPSDCSS AON Writing 0 to this disables software based control of output isolation for ULPSDCSS AON
2	R/W	BYPASS_ULPTASS_PWRCTL_ULP_AON	0	Writing 1 to this Enables software based control of output isolation for ULPTASS AON Writing 0 to this disables software based control of output isolation for ULPTASS AON
1:0	--	--	--	--

9.9.55 Analog_Power_Control

Table 9.67. Analog Power Control Register

Bit	Access	Function	Reset Value	Description
31:17	-	Reserved	-	It is recommended to write these bits to 0.
16	W	PWRCTRL_BOD	1	Writing 1 to this enables Power to Brown-Out Detector. Writing 0 to this disables Power to Brown-Out Detector.
15:14	-	Reserved	-	Reserved
13	-	Reserved	-	Reserved
12	-	Reserved	-	Reserved
11	W	PWRCTRL_AUXADC	1	Writing 1 to this enables Power to Auxillary ADC. Writing 0 to this disables Power to Auxillary ADC.
10:9	-	Reserved	-	It is recommended to write these bits to 0.
8	W	PWRCTRL_AUXDAC	1	Writing 1 to this enables Power to Auxillary DAC. Writing 0 to this disables Power to Auxillary DAC.
7:0	-	Reserved	-	It is recommended to write these bits to 0.

9.9.56 MCURET_QSPI_WR_OP_DIS

Table 9.68. MCURET_QSPI_WR_OP_DIS

Bit	Access	Function	Default Value	Description
31:2	--	Reserved	0	Reserved
1	R	TASS_QSPI_WRSR_WR_OP_DISABLE	0	Writing 1 to this disables NWP Write operation to Flash. Writing 0 to this enables NWP Write operation to Flash.
0	R/W	M4SS_QSPI_WRSR_WR_OP_DISABLE	0	Writing 1 to this disables M4SS Write operation to Flash. Writing 0 to this enables M4SS Write operation to Flash.

9.9.57 MCURET_BOOTSTATUS

Table 9.69. MCURET_BOOTSTATUS

Bit	Access	Function	Default Value	Description
31	--	--	--	--
30:0	R	BOOT_STATUS	0	Gives Boot Status/Configuration information to MCU

9.9.58 MCUAON_CTRL_REG4

Table 9.70. MCUAON_CTRL_REG4

Bit	Access	Function	Default Value	Description
31:9	--	Reserved	--	Reserved
8	R/W	mcu_tass_ref_clk_sel_mux_ctrl	0	1 - NWP has TASS ref clock control 0 - nwp_tass_ref_clk_sel_mux_ctrl determines the control. M4 has control only when both are zero
7:0	--	Reserved	--	Reserved

9.9.59 NPSS_GPIO_0_CTRLS

Table 9.71. NPSS_GPIO_0_CTRLS

Bit	Access	Function	Default Value	Description
31:17	NA	Reserved	0	Reserved
16	R/W	use_ulpss_pad_0	0	Writing 1 to this, maps ULPSS GPIO-0 to NPSS GPIO-0 Writing 0 to this, does not map ULPSS GPIO-0 to NPSS GPIO-0
15:9	NA	Reserved	0	Reserved
8	R/W	npss_gpio_0_polarity	0	NPSS GPIO 0 Polarity 1 - When signal is High 0 - When signal is Low
7	--	Reserved	0	--
6	R/W	npss_gpio_0_pad_select	0	NPSS GPIO 0 Pad Selection between M4 and NWP 0 - M4 has control over this GPIO output value 1 - NWP has control over this GPIO output value
5	R/W	npss_gpio_0_out	0	NPSS GPIO 0 Output value is written here.
4	R/W	npss_gpio_0_oen	1	Writing 0 to this enables NPSS GPIO 0 Output. Writing 1 to this enables NPSS GPIO 0 input.
3	R/W	npss_gpio_0_ren	0	Writing 1 to this enables NPSS GPIO 0 Input Buffer. Writing 0 to this disables NPSS GPIO 0 Input Buffer.
2:0	R/W	npss_gpio_0_mode	1	NPSS GPIO 0 mode select. Please refer to datasheet for npss gpio pin muxing

9.9.60 NPSS_GPIO_1_CTRLS

Table 9.72. NPSS_GPIO_1_CTRLS

Bit	Access	Function	Default Value	Description
31:17	NA	Reserved	0	Reserved
16	R/W	use_ulpss_pad_1	0	Writing 1 to this, maps ULPSS GPIO-1 to NPSS GPIO-1 Writing 0 to this, does not map ULPSS GPIO-1 to NPSS GPIO-1
15:9	NA	Reserved	0	Reserved
8	R/W	npss_gpio_1_polarity	0	NPSS GPIO 1 Polarity 1 - When signal is High 0 - When signal is Low
7	--	Reserved	0	--
6	R/W	npss_gpio_1_pad_select	0	NPSS GPIO 1 Pad Selection between M4 and NWP 0 - M4 has control over this GPIO output value 1 - NWP has control over this GPIO output value
5	R/W	npss_gpio_1_out	0	NPSS GPIO 1 Output value is written here.
4	R/W	npss_gpio_1_oen	1	Writing 0 to this enables NPSS GPIO 1 Output. Writing 1 to this enables NPSS GPIO 1 input.
3	R/W	npss_gpio_1_ren	0	Writing 1 to this enables NPSS GPIO 1 Input Buffer. Writing 0 to this disables NPSS GPIO 1 Input Buffer.
2:0	R/W	npss_gpio_1_mode	1	NPSS GPIO 1 mode select. Please refer to datasheet for npss gpio pin muxing

9.9.61 NPSS_GPIO_2_CTRLS

Table 9.73. NPSS_GPIO_2_CTRLS

Bit	Access	Function	Default Value	Description
31:17	NA	Reserved	0	Reserved
16	R/W	use_ulpss_pad_2	0	Writing 1 to this, maps ULPSS GPIO-2 to NPSS GPIO-2 Writing 0 to this, does not map ULPSS GPIO-2 to NPSS GPIO-2
15:9	NA	Reserved	0	Reserved
8	R/W	npss_gpio_2_polarity	0	NPSS GPIO 2 Polarity 1 - When signal is High 0 - When signal is Low
7	--	Reserved	0	--
6	R/W	npss_gpio_2_pad_select	0	NPSS GPIO 2 Pad Selection between M4 and NWP 0 - M4 has control over this GPIO output value 1 - NWP has control over this GPIO output value
5	R/W	npss_gpio_2_out	0	NPSS GPIO 2 Output value is written here.
4	R/W	npss_gpio_2_oen	1	Writing 0 to this enables NPSS GPIO 2 Output. Writing 1 to this enables NPSS GPIO 2 input.
3	R/W	npss_gpio_2_ren	0	Writing 1 to this enables NPSS GPIO 2 Input Buffer. Writing 0 to this disables NPSS GPIO 2 Input Buffer.
2:0	R/W	npss_gpio_2_mode	1	NPSS GPIO 2 mode select. Please refer to datasheet for npss gpio pin muxing

9.9.62 NPSS_GPIO_3_CTRLS

Table 9.74. NPSS_GPIO_3_CTRLS

Bit	Access	Function	Default Value	Description
31:17	NA	Reserved	0	Reserved
16	R/W	use_ulpss_pad_3	0	Writing 1 to this, maps ULPSS GPIO-3 to NPSS GPIO-3 Writing 0 to this, does not map ULPSS GPIO-3 to NPSS GPIO-3
15:9	NA	Reserved	0	Reserved
8	R/W	npss_gpio_3_polarity	0	NPSS GPIO 3 Polarity 1 - When signal is High 0 - When signal is Low
7	NA	Reserved	0	--
6	R/W	npss_gpio_3_pad_select	0	NPSS GPIO 3 Pad Selection between M4 and NWP 0 - M4 has control over this GPIO output value 1 - NWP has control over this GPIO output value
5	R/W	npss_gpio_3_out	0	NPSS GPIO 3 Output value is written here.
4	R/W	npss_gpio_3_oen	1	Writing 0 to this enables NPSS GPIO 3 Output. Writing 1 to this enables NPSS GPIO 3 input.
3	R/W	npss_gpio_3_ren	0	Writing 1 to this enables NPSS GPIO 3 Input Buffer. Writing 0 to this disables NPSS GPIO 3 Input Buffer.
2:0	R/W	npss_gpio_3_mode	0	NPSS GPIO 3 mode select. Please refer to datasheet for npss gpio pin muxing

10. Power Management Unit

10.1 Features

The PMU has following features,

1. PMU modes of Operation for high system efficiency
    a. Active mode
    b. Sleep mode
    c. Ultra sleep mode
    d. LDO switch mode

10.2 Functional Description

10.2.1 Block Diagram

Following is the block diagram of Power Management Unit

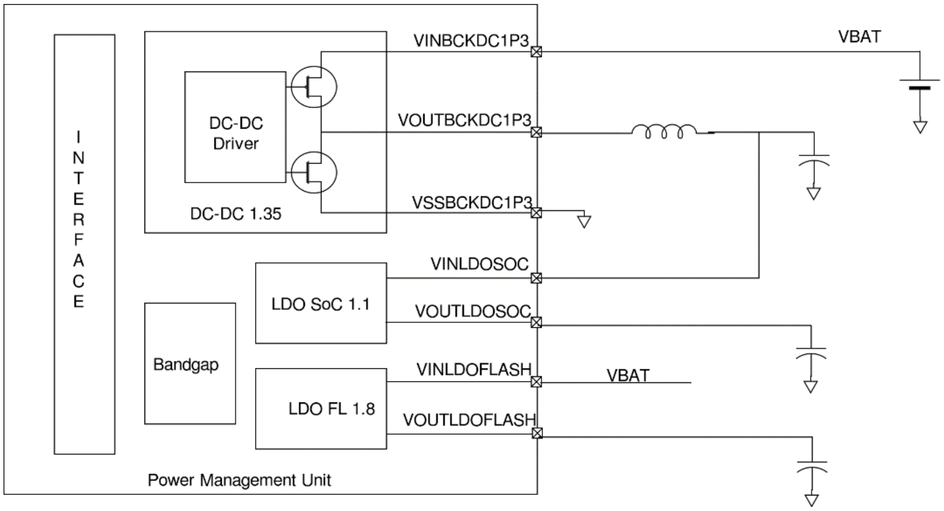


Figure 10.1. Block Diagram of Power Management Unit

10.2.2 Modes of Operation

PMU can be configured to one of the following modes of operation to optimize the overall system efficiency for various system load requirements.

10.2.2.1 Active Mode

PMU can be configured to Active mode trough APIs. In the active mode, PMU supports maximum power requirement of SOC. DC-DC 1.35 converter works in PWM mode with fixed frequency to achieve high efficiency at high load condition. The LDOs can support their maximum load current.

10.2.2.2 Sleep Mode

PMU can be configured to Sleep mode trough APIs. In the sleep mode, PMU supports no more than 50mA current for DC-DC 1.35 and LDO SOC 1.1. DC-DC 1.35 converter works in PFM mode with variable frequency to achieve high efficiency at low load condition. In this mode, LDO FL 1.8 can support max current.

10.2.2.3 Ultra Sleep Mode

PMU can be configured to Ultra Sleep mode trough APIs. In the Ultra sleep mode, both the LDOs are turned off and DC-DC 1.35 is configured in PFM mode. In this mode, the PMU consumes less than 1µA current and retain the buck output voltage to 1.2V.

10.2.2.4 LDO Switch Mode

Both the LDOs can be configured in LDO switch mode through APIs. In LDO switch mode, LDO is bypassed and power MOSFET used as switch to pass input voltage directly to the output voltage. It can be configured in PMU active or PMU sleep mode.

10.3 Register Summary

Base Address: 0x2405_8000

Register Name	offset	Description
Section 10.4.1 PMU_IP3_CTRL_REG	0x740	
Section 10.4.2 PMU_LDOSOC_REG	0x758	

10.4 Register Description

10.4.1 PMU_IP3_CTRL_REG

Bit	Access	Function	Default Value	Description	Dynamic Controllable
20:17	R/W	set_vref1p3	4'd11	Set DC-DC 1.35 output voltage. 0000 - 0.8V 0001 - 0.86V 0010 - 0.91V 0011 - 0.96V 0100 - 1.01V 0101 - 1.06V 0110 - 1.11 V 0111 - 1.16V 1000 - 1.21V 1001 - 1.26V 1010 - 1.31V 1011 - 1.36V 1100 - 1.41V 1101 - 1.46V 1110 - 1.51V 1111 - 1.56V	Yes

10.4.2 PMU_LDOSOC_REG

Bit	Access	Function	Default Value	Description	Dynamic Controllable
9:6	R/W	CLRL_LDOFLASH	4'd3	Set LDO FL 1.8 Output voltage 0000 - 1.6V 0001 - 1.68V 0010 - 1.76V 0011 - 1.84V 0100 - 1.92V 0101 - 2V 0110 - 2.08V 0111 - 2.16V 1000 - 2.24V 1001 - 2.32V 1010 - 2.4V 1011 - 2.48V 1100 - 2.56V 1101 - 2.64V 1110 - 2.72V 1111 - 2.8V	Yes
3:0	R/W	CTRL_LDOSOC	4'd11	Set LDO SOC 1.1 output voltage 0000 - 0.50 V 0001 - 0.55 V 0010 - 0.60 V 0011 - 0.65 V 0100 - 0.70 V 0101 - 0.75 V 0110 - 0.80 V 0111 - 0.90 V 1000 - 0.95 V 1001 - 1 V 1010 - 1.05 V 1011 - 1.10 V 1100 - 1.15 V 1101 - 1.2 V 1110 - 1.25 V 1111 - 1.3 V	Yes

10.5 PMU Good Time

10.5.1 Direct Battery Connected PMU Good Time

If the battery is directly conneted to VINBCKDC and VINLDOFLASH, then PMU takes following time to generate PMU power good. This data is based on simulation results

Blocks	Up Time (µs) (From Supply Rampup to PMU On)	Up Time (µs) (From Ultra Sleep mode to PMU Active mode)	Up Time (µs) (From LDO FL Off to On)
Supply Rampup[1]	10	-	-
Bandgap	225	225	-
Buck	170	25	-
LDOSOC	15	15	-
LDORF	20	20	20
PMU Total	430	285	20

Note:

1. If supply rampup time increased, PMU powergood time will also increased. Bandgap will start once supply reached to 1.6V.

10.5.2 Cascaded Power Supply PMU Good Time

If VINBCKDC and VINLDOFLASH are connected through an external power gate (PGATE) to battery or are powered by an external BOOST regulator, then add the above numbers to the Tstab (the voltage stabilization time of the BOOST or power gate) to arrive at the final PMU_GOOD_TIME. Tstab is defined as the time it takes for the BOOST or PGATE to charge the VINBCKDC and VINDLOFLASH to 1.6V.

For example, if Tstab is 20µs, then the final PMU_Good_Time from supply ramp up to PMU On is 450µs (430µs+20µs).

Save

11. ULP Regulators

11.1 General Description

Ultra-low-power (ULP) regulators are used to power low power Always-ON (AON) digital and analog power management circuitry inside the IC. Additionally, ULP regulators can be used to power the SRAMs and M4 core. The ULP regulators include two high power LDOs, a low-power LDO, and a switched capacitor DC-DC regulator.

11.2 Features

1. Support wide programmable output voltage range at very low quiescent current
2. Support wide current ranges (0.4 µA to 8mA)
3. Have two modes of operation:
    a. High Power Mode
    b. Low Power Mode
4. Support modes to bypass internal regulators

11.3 Functional Description

11.3.1 Block Diagram

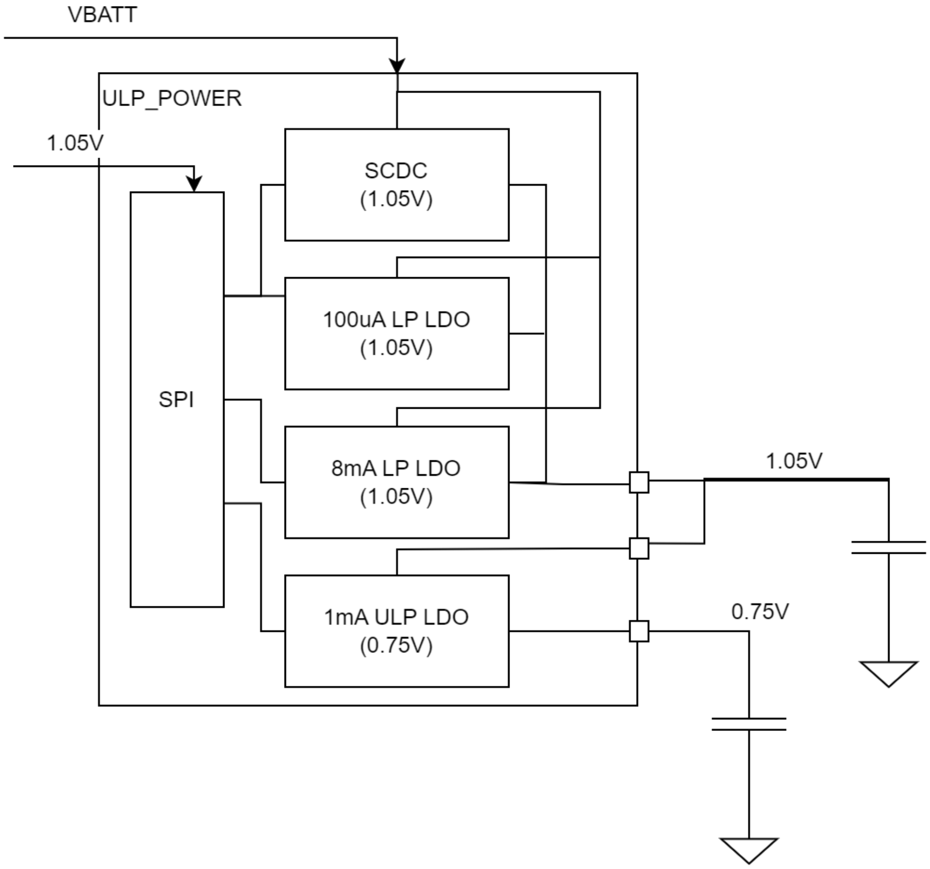


Figure 11.1. Block Diagram of ULP Regulators

Power Up:

Upon battery insertion, voltage VBATT ramps up by enabling the Bandgap (UULP_VBAT_Peripheral). The Bandgap generates a stable voltage reference to the 8 mA LP LDO, which in turn generates a stable 1.0 V output. Power On Control (POC), another UULP_VBAT_Peripheral, monitors the output of the regulator and holds the IC under reset until the output voltage of this regulator reaches a value that is high enough to facilitate a safe operation. Upon release of the POC reset, the LDO 0.7 V is enabled. This LDO can support a maximum load current of 1 mA.

Low Power State:

When the IC needs to enter a low power state, the provided APIs can be used to switch OFF the 8 mA LDO and transition over to the 100 µA LDO. This transition would enable lowering the quiescent current consumed by the LDO.

SC DC-DC Mode:

A switched capacitor DC-DC regulator can be used, instead of the 8 mA LP LDO, to improve the power conversion efficiency of the IC. The DC-DC converter loop automatically tracks the VBATT and changes the built-in gain to provide a high efficiency across a wide input voltage range (2.1 to 3.6 V). One can use the provided API to automatically transition over to the SC DC-DC mode after the first power up, and continue to remain in this mode until VBATT falls below 2.1 V. At VBATT lower than 2.1 V, the DC-DC regulator begins to operate in linear mode, hence the API automatically transitions back to the 8 mA LDO.

Bypass Options:

Each of the LDOs can be bypassed by over-driving the outputs with either a high efficiency buck regulator or a linear regulator

11.4 Register Summary

Base Address: 0x2405_A000

Register Name	Offset	Reset Value	Description
11.5.1 SCDC_CTRL_REG_0	0x498	22'h1E002F	SCDC-DC Algorithm Control Register
11.5.2 BG_SCDC_PROG_REG_1	0x49C	22'h200498	DC-DC / LDO Output Programmability Selection
11.5.3 BG_SCDC_PROG_REG_2	0x4A0	22'h000050	Enable Controls
11.5.4 BG_LDO_REG	0x4A4	22'h088000	LDO 0.7 V Controls

11.5 Register Description

11.5.1 SCDC_CTRL_REG_0

Table 11.1. SCDC_CTRL_REG_0 Register

Bit	Access	Function	Default Value	Description
21	R/W	ext_cap_en	1'b0	To change current trim bits to high or low through spi, based on high power or low power mode. When 0, curr prog value is 0.
20:17	R/W	fixed_curr_prog_high	4'd15	Current prog value to take when ext cap en is high and sel_high freq_ext_b is 0
16:13	R/W	fixed_curr_prog_low	4'd0	Current prog value to take when ext cap en is high and sel_high freq_ext_b is 1
12	R/W	bypass_trim_ro	1'b0	To program the trim value manually, irrespective of the fsm
11:7	R/W	fixed_trim_ro	5'd0	Manual trim word
6	R/W	fixed_mode	1'b0	Fixed mode
5:4	R/W	max_mode	2'd2	Maximum mode
3:0	R/W	count_reset	4'hF	Count reset value, count threshold will be double this value

11.5.2 BG_SCDC_PROG_REG_1

Table 11.2. BG_SCDC_PROG_REG_1 Register

Bit	Access	Function	Default Value	Description
21:19	R/W	bg_r_ptat	3'd2	Bandgap voltage programming
18:16	R/W	reserved	3'd0	Reserved
15	R/W	bg_en	1'b0	bg_en from spi
14	R/W	bg_sh_en	1'b0	bg_sh_en from spi
13	--	Reserved	0	Reserved
12:10	R/W	ref_sel_dcdc	3'd1	DCDC output programming vref_1p1/vref_1p05 3'd0 - 1.15/1.1 3'd1 - 1.1/1.05 3'd2 - 1.05/1.0 3'd3 - 1.0/0.95 3'd4 - 0.95/0.9 3'd5 - 0.9/0.85

Bit	Access	Function	Default Value	Description
9:7	R/W	ref_sel_lp_dcdc	3'd1	DCDC output programming in LDO high/low power mode 3'd0 -1.1 3'd1 - 1.15 3'd2 - 1.05 3'd3 - 1.0 3'd4 - 0.95 3'd5 - 0.9
6:5	-	Reserved	0	Reserved
4	R/W	bod_clks_ptat_en	1'b1	1 - To enable ptat currents to clocks and bod(cmp_npss)
3	R/W	an_perif_ptat_en	1'b1	1 - To enable ptat currents to analog peripherals
2:0	R/W	ref_sel_PMU	3'd0	3'd0 - 1.2 V 3'd1 - 1.15 V 3'd2 - 1.1 V 3'd3 - 1.05 V 3'd4 - 1.0 V 3'd5 - 0.95 V 3'd6 - 0.9 V 3'd7 - 0.85 V

11.5.3 BG_SCDC_PROG_REG_2

Table 11.3. BG_SCDC_PROG_REG_2 Register

Bit	Access	Function	Default Value	Description
21	R/W	scdcdc_sel	1'b0	To switch to SCDCDC mode from LDO mode. 1 - SCDC mode 0 - LDO mode
20	R/W	testmode_0_en	1'b0	Enable for output on to BG_TESTMODE0
19:18	R/W	testmode_0_sel	2'd0	2'd0: bg_sw_active 2'd1: scdcdc_sown 2'd2: scdcdc_lp_mode (sel_high_freq_ext_b) 2'd3: scdcdc_sel (To select ldo - scdcdc)
17	R/W	testmode_1_en	1'b0	To enable test mux for BG_TESTMODE1
16:15	R/W	testmode_1_sel	2'd0	2'd0: bg_sh_en 2'd1: scdcdc_up 2'd2: scdcdc_en (Enable for scdcdc block) 2'd3: scdcdc_lp_en (enable for 10uA LDO)
14	R/W	testmode_2_en	1'd0	To enable testmux for BG_TESTMODE2
13:11	R/W	testmode_2_sel	3'd0	3'd0: bg_en 3'd1: bg_comp_clk 3'd2: en_ldo_5m_b 3'd3: comp_clk 3'd4: scdcdc_conv_1b1 3'd5: scdcdc_conv_1b2 3'd6: scdcdc_conv_1b3 3'd7: 0

Bit	Access	Function	Default Value	Description
10:6	R/W	trim_clamp_lp	5'd1	Trim value lower clamp value when sel high freq_b is 1
5:1	R/W	trim_clamp_hp	5'd16	Trim value lower clamp value when sel high freq_b is 0
0	R/W	scdcdc_soft_reset	0	Soft reset signal for scdcdc fsm

11.5.4 BG_LDO_REG

Table 11.4. BG_LDO_REG Register

Bit	Access	Function	Default Value	Description
21	R/W	LDO_0P6_BYPASS	1'b0	bypass signal for DCDC1p1_lp_500uA
20:18	R/W	LDO_0P6_CTRL	3'd2	vref for DCDC1p1_lp_500uA 3'd0 - 0.8 V 3'd1 - 0.75 V 3'd2 - 0.7 V 3'd3 - 0.65 V 3'd4 - 0.6 V 3'd5 - 0.55 V
17	--	Reserved	0	Reserved
16	R/W	LDO_0P6_LP_MODE	1'b0	Enable low power mode, otherwise in high power mode
15	R/W	LDO_0P6_ENABLE	1'b1	Enable digital LDO
14:5	--	Reserved	0	Reserved
4	R/W	test_amux_en	1'b0	Enable analog mux to test reference voltages
3:1	R/W	test_amux_sel	3'd0	Select for analog mux 3'd0: Vbg_core 3'd1: vref_1p05 3'd2: vref_ulp 3'd3: vbg_lp_buff
0	--	Reserved	--	Reserved

12. Pad Configurations

12.1 General Description

There are a total of 45 GPIOs present. The number of GPIOs available varies between different packages. Refer to the GPIO available vs package table in the product data sheet for more details. Registers for GPIO pins that are not available on package are reserved. There are multiple processor sub-systems containing NWP, MCU High Performance (HP), and MCU Ultra-Low-Power (ULP) which share these common set of GPIO pads. These GPIO pads are controllable by either NWP, MCU HP, or MCU ULP. The PAD selection register must be programmed to control the PAD behavior for each GPIO.

The list below provides the registers to be configured for accessing any of the GPIO pads.

- PAD Selection Register.
- PAD Configuration Register.
- GPIO Mode Register.

More details about pad selection and pad configuration are described below.

12.2 Features

The 45 GPIOs are divided into 30 SoC GPIOs, 11 ULP GPIOs, and 4 Ultra ULP (UULP) Vbat GPIOs. The SoC GPIOs are available only in PS4/PS3 Active power states (as described in Section 9. Power Architecture), whereas ULP GPIOs are available in all the power states except for PS0 and sleep modes. The UULP Vbat GPIOs are available in all power states.

GPIOs Availability in Different Power States

The table below indicates the different GPIOs' availability in each of the power states.

S.No	Block	PS4	PS3	PS2	PS1	PS0	PS4-Sleep	PS3-Sleep	PS2-Sleep	PS4-Standby	PS3-Standby	PS2-Standby
1	SoC GPIOs	YES	YES	NO	NO	NO	NO	NO	NO	YES	YES	NO
2	ULP GPIOs	YES	YES	YES	YES	NO	NO	NO	NO	YES	YES	YES
3	UULP Vbat GPIOs	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES

The SoC GPIOs/NWP and MCUHP GPIOs and ULP GPIOs PAD are programmable, multi-voltage (1.8 V, 2.5 V, 2.8 V. 3.0 V, 3.3 V) general purpose, bi-directional I/O buffer with a selectable Low Voltage CMOS (LVCMOS) input or LVCMOS Schmitt trigger input and programmable pull-up/pull-down. In the full-drive mode, this buffer can operate in excess of 100 MHz frequency with 15 pF external load and 125 MHz with 10 pF load, but actual frequency is load and system dependent. A maximum of 200 MHz can be achieved under small capacitive loads.

The following PAD configurations can be controlled by software for SoC GPIOs and ULP GPIOs.

- Bi-directional IO capability
- Multi-voltage DVDD capability (1.8 V, 2.5 V, 2.8 V, 3.0 V, 3.3 V)
- Power-on-Start (POS) capable
- Optimized for EMC (low di/dt switching supply noise) with Simultaneous Switching Output (SSO) factor of 8
- Four (4) Programmable output drive strengths (rated 2 mA, 4 mA, 8 mA, and 12 mA)
- Selectable output slew-rate (slow/fast)
- Open drain output mode (logic low or high on input and use OEN as data input)
- LVCMOS/LVTTL compatible input with selectable hysteresis
- Programmable input options (pull-up, pull-down, repeater, or plain input)
- No power sequence requirements, I/Os are tri-stated when core power is not valid (POC control). These are tri-stated even if the system is under reset or in the deep sleep power state.

The following PAD configurations can be controlled by software for UULP Vbat GPIOs.

- Bi-directional IO capability
- Multi-voltage DVDD capability (1.8 V, 2.5 V, 2.8 V, 3.0 V, 3.3 V)

12.3 Functional Description

The figure below depicts the PAD model used for SoC-GPIOs and ULP-GPIOs.

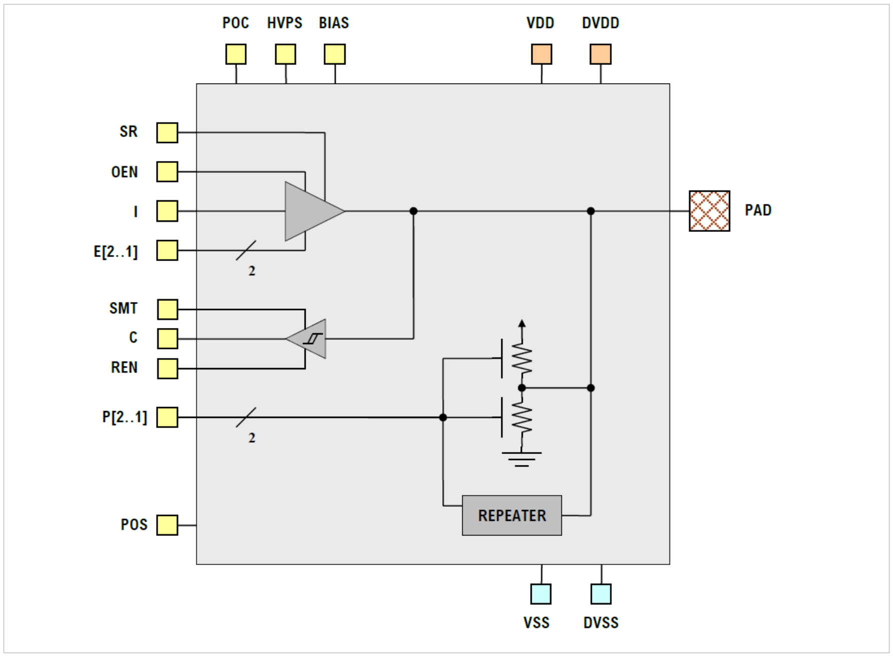


Figure 12.1. Pad Model

12.3.1 PAD Description

Table 12.1. Ports

Port Name	Direction	Description
PAD	INOUT	Pad pin (Bond pad)
C	OUTPUT	Data output to the core. The value on PAD will be assigned to C when REN is 1.
I	INPUT	Data input from the core logic. This value is assigned to PAD when OEN is 0.
OEN	INPUT	Active low output driver enable. 1 - Driver is Disabled. 0 - Driver is Enabled.
P[2..1]	INPUT	Driver disabled state control. 0 - Hi-Z 1 - Pull-up 2 - Pull-down 3 - Repeater
E[2..1]	INPUT	Drive Strength Selector 0: 2 mA 1: 4 mA 2: 8 mA 3: 12 mA
SR	INPUT	Slew Rate Control 0 - Slow (half frequency) 1 - Fast
REN	INPUT	Active High Receiver Enable 0 - Receiver disabled. 1 - Receiver enabled
SMT	INPUT	Active High Schmitt Trigger (Hysteresis) Select 0 - No hysteresis
POS	INPUT	Power-on-Start Enable 1 - Enables Active pull-down for invalid power. 0 - Disables Active pull-down capability. When one of the power supplies is invalid and active-high POS is set to 1, PAD is pulled to weak 0. When POS is set to 0, PAD remains in a high-Z state
POC	INPUT	Power-on Control
HVPS	INPUT	High Voltage Power Supply Signal
BIAS	INPUT	Bias Signal

Port Name	Direction	Description
VDD	INPUT	Core VDD
VSS	INPUT	Core VSS
DVDD	INPUT	I/O VDD
DVSS	INPUT	I/O VSS

PAD Port Description

Table 12.2. Output Enable (OEN) and Driver Disabled State Control (P2, P1) Truth Table

Inputs				Output
OEN	P2(MSB)	P1(LSB)	I	IO
0	-	-	0	0
			1	1
1	0	0	-	Z(Normal operation)
	0	1	-	Weak 1 (Pull-up)
	1	0	-	Weak 0 (Pull-down)
	1	1	-	Repeater (Bus keeper)

Truth Table for OEN, P1, P2

Table 12.3. Receiver Enable (REN) Truth Table

Inputs		Output
REN	PAD	C
1	0	0
1	1	1
0	-	0

Truth Table for REN

12.3.2 Programming Sequence

The SoC GPIOs (GPIO_0 to GPIO_57) (missing some in between) and ULP GPIOs (ULP_GPIO_0 to ULP_GPIO_11) are shared between NWP, MCU HP, and MCU ULP. The GPIOs configuration and functionality can be independently controlled by them. The UULP Vbat GPIOs (UULP_VBAT_GPIO_0 to UULP_VBAT_GPIO_5) are controlled by MCU ULP.

PAD Configuration

The PAD configuration for each GPIO can be done through NWP, MCU HP, or MCU ULP.

- The SoC GPIOs are shared and are configured by either MCU HP or NWP
- ULP GPIOs are configured by MCU ULP
- The PADs are configured through 12.5.5 PAD_CONFIG_REG_n (n = 0:57), 12.5.6 ULP_PAD_CONFIG_REG0, 12.5.7 ULP_PAD_CONFIG_REG1, and 12.5.8 ULP_PAD_CONFIG_REG2 Registers
- The UULP Vbat GPIOs are configured through 12.5.9 UULP_VBAT_GPIOn_CONFIG_REG Register (n=0:4)

  At power up, all shared GPIOs are controlled by NWP. The following control bits needs to be programmed corresponding to the particular PAD such that it can be configured by MCU HP.
- NWP_MCUHP_GPIO_CTRL1[21:0] control bits are configured through 12.5.1 MCUHP_PAD_SELECTION Register. The contents of this register are retained during sleep.
- NWP_MCUHP_GPIO_CTRL2 control bits accessible by 12.5.3 MEM_GPIO_ACCESS_CTRL_SET and 12.5.4 MEM_GPIO_ACCESS_CTRL_CLEAR Register. The contents of this register are retained during sleep.

  The table below indicates the PAD Selection control for each GPIO.

GPIO Index	Selection Control Signal
GPIO_0	NWP_MCUHP_GPIO_CTRL1[0]
GPIO_1	NWP_MCUHP_GPIO_CTRL1[0]
GPIO_2	NWP_MCUHP_GPIO_CTRL1[0]
GPIO_3	NWP_MCUHP_GPIO_CTRL1[0]
GPIO_4	NWP_MCUHP_GPIO_CTRL1[0]
GPIO_5	NWP_MCUHP_GPIO_CTRL1[0]
GPIO_6	NWP_MCUHP_GPIO_CTRL1[1]
GPIO_7	NWP_MCUHP_GPIO_CTRL1[2]
GPIO_8	NWP_MCUHP_GPIO_CTRL1[3]
GPIO_9	NWP_MCUHP_GPIO_CTRL1[4]
GPIO_10	NWP_MCUHP_GPIO_CTRL1[5]
GPIO_11	NWP_MCUHP_GPIO_CTRL1[6]
GPIO_12	NWP_MCUHP_GPIO_CTRL1[7]
GPIO_15	NWP_MCUHP_GPIO_CTRL1[8]
GPIO_25	NWP_MCUHP_GPIO_CTRL2
GPIO_26	NWP_MCUHP_GPIO_CTRL2
GPIO_27	NWP_MCUHP_GPIO_CTRL2
GPIO_28	NWP_MCUHP_GPIO_CTRL2
GPIO_29	NWP_MCUHP_GPIO_CTRL2
GPIO_30	NWP_MCUHP_GPIO_CTRL2
GPIO_31	NWP_MCUHP_GPIO_CTRL1[9]
GPIO_32	NWP_MCUHP_GPIO_CTRL1[9]
GPIO_33	NWP_MCUHP_GPIO_CTRL1[9]

GPIO Index	Selection Control Signal
GPIO_34	NWP_MCUHP_GPIO_CTRL1[9]
GPIO_46	NWP_MCUHP_GPIO_CTRL1[10]
GPIO_47	NWP_MCUHP_GPIO_CTRL1[11]
GPIO_48	NWP_MCUHP_GPIO_CTRL1[12]
GPIO_49	NWP_MCUHP_GPIO_CTRL1[13]
GPIO_50	NWP_MCUHP_GPIO_CTRL1[14]
GPIO_51	NWP_MCUHP_GPIO_CTRL1[15]
GPIO_52	NWP_MCUHP_GPIO_CTRL1[16]
GPIO_53	NWP_MCUHP_GPIO_CTRL1[17]
GPIO_54	NWP_MCUHP_GPIO_CTRL1[18]
GPIO_55	NWP_MCUHP_GPIO_CTRL1[19]
GPIO_56	NWP_MCUHP_GPIO_CTRL1[20]
GPIO_57	NWP_MCUHP_GPIO_CTRL1[21]
ULP_GPIO_0	Controlled by MCU ULP. *NWP_MCUHP_GPIO_CTRL1[22]
ULP_GPIO_1	Controlled by MCU ULP. *NWP_MCUHP_GPIO_CTRL1[23]
ULP_GPIO_2	Controlled by MCU ULP. *NWP_MCUHP_GPIO_CTRL1[24]
ULP_GPIO_3	Controlled by MCU ULP. *NWP_MCUHP_GPIO_CTRL1[25]
ULP_GPIO_4	Controlled by MCU ULP. *NWP_MCUHP_GPIO_CTRL1[26]
ULP_GPIO_5	Controlled by MCU ULP. *NWP_MCUHP_GPIO_CTRL1[27]
ULP_GPIO_6	Controlled by MCU ULP. *NWP_MCUHP_GPIO_CTRL1[28]
ULP_GPIO_7	Controlled by MCU ULP. *NWP_MCUHP_GPIO_CTRL1[29]
ULP_GPIO_8	Controlled by MCU ULP. *NWP_MCUHP_GPIO_CTRL1[30]
ULP_GPIO_9	Controlled by MCU ULP. *NWP_MCUHP_GPIO_CTRL1[31]
ULP_GPIO_10	Controlled by MCU ULP. *NWP_MCUHP_GPIO_CTRL1[32]
ULP_GPIO_11	Controlled by MCU ULP. *NWP_MCUHP_GPIO_CTRL1[33]
UULP_VBAT_GPIO_0	Controlled by MCU ULP

GPIO Index	Selection Control Signal
UULP_VBAT_GPIO_1	Controlled by MCU ULP
UULP_VBAT_GPIO_2	Controlled by MCU ULP
UULP_VBAT_GPIO_3	Controlled by MCU ULP
UULP_VBAT_GPIO_4	Controlled by MCU ULP

Note: ULP_GPIO_0 to ULP_GPIO_11 pads can be used for ULP peripherals as well as MCU peripheral functions (GPIO_64 to GPIO_75 is mapped to ULP_GPIO pads. Refer to Section for peripheral muxing for these GPIOs). When used as MCU peripheral functions, NWP_MCUHP_GPIO_CTRL1[33:22] is used to select between NWP and MCU.

PAD Configuration Control Signals

GPIO Register Programming

The 41 (SOC + ULP) general-purpose I/O (GPIO) pins are used in generating and capturing application-specific input and output signals. Each pin can be programmed as an output or as an input port for various functions. GPIO pins may have alternate input and output functions. A pin may be controlled by software or as an alternate function pin, but not as both at the same time.

Functionlity for all the GPIOs are shared between NWP, MCU HP, and MCU ULP. Each GPIO can be programmed through respective GPIO registers after configuring the GPIO mode as per the functional usage.

- GPIOs 0:57 are controlled from the MCU HP GPIO Registers
- ULP GPIOs 0:11 are controlled from the MCU ULP GPIO Registers

Each GPIO pin has a register that controls the behavior of the pin. Information about the pin, like the mode, direction of the pin, and type of signal detection required has to be programmed to this register. The GPIO mode for all GPIO pins are configured as per the Reset values table described below.

MCU HP GPIO Registers

The GPIO programming for GPIO_n (n=0:57) which are controlled by MCU HP are programmed as described in Section 16.6 Enhanced GPIO (EGPIO) of the MCU APB Peripherals section.

MCU ULP GPIO Registers

The GPIO mode for ULP_GPIO_n (n=0:11) which are controlled by MCU ULP are programmed as described in Section of the MCU ULP Peripherals section.

MCU UULP Vbat GPIO Registers

The configuration of UULP_VBAT_GPIO_n (n=0:4) which are controlled by MCU ULP can be done through the 12.5.9 UULP_VBAT_GPIOn_CONFIG_REG.

12.3.3 PAD Configuration and GPIO Mode Reset Values

The table below indicates the Reset values for the PAD configurations and GPIO modes of each GPIO.

Table 12.4. PAD Configuration and GPIO Mode Reset Values

GPPAD	P2	P1	SR	REN	SMT	POS	E2	E1	GPIO_MODE
GPIO_0	0	0	1	0	0	0	0	1	15
GPIO_1	0	0	1	0	0	0	0	1	15
GPIO_2	0	0	1	0	0	0	0	1	15
GPIO_3	0	0	1	0	0	0	0	1	15
GPIO_4	0	0	1	0	0	0	0	1	15
GPIO_5	0	0	1	0	0	0	0	1	15
GPIO_6	0	0	1	0	0	0	0	1	15
GPIO_7	0	0	1	0	0	0	0	1	15
GPIO_8	0	0	1	0	0	0	0	1	15
GPIO_9	0	0	1	0	0	0	0	1	15
GPIO_10	0	0	1	0	0	0	0	1	15
GPIO_11	0	0	1	0	0	0	0	1	15
GPIO_12	0	0	1	0	0	0	0	1	15
GPIO_15	0	0	1	0	0	0	0	1	15
GPIO_25	0	0	1	1	0	0	0	1	0
GPIO_26	0	0	1	1	0	0	1	0	0
GPIO_27	0	0	1	1	0	0	1	0	0
GPIO_28	0	0	1	1	0	0	1	0	0
GPIO_29	0	0	1	1	0	0	1	0	0
GPIO_30	0	1	1	1	0	0	1	0	0
GPIO_31	0	0	1	1	0	0	0	1	15
GPIO_32	0	0	1	1	0	0	0	1	15
GPIO_33	0	0	1	1	0	0	0	1	15
GPIO_34	0	1	1	1	0	0	0	1	15
GPIO_46	0	0	1	0	0	0	0	1	15
GPIO_47	0	0	1	0	0	0	0	1	15
GPIO_48	0	0	1	0	0	0	0	1	15
GPIO_49	0	0	1	0	0	0	0	1	15
GPIO_50	0	0	1	0	0	0	0	1	15
GPIO_51	0	0	1	0	0	0	0	1	15
GPIO_52	0	0	1	0	0	0	0	1	15
GPIO_53	0	0	1	0	0	0	0	1	15
GPIO_54	0	0	1	0	0	0	0	1	15

GPPAD	P2	P1	SR	REN	SMT	POS	E2	E1	GPIO_MODE
GPIO_55	0	0	1	0	0	0	0	1	15
GPIO_56	0	0	1	0	0	0	0	1	15
GPIO_57	0	0	1	0	0	0	0	1	15
ULP_GPIO_0	0	0	0	0	0	0	0	1	0
ULP_GPIO_1	0	0	0	0	0	0	0	1	0
ULP_GPIO_2	0	0	0	0	0	0	0	1	0
ULP_GPIO_3	0	0	0	0	0	0	0	1	0
ULP_GPIO_4	0	0	0	0	0	0	0	1	0
ULP_GPIO_5	0	0	0	0	0	0	0	1	0
ULP_GPIO_6	0	0	0	0	0	0	0	1	0
ULP_GPIO_7	0	0	0	0	0	0	0	1	0
ULP_GPIO_8	0	0	0	0	0	0	0	1	0
ULP_GPIO_9	0	0	0	0	0	0	0	1	0
ULP_GPIO_10	0	0	0	0	0	0	0	1	0
ULP_GPIO_11	0	0	0	0	0	0	0	1	0
UULP_VBAT_GPIO_0	—	—	—	0	—	—	—	—	1
UULP_VBAT_GPIO_1	—	—	—	0	—	—	—	—	1
UULP_VBAT_GPIO_2	—	—	—	0	—	—	—	—	1
UULP_VBAT_GPIO_3	—	—	—	0	—	—	—	—	0
UULP_VBAT_GPIO_4	—	—	—	0	—	—	—	—	0

12.4 Register Summary

12.4.1 PAD Selection Registers

Base Address: 0x4130_0000

Table 12.5. PAD Control Registers Summary

Register Name	Offset	Description
12.5.3 MEM_GPIO_ACCESS_CTRL_SET	0x000	Indicates the PAD Configuration Control for GPIO_25-GPIO_30.
12.5.4 MEM_GPIO_ACCESS_CTRL_CLEAR	0x004	Indicates the PAD Configuration Control for GPIO_25-GPIO_30.
12.5.1 MCUHP_PAD_SELECTION	0x610	Indicates the PAD Configuration Control for GPIO_0 to GPIO_57 pads except for GPIO_25-GPIO_30.
12.5.2 MCUHP_PAD_SELECTION_1	0x618	Indicates the PAD Configuration Control for ULP_GPIO_0 to ULP_GPIO_11 pads when used as SoC GPIO function for SoC_GPIO_64 to SoC_GPIO_75.

12.4.2 MCU HP GPIO PAD Configuration Registers

Base Address: 0x4600_4000

Table 12.6. MCUHP PAD Configuration Register Summary

Register Name	Offset	Description
12.5.5 PAD_CONFIG_REG_n	0x0 + 4*n	PAD Configuration Register for GPIO_n; n = 0,1,2, ... 63

12.4.3 MCU ULP GPIO PAD Configuration Registers

Base Address: 0x2404_A000

Table 12.7. MCUULP PAD Configuration Registers Summary

Register Name	Offset	Description
12.5.6 ULP_PAD_CONFIG_REG0	0x00	PAD Configuration Registers for ULP GPIOs (ULP_GPIO_0 to ULP_GPIO_11)
12.5.7 ULP_PAD_CONFIG_REG1	0x04	
12.5.8 ULP_PAD_CONFIG_REG2	0x08	

12.4.4 MCU UULP Vbat GPIO PAD Configuration Registers

Base Address: 0x2404_861C

Table 12.8. MCU UULP Vbat GPIO Configuration Registers

Register Name	Offset	Description
12.5.9 UULP_VBAT_GPIOn_CONFIG_REG	0x0 + 4*n	PAD Configuration Registers for UULP Vbat GPIOs UULP_VBAT_GPIOn (n=0:4)

12.5 Register Description

12.5.1 MCUHP_PAD_SELECTION

Table 12.9. MCUHP_PAD_SELECTION Description

Bit	Access	Function	Reset Value	Description
31:21	—	Reserved	—	It is recommended to write these bits to 0.
21:0	RW	NWP_MCUHP_GPIO_CTRL1[21:0]	0	PAD Configuration Controls between NWP and MCU HP. Writing 1 to a particular bit enables the MCU HP to configure the corresponding PADs. Writing 0 to a particular bit enables the NWP to configure the corresponding PADs. Details of the PADs corresponding to each bit are described in the GPIO Controls table above.

12.5.2 MCUHP_PAD_SELECTION_1

Table 12.10. MCUHP_PAD_SELECTION_1_Description

Bit	Access	Function	Reset Value	Description
31:12	—	Reserved	—	It is recommended to write these bits to 0.
11:0	RW	NWP_MCUHP_GPIO_CTRL1[33:22]	0	PAD Configuration Controls between NWP and MCU HP. Writing 1 to a particular bit enables the MCU HP to configure the corresponding PADs. Writing 0 to a particular bit enables the NWP to configure the corresponding PADs. Details of the PADs corresponding to each bit are described in the GPIO Controls table above.

12.5.3 MEM_GPIO_ACCESS_CTRL_SET

Table 12.11. MEM_GPIO_ACCESS_CTRL_SET Description

Bit	Access	Function	Reset Value	Description
31:6	—	Reserved	—	It is recommended to write these bits to 0.
5	RW	NWP_MCUHP_GPIO_CTRL2	1	Writing 1 to this enables NWP to configure the GPIO_25 to GPIO_30 Writing 0 to this has no effect.
4:0	—	Reserved	—	It is recommended to write these bits to 0.

12.5.4 MEM_GPIO_ACCESS_CTRL_CLEAR

Table 12.12. MEM_GPIO_ACCESS_CTRL_CLEAR Description

Bit	Access	Function	Reset Value	Description
31:6	—	Reserved	—	It is recommended to write these bits to 0.
5	RW	NWP_MCUHP_GPIO_CTRL2	1	Writing 1 to this enables MCU HP to configure the GPIO_25 to GPIO_30. Writing 0 to this has no effect.
4:0	—	Reserved	—	It is recommended to write these bits to 0.

12.5.5 PAD_CONFIG_REG_n

The Reset values for these registers are already provided in the 12.3.3 PAD Configuration and GPIO Mode Reset Values table above.

Table 12.13. PAD_CONFIG_REG_n Register Description

Bit	Access	Function	Description
31:8	—	Reserved	It is recommended to write these bits to 0.
7	RW	PADCONFIG_P2	P[2,1] – Driver disabled state control, 0-Hi-Z / 1-Pull-up / 2-Pull-down / 3-Repeater
6	RW	PADCONFIG_P1	P[2,1] – Driver disabled state control, 0-Hi-Z / 1-Pull-up / 2-Pull-down / 3-Repeater
5	RW	PADCONFIG_SR	Slew Rate Control; SR = 0 – Slow (half frequency); SR = 1 – Fast
4	RW	PADCONFIG_REN	Active high receiver enable; REN = 0 – Receiver disabled, C driven to 0 - REN = 1 – Receiver enabled
3	RW	PADCONFIG_SMT	Active high Schmitt trigger (Hysteresis) select; SMT=0 – No hysteresis; Default value for reset is 1'b1 and others is 1'b0
2	RW	PADCONFIG_POS	Power-on-Start enable POS = 1 – Enables active pull-down for invalid power; POS = 0 – Active pull-down capability disabled. When one of the power supplies is invalid and active-high POS is set to 1, PAD is pulled to weak 0. When POS is set to 0, PAD remains in a high-Z state. : Default 0
1	RW	PADCONFIG_E2	E[2,1] – Drive strength selector, 0-2 mA / 1-4 mA / 2-8 mA / 3-12 ma
0	RW	PADCONFIG_E1	E[2,1] – Drive strength selector, 0-2 mA / 1-4 mA / 2-8 mA / 3-12 ma

12.5.6 ULP_PAD_CONFIG_REG0

Table 12.14. MCUULP_PAD_CONFIG_REG0 Register Description

BIT	Access	Function	Reset Value	Description
31:16	—	Reserved	—	It is recommended to write these bits to 0.
15	RW	PADCONFIG_P2_2	0	P[2,1] – Driver disabled state control, 0-Hi-Z / 1-Pull-up / 2-Pull-down / 3-Repeater for ULP_GPIO_4 - ULP_GPIO_7
14	RW	PADCONFIG_P1_2	0	P[2,1] – Driver disabled state control, 0-Hi-Z / 1-Pull-up / 2-Pull-down / 3-Repeater for ULP_GPIO_4 - ULP_GPIO_7
13	RW	PADCONFIG_SR_2	0	Slew Rate Control; SR = 0 – Slow (half frequency); SR = 1 – Fast for ULP_GPIO_4 - ULP_GPIO_7
12	—	Reserved	—	It is recommended to write these bits to 0.
11	RW	PADCONFIG_SMT_2	0	Active high Schmitt trigger (Hysteresis) select; SMT=0 – No hysteresis; Default value for reset is 1'b1 and others is 1'b0 for ULP_GPIO_4 - ULP_GPIO_7
10	RW	PADCONFIG_POS_2	0	Power-on-Start enable: POS = 1 – Enables active pull-down for invalid power; POS = 0 – Active pull-down capability disabled. When one of the power supplies is invalid and active-high POS is set to 1, PAD is pulled to weak 0. When POS is set to 0, PAD remains in a high-Z state. : Default 0 for ULP_GPIO_4 - ULP_GPIO_7
9	RW	PADCONFIG_E2_2	0	E[2,1] – Drive strength selector, 0-2 mA / 1-4 mA / 2-8 mA / 3-12 mA for ULP_GPIO_4 - ULP_GPIO_7
8	RW	PADCONFIG_E1_2	1	E[2,1] – Drive strength selector, 0-2 mA / 1-4 mA / 2-8 mA / 3-12 mA for ULP_GPIO_4 - ULP_GPIO_7
7	RW	PADCONFIG_P2_1	0	P[2,1] – Driver disabled state control, 0-Hi-Z / 1-Pull-up / 2-Pull-down / 3-Repeater for ULP_GPIO_0 - ULP_GPIO_3
6	RW	PADCONFIG_P1_1	0	P[2,1] – Driver disabled state control, 0-Hi-Z / 1-Pull-up / 2-Pull-down / 3-Repeater for ULP_GPIO_0 - ULP_GPIO_3
5	RW	PADCONFIG_SR_1	0	Slew Rate Control; SR = 0 – Slow (half frequency); SR = 1 – Fast for ULP_GPIO_0 - ULP_GPIO_3
4	—	Reserved	—	It is recommended to write these bits to 0.

BIT	Access	Function	Reset Value	Description
3	RW	PADCONFIG_SMT_1	0	Active high Schmitt trigger (Hysteresis) select; SMT=0 – No hysteresis; Default value for reset is 1'b1 and others is 1'b0 for ULP_GPIO_0 - ULP_GPIO_3
2	RW	PADCONFIG_POS_1	0	Power-on-Start enable: POS = 1 – Enables active pull-down for invalid power; POS = 0 – Active pull-down capability disabled . When one of the power supplies is invalid and active-high POS is set to 1, PAD is pulled to weak 0. When POS is set to 0, PAD remains in a high-Z state. : Default 0 for ULP_GPIO_0 - ULP_GPIO_3
1	RW	PADCONFIG_E2_1	0	E[2,1] – Drive strength selector, 0-2 mA / 1-4 mA / 2-8 mA / 3-12 mA for ULP_GPIO_0 - ULP_GPIO_3
0	RW	PADCONFIG_E1_1	1	E[2,1] – Drive strength selector, 0-2 mA / 1-4 mA / 2-8 mA / 3-12 mA for ULP_GPIO_0 - ULP_GPIO_3

12.5.7 ULP_PAD_CONFIG_REG1

Table 12.15. MCUULP_PAD_CONFIG_REG1 Register Description

BIT	Access	Function	Reset Value	Description
31:16	—	Reserved	—	It is recommended to write these bits to 0.
15:8	—	Reserved	—	Reserved
7	RW	PADCONFIG_P2_1	0	P[2,1] – Driver disabled state control, 0-Hi-Z / 1-Pull-up / 2-Pull-down / 3-Repeater for ULP_GPIO_8 - ULP_GPIO_11
6	RW	PADCONFIG_P1_1	0	P[2,1] – Driver disabled state control, 0-Hi-Z / 1-Pull-up / 2-Pull-down / 3-Repeater for ULP_GPIO_8 - ULP_GPIO_11
5	RW	PADCONFIG_SR_1	0	Slew Rate Control ; SR = 0 – Slow (half frequency); SR = 1 – Fast for ULP_GPIO_8 - ULP_GPIO_11
4	—	Reserved	—	It is recommended to write these bits to 0.
3	RW	PADCONFIG_SMT_1	0	Active high Schmitt trigger (Hysteresis) select; SMT=0 – No hysteresis; Default value for reset is 1'b1 and others is 1'b0 for ULP_GPIO_8 - ULP_GPIO_11
2	RW	PADCONFIG_POS_1	0	Power-on-Start enable: POS = 1 – Enables active pull-down for invalid power; POS = 0 – Active pull-down capability disabled . When one of the power supplies is invalid and active-high POS is set to 1, PAD is pulled to weak 0. When POS is set to 0, PAD remains in a high-Z state. : Default 0 for ULP_GPIO_8 - ULP_GPIO_11
1	RW	PADCONFIG_E2_1	0	E[2,1] – Drive strength selector, 0-2 mA / 1-4 mA / 2-8 mA / 3-12 mA for ULP_GPIO_8 - ULP_GPIO_11
0	RW	PADCONFIG_E1_1	1	E[2,1] – Drive strength selector, 0-2 mA / 1-4 mA / 2-8 mA / 3-12 mA for ULP_GPIO_8 - ULP_GPIO_11

12.5.8 ULP_PAD_CONFIG_REG2

Table 12.16. MCUULP_PAD_CONFIG_REG2 Register Description

BIT	Access	Function	Reset Value	Description
31:12	—	Reserved	0	It is recommended to write these bits to 0.
11:0	RW	PADCONFIG_REN	0	Active high receiver enable; REN = 0 – Receiver disabled, C driven to 0 - REN = 1 – Receiver enabled for ULP_GPIO_11 - ULP_GPIO_0

12.5.9 UULP_VBAT_GPIOn_CONFIG_REG

Table 12.17. UULP_VBAT_GPIOn_CONFIG_REG Register Description

BIT	Access	Function	Reset Value	Description
31:9	—	Reserved	—	It is recommended to write these bits to 0.
8	RW	GPIO_POLARITY	0	Indicates the polarity of the UULP_VBAT_GPIO to be considered when used as a Wakeup source from any of the Sleep States as described in 9. Power Architecture 1 - When GPIO input is high 0 - When GPIO input is low
7	—	Reserved	-	It is recommended to write these bits to 0.
6	RW	GPIO_PAD_SELECT	0	UULP_VBAT_GPIO_n Pad selection between M4 and NWP 0 - M4 has control over this GPIO output value 1 - NWP has control over this GPIO output value
5	RW	GPIO_OUTPUT	0	Indicates the value to be driven on the PAD when configured to OUTPUT mode (GPIO Mode=0) for UULP_VBAT_GPIO_n (n=0:4)
4	RW	GPIO_OEN	1	Indicates the direction of the PAD for UULP_VBAT_GPIO_n (n=0:4) if configured to GPIO mode = 0 0 - Output 1 - Input
3	RW	GPIO_REN	0	Enables the Receiver of the PAD for UULP_VBAT_GPIO_n (n=0:4) 0 - Receiver Disabled 1 - Receiver Enabled
2:0	RW	GPIO_MODE	1	Indicates the GPIO Mode for UULP_VBAT_GPIO_n (n=0:4)

13. GPDMA

13.1 General Description

GPDMA is an AHB-based general purpose DMA Controller core that transfers data from a source peripheral or memory to a destination peripheral or memory over one or more AHB buses. GPDMA has two primariy interfaces. It can support 8 DMA channels. It can support a maximum of 64 peripherals.

13.2 Features

- Supports 8 channels (DMA links).
- Has two AHB primaries for parallel data transfer.
- Dynamically size configurable 8 channel SRAM based 64 x 64bit FIFO.
- A maximum of 64 peripherals to be supported.
- Primary selectable per channel and per source and destination.
- Selectable primary for descriptor fetch.
- Supports programmable Source and destination burst sizes (beats): (1,2,3..63). Burst Size is the number of beats transferred on a peripheral request.
- Supports programmable AHB bursts in beats (1,4,8.. 32 beats).
- Source and Destination address alignment. Can support byte-aligned 16 and 32-bit transfers.
- Programmable Transfer Types
  - Memory to Memory
  - Memory to Peripheral(Peripheral or DMA controlled)
  - Peripheral to Memory (Peripheral or DMA controlled)
- Programmable transfer length in bytes per descriptor.
- Linked descriptors
  - Move to next descriptor on completion of transfer and when next descriptor address is not 0.
  - Move to next descriptor after last transfer indicated by Peripheral.
  - Supports address increment/no-increment
- Interrupt Generation Options
  - Generate at the end of the descriptor when indicated by a descriptor bit.
  - When last transfer is indicated by the peripheral for peripheral controlled transfers.
  - At the end of the overall transfer when next descriptor address is 0.
- Supports Programmable priority encoded arbiter.
- DMA squash: DMA in progress will be having clean termination.(Any contents of the fifo will be lost).
- Support for memory Zero Fill and One Fill.
- Supports AHB secondary interface for programming.

13.3 Functional Description

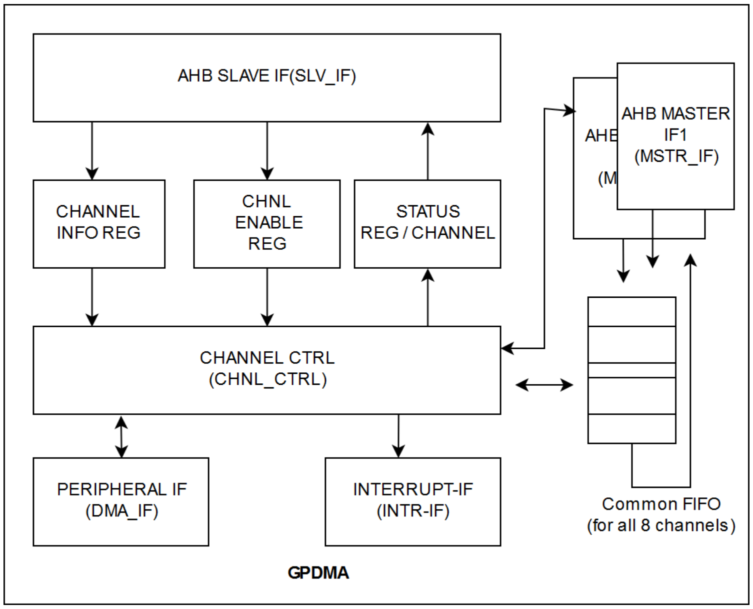


Figure 13.1. GPDMA Block Diagram

GPDMA fetch data from the source and write same data into the destination according to the programed source and destination locations. GPDMA has two AHB primaries. So, fetching data from source location and writing same data into destination location can be done in parallel by using these two AHB primaries. Once DMA transfer is done, status and interrupts are updated.

GPDMA supports the following three types transfers:

- Memory to Memory
- Memory to Peripheral (DMA controlled)
- Peripheral to Memory (Peripheral or DMA controlled)

MCU peripherals are assigned with following PERIPHERAL CODEs in GPDMA.

Table 13.1. Peripheral codes for GPDMA

PERIPHERAL Code	Peripheral
0	SDIO MF (Source)
1	SDIO MF (Destination)
2	SSI Secondary 1 (Source)
3	SSI Secondary 1 (Destination)
4	Reserved
5	Reserved
6	Reserved
7	Reserved
8	UART 0 (Source)
9	UART 0 (Destination)
10	GSPI Primary 1 (Source)
11	GSPI Primary 1 (Destination)
12	Reserved
13	Reserved

PERIPHERAL Code	Peripheral
14	I2C Secondary 1 (Source)
15	I2C Secondary 1 (Destination)
16	QSPI Q1 (Source)
17	QSPI Q2 (Destination)
18	MVP
19	QSPI Q2 (Source)
20	QSPI Q2 (Destination)
21	Reserved
22	Reserved
23	Reserved
24-25	Reserved
26	Reserved
27	Reserved
28	Reserved
29	Reserved
30	SDIO (Source)
31	SDIO (Destination)
32	Reserved
33	Reserved
34	USART 1 (Source)
35	USART 1 (Destination)
36-45	Reserved
46	Reserved
47	CRC (Destination)
48-63	Reserved

13.4 Programming Sequence

Before doing any thing, the programmer must understand the dependability of transfer size on FIFO size. Based on FIFO size, the programmer must select correct source/destination burst size and data width and AHB burst. Source/destination burst or ahb_burst must never exceed FIFO size. If the programmer do not follow this rule, FIFO will over run or under run.

The programming sequence is as follows:

- Program FIFO size
- Update channel SA/DA registers
- Update Link List Ptr Register
- Channel control register.
- Misc_channel_ctrl_register
- Channel and FIFO config register
- Config link list descriptors
- Write DMA channel enable register
- Depending on DMA transfer types, GPDMA perform DMA transfers.
- Once DMA transfer is done, status and interrupts are updated.

It supports two types of DMA transfers. They are as follows:

- Link listed mode
- Non link listed mode

In the case of link listed mode, MCU writes the link list fetch address into LINK_LIST_PTR_REG. After the channel arbitration, hardware fetches the descriptor from the address pointed by LINK_LIST_PTR_REG.ancd saves the contents of the descriptor into the appropriate control registers. Based on the control register settings, it will perform DMA operation.

In case of non-link listed operation, MCU writes the link list fetch address into LINK_LIST_PTR_REG as 0. MCU also writes all the control registers and SA/DA registers. After the channel arbitration, hardware reads control registers. Based on the control register settings, it will perform DMA operation.

13.4.1 Interrupt Configurations of GPDMA

1. Mask Transfer done interrupt and Unmask Linked List descriptor Done Interrupt.
   In this case, INTR_MASK_REG should be written with 0x00FF00FF and get LINK_LIST_FETCH_DONE interrupt.
2. Unmask Transfer done interrupt and Mask Linked List descriptor Done Interrupt.
   In this case, INTR_MASK_REG should be written with 0x0000FFFF and get TFR_DONE interrupt.
3. Mask Transfer done interrupt and Linked List descriptor Done Interrupts
   . In this case, INTR_MASK_REG should be written with 0x00FFFFFF. Any interrupts and their status in INTR_STAT_REG are not observed because of their masking.
4. Unmask Transfer done and Linked List descriptor Done Interrupts.
   In this case, INTR_MASK_REG should be written with 0x000000FF. Here LINK_LIST_FETCH_DONE interrupt is observed after every link descriptor fetch done and TFR_DONE interrupt after completion of last link transfer.
5. Unmask Transfer done and Linked List descriptor Done Interrupts and Mask GPDMA INT ENABLE to NVIC.
   In this case, interrupt is masked from source GPDMA to processor by masking at M4SS_GPDMA_INTR_SEL register so that the status register bits should be polled for interrupt status.

Flow control error

When flow control error happens, rises an interrupt if interrupt is not masked, error bit for the channel is set and DMA will be terminated and channel enable is reset for this channel. FIFOs will be reset and a new arb grant is requested.

13.4.2 Transfer Size Less than Burst Size Error

For a non-link listed dma transfers, it is expected transfer size is always equal to data_width x burst_size . If this condition is not met, this error will terminate DMA transfers and channel enable will be reset for this channel. Also causes an interrupt if interrupt is not masked. FIFOs will be reset and a new arb grant is requested.

13.4.3 FIFO Re-Configuration

Dynamic resize of the FIFO is supported in GPDMA. In order to do this, the Programmer needs to know if all the channels are cleared. Alternatively, the programmer can set DMA_SQUASH bit in channel control register. This ensures clean termination of the FIFO and hardware functionality. A done bit will be provided after completion of this operation.

13.5 Register Summary

Table 13.2. Register Summary Table

Base Address: 0x0x2108_0000

Register Name	Offset	Description
INTERRUPT_REG	0x1084	Interrupt Register
INTERRUPT_MASK_REG	0x1088	Interrupt Mask Register
INTERRUPT_STAT_REG	0x108C	Interrupt Status Register
DMA_CHNL_ENABLE_REG	0x1090	DMA Channel Enable Register
DMA_CHNL_SQUASH_REG	0x1094	DMA Channel Squash Register
DMA_CHNL_LOCK_REG	0x1098	DMA Channel Lock Register
LINK_LIST_PTR_REG_CHNL_0	0x1004	Pointer Register of Channel 0
SRC_ADDR_REG_CHNL_0	0x1008	Source Address Register of Channel 0
DEST_ADDR_REG_CHNL_0	0x100C	Destination Address Register of Channel 0
CHANNEL_CTRL_REG_CHNL_0	0x1010	Channel Control Register for Channel 0
MISC_CHANNEL_CTRL_REG_CHNL_0	0x1014	Miscellaneous Control Register for Channel 0
FIFO_CONFIG_REG_CHNL_0	0x1018	FIFO Configuration Register for Channel 0
PRIORITY_LEVEL_REG_CHNL_0	0x101C	Priority Level for Channel 0
LINK_LIST_PTR_REG_CHNL_1	0x1104	Pointer Register of Channel 1
SRC_ADDR_REG_CHNL_1	0x1108	Source Address Register of Channel 1
DEST_ADDR_REG_CHNL_1	0x110C	Destination Address Register of Channel 1
CHANNEL_CTRL_REG_CHNL_1	0x1110	Channel Control Register for Channel 1
MISC_CHANNEL_CTRL_REG_CHNL_1	0x1114	Miscellaneous Control Register for Channel 1
FIFO_CONFIG_REG_CHNL_1	0x1118	FIFO Configuration Register for Channel 1
PRIORITY_LEVEL_REG_CHNL_1	0x111C	Priority Level for Channel 1
LINK_LIST_PTR_REG_CHNL_2	0x1204	Pointer Register of Channel 2
SRC_ADDR_REG_CHNL_2	0x1208	Source Address Register of Channel 2
DEST_ADDR_REG_CHNL_2	0x120C	Destination Address Register of Channel 2
CHANNEL_CTRL_REG_CHNL_2	0x1210	Channel Control Register for Channel 2
MISC_CHANNEL_CTRL_REG_CHNL_2	0x1214	Miscellaneous Control Register for Channel 2
FIFO_CONFIG_REG_CHNL_2	0x1218	FIFO Configuration Register for Channel 2
PRIORITY_LEVEL_REG_CHNL_2	0x121C	Priority Level for Channel 2
LINK_LIST_PTR_REG_CHNL_3	0x1304	Pointer Register of Channel 3
SRC_ADDR_REG_CHNL_3	0x1308	Source Address Register of Channel 3
DEST_ADDR_REG_CHNL_3	0x130C	Destination Address Register of Channel 3
CHANNEL_CTRL_REG_CHNL_3	0x1310	Channel Control Register for Channel 3
MISC_CHANNEL_CTRL_REG_CHNL_3	0x1314	Miscellaneous Control Register for Channel 3
FIFO_CONFIG_REG_CHNL_3	0x1318	FIFO Configuration Register for Channel 3

Register Name	Offset	Description
PRIORITY_LEVEL_REG_CHNL_3	0x131C	Priority Level for Channel 3
LINK_LIST_PTR_REG_CHNL_4	0x1404	Pointer Register of Channel 4
SRC_ADDR_REG_CHNL_4	0x1408	Source Address Register of Channel 4
DEST_ADDR_REG_CHNL_4	0x140C	Destination Address Register of Channel 4
CHANNEL_CTRL_REG_CHNL_4	0x1410	Channel Control Register for Channel 4
MISC_CHANNEL_CTRL_REG_CHNL_4	0x1414	Miscellaneous Control Register for Channel 4
FIFO_CONFIG_REG_CHNL_4	0x1418	FIFO Configuration Register for Channel 4
PRIORITY_LEVEL_REG_CHNL_4	0x141C	Priority Level for Channel 4
LINK_LIST_PTR_REG_CHNL_5	0x1504	Pointer Register of Channel 5
SRC_ADDR_REG_CHNL_5	0x1508	Source Address Register of Channel 5
DEST_ADDR_REG_CHNL_5	0x150C	Destination Address Register of Channel 5
CHANNEL_CTRL_REG_CHNL_5	0x1510	Channel Control Register for Channel 5
MISC_CHANNEL_CTRL_REG_CHNL_5	0x1514	Miscellaneous Control Register for Channel 5
FIFO_CONFIG_REG_CHNL_5	0x1518	FIFO Configuration Register for Channel 5
PRIORITY_LEVEL_REG_CHNL_5	0x151C	Priority Level for Channel 5
LINK_LIST_PTR_REG_CHNL_6	0x1604	Pointer Register of Channel 6
SRC_ADDR_REG_CHNL_6	0x1608	Source Address Register of Channel 6
DEST_ADDR_REG_CHNL_6	0x160C	Destination Address Register of Channel 6
CHANNEL_CTRL_REG_CHNL_6	0x1610	Channel Control Register for Channel 6
MISC_CHANNEL_CTRL_REG_CHNL_6	0x1614	Miscellaneous Control Register for Channel 6
FIFO_CONFIG_REG_CHNL_6	0x1618	FIFO Configuration Register for Channel 6
PRIORITY_LEVEL_REG_CHNL_6	0x161C	Priority Level for Channel 6
LINK_LIST_PTR_REG_CHNL_7	0x1704	Pointer Register of Channel 7
SRC_ADDR_REG_CHNL_7	0x1708	Source Address Register of Channel 7
DEST_ADDR_REG_CHNL_7	0x170C	Destination Address Register of Channel 7
CHANNEL_CTRL_REG_CHNL_7	0x1710	Channel Control Register for Channel 7
MISC_CHANNEL_CTRL_REG_CHNL_7	0x1714	Miscellaneous Control Register for Channel 7
FIFO_CONFIG_REG_CHNL_7	0x1718	FIFO Configuration Register for Channel 7
PRIORITY_LEVEL_REG_CHNL_7	0x171C	Priority Level for Channel 7

13.6 Register Description

13.6.1 INTERRUPT_REG

Table 13.3. Interrupt Register Description

Bit	Access	Function	Reset Value	Description
31:8	R	Reserved	-	Reserved

Bit	Access	Function	Reset Value	Description
7:0	R/W	GPDMAC_INT_STAT	0x00	This bit indicates the status of transfer done interrupt or linked link descriptor fetch interrupt. Transfer done interrupt is set only when corresponding TFR_DONE_MASK(controlled via INTERRUPT_MASK_REG register) is enabled. Linked list descriptor fetch interrupt is set only when corresponding LINK_LIST_FETCH_MASK(controlled via INTERRUPT_MASK_REG register) is enabled and LINK_INTERRUPT(controlled via CHANNEL_CTRL_REG_CHNL_n register) is enabled. Bit 0 = 1: channel 0- interrupt event ..... Bit 7 = 1: channel 7, interrupt event has happened. To clear this interrupt, register write data from CPU is set to "1" and actual register value is set to '0'(inverted value of write data). That means, if this register is read back, '0' will be read upon writing "1" from CPU. Clearing up this interrupt also resets the values of INTERRUPT_STAT_REG bits for that corresponding channel.

13.6.2 INTERRUPT_MASK_REG

Table 13.4. Interrupt Mask Register Description

Bit	Access	Function	Reset Value	Description
31:24	R/W	RSVD	8'h0	Reserved
23:16	R/W	TFR_DONE_MASK	8'h0	Transfer done interrupt bit mask control. By default, transfer done interrupt is unmasked. 0 stands for unmask and 1 stands for mask. 16-bit – Channel 0 17-bit – Channel 1 18-bit – Channel 2 19-bit – Channel 3 20-bit – Channel 4 21-bit – Channel 5 22-bit – Channel 6 23-bit – Channel 7

Bit	Access	Function	Reset Value	Description
15:8	R/W	LINK_LIST_FETCH_MASK	8'hff	Linked list fetch done interrupt bit mask control. By default, descriptor fetch done interrupt is masked. Each bit is used to mask the interrupt per channel. Writing value “1” in to the bit MASKS that particular interrupt. 8-bit – Channel 0 9-bit – Channel 1 10-bit – Channel 2 11-bit – Channel 3 12-bit – Channel 4 13-bit – Channel 5 14-bit – Channel 6 15-bit – Channel 7
7:0	R/W	Reserved	8'hff	Reserved

13.6.3 INTERRUPT_STAT_REG

Table 13.5. Interrupt Status Register Description

Bit	Access	Function	Reset Value	Description
31	R	GPDMAC_ERR7	1'b0	(1) transfer size/burst size /h size mismatch (2)flow_ctrl_err
30	R	TFR_DONE7	1'b0	This bit indicates the status of DMA transfer done interrupt for channel 7. This bit is set only when corresponding TFR_DONE_MASK(controlled via INTERRUPT_MASK_REG register) is enabled.
29	R	LINK_LIST_FETCH_DONE7	1'b0	This bit indicates the status of linked list descriptor fetch done for channel 7. This bit is set only when corresponding LINK_LIST_FETCH_MASK(controlled via INTERRUPT_MASK_REG register) is enabled and LINK_INTERRUPT(controlled via CHANNEL_CTRL_REG_CHNL_n register)
28	R	HRESP_ ERR7	1'b0	1: channel 7 , dma error
27	R	GPDMAC_ERR6	1'b0	(1) transfer size/burst size /h size mismatch (2) flow_ctrl_err
26	R	TFR_DONE6	1'b0	This bit indicates the status of DMA transfer done interrupt for channel 6. This bit is set only when corresponding TFR_DONE_MASK(controlled via INTERRUPT_MASK_REG register) is enabled.

Bit	Access	Function	Reset Value	Description
25	R	LINK_LIST_FETCH_DONE6	1'b0	This bit indicates the status of linked list descriptor fetch done for channel 6. This bit is set only when corresponding LINK_LIST_FETCH_MASK(controlled via INTERRUPT_MASK_REG register) is enabled and LINK_INTERRUPT(controlled via CHANNEL_CTRL_REG_CHNL_n register)
24	R	HRESP_ ERR6	1'b0	1: channel 6, dma error
23	R	GPDMAC_ERR5	1'b0	(1) transfer size/burst size /h size mismatch (2) flow_ctrl_err
22	R	TFR_DONE5	1'b0	This bit indicates the status of DMA transfer done interrupt for channel 5. This bit is set only when corresponding TFR_DONE_MASK(controlled via INTERRUPT_MASK_REG register) is enabled.
21	R	LINK_LIST_FETCH_DONE5	1'b0	This bit indicates the status of linked list descriptor fetch done for channel 5. This bit is set only when corresponding LINK_LIST_FETCH_MASK(controlled via INTERRUPT_MASK_REG register) is enabled and LINK_INTERRUPT(controlled via CHANNEL_CTRL_REG_CHNL_n register)
20	R	HRESP_ ERR5	1'b0	1: channel 5, dma error
19	R	GPDMAC_ERR4	1'b0	(1) transfer size/burst size /h size mismatch (2) flow ctrl err
18	R	TFR_DONE4	1'b0	This bit indicates the status of DMA transfer done interrupt for channel 4. This bit is set only when corresponding TFR_DONE_MASK(controlled via INTERRUPT_MASK_REG register) is enabled.
17	R	LINK_LIST_FETCH_DONE4	1'b0	This bit indicates the status of linked list descriptor fetch done for channel 4. This bit is set only when corresponding LINK_LIST_FETCH_MASK(controlled via INTERRUPT_MASK_REG register) is enabled and LINK_INTERRUPT(controlled via CHANNEL_CTRL_REG_CHNL_n register)
16	R	HRESP_ ERR4	1'b0	1: channel 4, dma error
15	R	GPDMAC_ERR3	1'b0	(1) transfer size/burst size /h size mismatch (2) flow_ctrl_err
14	R	TFR_DONE3	1'b0	This bit indicates the status of DMA transfer done interrupt for channel 3. This bit is set only when corresponding TFR_DONE_MASK(controlled via INTERRUPT_MASK_REG register) is enabled.
13	R	LINK_LIST_FETCH_DONE3	1'b0	This bit indicates the status of linked list descriptor fetch done for channel 3. This bit is set only when corresponding LINK_LIST_FETCH_MASK(controlled via INTERRUPT_MASK_REG register) is enabled and LINK_INTERRUPT(controlled via CHANNEL_CTRL_REG_CHNL_n register)
12	R	HRESP_ ERR3	1'b0	1: channel 3, dma error

Bit	Access	Function	Reset Value	Description
11	R	GPDMAC_ERR2	1'b0	(1) transfer size/burst size /h size mismatch (2) flow_ctrl_err
10	R	TFR_DONE2	1'b0	This bit indicates the status of DMA transfer done interrupt for channel 2. This bit is set only when corresponding TFR_DONE_MASK(controlled via INTERRUPT_MASK_REG register) is enabled.
9	R	LINK_LIST_FETCH_DONE2	1'b0	This bit indicates the status of linked list descriptor fetch done for channel 2. This bit is set only when corresponding LINK_LIST_FETCH_MASK(controlled via INTERRUPT_MASK_REG register) is enabled and LINK_INTERRUPT(controlled via CHANNEL_CTRL_REG_CHNL_n register)
8	R	HRESP_ ERR2	1'b0	1: channel 2, dma error
7	R	GPDMAC_ERR1	1'b0	(1) transfer size/burst size /h size mismatch (2) flow_ctrl_err
6	R	TFR_DONE1	1'b0	This bit indicates the status of DMA transfer done interrupt for channel 1. This bit is set only when corresponding TFR_DONE_MASK(controlled via INTERRUPT_MASK_REG register) is enabled.
5	R	LINK_LIST_FETCH_DONE1	1'b0	This bit indicates the status of linked list descriptor fetch done for channel 1. This bit is set only when corresponding LINK_LIST_FETCH_MASK(controlled via INTERRUPT_MASK_REG register) is enabled and LINK_INTERRUPT(controlled via CHANNEL_CTRL_REG_CHNL_n register)
4	R	HRESP_ ERR1	1'b0	1: channel 1, dma error
3	R	GPDMAC_ERR0	1'b0	(1) transfer size/burst size /h size mismatch (2) flow_ctrl_err
2	R	TFR_DONE0	1'b0	This bit indicates the status of DMA transfer done interrupt for channel 0. This bit is set only when corresponding TFR_DONE_MASK(controlled via INTERRUPT_MASK_REG register) is enabled.
1	R	LINK_LIST_FETCH_DONE0	1'b0	This bit indicates the status of linked list descriptor fetch done for channel 0. This bit is set only when corresponding LINK_LIST_FETCH_MASK(controlled via INTERRUPT_MASK_REG register) is enabled and LINK_INTERRUPT(controlled via CHANNEL_CTRL_REG_CHNL_n register)
0	R	HRESP_ ERR0	1'b0	1: channel 0, dma error

13.6.4 DMA_CHNL_ENABLE_REG

Table 13.6. DMA Channel Enable Register Description

Bit	Access	Function	Reset Value	Description
31:8	R	Reserved		Reserved
7:0	R/W	CH_ENB	0x00	When a bit is set to one, it indicates, corresponding channel is enabled for dma operation. "CPU Will be masked to write zeros, CPU is allowed write 1 only. Hardware will set these bits to zero when the DMA operation is done. These bits feed in to channel arbitration logic. Bit 0 = 1: channel 0 enabled ..... Bit 7 = 1: channel 7 is enabled.

13.6.5 DMA_CHNL_SQUASH_REG

Table 13.7. DMA Channel Squash Register Description

Bit	Access	Function	Reset Value	Description
31:8	R	Reserved		Reserved
7:0	R/W	CH_DIS	0x00	CPU Will be masked to write zeros, CPU is allowed write 1 only. Hardware will set these bits to zero when the DMA Squash is done. Setting this bit ensures clean termination of fifo, fsm, etc. Bit 0 = 1: Channel 1 fifo will be cleared, FSM will be reset. ..... Bit 7 = 1: Channel 1 fifo will be cleared, FSM will be reset.

13.6.6 DMA_CHNL_LOCK_REG

Table 13.8. DMA Channel Lock Register Description

Bit	Access	Function	Reset Value	Description
31:8	R	Reserved		Reserved
7:0	R/W	CHNL_LOCK	0x00	When set entire DMA block transfer is done, before other DMA request is serviced. When set entire DMA transfer is done, before other DMA request is serviced. This bit is reset to zero up on completion of DMA.

13.6.7 LINK_LIST_PTR_REG_CHNL_n

Table 13.9. Link List Pointer Register Description

Bit	Access	Function	Reset Value	Description
31:0	R/W	LINK_LIST_ PTR_REG_CHNL	0x00	This is the address of the memory location from which next descriptor is obtained.

13.6.8 SRC_ADDR_REG_CHNL_n

Table 13.10. Source Address Register Description

Bit	Access	Function	Reset Value	Description
31:0	R/W	SRC_ADDR	0x00	This is the source address from which the data is fetched.

13.6.9 DEST_ADDR_REG_CHNL_n

Table 13.11. Destination Address Register Description

Bit	Access	Function	Reset Value	Description
31:0	R/W	DEST_ADDR	0x00	This is the destination address from which the data is fetched.

13.6.10 CHANNEL_CTRL_REG_CHNL_n

Table 13.12. Channel Control Register Description

Bit	Access	Function	Reset Value	Description
31	R	Reserved	0x0	Reserved
30	R/W	DEST_FIFO_MODE	0x0	If set to 1; destination address will not be incremented, if destination is peripheral (means fifo mode for destination). For memories address will be incremented despite the value 1.
29	R/W	SRC_FIFO_MODE	0x0	If set to 1; source address will not be incremented, if source is peripheral (means fifo mode for source). For memories address will be incremented despite the value 1.
28	R/W	LINK_INTERRUPT	0x0	This bit is set in link list descriptor. Hard ware will send an interrupt when the DMA transfer is done for the corresponding link list address.
27	R/W	RETRY_ON_ERROR	0x0	When this bit is set, if we recieve HRESPERR, We will retry the DMA for that channel.
26	R/W	DEST_ADDR_CONTIGUOUS	0x0	1: Indicates Address is contiguous from previous.
25	R/W	SRC_ADDR_CONTIGUOUS	0x0	1: Indicates Address is contiguous from previous.
24	R/W	LINK_LIST_MSTR_SEL	0x0	0 : M0 will be used to fetch desc. 1: M1 will be used to fetch desc.
23	R/W	LINK_LIST_ON	0x0	This mode is set, when we do link listed operation.
22	R/W	SRC_ALIGN	0x0	Reserved.Value set to 0 We do not do any singles. We just do burst, save first 3 bytes in to residue buffer in one cycle, In the next cycle send 4 bytes to fifo, save 3 bytes in to residue. This continues on.

Bit	Access	Function	Reset Value	Description
21:20	R/W	SRC_DATA_WIDTH	0x0	Data transfer to destination. 00: 08 bits of data on the bus 01: 16 bits of data on the bus 10: 32 bits of data on the bus 11: reserved
19:18	R/W	DEST_DATA_WIDTH	0x0	Data transfer to destination. 00: 08 bits of data on the bus 01: 16 bits of data on the bus 10: 32 bits of data on the bus 11: reserved
17	R/W	MSTR_IF_SEND_SEL	0x0	This selects the MASTER IF from which data to be sent 0: MSTR-0 for send (to destination) 1: MSTR-1 for send (to destination)
16	R/W	MSTR_IF_FETCH_SEL	0x0	This selects the MASTER IF from which data to be fetched . 0: MSTR-0 for fetch (from src) 1: MSTR-1 for fetch (from src)
15:14	R/W	DMA_FLOW_CTRL	0x0	00: gpdmaC :can be set for any type of transfers 01: source peripheral : typically set for peripheral to memory 10: destination peripheral : typically set for memory to peripheral 11: src_and_dest peripheral : Typically set for peripheral to peripheral.
13:12	R/W	TRNS_TYPE	0x0	DMA transfer type 00: memory to memory 01: memory to peripheral 10: peripheral to memory 11: peripheral to peripheral.
11:0	R/W	DMA_BLK_SIZE	0x000	This is data to be transmitted. User should program non-zero value. Loaded at the beginning of the DMA transfer and decremented at every dma transaction. Zero size length is not supported. If zero is programmed, DMA will get stuck.

13.6.11 MISC_CHANNEL_CTRL_REG_CHNL_n

Table 13.13. Miscellaneous Channel Control Register Description

Bit	Access	Function	Reset Value	Description
31	R/W	MEM_ONE_FILL	0x0	Select for memory filling with either 1's or 0's. 0 – Memory fill with 0's. 1 - Memory fill with 1's.
30	R/W	MEM_FILL_ENABLE	0x0	Enable for memory filling with either 1's or 0's. 0 – Disabled 1 – Enabled the memory filling. When memory fill is enabled, transfer size has to be multiples of hsize (destination data-width). This is a mandate requirement.

Bit	Access	Function	Reset Value	Description
29:27	R/W	DMA_PROT	0x0	Protection level to go with the data. It will be concatenated with 1'b1 as there will be no opcode fetching and directly assign to hprot in AHB interface.
26:21	R/W	SRC_CHNL_ID	0x0	This is the source channel Id, from which the data is fetched. must be set up prior to DMA_CHANNEL_ENABLE.
20:15	R/W	DEST_CHNL_ID	0x0	This is the destination channel Id to which the data is sent. Must be set up prior to DMA_CHANNEL_ENABLE.
14:9	R/W	SRC_DATA_BURST	0x0	Burst writes in beats from source 000000: 64 beats 000001: 1 beat .... 001111: 15 beats 111111: 63 beats
8:3	R/W	DEST_DATA_BURST	0x0	Burst writes in beats to destination. 000000: 64 beats .00001: 1 beat .... 001111: 15 beats 111111: 63 beats
2:0	R/W	AHB_BURST_SIZE	0x0	000 : 1 beat 001 : 4 beat 010 : 8 011 : 16 100 : 20 101: 24 110: 28 111 : 32

13.6.12 FIFO_CONFIG_REG_CHNL_n

Table 13.14. FIFO Configuration Register Description

Bit	Access	Function	Reset Value	Description
31:12	R	Reserved	0x0	Reserved
11:6	R/W	FIFO_SIZE	0x00	Channel size. Configure FIFO size before starting DMA operation.
5:0	R/W	FIFO_STRT_ADDR	0x00	Starting row address of channel

13.6.13 PRIORITY_LEVEL_REG_CHNL_n

Table 13.15. Priority Level Channel Register Description

Bit	Access	Function	Reset Value	Description
31:2	R	Reserved	0x0	Reserved
1:0	R/W	PRIORITY_ CH	0x00	Indicates the level of priority Four levels of priority is supported • 00 – first level priority • 01 – second level priority • 10 – third level priority • 11 – four level priority .

14. SPI Flash/PSRAM Controller

14.1 General Description

The SPI Controller is a 1/2/4-wired interface for serial access of data from Flash or PSRAM. Dedicated SPI controllers are present for Flash and PSRAM. It can be used in either Single, Dual, or Quad modes with support for Single Data Rate(SDR) to read the Processor's instructions and for data transfers to/from the Flash or PSRAM. The Controller supports inline decryption of encrypted instructions read from the Flash before they are passed on to the Processor's Instruction Cache. Instructions are read using the Direct Access mode while data transfers use the Indirect Access mode in case of Flash. Both data read and write along with decryption and encryption are possible during Direct and Indirect Access modes for PSRAM.

14.2 Features

- Supports Single/Dual/Quad (S/D/Q) modes for reading processor instructions and data transfers to/from Flash and PSRAM.
- Support for SPI Mode-0.
- Supports full duplex mode in single-bit SPI mode. Support for HOST SPI slave interface.
- Support for SDR mode Flashes/PSRAMs.
- Supports both 8 and 16-bit Flash commands.
- Support 16, 24 and 32-bit addressing modes
- Supports inline decryption (AES) in XTS/CTR mode with 128-bit and 256-bit key sizes while reading encrypted instructions from the Flash.
- Supports only AES CTR mode encryption and decryption of PSRAM data with 128-bit and 256-bit key sizes.
- Supports up to two Flashes/PSRAMs connected to CSN0 and CSN1.
- Supports Direct mode write for PSRAM only.
- Supports semi direct mode read operation for flash and PSRAM.
- Direct Access Mode:
  - Instructions are read from Flash and Data transfer from/to PSRAM using the Direct Access mode which does not need any processor involvement after the initial configuration of the Controller. The read command used for this mode is programmable depending on the Flash/PSRAM used.
  - Direct Access mode supports Wrap / Incremental / Single read operations.
  - Supports prefetch option - enabling this option makes the SPI Controller prefetch the next instruction before the request is posted on the internal AHB bus. If the address for the next instruction is different from the prefetch address, the instruction is scrapped.
  - Supports continuous fetch option to reduce instruction fetch delay from Flash/PSRAM - this option makes the SPI controller to post the Command and Address only once on the bus to read contiguous instructions by controlling only the CSN.
  - Supports programmable CSN high time.
- Indirect Access Mode:
  - Configuration of Flash/PSRAM and reading/writing data from/to the Flash/PSRAM uses the Indirect Access mode which requires the processor to program the SPI controller for each access.
  - Supports reading of up to 32KB bytes of data from Flash/PSRAM in a single read operation.
  - In addition to 24 and 32-bit addressing, the SPI Controller supports 9, 10 and 16-bit addressing in this mode.
- Common flash mode - Flash can be accessed by both MCU and NWP simultaneously.
- Clock Configuration
  - Support for selection of source clock between AHB bus clock and PLL clock.
  - Support for even division factors up to 64 to generate the SPI clock from the source clock.
- Transmission of Extra-byte after the address phase is supported. The contents of this byte are programmable. There is also an option to only transmit the first nibble of the extra byte and maintain a Hi-z on the bus for the next nibble.
- Each phase of a Read operation (Command, Address, Dummy Byte, Extra Byte, Read Data) can be in any of the S/D/Q/O modes depending on the Flash requirements.
- The number of dummy bytes is programmable and can be programmed as per the instruction and the mode of operation.
- Supports DMA flow control and programmable FIFO thresholds
- Supports dual Flash mode - reading of data from two flashes simultaneously and reading and writing from/to two PSRAM simultaneously.
- Supports Flash Write Protect
- Supports interrupt generation based on different events

14.3 Functional Description

The SPI Controller block diagram is shown below:

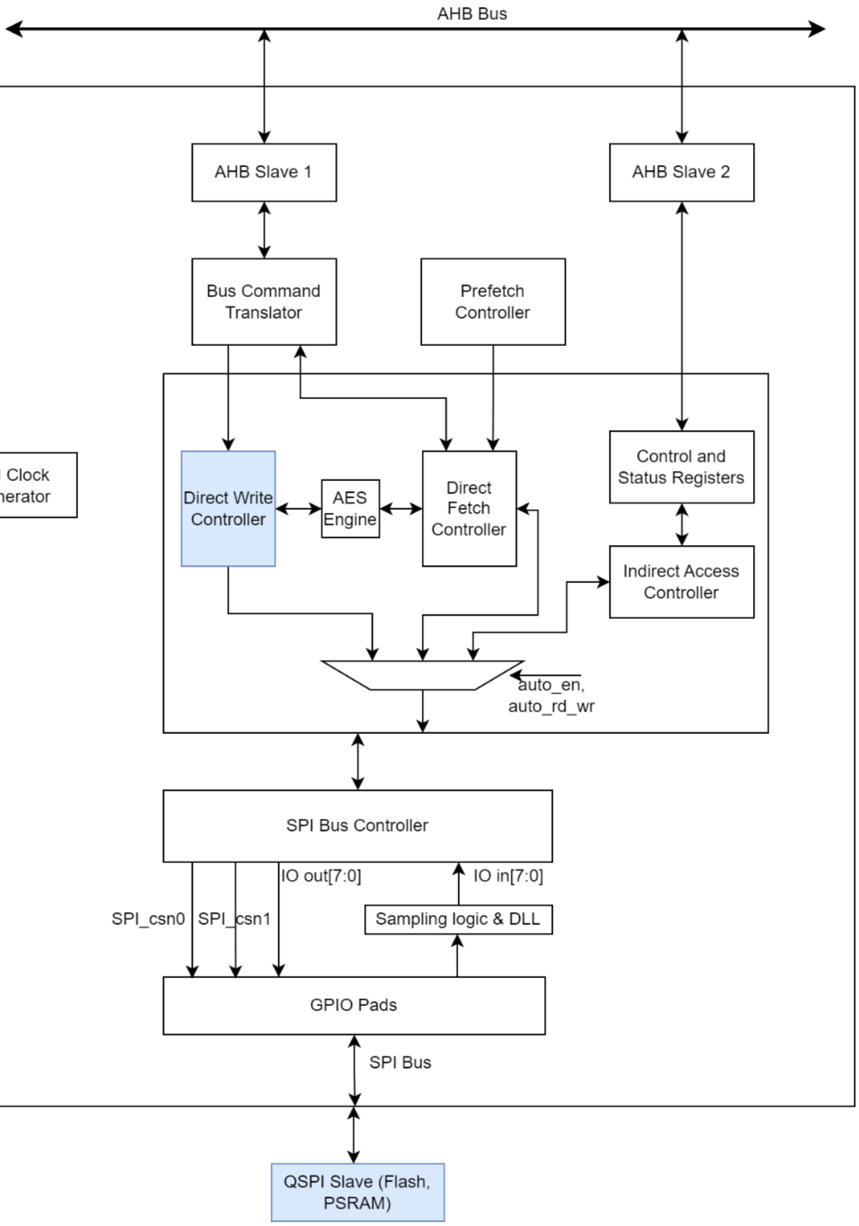


Figure 14.1. SPI Flash Controller Block Diagram

The Direct Access mode is used to read instructions and data directly from Flash and both read from and write to PSRAM without any processor intervention . It supports inline decryption using an AES engine in XTS/CTR mode for the instructions stored in Flash and only AES CTR encryption/decryption for data transfer with PSRAM. The Indirect Access mode is used to read and write data/instructions from Flash/PSRAM. The two modes - Direct Access and Indirect Access - can be used to access the same Flash/PSRAM or two different Flashes/PSRAM (using CSN0 and CSN1) at a time by enabling hardware controlled mode. The SPI Controllers have independent AHB slaves for these modes of access.

14.3.1 Programming sequence

Register Summary

NWP Flash controller base address : 0x1000_0000

MCU flash controller base address : 0x1200_0000

MCU PSRAM controller base address : 0x1204_0000

Offset Address	Register Name	Function
0x00	QSPI Clock Configuration Register	This register is used for the Configuration of QSPI clocks.
0x04	QSPI Bus Mode Resister	This register is used to program the Bus Mode of QSPI.
0x08	QSPI_AUTO_CONFIG_1	These registers are used to Configure the QSPI in Auto mode
0x0C	QSPI_AUTO_CONFIG_2	
0x10	QSPI_MANUAL_CONFIG1	These registers are used to Configure the QSPI in Manual Mode.
0x14	QSPI_MANUAL_CONFIG2	
0x80	QSPI_MANUAL_WRITE_DATA2	This register is used to Configure the QSPI in Manual Mode for Write data2.
0x1C	QSPI_FIFO_THRLD	This register is used to configure the threshold levels.
0x20	QSPI_MANUAL_STATUS Register	This register is used to know the Status of QSPI, FIFO depth
0x24	QSPI_INTR_MASK	This register is used to Mask the interrupts of the QSPI Controller
0x28	QSPI_INTR_UNMASK	This register is used to UnMask the interrupts of the QSPI Controller
0x2C	QSPI_INTR_STS	This register is used to know the status of the Interrupt of the QSPI Controller.
0x30	QSPI_INTR_ACK	This register is used to give ACK for the FIFO's and the interrupt of the QSPI Controller.
0x34	QSPI_STS_MC	This register is used to know the state of the QSPI Controller.
0x38	QSPI_AUTO_CONFIG_1_CSN1	This register is used to configure the QSPI Controller in auto mode for slave on CSN1.
0x3C	QSPI_AUTO_CONFIG_2_CSN1	
0x90	QSPI_AUTO_CONFIG3	This register is used to configure the QSPI Controller in auto configuration mode.
0x94	QSPI_AUTO_CONFIG3_CSN1	
0xA0	QSPI_AUTO_BASE_ADDR_CSN0	This register is used to configure the QSPI Controller auto Base address configuration.
0xA4	QSPI_AUTO_BASE_ADDR_CSN1	
0xB4	QSPI_AUTO_BASE_ADDR_UNMASK_CSN0	This register is used to configure the QSPI in Auto Base address mode for unmasking.
0xB8	QSPI_AUTO_BASE_ADDR_UNMASK_CSN1	
0XC8	QSPI_AES_CONFIG	This register is used configure the QSPI Controller AES Mode, Key Size and Context switching
0XCC	QSPI_AES_KEY_IV_VALID	This register is used configure the QSPI Controller AES write Enables for Keys and IVs
0xD0	QSPI_CMNFLASH_STS	This register is used to know the M4 QSPI Status in Common flash modes. (Present only in M4)

Offset Address	Register Name	Function
0xD4	QSPI_AES_LB_DATA_0_3	This register is data in out register QSPI AES in LB (Standalone mode) for [31:0]
0xD8	QSPI_AES_LB_DATA_4_7	This register is data in out register QSPI AES in LB (Standalone mode) for [63:32]
0xDC	QSPI_AES_LB_DATA_8_B	This register is data in out register QSPI AES in LB (Standalone mode) for [95:64]
0xE0	QSPI_AES_LB_DATA_C_F	This register is data in out register QSPI AES in LB (Standalone mode) for [127:96]
0xE4	QSPI_AES_SEC_SEG_LS_ADDR_1	This register is used to configure the QSPI Controller AES lower boundary address of 1st segment
0xE8	QSPI_AES_SEC_SEG_MS_ADDR_1	This register is used to configure the QSPI Controller AES upper boundary address of 1st segment
0xEC	QSPI_AES_SEC_SEG_LS_ADDR_2	This register specifies the lower boundary address of 2nd segment
0xF0	QSPI_AES_SEC_SEG_MS_ADDR_2	This register specifies the upper boundary address of 2nd segment
0xF4	QSPI_AES_SEC_SEG_LS_ADDR_3	This register specifies the lower boundary address of 3rd segment
0xF8	QSPI_AES_SEC_SEG_MS_ADDR_3	This register specifies the upper boundary address of 3rd segment
0xFC	QSPI_AES_SEC_SEG_LS_ADDR_4	This register specifies the lower boundary address of 4th segment
0x100	QSPI_AES_SEC_SEG_MS_ADDR_4	This register specifies the upper boundary address of 4th segment
0x104	QSPI_SRAM_CTRL_CSN0_REG	This register is used to configure the SRAM CTRL register of the QSPI Controller.
0x108	QSPI_SRAM_CTRL_CSN1_REG	
0x11C	SEMI_AUTO_MODE_ADDR_REG	This register is used to configure the QSPI Controller in SEMI AUTO mode for address
0x120	SEMI_AUTO_MODE_CONFIG_REG	This register is used to configure the QSPI Controller in SEMI AUTO mode for Configuration
0x124	SEMI_AUTO_MODE_CONFIG2_REG	
0x128	QSPI_BUS_MODE2_REG	This register is used to configure the QSPI Controller Bus in mode2.
0x12C	QSPI_AES_SEC_KEY_FRM_KH_REG	This register is used to configure the QSPI Controller AES SEC KEY FRM_KH mode.
0x130	QSPI_AUTO_CONITNUE_FETCH_CTRL_REG	This register is used to configure the QSPI Controller in Auto Continue Fetch mode.
0x134	QSPI_AES_KEY1_0_3	This register is used configure the QSPI Controller AES KEY1 from 0 to 3 bytes
0x138	QSPI_AES_KEY1_4_7	This register is used to configure the QSPI Controller AES KEY1 from 4 to 7 bytes
0x13C	QSPI_AES_KEY1_8_B	This register is used to configure the QSPI Controller AES KEY1 from 8 to B bytes
0x140	QSPI_AES_KEY1_C_F	This register is used to configure the QSPI Controller AES KEY1 from C to F bytes
0x144	QSPI_AES_KEY1_10_13	This register is used to configure the QSPI Controller AES KEY1 from 10 to 13 bytes
0x148	QSPI_AES_KEY1_14_17	This register is used to configure the QSPI Controller AES KEY1 from 14 to 17 bytes

Offset Address	Register Name	Function
0x14C	QSPI_AES_KEY1_18_1B	This register is used to configure the QSPI Controller AES KEY1 from 18 to 1B bytes
0x150	QSPI_AES_KEY1_1C_1F	This register is used to configure the QSPI Controller AES KEY1 from 1C to 1F bytes
0x154	QSPI_AES_KEY2_0_3	This register is used to configure the QSPI Controller AES KEY2 from 0 to 3 bytes
0x158	QSPI_AES_KEY2_4_7	This register is used to configure the QSPI Controller AES KEY2 from 4 to 7 bytes
0x15C	QSPI_AES_KEY2_8_B	This register is used to configure the QSPI Controller AES KEY2 from 8 to B bytes
0x160	QSPI_AES_KEY2_C_F	This register is used to configure the QSPI Controller AES KEY2 from C to F bytes
0x164	QSPI_AES_KEY2_10_13	This register is used to configure the QSPI Controller AES KEY2 from 10 to 13 bytes
0x168	QSPI_AES_KEY2_14_17	This register is used to configure the QSPI Controller AES KEY2 from 14 to 17 bytes
0x16C	QSPI_AES_KEY2_18_1B	This register is used to configure the QSPI Controller AES KEY2 from 18 to 1B bytes
0x170	QSPI_AES_KEY2_1C_1F	This register is used to configure the QSPI Controller AES KEY2 from 1C to 1F bytes
0x174	QSPI_AES_IV1_0_3	This register is used to configure the QSPI Controller AES IV1 from 0 to 3 bytes in LB mode
0x178	QSPI_AES_IV1_4_7	This register is used to configure the QSPI Controller AES IV1 from 4 to 7 bytes in LB mode
0x17C	QSPI_AES_IV1_8_B	This register is used to configure the QSPI Controller AES IV1 from 8 to B bytes in LB mode
0x180	QSPI_AES_IV1_C_F	This register is used to configure the QSPI Controller AES IV1 from C to F bytes in LB mode
0x184	QSPI_LB_STATUS	This register is QSPI AES status register in LB mode

Register Description

QSPI Clock Configuration Register

QSPI_CLK_CONFIG				
Address : 0x00				
Bits	Access	Mnemonic	Default value	Description
31-29	R/W	Reserved	0	Reserved
28	R/W	Qspi_rx_dqs_dll_calib	0	It enables the QSPI DLL calibration mode. It is used for both TX and RX DLL calibration.
27-22	R/W	Spi_clk_delay_val_tx	0	Delay value programmed to TX QSPI DLL in write path. This delay is used to delay the qspi clock output according to the requirement
21	R/W	Qspi_dll_enable_tx	0	Enable for TX QSPI DLL in write path. This is used in M4SS QSPI DDR pads to delay the qspi clock output. 0 – DLL is disabled/bypassed. 1 – DLL is enabled
20	R/W	ddr_clk_polarity_from_reg	0	Used this bit to sample the data at posedge/negedge after interface FFs with internal qspi clock 0- Sample at negedge 1- Sample at posedge
19	R	Reserved	-	
18	R	Reserved	-	
17-12	R	Reserved	-	
11-9	R/W	Reserved	0	Reserved
8	R/W	QSPI_CLK_EN_SCLK	1	QSPI clock enable for sclock. 0 – Dynamic clock gating is enabled in side QSPI controller. 1 – Full time clock is enabled for QSPI controller.
7:6	R/W	Reserved	0	Reserved
5	R/W	Reserved	0	If the clock frequency to FLASH(spi_clk) and QSPI(hclk) controller is same, this bit can be set to one to by-pass the syncros results in time consumption. Bypass sync logic bit. 1: Sync (synchros) logic is bypassed. 0: Sync logic is enabled.
4:0	R/W	QSPI_auto_csn_high_cnt	31	Minimum SOC clock cycles, during which QSPI auto csn should be high between consecutive CSN assertions. Range is 0 to 31.

QSPI Bus Mode Resister

QSPI_BUS_MODE				
Address : 0x04				
Bits	Access	Mnemonic	Default values	Description
31	R/W	QSPI_D3_DATA_CSN3	0	Value of SPI_IO3 in case of dual/single mode for chip select3 (cs_n3). It is used both in Auto and Manual Mode.
30	R/W	QSPI_D2_DATA_CSN3	0	Value of SPI_IO2 in case of dual/single mode for chip select3 (cs_n3). It is used both in Auto and Manual Mode.
29	R/W	QSPI_D3_OEN_CSN3	0	Direction Control for SPI_IO3 in case of dual/single mode for chip select3 (cs_n3). It is used both in Auto and Manual Mode.
28	R/W	QSPI_D2_OEN_CSN3	0	Direction Control for SPI_IO2 in case of dual/single mode for chip select3 (cs_n3). It is used both in Auto and Manual Mode.
27	R/W	QSPI_D3_DATA_CSN2	0	Value of SPI_IO3 in case of dual/single mode for chip select2 (cs_n2). It is used both in Auto and Manual Mode.
26	R/W	QSPI_D2_DATA_CSN2	0	Value of SPI_IO2 in case of dual/single mode for chip select2 (cs_n2). It is used both in Auto and Manual Mode.
25	R/W	QSPI_D3_OEN_CSN2	0	Direction Control for SPI_IO3 in case of dual/single mode for chip select2 (cs_n2). It is used both in Auto and Manual Mode.
24	R/W	QSPI_D2_OEN_CSN2	0	Direction Control for SPI_IO2 in case of dual/single mode for chip select2 (cs_n2). It is used both in Auto and Manual Mode.
23	R/W	Reserved	0	Reserved
22	R/W	Flash_sec_aes_ls_en	0	Qspi flash security fifo light sleep enable.
21	R/W	Flash_aw_fifo_ls_en	0	Qspi flash auto write fifo light sleep enable.
20	R/W	QSPI_CLK_MODE_CSN3	0	0 – Mode 0 , QSPI_CLK is low when QSPI_CS is high for chip select3 (csn3)1 – Mode 3 , QSPI_CLK is high when QSPI_CS is high for chip select3 (csn3)
19	R/W	QSPI_CLK_MODE_CSN2	0	0 – Mode 0, QSPI_CLK is low when QSPI_CS is high for chip select2 (csn2)1 – Mode 3, QSPI_CLK is high when QSPI_CS is high for chip select2 (csn2)
18	R/W	QSPI_CLK_MODE_CSN1	0	0 – Mode 0, QSPI_CLK is low when QSPI_CS is high for chip select1 (csn1)1 – Mode 3, QSPI_CLK is high when QSPI_CS is high for chip select1 (csn1)
17	R/W	QSPI_CLK_MODE_CSN0	0	0 – Mode 0, QSPI_CLK is low when QSPI_CS is high for chip select0 (csn0)1 – Mode 3, QSPI_CLK is high when QSPI_CS is high for chip select0 (csn0)
16	R/W	QSPI_DATA_SAMPLE_EDGE	0	Samples MISO data on clock edges. This should be **ZERO** for mode3 clock. 0 – Posedge of loop back spi_pad_clk. Use for low speed mode (sclk freq =< 40 MHz) 1 – Negedge of loop back spi_pad_clk. Use for high speed mode (sclk freq >= 40 MHz)
15	R/W	QSPI_D3_DATA_CSN1	0	Value of SPI_IO3 in case of dual/single mode for chip select1 (cs_n1). It is used both in Auto and Manual Mode.
14	R/W	QSPI_D2_DATA_CSN1	0	Value of SPI_IO2 in case of dual/single mode for chip select1 (cs_n1). It is used both in Auto and Manual Mode.

13	R/W	QSPI_D3_OEN_CSN1	0	Direction Control for SPI_IO3 in case of dual/single mode for chip select1 (cs_n1). It is used both in Auto and Manual Mode.
12	R/W	QSPI_D2_OEN_CSN1	0	Direction Control for SPI_IO2 in case of dual/single mode for chip select1 (cs_n1). It is used both in Auto and Manual Mode.
11	R/W	QSPI_D3_DATA_CSN0	0	Value of SPI_IO3 incase of dual/single mode for chip select0 (cs_n0). It is used both in Auto and Manual Mode.
10	R/W	QSPI_D2_DATA_CSN0	0	Value of SPI_IO2 incase of dual/single mode for chip select0 (cs_n0). It is used both in Auto and Manual Mode.
9	R/W	QSPI_D3_OEN_CSN0	0	Direction Control for SPI_IO3 incase of dual/single mode for chip select0 (cs_n0). It is used both in Auto and Manual Mode.
8	R/W	QSPI_D2_OEN_CSN0	0	Direction Control for SPI_IO2 incase of dual/single mode for chip select0 (cs_n0). It is used both in Auto and Manual Mode.
7	R/W	Programmable_auto_csn_base_addr_en	0	Programmable auto csn mode enable. 1- programmable auto csn mode is enabled. 0- programmable auto csn mode is diabled.
6	R/W	QSPI_AUTO_MODE_FRM_REG	0	Mode of Operation. 0 – Manual Mode is selected 1 – Auto Mode is selected. This is valid only when HW_CTRLD_QSPI_MODE_CTRL is zero. Before switching from MANUAL to AUTO_MODE, MANUAL should be IDLE. During transition from AUTO_MODE to MANUAL_MODE(After resetting this bit to zero) we have to make sure AUTO mode operations should be completed by checking the STATUS bit in MANUAL_STATUS register[12] (0 - Idle).
5	R/W	QSPI_WRAP_EN	0	Model wrap is considered with this bit and uses wrap instruction to read from FLASH, loaded into corresponding auto_config (csn0, csn1) register. 0 – Wrap mode is disabled (AHB WRAP can be used). 1 – Wrap mode is enabled. Note : Wrap mode is common in AUTO_MODE for both Flashes connected to csn0 and csn1.
4	R/W	QSPI_PREFETCH_EN	0	0 – Pre-fetch mode is disabled. 1 – Pre-fetch mode is enabled. Pre-fetch of data from the model which is connected to QSPI, automatically with out reading on AHB and is supplied to AHB, when address is matched with AHB read transaction address. Note : Pre-fetch mode is common in AUTO_MODE for both Flashes connected to csn0 and csn1.
3	R/W	AUTO_MODE_RESET	0	QSPI Auto controller reset. This is not a Self clearing bit. 0 – Auto mode is active. 1 – Auto mode is inactive(In soft-reset). Auto mode FIFO also get reset. Prefetch should be disabled while going to reset. The controller should be in normal (Not a HW_mode) Manual mode.

2:1	R/W	QSPI_MAN_MODE_CONF_CSN0	0	Configures the QSPI flash for Single/Dual/Quad mode operation in manual mode for chip selec0 (csn0). 00 - Single 01 - Dual 10 - Quad 11 – Reserved
0	R/W	Reserved	0	

QSPI Auto Controller Configuration 1 Register

QSPI_AUTO_CONFIG_1				
Address : 0x08				
Bits	Access	Mnemonic	Default values	Description
31-28	R/W	QSPI_DUMMY_BYTES_WRAP_CSN0	0	Specifies the number of dummy bytes (0 –3) for the selected SPI mode in case of wrap instruction.
27-24	R/W	QSPI_DUMMY_BYTES_INCR_CSN0	0	Specifies the number of dummy bytes (0 –3) for the selected SPI mode.
23	R/W	QSPI_PG_JUMP_CSN0	0	0 – Do not use Index jump instruction.1 – Use Index jump instruction specified by QSPI_PG_JUMP_INST.
22	R/W	Reserved	0	Reserved
21:20	R/W	Qspi_wrap-size	0	Qspi auto wrap size
19:18	R/W	QSPI_EXTRA_BYTE_EN_CSN0	0	00 – Do not transmit extra byte.01 – Transmit Extra byte after address phase.10 – Transmit only first nibble of the byte and maintain Hi-Z on the IO bus for next nibble.11 - Reserved
17:10	R/W	QSPI_EXTRA_BYTE_CSN0	0	Value of the extra byte to be transmitted, if the extra byte mode is enabled.
9:8	R/W	QSPI_DATA_MODE_CSN0	0	Mode of operation of QSPI in DATA phase.00 – SPI01 – Dual SPI10 – Quad SPI11 – Reserved
7:6	R/W	QSPI_CMD_MODE_CSN0	0	Mode of operation of QSPI in instruction phase.00 – SPI01 – Dual SPI10 – Quad SPI11 – Reserved
5:4	R/W	QSPI_ADDR_MODE_CSN0	0	Mode of operation of QSPI in instruction phase.00 – SPI01 – Dual SPI10 – Quad SPI11 – Reserved
3:2	R/W	QSPI_DUMMY_MODE_CSN0	0	Mode of operation of QSPI in instruction phase.00 – SPI01 – Dual SPI10 – Quad SPI11 – Reserved
1:0	R/W	QSPI_EXT_BYTE_MODE_CSN0	0	Mode of operation of QSPI in the extra byte phase.00 – SPI01 – Dual SPI10 – Quad SPI11 – Reserved

QSPI Auto Controller Configuration 2 Register

QSPI_AUTO_CONFIG_2				
Address : 0x0C				
Bits	Access	Mnemonic	Default values	Description
31:24	R/W	QSPI_PG_JMP_INST_CSN0	0	Read instruction to be used, when Page jump is to be used.
23:16	R/W	QSPI_RD_WRAP_INST_CSN0	0	Read instruction to be used, when wrap mode is supported by QSPI flash.
15:8	R/W	QSPI_RD_INST_CNS0_LSB	3	Read instruction LS byte to be used for the selected SPI modes and when wrap is not needed or supported.
7:4	R/W	QSPI_DUMMY_BYTES_JMP_CSN0	0	Dummy cycles to be selected in case of JUMP
3	R/W	DUMMY_BYTES_WR_RD_CSN0	0	Dummy bytes to the model to be read or to be write. 0 – Dummy bytes will be read. 1 – Dummy bytes to be write.

2	R/W	QSPI_CONTI_RD_EN_CSN0	0	Continuous read enable bit. 1 – Continuous read enabled. 0 – Continuous read disabled.
1	R/W	QSPI_ADR_SIZE_16BIT_AUTO_MODE_CNS0	0	1 – 16 Bit address is sent to model. 0 – 24 bit address is sent to model
0	R/W	QSPI_RD_DATA_SWAP_AUTO_CSN0	1	1 – Swap the auto read data in auto mode. 0 – Do not swap the read data in auto mode

QSPI Manual Configuration1 Register

QSPI_MANUAL_CONFIG1				
Address : 0x10				
Bits	Access	Mnemonic	Default values	Description
31:27	R/W	QSPI_MANUAL_RD_CNT[14:10]	0	Indicates total number of bytes or bits (depending on Qspi_manual_dummy_byte_or_bit_mode bit) to be read along with 12:3 bits of this register. Maximum length supported is 32k bytes.
26	R/W	QSPI_MANUAL_GSPI_MODE	0	Internally the priority is given to manual mode. 0 – SPI mode. When HW_CTRLD_MODE is enabled only, priority is given to manual mode (Breaking of manual mode operations won't be happen even though auto mode request is pending). It gives grant to AUTO mode requests after completion of current manual mode request (csn low to csn high). 1 – Host SPI mode. When HW_CTRLD_MODE is enabled, priority is given to auto mode (Breaking of manual mode operation can be happen). Break happens after the beat completion. After breaking, serves AUTO_MODE requests and continues the manual mode operation(already break). Beat refers to write or read size.
25	R/W	HW_CTRLD_QSPI_MODE_CTRL	0	Hardware controlled qspi mode in between AUTO and manual. 0 – Hardware control is disabled. 1 – Hardware control is enabled. Before setting HW_CTRL_MODE , MANUAL should be IDLE. After resetting HW_CTRL_MODE this bit to zero we have to make sure AUTO mode operations should be completed by checking the STATUS bit in MANUAL_STATUS register[14] (0 - Idle).
24:23	R/W	Reserved	0	Reserved

22	R/W	QSPI_FULL_DUPLEX_EN	0	Full duplex mode enable. 0 – Full duplex mode disabled. 1 – Full duplex mode enabled. Full duplex mode means reading while writing. When this bit is enabled, while writing the data to slaves connected to QSPI controller, reads the data from slave selected among the slaves connected to QSPI controller and stores in read_fifo. This fifo will get automatically flush after 16 reads and the fifo is not empty to write into.
21	R/W	TAKE_QSPI_MANUAL_WR_SIZE_FRM_REG	0	1 – Take write size from Manual config register1[20:19]. 0 – No action. Takes write size from fifo [36:35].
20:19	R/W	QSPI_MANUAL_SIZE_FRM_REG	3	Manual reads and manual writes(If take_manual_size_from_reg bit is 1) follow this size. 00 – 1 Byte (8 – bit mode) 01 – 2 Bytes (16 – bit mode) 10 – 3 Bytes (24 – bit mode) 11 – 4 Bytes (32 – bit mode) Above configuration is valid if Qspi_manual_dummy_byte_or_bit_mode bit is set to byte mode during read operation. If Qspi_manual_dummy_byte_or_bit_mode bit is set bti mode, following configuration is valid for read operation. 00 – 1 bit 01 – 2 bit 10 – 3 bit 11 – 4 bit For ddr mode, 01 and 11 are valid in dummy bit mode read.
18:15	R/W	Reserved	0	Reserved
14:13	R/W	QSPI_MANUAL_CSN_SELECT	0	Indicates which CSn is valid. Can be programmable in manual mode. Note : In auto mode csn select is decoded from the address itself, AHB_addr [25:24].
12:3	R/W	QSPI_MANUAL_RD_CNT[9:0]	0	Indicates total number of bytes to be read along with 31:27 bits of this register. Maximum length supported is 32k bytes.
2	R/W	QSPI_MANUAL_RD	0	Read enable for manual mode when CS is low.
1	R/W	QSPI_MANUAL_WR	0	Write enable for manual mode when CS is low.
0	R/W	QSPI_MANUAL_CSN	1	SPI CS in manual mode.

QSPI Manual Configuration2 Register

QSPI_MANUAL_CONFIG2				
Address : 0x14				
Bits	Access	Mnemonic	Default values	Description
31:26	R/W	Reserved	0	Reserved
25	R/W	Qspi_manual_dummy_byte_or_bit_mode	0	Indicates qspi_manual_rd_cnt values are dummy bytes or bits in manual mode. 1 means dummy bits mode 0 means dummy bytes mode It is used to provide proper dummy cycles in manual read mode.
24:15	-	Reserved	-	-
13:12	R/W	QSPI_MAN_MODE_CONF_CSN3	0	Configures the QSPI flash for Single/Dual/Quad mode operation in manual mode for chip select3 (csn3). 00 - Single 01 - Dual 10 - Quad 11 – Reserved
11:10	R/W	QSPI_MAN_MODE_CONF_CSN2	0	Configures the QSPI flash for Single/Dual/Quad mode operation in manual mode for chip select2 (csn2). 00 - Single 01 - Dual 10 - Quad 11 – Reserved
9:8	R/W	QSPI_MAN_MODE_CONF_CSN1	0	Configures the QSPI flash for Single/Dual/Quad mode operation in manual mode for chip select1 (csn1). 00 - Single 01 - Dual 10 - Quad 11 – Reserved
7	R/W	QSPI_RD_DATA_SWAP_MNL_CSN3	1	Swap the read data inside the QSPI controller it-self. 0 – Manual read data swap is disabled for csn3. 1 – Manual read data swap is enabled for csn3.
6	R/W	QSPI_RD_DATA_SWAP_MNL_CSN2	1	Swap the read data inside the QSPI controller it-self. 0 – Manual read data swap is disabled for csn2. 1 – Manual read data swap is enabled for csn2.
5	R/W	QSPI_RD_DATA_SWAP_MNL_CSN1	1	Swap the read data inside the QSPI controller it-self. 0 – Manual read data swap is disabled for csn1. 1 – Manual read data swap is enabled for csn1.

4	R/W	QSPI_RD_DATA_SWAP_MNL_CSN0	1	Swap the read data inside the QSPI controller it-self. 0 – Manual read data swap is disabled for csn0. 1 – Manual read data swap is enabled for csn0.
3	R/W	QSPI_WR_DATA_SWAP_MNL_CSN3	0	Swap the write data inside the QSPI controller it-self. 0 – Manual write data swap is disabled for csn3. 1 – Manual write data swap is enabled for csn3.
2	R/W	QSPI_WR_DATA_SWAP_MNL_CSN2	0	Swap the write data inside the QSPI controller it-self. 0 – Manual write data swap is disabled for csn2. 1 – Manual write data swap is enabled for csn2.
1	R/W	QSPI_WR_DATA_SWAP_MNL_CSN1	0	Swap the write data inside the QSPI controller it-self. 0 – Manual write data swap is disabled for csn1. 1 – Manual write data swap is enabled for csn1.
0	R/W	QSPI_WR_DATA_SWAP_MNL_CSN0	0	Swap the write data inside the QSPI controller it-self. 0 – Manual write data swap is disabled for csn0. 1 – Manual write data swap is enabled for csn0.

QSPI Manual Write Data 2 Register

QSPI_MANUAL_WRITE_DATA2				
Address : 0x80				
Bits	Access	Mnemonic	Default values	Description
31:9	R/W	Reserved	0	Reserved
8	R/W	Qspi_clk_enable_hclk	1	Static clock enable for qspi hclock
7	R/W	Use_prev_length	0	Use previous length. 1 – Uses previously programmed length in [4:0] of this register for next writes. 0 – No action. Note : TAKE_WR_SIZE_FRM_REG bit should be zero to consider this register.
6:5	R/W	Reserved	0	Reserved
4:0	R/W	QSPI_MANUAL_WRITE_DATA2	0	Number of bits to be written in write mode.

QSPI FIFO Threshold Register

QSPI_FIFO_THRLD				
Address : 0x1C				
Bits	Access	Mnemonic	Default values	Description
31:10	R/W	Reserved	0	Reserved
9	R/W	RFIFO_RESET	0	Read fifo reset
8	R/W	WFIFO_RESET	0	Write fifo reset
7:4	R/W	FIFO_AFULL_THRLD	12	FIFO almost full threshold
3:0	R/W	FIFO_AEMPTY_THRLD	7	FIFO almost empty threshold

QSPI Manual Read Data

The data is read from a 16x32 FIFO. The address is 0x40 to 0x7C.

QSPI Manual Write Data

The data is written into a 16x37 FIFO, least 32 bits. The address is 0x40 to 0x7C.

Writing and Reading methods

1) Manual mode supports read and write.

2) Auto mode supports only read. If writes comes on AUTO_AHB, writes will be neglected by QSPI controller.

3) Manual Write :

- Write valid number of bits into the wr_data2 register and then write the data in wr_data register.
- Continuous programming con be done as above for all the writes or USE_PREV_LNTH bit in wr_data2 register can be used for future length to use present programmed length.
- Write valid number of bytes minus one into manual config register [20:19] and set the bit TAKE_WR_SIZE_FRM_REG to consider this byte ordered length to write into the flash.

4) Auto Write :

- Enable the auto mode bit in MANUAL_CONFIG_REG and configure the AUTO_CONFIG registers
- Program the QSPI_SRAM_CTRL_CSN1_REG
- Start writing to the PSRAM with AUTO_MODE_BASE_ADDRESS plus PSRAM location (0x09000000 +PSRAM location).

5) Manual Read :

- Write the number bytes to read in MANUAL_CONFIG_REG [12:3] MANUAL_RD_CNT and keep the number bytes minus one in MANUAL_CONFIG_REG[20:19] and trigger read.
- Poll for rfifo not empty and read from the fifo.

6) Auto Read :

- Enable the auto mode bit in MANUAL_CONFIG_REG and configure the AUTO_CONFIG registers and then start reading from the flash or model connected to QSPI with AUTO_MODE_BASE ADDRRESS plus flash location at which you want to read.
- The valid range of address to read from the flashes on AHB is 0x0400_0000 to 0x07ff_ffff .

QSPI Application modes:

1. Read data in auto mode(Use csn0 or csn1) and read or write data in manual mode (Use csn0 or csn1 or csn2 or csn3).
2. Both manual and auto mode can access different flashes in HW_CTRLD mode enable at a time.
3. Both manual and auto modes can access same csn at a time in HW_CTRLD mode enable. Manual mode can access another flash.
4. Auto mode can access partly one csn(csn0) and partly another csn(csn1) Manual mode can access csn1.
5. Auto mode can access one csn(csn0) and partly another csn(csn1) and manual mode can access (csn1 and csn2) of the flash accessing by auto mode.
6. Auto mode can access any csn(csn0 or csn1) and manual mode can work with HOST SPI and QSPI.

Flash / PSRAM Security:

To support the flash security, 128 bit AES XTS/CTR mode decryption is available inline flash read. Decryption Supports in only Auto Mode. While burning into the flash, data is encrypted by considering the address as a counter in CTR mode and tweak value in XTS mode, so counter has to increment for every 128 bits of data chunk. To support the security, normal mode and mixed mode, security enabled regions can be mapped up to 4 regions. Security can be enabled on any region or all. .

Procedure:

1. Configure the AES mode in QSPI_AES_CONFIG register
2. Load the 128 bit key into the QSPI_AES_KEYx_x_x registers
3. Load the initialization value into the QSPI_AES_IVx_x_x registers.
4. Load the valid bytes of keys and IVs in QSPI_AES_KEY_IV_VALID register .
5. Load the registers with segment boundaries
   a. Lower Boundary Address: QSPI_AES_SEC_SEG_LS_ADDR_x.
   b. Upper Boundary Address: QSPI_AES_SEC_SEG_MS_ADDR_x
6. Set the security enable for required segment among 4 segments or all segments in the spi bus controller2 register[15:12] respectively.
7. Set the security enable bit [2] in the bus controller 2 register

QSPI AES Standalone mode :

QSPI AES module supports standalone mode (loop back mode) in XTS and CTR modes for both encryption and decryption of data.

Procedure:

1. Configure the AES standalone mode, encryption/decryption, data endian flipping in QSPI_AES_CONFIG register
2. Load the 128 bit key into the QSPI_AES_KEYx_x_x registers
3. Load the initialization value into the QSPI_AES_IV1_0_3 registers.
4. Load the valid bytes of keys and IVs in QSPI_AES_KEY_IV_VALID register .
5. Configure invalid bytes in QSPI_AES_LB_STATUS **(Only in CTR mode. XTS mode all bytes are valid).**
6. Check for din_ready status in QSPI_AES_LB_STATUS
7. FEED the data through QSPI_AES_LB_DATA_x_x register in the order of C_F, 8_B, 7_4, 3_0 for valid bytes. Valid bytes are always at MSB of the D-word.
8. Configure dout ready in QSPI_AES_LB_STATUS
9. Check for dout valid status in QSPI_AES_LB_STATUS.
10. Correspondingly read the data from QSPI_AES_LB_DATA_x_x in the order of C_F, 8_B, 7_4, 3_0 for valid bytes.

Notes :

Flash / PSRAM security is not supported for Wrap commands in them. It is supported only for AHB wrap to auto AHB slave.

For XTS mode decryption,

1. In standalone mode, after **step 4** make key valids key1_valid (QSPI_AES_KEY_IV_VALID[3:0]) and key2_valid (QSPI_AES_KEY_IV_VALID[11:8]) zeros. If they are high pre-calculated keys are not used.
2. In standalone mode, Program deckeycal QSPI_AES_CONFIG[9] zero for subsequent transactions with same keys, as pre calculation is already done and that pre-calculated key is used in subsequent decryption transaction.

Programming specific to security enabled cases.

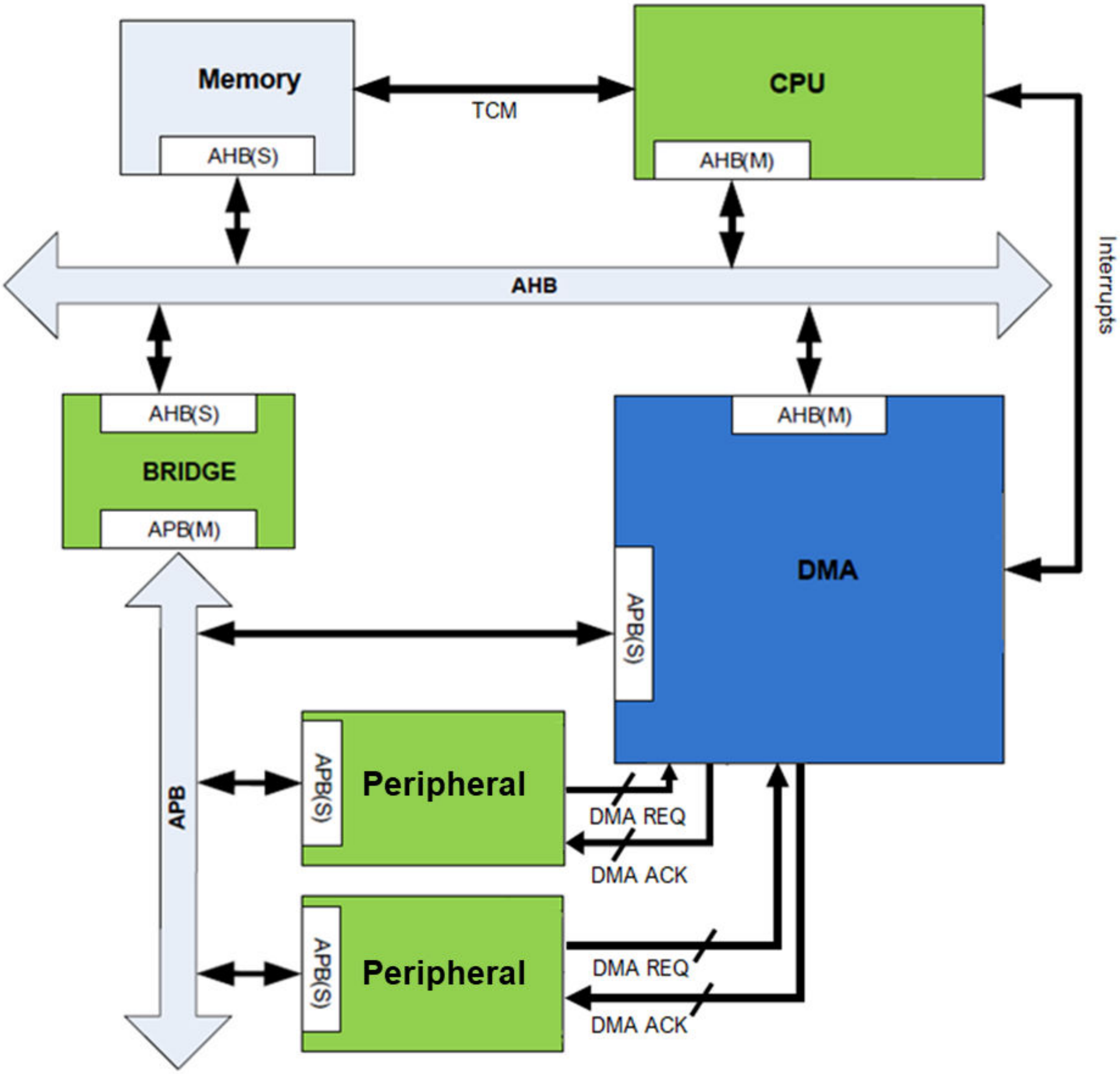
- Data swap sholud be enabled to get proper data in CTR 16-bit and CTR 32-bit mode and in all XTS mode scenarios. This should taken care in firmware.
- This swapped data is input for decryption.
- So, for 1-byte decryption, data will be palced as follows.
- 3rd MSB - 0th LSB

15. UDMA

15.1 General Description

The UDMA is an Advanced Microcontroller Bus Architecture (AMBA) compliant System-on-Chip (SoC) peripheral. It is a very low gate count DMA controller that is compatible with the AMBA AHB-Lite protocol. There are two UDMAs - MCU HP UDMA (for MCU HP peripherals) and MCU ULP UDMA (for MCU ULP peripherals)

uDMA Achitectural Diagram



DMA configuration is done through APB secondary interface. After the configuration is done and channel is enabled, the DMA transfers are excuted as follows

1. For memory to memory transfers DMA directly fetches the data from source memory and writes it to destination memory through the AHB Primary interface.
2. For memory to peripheral transfers, DMA waits for the destination peripheral to generate a request when it is ready to accept the data. Once transfer is complete acknowledgment is generated to the peripheral
3. For peripheral to memory transfers, DMA waits for the source peripheral to generate a request when it has the data available. Once transfer is complete acknowledgment is generated to the peripheral

15.2 Features

- It is compatible with AHB-Lite for the DMA transfers.
- It is compatible with APB for programming the registers.
- It has a single AHB-Lite master for transferring data using a 32-bit address bus and 32-bit data bus.
- Each DMA channel has dedicated handshake signals.
- Each DMA channel has a programmable priority level.
- Each priority level arbitrates using a fixed priority that is determined by the DMA channel number.
- It supports multiple transfer types: memory-to-memory ,memory-to-peripheral ,peripheral-to-memory.
- It supports multiple DMA cycle types.
- It supports multiple DMA transfer data widths.
- It supports both Memory and peripheral scatter-gather modes.
- Each DMA channel can access a primary, and alternate, channel control data structure.
- All the channel control data is stored in system memory in little-endian format.
- It performs all DMA transfers using the SINGLE AHB-Lite burst type.
- The destination data width is equal to the source data width.
- The number of transfers in a single DMA cycle can be programmed from 1 to 1024.
- The transfer address increment can be greater than the data width.
- It has a single output to indicate when an ERROR condition occurs on the AHB bus.
- HP UDMA has 32 channels. Out of which last 24 are dedicated channels for particular peripherals. First 8 channels can support 32 different peripherals.
- ULP UDMA has 12 channels.

15.3 Functional Description

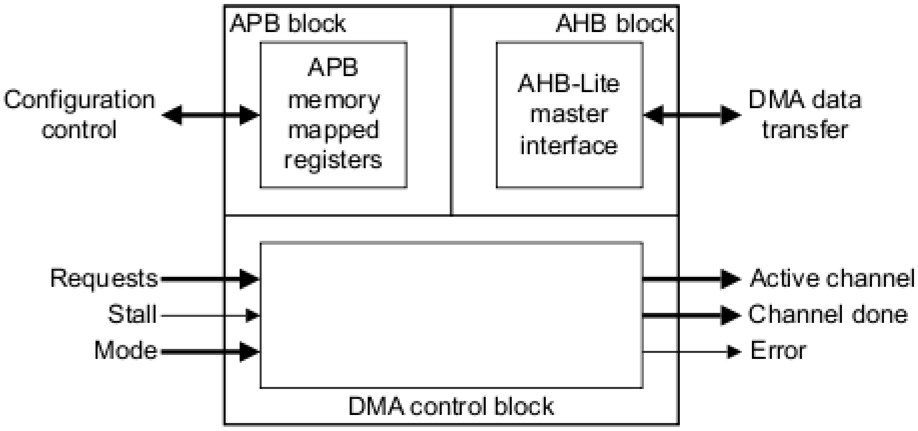


Figure 15.1. µDMA Block Diagram

µDMA has DMA control block, µDMA control and status register block and AHB-Lite primary as shown above. The µDMA control and status register block contains the registers that enable to configure the controller by using the APB secondaryu interface. The AHB-Lite primary transfer data from a source AHB secondary to a destination AHB secondary using a 32-bit data bus. DMA control block do the following functions:

- Arbitrates the incoming requests
- Indicates which channel is active
- Indicates when a channel is complete
- Indicates when an ERROR has occurred on the AHB-Lite interface
- Enables slow peripherals to stall the completion of a DMA cycle
- Waits for a request to clear before completing a DMA cycle
- Performs multiple or single DMA transfers for each request

HP UDMA Peripheral Mapping

All MCU HP peripherals are assigned with following PERIPHERAL CODEs in HP UDMA.

Peripheral Code	Acting as Source(Channel Number)	Acting as Destination(Channel Number)
CT0	0 (CT0_0 - destination)	1 (Reserved)
I2C 1	2	3
QSPI2/CRC	4 (QSPI2)	5 (CRC - destination) or (QSPI2)
SOC PLL SPI	6	7
MVP/CT0	8 (CT0_1 - destination)	9 (Reserved) or (MVP)
GSPI Master 1	10	11
I2S Channel 0	14	15
I2S Channel 1	16	17
SDIO	18	19
QSPI	20	21
SSI Slave	22	23
USART 0	24	25
UART 1	26	27
SSI Master	28	29
I2C 0	30	31

Note: By default CRC and CT0 channel will be selected for respective channel numbers. To configure it to QSPI2 and MVP respectively, we need to refer to mentioned register: DMA_DEVICE_SELECT_REG in Section 16.11 MCU Configuration Registers

ULP UDMA Peripheral Mapping

There are 12 dma channels present and are mapped to the following peripheral events

Chaneel number	Channel request signal
0	UART udma 0x
1	UART udma tx
2	SSI udma rx
3	SSI udma tx
4	I2C udma rx
5	I2C udma tx
6	I2S rx da0
7	I2S tx emp0
8	Timer interrupt
9	Cap sensing fifo empty
10	DAC fifo afull
11	ADC fifo aempty

DMA Transfer complete acknowledgment

The below mentioned peripherals require acknowledgment from UDMA after the DMA transfer is complete. So, acknowledgment is generated for these peripherals

1. SSI Primary
2. SSI Secondary
3. USART 0
4. UART 1
5. I2C 0
6. I2C 1
7. ULP SSI Primary
8. ULP UART
9. ULP I2C

15.4 Programming Sequence

The following are the steps needed to program µDMA to do transfers:

1. Program the source end address.
2. Program the destination end address.
3. Program the configuration information (the type of transfer and no of transfers).
4. Enable the particular channel for programming by writing to '1' at particular bit in enable register.
5. Enable the controller by writing '1' at 0th position of dma_config register.
6. Give the request to transfer. Either write the '1' at a particular position of channel software request register or through dma_req or dma_sreq.

15.5 Register Summary

Table 15.1. Register Summary Table

HP UDMA Base Address: 0x4403_0000 and **ULP UDMA Base Address: 0x2407_8000**

Register Name	Offset	Description
dma_status	0x000	DMA Status Register
dma_cfg	0x004	DMA Configuration Register
ctrl_base_ptr	0x008	Channel Control Data Base Pointer
alt_ctrl_base_ptr	0x00C	Channel Alternate Control Data Base Pointer
dma_waitonreq_status	0x010	Channel Wait on request status
chnl_sw_request	0x014	Channel Software Request
chnl_useburst_set	0x018	UDMA Channel useburst set
chnl_useburst_clr	0x01C	UDMA Channel useburst clear
chnl_req_mask_set	0x020	UDMA Channel request mask set
chnl_req_mask_clr	0x024	UDMA Channel request mask clear
chnl_enable_set	0x028	UDMA Channel enable register
chnl_enable_clr	0x02C	UDMA Channel enable clear register
chnl_pri_alt_set	0x030	UDMA Channel primary –alternate set
chnl_pri_alt_clr	0x034	UDMA Channel primary –alternate clear
chnl_priority_set	0x038	UDMA Channel Priority Set
chnl_priority_clr	0x03C	UDMA Channel Priority Clear
err_clr	0x04C	UDMA Bus Error Clear Register
skip_desc_fetch	0x050	UDMA Skip Descriptor Register
UDMA_done_status	0x800	UDMA Done Status Register
Channel_Status	0x804	Channel Status Register
Peripheral Select channel 0	0x808	Peripheral_select_channel0 register
Peripheral Select channel 1	0x80C	Peripheral_select_channel1 register
Peripheral Select channel 2	0x810	Peripheral_select_channel2 register
Peripheral Select channel 3	0x814	Peripheral_select_channel3 register
Peripheral Select channel 4	0x818	Peripheral_select_channel4 register
Peripheral Select channel 5	0x81C	Peripheral_select_channel5 register
Peripheral Select channel 6	0x820	Peripheral_select_channel6 register
Peripheral Select channel 7	0x824	Peripheral_select_channel7 register
UDMA_config_ctrl	0x828	UDMA Configuration Control Register

15.6 Register Description

Channel corresponding bits are 32 in case HP UDMA register and only 12 in case ULP UDMA registers. Remaining 20 bits ([31:12]) in ULP UDMA registers are reserved

15.6.1 DMA_STATUS

Table 15.2. DMA Status Register Description

Bit	Access	Function	Reset Value	Description
31:28	R	test_status	0x0	To reduce the gate count you can configure the controller, to exclude the integration test logic. Read as: 0x0 - controller does not include the integration test logic 0x1 - controller includes the integration test logic 0x2 - 0xF - undefined.
27:21	R	Reserved	0x0	Reserved
20:16	R	chnls_minus1		Number of available DMA channels minus one. For example: b00000 - controller configured to use 1 DMA channel b00001 - controller configured to use 2 DMA channels b00010 - controller configured to use 3 DMA channels . . . b11111 - controller configured to use 32 DMA channels.
15:8	R	Reserved	0x0	Reserved
7:4	R	state	0x0	Current state of the control state machine. State can be one of the following: b0000 - idle b0001 - reading channel controller data b0010 - reading source data end pointer b0011 - reading destination data end pointer b0100 - reading source data b0101 - writing destination data b0110 - waiting for DMA request to clear b0111 - writing channel controller data b1000 - stalled b1001 - done b1010 - peripheral scatter-gather transition b1011-b1111 - undefined.
3:1	R	Reserved	0x0	Reserved
0	R	master_enable	0x0	Enable status of the controller: 0 - controller is disabled 1 - controller is enabled.

15.6.2 DMA_CFG

Table 15.3. DMA Configuration Register Description

Bit	Access	Function	Reset Value	Description
31:8	W	Reserved	0x0	Undefined. Write as zero.
7:5	W	chnl_prot_ctrl		Sets the AHB-Lite protection by controlling the HPROT[3:1] signal levels as follows: Bit [7] -Controls HPROT[3] to indicate if a cacheable access is occurring. Bit [6] -Controls HPROT[2] to indicate if a bufferable access is occurring. Bit [5] -Controls HPROT[1] to indicate if a privileged access is occurring.
4:1	W	Reserved	0x0	Undefined. Write as zero.

Bit	Access	Function	Reset Value	Description
0	W	master_enable	0x0	Enable for the controller: 0 - disables the controller 1 - enables the controller.

15.6.3 CTRL_BASE_PTR

Table 15.4. Channel Control Data Base Pointer Register Description

Bit	Access	Function	Reset Value	Description
31:10	R/W	ctrl_base_ptr	0x0	Pointer to the base address of the primary data structure.
9:0	W	Reserved	0x0	Undefined. Write as zero.

15.6.4 ALT_CTRL_BASE_PTR

Table 15.5. Channel Alternate Control Data Base Pointer Register Description

Bit	Access	Function	Reset Value	Description
31:0	R	alt_ctrl_base_ptr	0x0	Base address of the alternate data structure

15.6.5 DMA_WAITONREQUEST_STATUS

Table 15.6. Channel Wait On Request Status Register Description

Bit	Access	Function	Reset Value	Description
31:0	R	dma_waitonreq_status	0x0	Per Channel wait on request status(where C specifies channel number). Read as: Bit [C] - 0 dma_waitonreq[C] is LOW. Bit [C] - 1 dma_waitonreq[C] is HIGH.

15.6.6 CHNL_SW_REQUEST

Table 15.7. Channel Software Request Register Description

Bit	Access	Function	Reset Value	Description
31:0	W	chnl_sw_request	0x0	Set the appropriate bit to generate a software DMA request on the corresponding DMA channel(C specifies channel number). Write as: Bit [C] - 0 Does not create a DMA request for channel C. Bit [C] - 1 Creates a DMA request for channel C. Writing to a bit where a DMA channel is not implemented does not create a DMA request for that channel.

15.6.7 CHNL_USEBURST_SET

Table 15.8. UDMA Channel Useburst Set Register Description

Bit	Access	Function	Reset Value	Description
31:0	R/W	Chnl_useburst_set	0x0	Returns the useburst status, or disables dma_sreq[C] from generating DMA requests. Read as: Bit [C] - 0 DMA channel C responds to requests that it receives on dma_req[C] or dma_sreq[C]. The controller performs 2R, or single, bus transfers. Bit [C] - 1 DMA channel C does not respond to requests that it receives on dma_sreq[C]. The controller only responds to dma_req[C] requests and performs 2R transfers. Write as: Bit [C] - 0 No effect. Use the chnl_useburst_clr Register to set bit [C] to 0. Bit [C] - 1 Disables dma_sreq[C] from generating DMA requests. The controller performs 2R transfers. Writing to a bit where a DMA channel is not implemented has no effect.

15.6.8 CHNL_USEBURST_CLR

Table 15.9. UDMA Channel Useburst Clear Register Description

Bit	Access	Function	Reset Value	Description
31:0	W	chnl_useburst_clr	0x0	Set the appropriate bit to enable dma_sreq[] to generate requests. Write as: Bit [C] - 0 No effect. Use the chnl_useburst_set Register to disable dma_sreq[] from generating requests. Bit [C] - 1 Enables dma_sreq[C] to generate DMA requests. Writing to a bit where a DMA channel is not implemented has no effect.

15.6.9 CHNL_REQ_MASK_SET

Table 15.10. UDMA Channel Request Mask Set Register Description

Bit	Access	Function	Reset Value	Description
31:0	R/W	chnl_req_mask_set	0x0	Returns the request mask status of dma_req[] and dma_sreq[], or disables the corresponding channel from generating DMA requests. Read as: Bit [C] - 0 External requests are enabled for channel C. Bit [C] - 1 External requests are disabled for channel C. Write as: Bit [C] - 0 No effect. Use the chnl_req_mask_clr Register to enable DMA requests. Bit [C] - 1 Disables dma_req[C] and dma_sreq[C] from generating DMA requests. Writing to a bit where a DMA channel is not implemented has no effect.

15.6.10 CHNL_REQ_MASK_CLR

Table 15.11. UDMA Channel Request Mask Clear Register Description

Bit	Access	Function	Reset Value	Description
31:0	W	chnl_req_mask_clr	0x0	Returns the request mask status of dma_req[] and dma_sreq[], or disables the corresponding channel from generating DMA requests. Read as: Bit [C] - 0 External requests are enabled for channel C. Bit [C] - 1 External requests are disabled for channel C. Write as: Bit [C] - 0 No effect. Use the chnl_req_mask_clr Register to enable DMA requests. Bit [C] - 1 Disables dma_req[C] and dma_sreq[C] from generating DMA requests. Writing to a bit where a DMA channel is not implemented has no effect.

15.6.11 CHNL_ENABLE_SET

Table 15.12. UDMA Channel Enable Set Register Description

Bit	Access	Function	Reset Value	Description
31:0	R/W	chnl_enable_set	0x0	Returns the enable status of the channels, or enables the corresponding channels. Read as: Bit [C] - 0 Channel C is disabled. Bit [C] - 1 Channel C is enabled. Write as: Bit [C] - 0 No effect. Use the chnl_enable_clr Register to disable a channel. Bit [C] - 1 Enables channel C. Writing to a bit where a DMA channel is not implemented has no effect.

15.6.12 CHNL_ENABLE_CLR

Table 15.13. UDMA Channel Enable Clear Register Description

Bit	Access	Function	Reset Value	Description
31:0	W	chnl_enable_clr	0x0	Set the appropriate bit to disable the corresponding DMA channel. Write as: Bit [C] - 0 No effect. Use the chnl_enable_set Register to enable DMA channels. Bit [C] - 1 Disables channel C. Writing to a bit where a DMA channel is not implemented has no effect. **Note:** The controller disables a channel, by setting the appropriate bit, when either: it completes the DMA cycle; it reads a channel_cfg memory location which has cycle_ctrl - b000; an ERROR occurs on the AHB-Lite bus.

15.6.13 CHNL_PRI_ALT_SET

Table 15.14. UDMA Channel Primary –Alternate Set Register Description

Bit	Access	Function	Reset Value	Description
31:0	R/W	chnl_pri_alt_set	0x0	Returns the channel control data structure status, or selects the alternate data structure for the corresponding DMA channel. Read as: Bit [C] - 0 DMA channel C is using the primary data structure. Bit [C] - 1 DMA channel C is using the alternate data structure. Write as: Bit [C] - 0 No effect. Use the chnl_pri_alt_clr Register to set bit [C] to 0. Bit [C] - 1 Selects the alternate data structure for channel C. Writing to a bit where a DMA channel is not implemented has no effect. **Note:** The controller toggles the value of the chnl_pri_alt_set [C] bit after it completes: the four transfers that the primary data structure specifies for a memory scatter-gather, or peripheral scatter-gather, DMA cycle; all the transfers that the primary data structure specifies for a ping-pong DMA cycle; all the transfers that the alternate data structure specifies for the following DMA cycle types: — ping-pong — memory scatter-gather — peripheral scatter-gather.

15.6.14 CHNL_PRI_ALT_CLR

Table 15.15. UDMA Channel Primary –Alternate Clear Register Description

Bit	Access	Function	Reset Value	Description
31:0	W	chnl_pri_alt_clr	0x0	Set the appropriate bit to select the primary data structure for the corresponding DMA channel. Write as: Bit [C] - 0 No effect. Use the chnl_pri_alt_set Register to select the alternate data structure. Bit [C] - 1 Selects the primary data structure for channel C. Writing to a bit where a DMA channel is not implemented has no effect. **Note:** The controller toggles the value of the chnl_pri_alt_clr [C] bit after it completes: the four transfers that the primary data structure specifies for a memory scatter-gather, or peripheral scatter-gather, DMA cycle; all the transfers that the primary data structure specifies for a ping-pong DMA cycle; all the transfers that the alternate data structure specifies for the following DMA cycle types: — ping-pong — memory scatter-gather — peripheral scatter-gather.

15.6.15 CHNL_PRIORITY_SET

Table 15.16. UDMA Channel Priority Set Register Description

Bit	Access	Function	Reset Value	Description
31:0	R/W	chnl_priority_set	0x0	Set corresponding bit to set the priority level of a channel Read as : Bit [C] - 0 channel priority is not prioritized. Bit [C] - 1 channel is prioritized Write as: Bit [C] - 0 No effect. Bit [C] - 1 sets the channel priority to 1 **Note:** By default Channel 0 has the highest priority. Channel 31 has the lowest priority If multiples channel has the same priority level, the channel with lowest number will have the highest priority among them

15.6.16 CHNL_PRIORITY_CLR

Table 15.17. UDMA Channel Priority Clear Register Description

Bit	Access	Function	Reset Value	Description
31:0	W	chnl_priority_clr	0x0	Set corresponding bit to set the priority level of a channel Read as : Bit [C] - 0 channel priority is not prioritized. Bit [C] - 1 channel is prioritized Write as: Bit [C] - 0 No effect. Bit [C] - 1 clears the channel priority to 0 **Note:** By default Channel 0 has the highest priority. Channel 31 has the lowest priority If multiples channel has the same priority level, the channel with lowest number will have the highest priority among them

15.6.17 ERR_CLR

Table 15.18. UDMA Bus Error Clear Register Description

Bit	Access	Function	Reset Value	Description
31:1	R/W	Reserved	0x0	Reserved
0	R/W	Err_clr	0x0	Returns the status of dma_err, or sets the signal LOW. Read as: 0 - dma_err is LOW 1 - dma_err is HIGH. Write as: 0 - No effect, status of dma_err is unchanged. 1 - Sets dma_err LOW. **Note:** If you deassert dma_err at the same time as an ERROR occurs on the AHB-Lite bus, then the ERROR condition takes precedence and dma_err remains asserted.

15.6.18 SKIP_DESC_FETCH

Table 15.19. UDMA Skip Descriptor Register Description

Bit	Access	Function	Reset Value	Description
31:0	R/W	skip_desc_fetch	0x0	Bit[C] - 1, enables the skipping of descriptor for each transfer for channel C UDMA by default fetches the source and destination addresses for each transfer, even during a burst. Setting above bit will avoid these repeated fetches within burst. We will buffer them and use within burst. This will help to improve the performance of transfer and saves bus cycles. This features has to be enabled always.

15.6.19 UDMA_DONE_STATUS

Table 15.20. UDMA Done Status Register Description

Bit	Access	Function	Reset Value	Description
31	R/W	Done_status_channel_31	0x0	Reading '1' indicates the transfer is completed for channel 31st . Writing '1' will clear the bit. Writing 0 will have no effect.
30	R/W	Done_status_channel_30	0x0	Reading '1' indicates the transfer is completed for channel 30th . Writing '1' will clear the bit. Writing 0 will have no effect.
29	R/W	Done_status_channel_29	0x0	Reading '1' indicates the transfer is completed for channel 29th . Writing '1' will clear the bit. Writing 0 will have no effect.
28	R/W	Done_status_channel_28	0x0	Reading '1' indicates the transfer is completed for channel 28th. Writing '1' will clear the bit. Writing 0 will have no effect.
27	R/W	Done_status_channel_27	0x0	Reading '1' indicates the transfer is completed for channel 27th . Writing '1' will clear the bit. Writing 0 will have no effect.

Bit	Access	Function	Reset Value	Description
26	R/W	Done_status_channel_26	0x0	Reading '1' indicates the transfer is completed for channel 26th . Writing '1' will clear the bit. Writing 0 will have no effect.
25	R/W	Done_status_channel_25	0x0	Reading '1' indicates the transfer is completed for channel 25th . Writing '1' will clear the bit. Writing 0 will have no effect.
24	R/W	Done_status_channel_24	0x0	Reading '1' indicates the transfer is completed for channel 24th . Writing '1' will clear the bit. Writing 0 will have no effect.
23	R/W	Done_status_channel_23	0x0	Reading '1' indicates the transfer is completed for channel 23rd . Writing '1' will clear the bit. Writing 0 will have no effect.
22	R/W	Done_status_channel_22	0x0	Reading '1' indicates the transfer is completed for channel 22nd . Writing '1' will clear the bit. Writing 0 will have no effect.
21	R/W	Done_status_channel_21	0x0	Reading '1' indicates the transfer is completed for channel 21st. Writing '1' will clear the bit. Writing 0 will have no effect.
20	R/W	Done_status_channel_20	0x0	Reading '1' indicates the transfer is completed for channel 20th . Writing '1' will clear the bit. Writing 0 will have no effect.
19	R/W	Done_status_channel_19	0x0	Reading '1' indicates the transfer is completed for channel 19th . Writing '1' will clear the bit. Writing 0 will have no effect.
18	R/W	Done_status_channel_18	0x0	Reading '1' indicates the transfer is completed for channel 18th . Writing '1' will clear the bit. Writing 0 will have no effect.
17	R/W	Done_status_channel_17	0x0	Reading '1' indicates the transfer is completed for channel 17th . Writing '1' will clear the bit. Writing 0 will have no effect.
16	R/W	Done_status_channel_16	0x0	Reading '1' indicates the transfer is completed for channel 16th . Writing '1' will clear the bit. Writing 0 will have no effect.
15	R/W	Done_status_channel_15	0x0	Reading '1' indicates the transfer is completed for channel 15th . Writing '1' will clear the bit. Writing 0 will have no effect.
14	R/W	Done_status_channel_14	0x0	Reading '1' indicates the transfer is completed for channel 14th . Writing '1' will clear the bit. Writing 0 will have no effect.
13	R/W	Done_status_channel_13	0x0	Reading '1' indicates the transfer is completed for channel 13th . Writing '1' will clear the bit. Writing 0 will have no effect.
12	R/W	Done_status_channel_12	0x0	Reading '1' indicates the transfer is completed for channel 12th. Writing '1' will clear the bit. Writing 0 will have no effect.
11	R/W	Done_status_channel_11	0x0	Reading '1' indicates the transfer is completed for channel 11th . Writing '1' will clear the bit. Writing 0 will have no effect.
10	R/W	Done_status_channel_10	0x0	Reading '1' indicates the transfer is completed for channel 10th . Writing '1' will clear the bit. Writing 0 will have no effect.

Bit	Access	Function	Reset Value	Description
9	R/W	Done_status_channel_9	0x0	Reading '1' indicates the transfer is completed for channel 9th . Writing '1' will clear the bit. Writing 0 will have no effect.
8	R/W	Done_status_channel_8	0x0	Reading '1' indicates the transfer is completed for channel 8th . Writing '1' will clear the bit. Writing 0 will have no effect.
7	R/W	Done_status_channel_7	0x0	Reading '1' indicates the transfer is completed for channel 7th . Writing '1' will clear the bit. Writing 0 will have no effect.
6	R/W	Done_status_channel_6	0x0	Reading '1' indicates the transfer is completed for channel 6th . Writing '1' will clear the bit. Writing 0 will have no effect.
5	R/W	Done_status_channel_5	0x0	Reading '1' indicates the transfer is completed for channel 5th . Writing '1' will clear the bit. Writing 0 will have no effect.
4	R/W	Done_status_channel_4	0x0	Reading '1' indicates the transfer is completed for channel 4th . Writing '1' will clear the bit. Writing 0 will have no effect.
3	R/W	Done_status_channel_3	0x0	Reading '1' indicates the transfer is completed for channel 3rd . Writing '1' will clear the bit. Writing 0 will have no effect.
2	R/W	Done_status_channel_2	0x0	Reading '1' indicates the transfer is completed for channel 2nd . Writing '1' will clear the bit. Writing 0 will have no effect.
1	R/W	Done_status_channel_1	0x0	Reading '1' indicates the transfer is completed for channel 1st . Writing '1' will clear the bit. Writing 0 will have no effect.
0	R/W	Done_status_channel_0	0x0	Reading '1' indicates the transfer is completed for channel 0th . Writing '1' will clear the bit. Writing 0 will have no effect.

15.6.20 CHANNEL_STATUS

Table 15.21. UDMA Channel Status Register Description

Bit	Access	Function	Reset Value	Description
31	R	Busy or ideal status_channel_31	0x0	Reading '1' indicates that the channel 31 is busy.
30	R	Busy or ideal status_channel_30	0x0	Reading '1' indicates that the channel 30 is busy.
29	R	Busy or ideal status_channel_29	0x0	Reading '1' indicates that the channel 29 is busy.
28	R	Busy or ideal status_channel_28	0x0	Reading '1' indicates that the channel 28 is busy.
27	R	Busy or ideal status_channel_27	0x0	Reading '1' indicates that the channel 27 is busy.
26	R	Busy or ideal status_channel_26	0x0	Reading '1' indicates that the channel 26 is busy.
25	R	Busy or ideal status_channel_25	0x0	Reading '1' indicates that the channel 25 is busy.
24	R	Busy or ideal status_channel_24	0x0	Reading '1' indicates that the channel 24 is busy.
23	R	Busy or ideal status_channel_23	0x0	Reading '1' indicates that the channel 23 is busy.
22	R	Busy or ideal status_channel_22	0x0	Reading '1' indicates that the channel 22 is busy.

Bit	Access	Function	Reset Value	Description
21	R	Busy or ideal status_channel_21	0x0	Reading '1' indicates that the channel 21 is busy.
20	R	Busy or ideal status_channel_20	0x0	Reading '1' indicates that the channel 20 is busy.
19	R	Busy or ideal status_channel_19	0x0	Reading '1' indicates that the channel 19 is busy.
18	R	Busy or ideal status_channel_18	0x0	Reading '1' indicates that the channel 18 is busy.
17	R	Busy or ideal status_channel_17	0x0	Reading '1' indicates that the channel 17 is busy.
16	R	Busy or ideal status_channel_16	0x0	Reading '1' indicates that the channel 16 is busy.
15	R	Busy or ideal status_channel_15	0x0	Reading '1' indicates that the channel 15 is busy.
14	R	Busy or ideal status_channel_14	0x0	Reading '1' indicates that the channel 14 is busy.
13	R	Busy or ideal status_channel_13	0x0	Reading '1' indicates that the channel 13 is busy.
12	R	Busy or ideal status_channel_12	0x0	Reading '1' indicates that the channel 12 is busy.
11	R	Busy or ideal status_channel_11	0x0	Reading '1' indicates that the channel 11 is busy.
10	R	Busy or ideal status_channel_10	0x0	Reading '1' indicates that the channel 10 is busy.
9	R	Busy or ideal status_channel_9	0x0	Reading '1' indicates that the channel 9 is busy.
8	R	Busy or ideal status_channel_8	0x0	Reading '1' indicates that the channel 8 is busy.
7	R	Busy or ideal status_channel_7	0x0	Reading '1' indicates that the channel 7 is busy.
6	R	Busy or ideal status_channel_6	0x0	Reading '1' indicates that the channel 6 is busy.
5	R	Busy or ideal status_channel_5	0x0	Reading '1' indicates that the channel 5 is busy.
4	R	Busy or ideal status_channel_4	0x0	Reading '1' indicates that the channel 4 is busy.
3	R	Busy or ideal status_channel_3	0x0	Reading '1' indicates that the channel 3 is busy.
2	R	Busy or ideal status_channel_2	0x0	Reading '1' indicates that the channel 2 is busy.
1	R	Busy or ideal status_channel_1	0x0	Reading '1' indicates that the channel 1 is busy.
0	R	Busy or ideal status_channel_0	0x0	Reading '1' indicates that the channel 0 is busy.

15.6.21 UDMA_PERIPHERAL_SEL_CHn_REG

Bit	Access	Function	Default Value	Description
31:5	-	Reserved	-	
4:0	R/W	peripheral_select_channel_n_	0x0	Selects which peripheral will be selected for the channel n. For peripheral codes, refer to peripheral code decoding table.

15.6.22 UDMA_CONFIG_CTRL

Table 15.22. UDMA Configuration Control Register Description

Bit	Access	Function	Reset Value	Description
31:1	R/W	Reserved	0x0	Reserved

Bit	Access	Function	Reset Value	Description
0	R/W	Single_request_enable	0x0	Enabled signal for single request 0 - Single request will be disabled 1 - Single request will be enabled Connect dma_waitonreq port to zero. DMA will consider single requests (empty, full signals from peripherals) only when dma_waitonreq is high. When single_request_enable is set, added logic will ignore dma_waitonreq and consider single requests. If this is not enabled, the following case will not work . Read 10 bytes from UART and beat size/fifo threshold configured as 4 bytes. Only 8 bytes will be read and DMA will not read remaining two bytes as dma_req will not come.

16. MCU Peripherals

16.1 HSPI Secondary

16.1.1 General Description

The HSPI Secondary Interface is a full duplex serial host interface, which supports 8-bit and 32-bit data granularity. It also supports the gated mode of the SPI clock and the low, high, and very high frequency modes. In case of low frequency host, the data is driven on the falling edge and sampled on the rising edge and hence, it should be ensured that a valid data is present on the bus at the immediate rising edge after the SPI chip select is driven low. For high frequency transmission, the data is driven as well as sampled on rising edge.

16.1.2 Features

- 4-pin serial interface
- Supports 8-bit and 32-bit data granularity
- SPI clock can be at the max 4 times higher than AHB clock
- Supports DMA flow control signals
- Supports AHB interface for accessing data from SOC
- Supports system soft reset from host

16.1.3 Functional Description

The SPI Secondary interface is invoked by the host using a specific pattern in the first 32-bits of each transfer. This 32-bit pattern is divided into four 8-bit patterns and each 8-bit set is termed as one command.
Transmissions up to 25 MHz are termed low-speed transmissions, and during these transmissions, the data is driven on the falling edge of the clock and read on the rising edge of the clock.

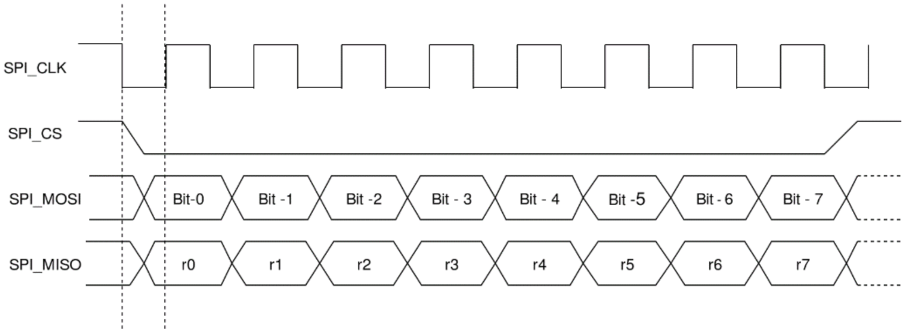


Figure 16.1. SPI Low-Speed Transmission

Transmissions above 25 MHz are high-speed transmissions, and during these transmissions, the data is driven on the rising edge of the clock and read on following rising edge of the clock.

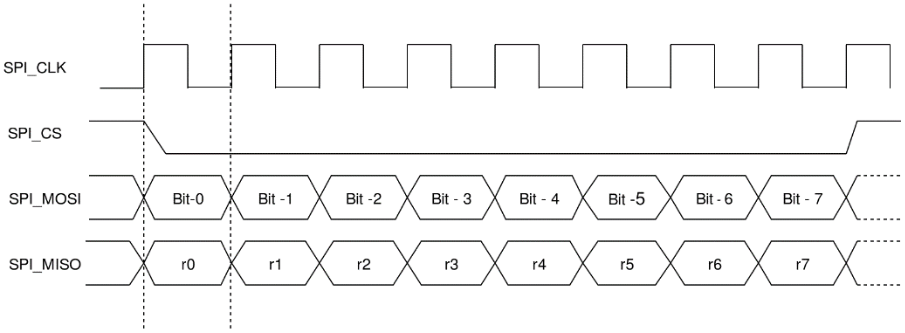


Figure 16.2. SPI High-Speed Transmission

However, initialization is done only in low-speed mode. To enable the SPI interface for a high-speed transmission, the host has to first initialize the SPI Interface and then perform a SPI Secondary Internal Register Write operation to SPI_MODE register to set the SPI_MODE [SPI_OP_MODE] bit. The register write is also performed in the low-speed mode. The host can do a high-speed transfer only if the SPI_MODE [SPI_OP_MODE] bit is set.

16.1.3.1 SPI Commands

The SPI interface is programmed to perform a certain transfer using four commands and a 32-bit address. The host transfers all the Commands and Addresses with 8-bit granularity. At the end of all Commands and Addresses, the host should be reconfigured to transfer data with 8-bit or 32-bit granularity depending on the commands issued. The secondary responds to all the commands with a certain response pattern. For transferring the response, start-token, and data, the SPI interface follows the same protocol as followed by the host.
The four commands C1, C2, C3, C4 indicate to the SPI interface all the aspects of the transfer.

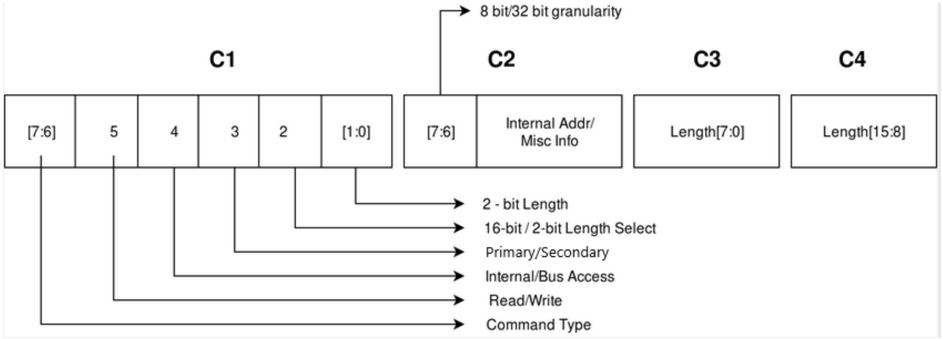


Figure 16.3. SPI Command Description

The command description is as follows:

Table 16.1. SPI Command Description

C1	[7:6]	Command Type "00"- Initialization Command "01"- Read/Write Command "10", "11"- Reserved for future use
	5	Read/Write '0'- Read Command '1'- Write Command
	4	Internal/Bus Access '0'- SPI Secondary Internal Access '1'- AHB Bus Access
	3	Primary/Secondary Access '0'- AHB Primary Access '1'- AHB Secondary Access
	2	2-bit or 16-bit length for the transfer '0'- 2-bit length for the transfer '1'- 16-bit length for the transfer

	1:0	2-bit length (in terms of bytes) for the transfer (if bit 2 is cleared) "00"- 4 Bytes length "01"- 1 Byte length "10"- 2 Bytes length "11"- 3 Bytes length
C2	7:6	Data granularity. Indicates the granularity of the write/read data. Note: The Primary commands and addresses will always be 8-bit irrespective of this value. "00"- 8-bit granularity "01"- 32-bit granularity "10", "11"- Reserved for future use
	5:0	Internal Address or miscellaneous Info. This carries the SPI Secondary's internal address if bit 4 for Command C1 is cleared. Otherwise, it carries miscellaneous information for the Godavari chip.
C3	7:0	Length (15:8) MSB of the transfer's length (which is in terms of bytes) in case bit 2 of C1 is set. This command is skipped if bit 2 of C1 is cleared i.e., if 2-bit length is selected
C4	7:0	Length (7:0) LSB of the transfer's length (which is in terms of bytes) in case bit 2 of C1 is set. This command is skipped if bit 2 of C1 is cleared.

Depending on the above four commands, the SPI interface can be initialized or made to do a read or write operation to an AHB Primary or an AHB Secondary. The SPI interface responds with set of unique responses to all these commands.

16.1.3.2 Secondary Response to Commands

The SPI Secondary gives simultaneous responses to the SPI Primary's requests. These are as follows:

- • An 8-bit Success/Failure response at the end of receiving the first 8-bits of the Command. This response is continuously driven in the case of a Write Request starting from the time when the first 8-bits are received. This response is driven with 8-bit granularity during the Command and Address phase and then is switched to 8-bit or 32-bit granularity during the Data phase, according to the command issued
  - Success: 0x58
  - Failure: 0x52
- An 8-bit or 32-bit start token is transmitted once the four commands indicating a read request are received and the secondary is ready to transmit data. The start token is immediately followed by the read-data
  - Start Token: 0x55
- An 8-bit busy response is driven by the SPI interface in case a new transaction is initiated by the host while the previous transaction is still pending from the system side. In this case, the host has to retry the commands.
  - Busy Response: 0x54

16.1.3.3 Initialization

The Initialization Command is given to the secondary to initialize the SPI interface. The SPI interface remains non-functional to any command before initialization and responds only after successful initialization. Initialization should be done only once after the power-on. The SPI Secondary treats any subsequent initialization commands before the reset as errors.

For the initialization command, the host drives C1 command, followed by an 8-bit miscellaneous data. Bits 7:6 of C1 are cleared and 0x15 is driven on bits 5:0. Status response from the SPI Interface is driven during the transmission of the miscellaneous data i.e., after the transfer of 8-bits of command C1.

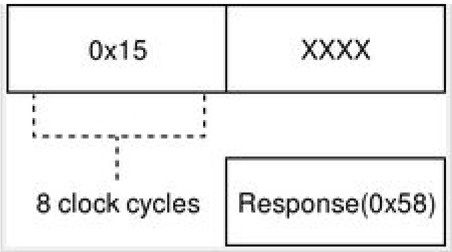


Figure 16.4. SPI Secondary Initialization

16.1.3.4 Busy Condition

The busy condition arises when the a new transaction is initiated by the host while the previous transaction is still pending from the system side. The SPI Interface indicates this to the host with a Busy Response 0x54 to the command C1. In such cases, the host should re-transmit the commands until it receives a success response 0x58 from the SPI Secondary Interface.

16.1.3.5 AHB Primary Write

During a write operation to the AHB Primary, the host first transfers the four commands, then transfers a valid 32-bit address to which write has to be done and then the data. The secondary responds to all the commands and data with an appropriate response.

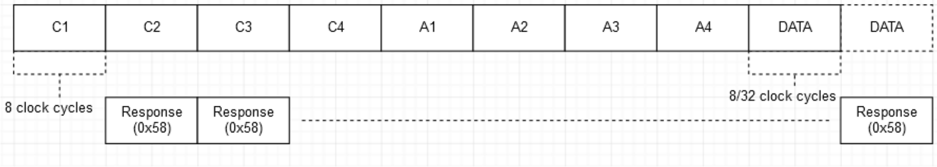


Figure 16.5. SPI Secondary AHB Primary Write

16.1.3.6 AHB Secondary Write

This is similar to the AHB Primary write except that bit 3 of C1 is set and the Address phase (A1, A2, A3& A4) is skipped.

16.1.3.7 SPI Interface Internal Registers Write

This is similar to the AHB Primary write except that bit 4 of C1 is cleared and A1, A2, A3, and A4 are skipped. The Valid Data phase starts immediately after C4 is transferred.

16.1.3.8 AHB Primary/Secondary, Internal Register Write with 2-bit Length

This, too, is similar to the AHB Primary write except that bit 2 of C1 is cleared and C3 and C4 are both skipped. If it's an internal register or AHB Seondary access then A1, A2, A3, and A4 are also skipped after C2. The number of bytes to be transferred is inferred from bits [1:0] of C1.

16.1.3.9 AHB Primary Read

During the AHB Primary read operation, first the four commands are transmitted, then the 32-bit address, and then after a variable wait period, a start token is transmitted. The start token is followed by valid data. The start token indicates to the host the beginning of valid data.

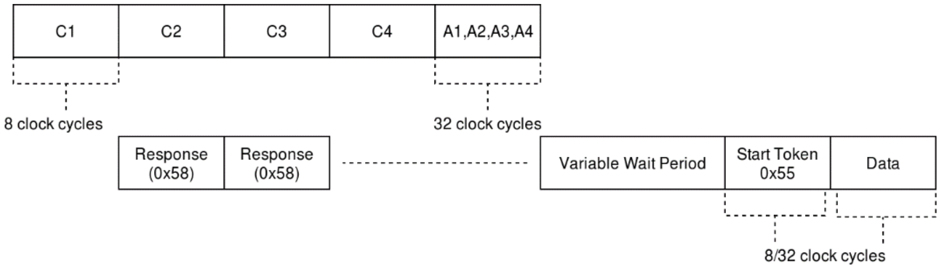


Figure 16.6. SPI Secondary AHB Primary Read

16.1.3.10 AHB Secondary Read

This is similar to the AHB Primary Read except that bit 3 of C1 is set and the Address phase (A1, A2, A3, and A4) is skipped.

16.1.3.11 SPI Interface's Internal Registers Read

This is similar to the AHB Primary Read except that bit 4 of C1 is cleared and A1, A2, A3, and A4 are skipped. The Variable Wait Period starts immediately after C4 is transferred.

16.1.3.12 AHB Primary/AHB Secondary/ Internal Register Write with 2-bit Length

This register is also similar to the AHB Primary Read except that bit 2 of C1 is cleared and C3 and C4 are both skipped. If it's an internal register or AHB secondary access, then A1, A2, A3, and A4 are also skipped after C2. The number of bytes to be transferred is inferred from bits 1:0 of C1.

16.1.4 Register Summary

Base Address: 0x2020_0000

Table 16.2. Register Summary Table

Register Name	Offset	Description	Access by MCU / SOC	Access by Host
SPI_HOST_INTR	0x0	SPI Host Interrupt Register	Yes	Yes
SPI_RFIFO_START	0x2	SPI RFIFO Start Level Register	Yes	Yes
SPI_RFIFO_AFULL_LEV	0x4	SPI RFIFO Almost Full Register	Yes	Yes
SPI_RFIFO_AEMPTY_LEV	0x6	SPI WFIFO Almost Empty Register	Yes	Yes
SPI_MODE	0x8	SPI Mode Register	Yes	Yes
SPI_INTR_STATUS	0xA	SPI Interrupt Status/Clear Register	Yes	Yes
SPI_INTR_EN	0xC	SPI Interrupt Enable Register	Yes	Yes
SPI_INTR_MASK	0xE	Interrupt Mask Register	Yes	Yes
SPI_INTR_UNMASK	0x10	SPI/MMIO Interrupt Unmask Register	Yes	Yes
SPI_LENGTH	0x12	SPI Length Register	Yes	Yes
SPI_COMMAND	0x14	SPI Command Register	Yes	Yes
SPI_DEV_ID	0x16	SPI Device ID Register	Yes	Yes
SPI_VER_NO	0x18	SPI Version Number Register	Yes	Yes
SPI_STATUS	0x1A	SPI Status Register	Yes	Yes
SPI_BUS_CONTROLLER_STATE	0x1C	SPI Bus Controller State Register	Yes	Yes
SPI_CONFIG_1	0x20	SPI Configuration 1 Register	Yes	Yes
SPI_CONFIG_2	0x22	SPI Configuration 2 Register	Yes	Yes
SPI_CONFIG_3	0x24	SPI Configuration 3 Register	Yes	Yes
SPI_CONFIG_4	0x26	SPI Configuration 4 Register	Yes	Yes
SPI_CONFIG_5	0x28	SPI Configuration 5 Register	Yes	Yes
SPI_CONFIG_6	0x2A	SPI Configuration 6 Register	Yes	Yes
SPI_CONFIG_7	0x2C	SPI Configuration 7 Register	Yes	Yes
SPI_CONFIG_8	0x2E	SPI Configuration 8 Register	Yes	Yes
SPI_SYS_RESET_REQ	0x3E	SPI Reset Register	No	Yes
SPI_WAKE_UP	0x3F	SPI Wake up Register	No	Yes
SPI_RFIFO_DATA	0x380 – 0x3BF	SPI RFIFO Data Register	Yes	Yes
SPI_WFIFO_DATA	0x3C0 – 0x3FF	SPI WFIFO Data Register	Yes	Yes

16.1.5 Register Description

Legend:
R = Read-only, W = Write-only, R/W = Read/Write

16.1.5.1 SPI Host Interrupt Register

Table 16.3. SPI_HOST_INTR Register Description

Bit	Access	Function	POR Value	Description
15:8	R	Reserved	0x0	Reserved
7:0	R/W	SPI_HOST_INTR	0x0	These bits indicate the interrupt vector value coming from system side.

16.1.5.2 SPI RFIFO Start Level Register

Table 16.4. SPI_RFIFO_ST Register Description

Bit	Access	Function	POR Value	Description
7:0	R/W	SPI_RFIFO_ST	0x10	These bits are used to program the minimum FIFO occupancy level before which data will not be sent out.

16.1.5.3 SPI RFIFO Almost Full Register

Table 16.5. SPI_RFIFO_AFULL_LEV Register Description

Bit	Access	Function	POR Value	Description
15:8	R	Reserved	0x0	Reserved
7:0	R/W	SPI_RFIFO_AFULL_LEV	0x40	These bits are used to program the FIFO occupancy level to trigger the Almost Full indication.

16.1.5.4 SPI WFIFO Almost Empty Register

Table 16.6. SPI_WFIFO_AEMPTY_LEV Register Description

Bit	Access	Function	POR Value	Description
15:8	R	Reserved	0x0	Reserved
7:0	R/W	SPI_RFIFO_AEMPTY_LEV	0x40	These bits are used to program the occupancy level to trigger the Almost Empty indication.

16.1.5.5 SPI Mode Register

Table 16.7. SPI_MODE Register Description

Bit	Access	Function	POR Value	Description
15:5	R	Reserved	0x0	Reserved for future use
4	R/W	EN_FOR_HIGH_CLK_RATIOS_FIX	0x0	This bit must be set when SOC clock to SPI clock ratio is high. If the host SPI and AHB clock do not satisfy the following equation, then this bit must be set. SPI clock period * (Number of clock cycles after C2 in first command + Number clock cycles in second command C1(8) - double ranking delay(2)) >= AHB clock period * (pulse synchro delay(3)) 0 - Disable the fix 1- Enable the fix
3	R/W	BYPASS_INIT	0x0	This bit is used to bypass the SPI initialization. 0 - Doesn't bypass 1 - Bypasses SPI initialization
2	R/W	VHS_EN	0x0	This bit is used to enable very high speed mode (120 MHz) '0' – Doesn't enable '1' – Enable
1	R/W	SPI_FIX_EN	0x1	This bit is used to enable the fix made for bus_ctrl_busy being asserted when success_state is being asserted, getting deasserted when FSM has decided to move to BUSY_STATE or not '0' – Doesn't enable the fix '1' – Enables the fix
0	R/W	SPI_OP_MODE	0x0	This bit is used to program the mode of working of SPI Interface '0'- Low speed mode '1'- High speed mode

16.1.5.6 SPI Interrupt Status/Clear Register

Table 16.8. SPI_INTR_STATUS Register Description

Bit	Access	Function	POR Value	Description
15:3	R	Reserved	0x0	Reserved for future use.
2	R/W	SPI_CS_DEASSERT	0x0	SPI_CS deasserted without transferring the correct number of bits. This can happen because of glitches in the clock or because of a wrong de-assertion of CS.
1	R/W	SPI_RD_REQ	0b0	Read request received
0	R/W	SPI_WR_REQ	0b0	Write request received

16.1.5.7 SPI Interrupt Enable Register

Table 16.9. SPI_INTR_EN Register Description

Bit	Access	Function	POR Value	Description
15:3	R	Reserved	0x0	Reserved for future use.
2	R/W	SPI_CS_DEASSERT_INT_EN	0x0	This bit is used to enable the interrupt due to wrong de-assertion of CS.
1	R/W	SPI_RD_INTR_EN	0x0	This bit is used to enable the read interrupt.
0	R/W	SPI_WR_INTR_EN	0x0	This bit is used to enable the write interrupt.

16.1.5.8 Interrupt Mask Register

Table 16.10. SPI_INTR_MASK Register Description

Bit	Access	Function	POR Value	Description
15:3	R	Reserved	0x0	Reserved for future use.
2	R/W	SPI_CS_DEASSERT_INT_MSK	0x0	This bit is used to mask the CS de-assertion interrupt. '1' – Mask interrupt
1	R/W	SPI_RD_INTR_MSK	0x0	This bit is used to mask the read interrupt. '1'- Mask interrupt
0	R/W	SPI_WR_INTR_MSK	0x0	This bit is used to mask the write interrupt. '1'- Mask interrupt

16.1.5.9 SPI / MMIO Interrupt Unmask Register

Table 16.11. SPI_INTR_UNMASK Register Description

Bit	Access	Function	POR Value	Description
15:3	R	Reserved	0x0	Reserved for future use.
2	R/W	SPI_CS_DEASSERT_INT_UNMSK	0x0	This bit is used to unmask the CS de-assertion interrupt '1' – Unmask interrupt
1	R/W	SPI_RD_INTR_UNMSK	0x0	This bit is used to unmask the read interrupt. '1'- Unmask interrupt
0	R/W	SPI_WR_INTR_UNMSK	0x0	This bit is used to unmask the write interrupt. '1'- Unmask interrupt

16.1.5.10 SPI Length Register

Table 16.12. SPI_LENGTH Register Description

Bit	Access	Function	POR Value	Description
15:0	R	SPI_LEN	0x0	These bits indicate the length of the transfer as transmitted in the Commands C3 and C4.

16.1.5.11 SPI Command Register

Table 16.13. SPI_COMMAND Register Description

Bit	Access	Function	POR Value	Description
15:8	R	SPI_C2	0x0	These bits store the received command C2.
7:0	R	SPI_C1	0x0	These bits store the received command C1.

16.1.5.12 SPI Device ID Register

Table 16.14. SPI_DEV_ID Register Description

Bit	Access	Function	POR Value	Description
15:0	R	SPI_DEV_ID	0x917	These bits store the Device ID information.

16.1.5.13 SPI Version Number Register

Table 16.15. SPI_VER_NO Register Description

Bit	Access	Function	POR Value	Description
15:8	R	Reserved	0x00	Reserved for future use.
7:0	R	SPI_VER_NO	0x1	These bits store the version number.

16.1.5.14 SPI Status Register

Table 16.16. SPI_STATUS Register Description

Bit	Access	Function	POR Value	Description
15:8	R	Reserved	0x0	Reserved for future use.
7	R	SPI_WFIFO_AFULL	0x0	This bit indicates if write FIFO is almost full. (Write from Host to SOC) '0'- write FIFO is not almost full '1'- write FIFO is almost full
6	R	SPI_WFIFO_FULL	0x0	This bit indicates if write FIFO is full. (Write from Host to SOC) '0'- write FIFO is not full '1'- write FIFO is full
5	R	SPI_RFIFO_AEMPTY	0x1	This bit indicates if read FIFO is almost empty. (Read from SOC to host) '0'- read FIFO is not almost empty '1'- read FIFO is empty
4	R	SPI_RFIFO_EMPTY	0x1	This bit indicates if read FIFO is empty. (Read from SOC to host) '0'- read FIFO is not empty '1'- read FIFO is empty
3	R	SPI_WFIFO_AEMPTY	0x1	This bit indicates if write FIFO is almost empty. '0'- Write FIFO is almost empty '1'- Write FIFO is not almost empty
2	R	SPI_WFIFO_EMPTY	0x1	This bit indicates if write FIFO is empty. '0'- Write FIFO is empty '1'- Write FIFO is not empty
1	R	SPI_RFIFO_AFULL	0x0	This bit indicates if the read FIFO is almost full. '1'- Almost full '0'- Not almost full
0	R	SPI_RFIFO_FULL	0x0	This bit indicates if the read FIFO is almost full. '1'- Almost full '0'- Not almost full

16.1.5.15 SPI Bus Controller State Register

Table 16.17. SPI_BC_STATE Register Description

Bit	Access	Function	POR Value	Description
15:14	R	Reserved	0x0	Reserved for future use.
13:0	R	SPI_BC	0x0	These bits indicate the Bus Controller FSM state. (One-Hot Coding) "0th bit" - BC_IDLE "1st bit" - BC_EN "2nd bit" - BC_CTRL_BUSY "3rd bit" - BC_INT_REG_RD "4th bit" - BC_INT_REG_FIFO_WR "5th bit" - BC_INT_REG_WR_WAIT "6th bit" - BC_INT_REG_FIFO_RD "7th bit" - BC_ INT_REG_WR "8th bit" - BC_AHB_MASTER "9th bit" - BC_ AHB_MASTER_WAIT "10th bit" - BC_AHB_SLAVE_WR_LEN "11th bit" - BC_AHB_SLAVE_WR_CMD "12th bit" - BC_AHB_SLAVE_WR_INTR "13th bit" - BC_AHB_SLAVE_WAIT

16.1.5.16 SPI_CONFIG_n

SPI_CONFIG_1 to SPI_CONFIG_8 are general purpose registers used by firmware and Host Driver.

Table 16.18. SPI_CONFIG_n Register Description

Bit	Access	Function	POR Value	Description
31:0	R/W	General purpose bits	0x0	These bits are used by firmware and Host Driver.

16.1.5.17 SPI SYS Reset Req Register

Table 16.19. SPI_SYS_RESET_REQ Register Description

Bit	Access	Function	POR Value	Description
15:1	R	Reserved	0x0	Reserved for future use.
0	R/W	SPI_SYS_RESET_REQ	0x0	When set generates system reset request to reset controller. This gets reset once, reset controller generates reset. Host shouldn't reset this bit. With this reset request, reset controller generates non por reset.

16.1.5.18 SPI Wakeup Register

Table 16.20. SPI_WAKE_UP Register Description

Bit	Access	Function	POR Value	Description
15:2	R	Reserved	0x0	Reserved for future use.
1	R/W	SPI_DEEP_SLEEP_ST	0x0	Deep Sleep Start - Indicates the device to enter Deep Sleep state for maximum power save
0	R/W	SPI_WAKEUP	0x0	Wakeup Interrupt - Interrupt for waking up the system from Deep Sleep.

16.1.5.19 SPI RFIFO Data Register

Table 16.21. SPI_RFIFO_DATA Register Description

Bit	Access	Function	POR Value	Description
31:0	R	SPI_RFIFO	0x0	These bits store the data received from the host

16.1.5.20 SPI WFIFO Data Register

Table 16.22. SPI_WFIFO_DATA Register Description

Bit	Access	Function	POR Value	Description
31:0	W	SPI_WFIFO	0x0	These bits are used to write, the data to be sent to the host.

16.2 SDIO Secondary

16.2.1 General Description

The Secure Digital I/O (SDIO) Secondary module implements the functionality of the SDIO card based on the SDIO specifications version 2.0, released by SD Association. During the normal initialization and interrogation of the card by the host, the card identifies itself as an SDIO card. The host software then obtains the card information in a tuple (linked list) format and determines if that card’s I/O function(s) are acceptable to activate. This decision is based on such parameters as power requirements or the availability of appropriate software drivers. If the card is acceptable, it is allowed to power up fully and start the I/O function(s) built into it. The salient SDIO Secondary features are described in the following section.

16.2.2 Features

- Full throughput with SDIO 1.2 as well as with SDIO 2.0
- Supports full-speed* and high speed modes
- Supports SD-1 bit and SD-4 bit modes
- Supports interrupt for host abort, CRC Error, CMD52 and CMD53 interrupts
- Supports single as well as multiple block transfers for CMD53 access
- Supports CMD52 while CMD53 data transfer is in progress
- Supports CMD52 Abort
- Supports Read Wait
- Does not support Suspend/Resume
- Supports system soft reset from host

There is a constraint on the minimum SoC clock relative to SDIO clock. SoC clock has to be a minimum half of SDIO clock. This constraint is due to the synchronization mechanism used between the SoC clock domain and SDIO clock domain.

16.2.3 Functional Description

The figure below illustrates the block diagram of the SDIO Secondary module. It contains Command and data control logic, WFIFO, RFIFO, Card Information Structure(CIS) registers, Card Common Control Registers (CCCR), Function Basic Registers (FBR), and AHB Primary and Secondary interfaces.

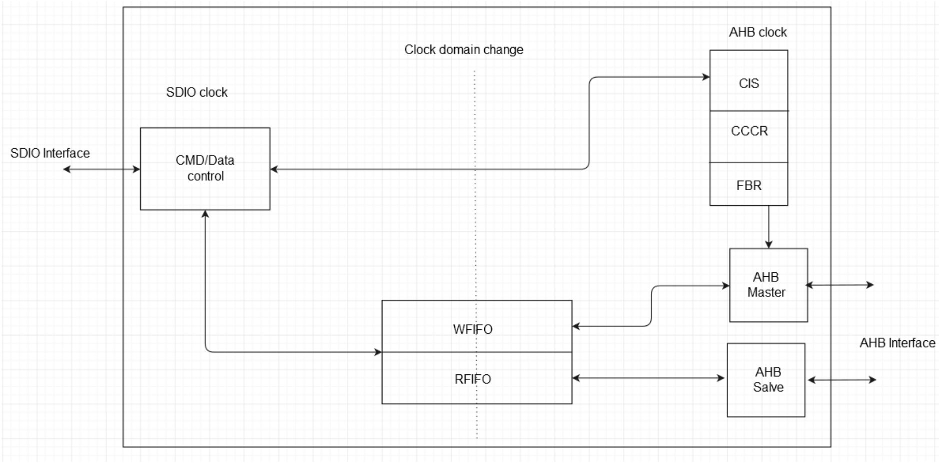


Figure 16.7. SDIO Secondary Block Diagram

16.2.3.1 SDIO Card Initialization

An SDIO card shall not cause non-I/O aware hosts to fail when inserted. To prevent operation of I/O functions in non-I/O aware hosts, a change to the SD card identification mode flowchart is needed. A new command (IO_SEND_OP_COND, CMD5) is added to replace the ACMD41 for SDIO initialization by I/O aware hosts. After reset or power-up, all I/O functions on the card are disabled and the I/O portion of the card shall not execute any operation except CMD5 or CMD0 with CS = low. If there is SD memory installed on the card (also called a combo card), that memory shall respond normally to all normal mandatory memory commands. An I/O only card shall not respond to the ACMD41 and thus appear initially as an MMC card.
The I/O only card shall also not respond to the CMD1 used to initialize the MMC cards and appear as a non-responsive card. The host then gives up and disables this card. Thus, the non-aware host receives no response from an I/O only card and forces it to the inactive state. The operation of an I/O card with a non-I/O aware host is shown in the following figure.

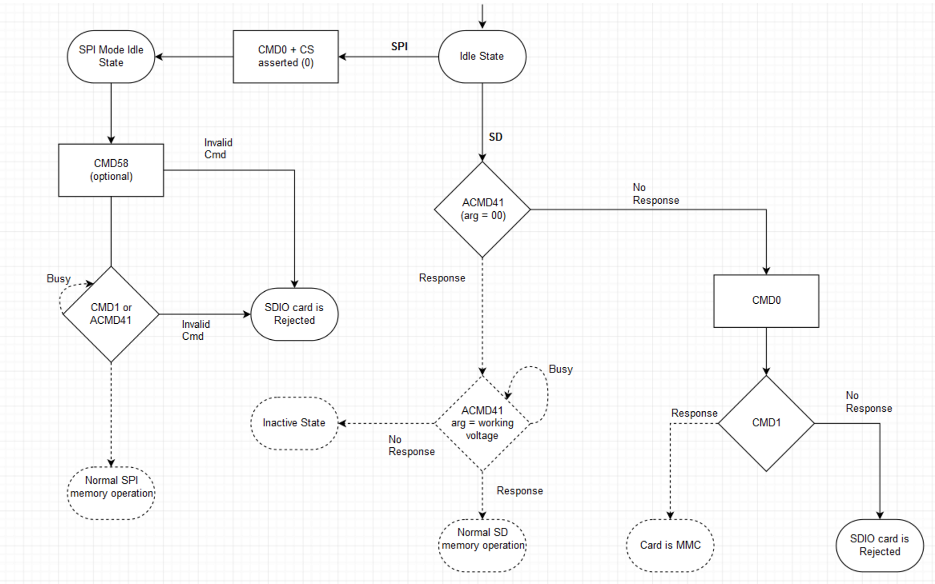
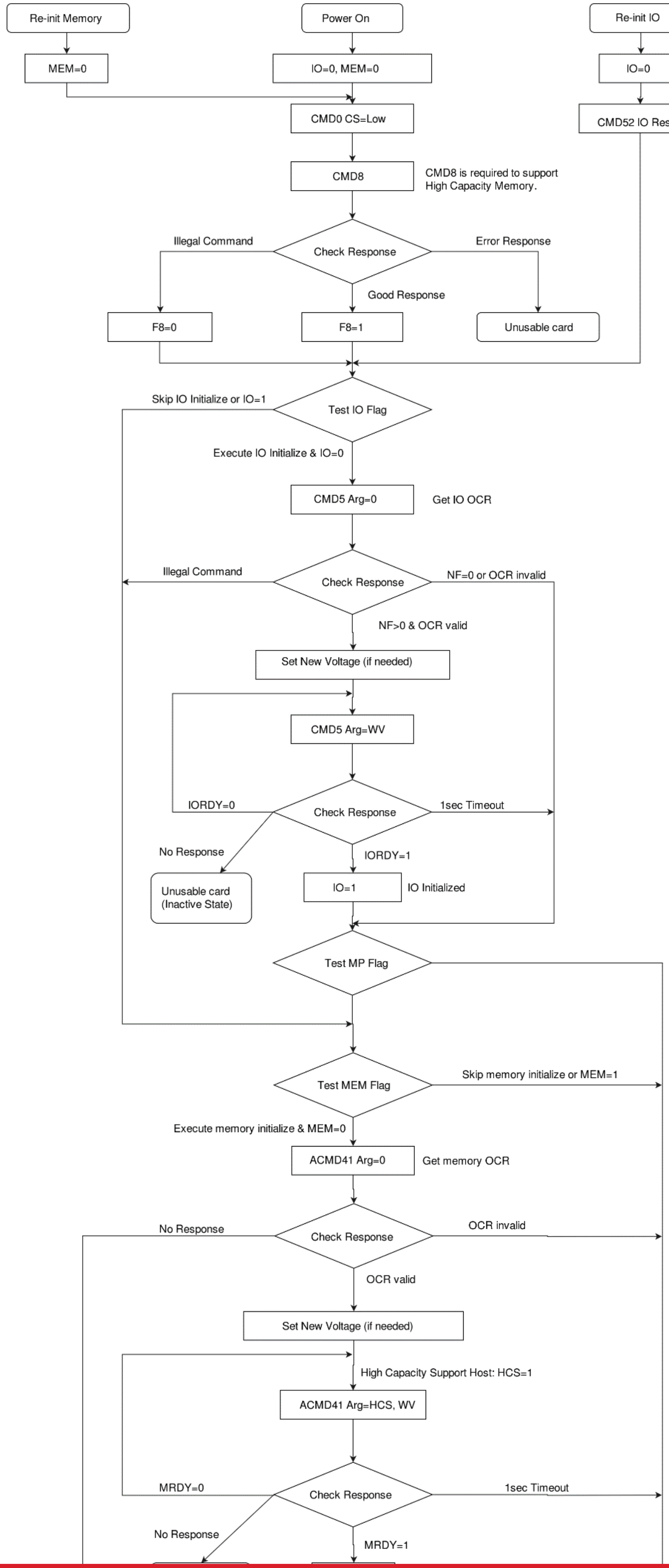


Figure 16.8. SDIO Response to Non-I/O Aware Initialization

An SDIO aware host sends CMD5 prior to the CMD55/ACMD41 pair, and thus would receive a valid OCR in the R4 response to CMD5 and continue to initialize the card. Figure 16.9 Card Initialization Flow in SD Mode (SDIO Aware Host) on page 331 shows the operation of an SDIO aware host operating in the SD modes. If the I/O portion of a card has received no CMD5, the I/O section remains inactive and shall not respond to any command except CMD5. A combo card stays in the memory-only mode. If no memory is installed on the card (i.e., an I/O only card in a non-SDIO aware host), the card would not respond to any memory command. This satisfies the condition where a user uses some I/O function on the card. The card is then removed and inserted into a non-SDIO aware host. That host would not enable the I/O function (no CMD5) so would appear to the player as a memory-only card. If the host were I/O aware, it would send the CMD5 to the card and the card would respond with R4. The host reads that R4 value and knows the number of available I/O functions and about the existence of any SD memory. After the host has initialized the I/O portion of the card, it then reads the Common Information Area (CIA) of the card. This is done by issuing a read command, starting with the byte at address 0x00, of I/O function 0. The CIA contains the Card Common Control Registers (CCCR) and the Function Basic Registers (FBR). Also included in the CIA are pointers to the card's common Card Information Structure (CIS) and each individual function's CIS. The CIS includes information on power, function, manufacturer, and other things the host needs to determine if the I/O function(s) is appropriate to power-up. If the host determines that the card should be activated, a register in the CCCR area enables the card and each individual function. At this time, all functions of the I/O card are fully available. In addition, the host can control the power consumption and enable/disable interrupts on a function-by-function basis. Combo Cards can accept CMD15 with RCA = 0000, but there is an exception for SD memory only cards. Memory only cards require a non-zero RCA before the host may issue CMD15. Thus, CMD15 shall be issued after CMD3 in the Standby state. In the case of ACMD41, it shall accept RCA = 0x0000.

As shown in the figures below, an SDIO aware host shall send CMD5 arg = 0 as part of the initialization sequence after either Power On or a CMD 52 with write to I/O Reset. Sending CMD5 arg = 0 that has not been preceded by one of these two reset conditions shall not result in either the host or card entering the initialization sequence.


Re-init Memory
Power On
Re-init IO
MEM=0
IO=0, MEM=0
IO=0
CMD0 CS=Low
CMD52 IO Reset
CMD8
CMD8 is required to support High Capacity Memory.
Illegal Command
Check Response
Error Response
Good Response
F8=0
F8=1
Unusable card
Skip IO Initialize or IO=1
Test IO Flag
Execute IO Initialize & IO=0
CMD5 Arg=0
Get IO OCR
Illegal Command
Check Response
NF=0 or OCR invalid
NF>0 & OCR valid
Set New Voltage (if needed)
CMD5 Arg=WV
IORDY=0
Check Response
1sec Timeout
No Response
IORDY=1
Unusable card (Inactive State)
IO=1
IO Initialized
Test MP Flag
Test MEM Flag
Skip memory initialize or MEM=1
Execute memory initialize & MEM=0
ACMD41 Arg=0
Get memory OCR
No Response
Check Response
OCR invalid
OCR valid
Set New Voltage (if needed)
High Capacity Support Host: HCS=1
ACMD41 Arg=HCS, WV
MRDY=0
Check Response
1sec Timeout
No Response
MRDY=1

SDIO cards may transfer data in either a multi-byte (1 to 256 bytes) or an optional block format. Any block size from 1 byte to 2048 bytes is possible in order to accommodate the various natural block sizes for I/O functions.

16.2.3.2 Block Mode

SDIO read or write operation shall be performed on a block basis, rather than the normal byte basis. If Block Mode bit in CMD53 is set, the Byte/Block count value shall contain the number of blocks to be read/written. The block size for functions is set by writing the block size to the I/O block size register in the FBR. The block size for function 0 is set by writing to the FN0 Block Size register in the CCCR. Card and host support of the block I/O mode is optional. The host can determine if a card supports block I/O by reading the card supports MBIO bit (SMB) in the CCCR. The block size used when Block Mode = 1 and the maximum byte count per command used when Block Mode = 0 can be read from the CIS in the tuple TPLFE_MAX_BLK_SIZE on a per-function basis.

SDIO Read (Multi-Block)

The figure below illustrates the SDIO bus timings for multi-block read operation. For multi block mode, block count is mentioned in CMD53. The host does not need to stop the transfer, as it continues until the block count is satisfied. If the block count is set to zero, the operation is identical to the memory mode in that the host must stop the transfer.

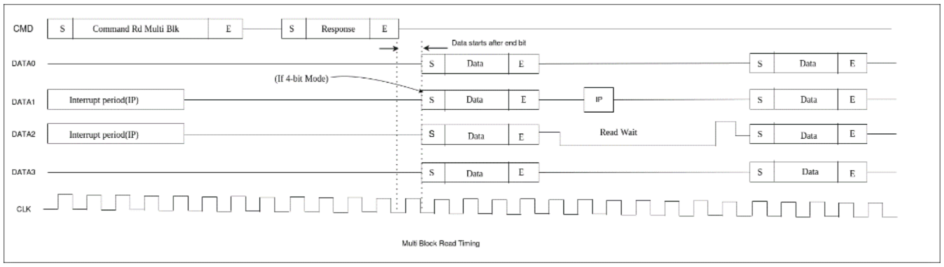


Figure 16.10. SDIO Multi-Block Read

SDIO Write (Multi-Block)

The figure below illustrates the SDIO bus timings for multi-block write operation. Multiple block write command shall be used rather than continuous single write command to make faster write operation.

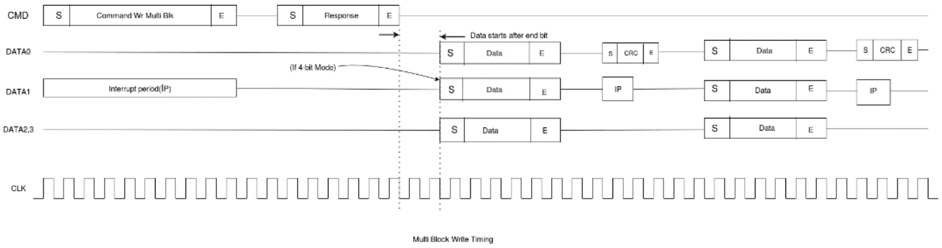


Figure 16.11. SDIO Multi-Block Write

Write Operation Flow Diagram

The figures below illustrates the example flow diagrams to be followed by the host driver to send a packet to the SDIO Secondary in non-block mode and block mode.

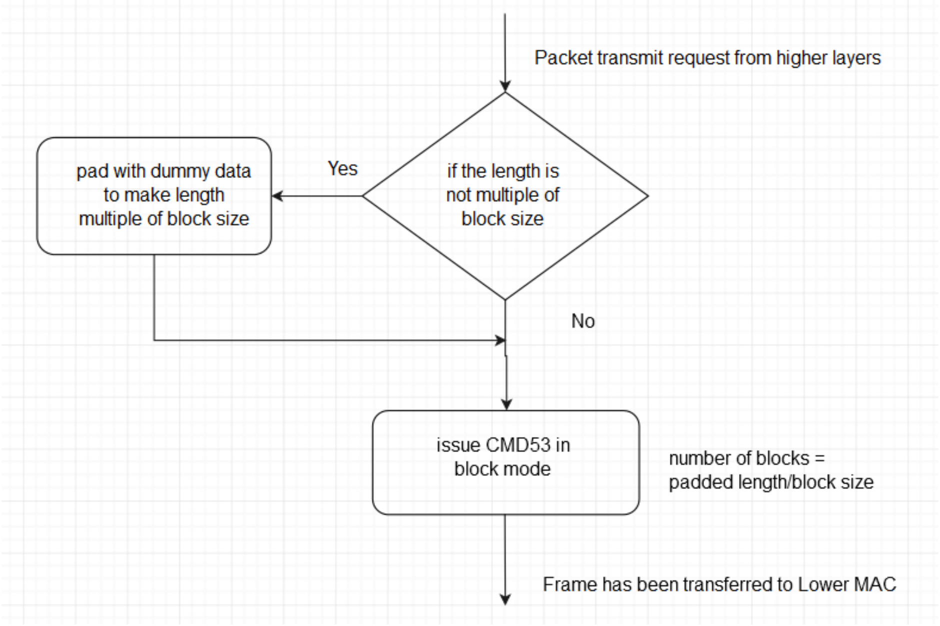

Packet transmit request from higher layers
if the length is not multiple of block size
Yes
pad with dummy data to make length multiple of block size
No
issue CMD53 in block mode
number of blocks = padded length/block size
Frame has been transferred to Lower MAC

Figure 16.12. SDIO Writing Packet in Block Mode

In Non-block mode, Block Mode bit is set to zero in CMD53. The size of the data payload is in the range of 1-256 bytes (due to FIFO size restriction in SDIO Secondary) in non-block mode. The byte count for this transfer is set in the command (CMD53), rather than the fixed block size.

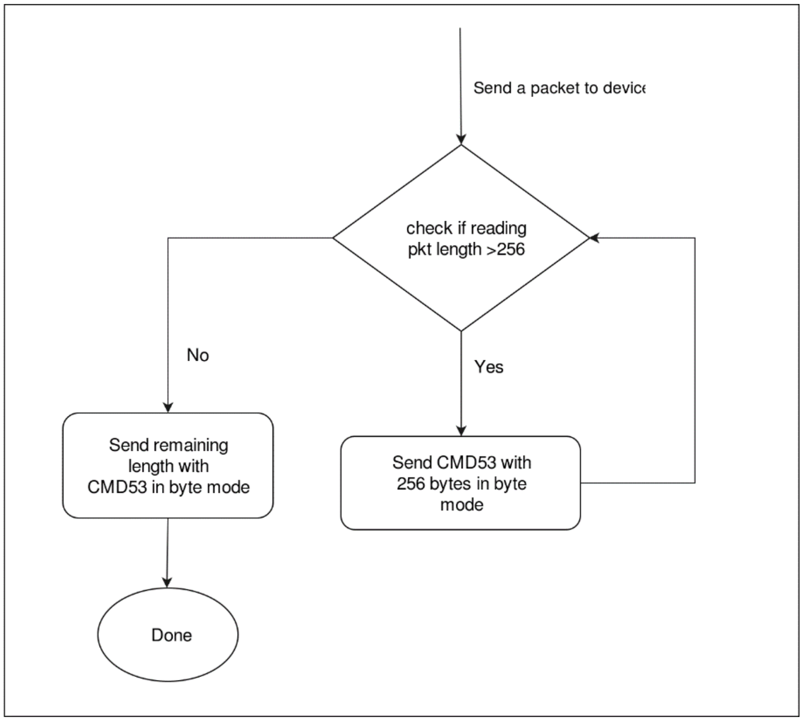

Send a packet to devic
check if reading pkt length >256
No
Yes
Send remaining length with CMD53 in byte mode
Send CMD53 with 256 bytes in byte mode
Done

Figure 16.13. SDIO Writing Packet in Non-block Mode

Read Operation Flow Diagram

The following figure illustrates the example flow diagrams to be followed by the Host Driver for reading packets from the SDIO Secondary in block mode.

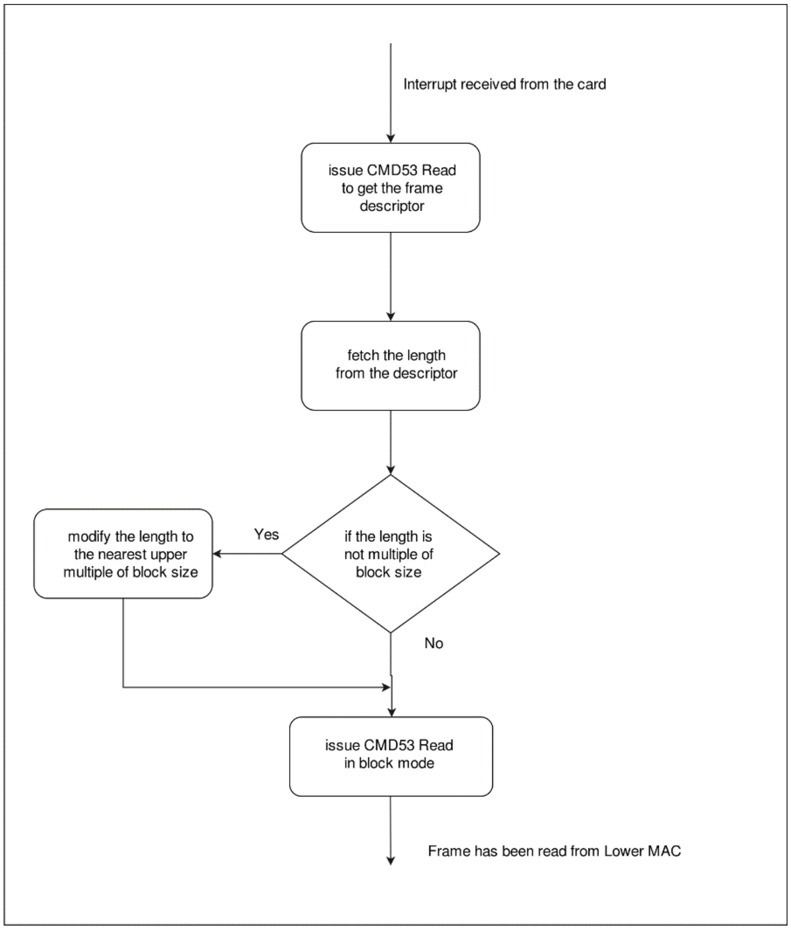

Interrupt received from the card
issue CMD53 Read to get the frame descriptor
fetch the length from the descriptor
if the length is not multiple of block size
Yes
modify the length to the nearest upper multiple of block size
No
issue CMD53 Read in block mode
Frame has been read from Lower MAC

Figure 16.14. SDIO Reading Packet in Block Mode

The following figure illustrates the example flow diagrams to be followed by the Host Driver for reading packets from the SDIO Secondary in non-block mode.

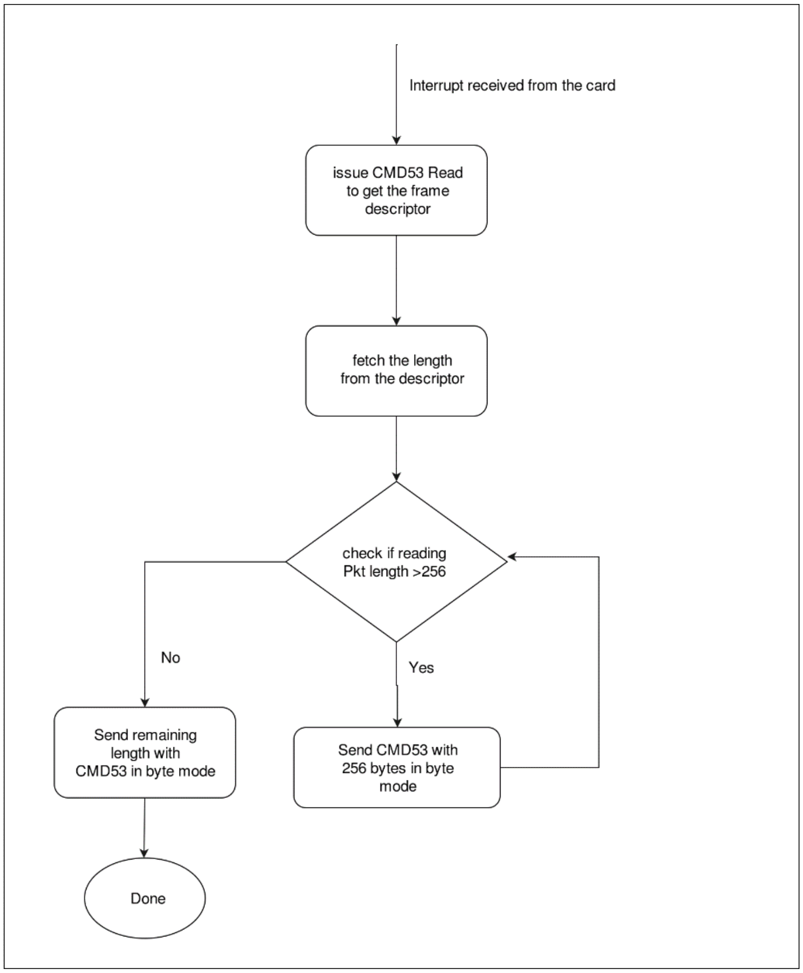

Interrupt received from the card
issue CMD53 Read to get the frame descriptor
fetch the length from the descriptor
check if reading Pkt length >256
No
Yes
Send remaining length with CMD53 in byte mode
Send CMD53 with 256 bytes in byte mode
Done

Figure 16.15. SDIO Reading Packet in Non-Block Mode

16.2.3.3 Address Mapping

The following table shows SDIO address mapping and the allowed commands in the given address space.

Table 16.23. SDIO Address Mapping

17-Bit Address Field	Function Number	CMD52 Access	CMD53 Access	Description
0x00000 – 0x1FFFF	0	Yes	No	SDIO specific registers
0x10000 – 0x1FFFF	1-5	Yes	Yes	AHB bus access will be done with lower 16-bits being picked up from 17-bit address and upper 16-bits being picked up from the fn0 registers
0x00000 – 0x0FFFF	1-5	Yes	Yes	An interrupt will be raised to the system side and data has to be read/written through AHB Secondary by the system side bus primaries.

16.2.4 Register Summary

Base Address: 0x2020_0000

The table below describes the vendor specific registers. For description about other registers SDIO standard 2.0 can be referred.

Table 16.24. SDIO Function 0 Vendor Specific Register Summary

Register Name	Offset	Description
OCR [7:0]	0x000F0	Operational Conditions Register[7:0]
OCR [15:8]	0x000F1	Operational Conditions Register[15:8]
OCR [23:16]	0x000F2	Operational Conditions Register[23:16]
—	0x000F3	Reserved
RD_NXT_DELAY1	0x000F4	Read Next Delay Register1
RD_NXT_DELAY2	0x000F5	Read Next Delay Register2
DEVICE_ID[7:0]	0x000F6	Device ID register
DEVICE_ID[15:8]	0x000F7	Device ID register
VER_NO	0x000F8	Version number register
INTR_STATUS_REG	0x000F9	Function 1 Interrupt Register
AHB_MASTER_ACC_ADDR_LSB	0x000FA	Master Access Address Least Significant Byte Register
AHB_MASTER_ACC_ADDR_MSB	0x000FB	Master Access Address Most Significant Byte Register
RFIFO_START_LEVEL	0x000FC	RFIFO Start Level Register
RFIFO_AFULL_LEVEL	0x000FD	RFIFO Almost Full Level Register
WFIFO_AEMPTY_LEVEL	0x000FE	WFIFO Almost Empty Level Register
WAKEUP_REG	0x000FF	Wakeup Register

Table 16.25. SDIO AHB Secondary Register Summary

Register Name	Offset	Description
SDIO_INTR_FN1_REG	0x00	SDIO Interrupt Function1 Register
SDIO_INTR_FN1_ENABLE_REG	0x04	SDIO Interrupt Function1 Enable Register
SDIO_INTR_FN1_MASK_REG	0x08	SDIO Interrupt Function1 Mask Register
SDIO_INTR_FN1_UNMASK_REG	0x0C	SDIO Interrupt Function1 Unmask Register
SDIO_BLK_LEN_REG	0x10	SDIO Block Length Register
SDIO_BLK_CNT_REG	0x14	SDIO Block Count Register
SDIO_ADDRESS_REG	0x18	SDIO Address Register
SDIO_CMD52_RDATA_REG	0x1C	SDIO Command52 Read Data Regsiter
SDIO_CMD52_WDATA_REG	0x20	SDIO Command52 Write Data Register
SDIO_INTR_REG	0x24	SDIO Interrupt Register
SDIO_INTR_FN_NUMER_REG	0x28	SDIO Interrupt Function Number Register
SDIO_FIFO_STATUS_REG	0x2C	SDIO FIFO Status Register
SDIO_FIFO_OCC_REG	0x30	SDIO FIFO Occupancy Register

Register Name	Offset	Description
SDIO_HOST_INTR_SET_REG	0x34	SDIO Host Interrupt Set Register
SDIO_HOST_INTR_CLEAR_REG	0x38	SDIO Host Interrupt Clear Register
SDIO_RFIFO_DATA_REG	0x40 – 0x7E	SDIO Read FIFO Data Register
SDIO_WFIFO_DATA_REG	0x80 – 0xBE	SDIO Write FIFO Data Register
SDIO_INTR_FN2_REG	0xC0	SDIO Interrupt Function2 Register
SDIO_INTR_FN2_ENABLE_REG	0xC4	SDIO Interrupt Function2 Enable Register
SDIO_INTR_FN2_MASK_REG	0xC8	SDIO Interrupt Function2 Mask Register
SDIO_INTR_FN2_UNMASK_REG	0xCC	SDIO Interrupt Function2 Unmask Register
SDIO_INTR_FN3_REG	0xD0	SDIO Interrupt Function3 Register
SDIO_INTR_FN3_ENABLE_REG	0xD4	SDIO Interrupt Function3 Enable Register
SDIO_INTR_FN3_MASK_REG	0xD8	SDIO Interrupt Function3 Mask Register
SDIO_INTR_FN3_UNMASK_REG	0xDC	SDIO Interrupt Function3 Unmask Register
SDIO_INTR_FN4_REG	0xE0	SDIO Interrupt Function4 Register
SDIO_INTR_FN4_ENABLE_REG	0xE4	SDIO Interrupt Function4 Enable Register
SDIO_INTR_FN4_MASK_REG	0xE8	SDIO Interrupt Function4 Mask Register
SDIO_INTR_FN4_UNMASK_REG	0xEC	SDIO Interrupt Function4 Unmask Register
SDIO_INTR_FN5_REG	0xF0	SDIO Interrupt Function5 Register
SDIO_INTR_FN5_ENABLE_REG	0xF4	SDIO Interrupt Function5 Enable Register
SDIO_INTR_FN5_MASK_REG	0xF8	SDIO Interrupt Function5 Mask Register
SDIO_INTR_FN5_UNMASK_REG	0xFC	SDIO Interrupt Function5 Unmask Register
SDIO_ERROR_COND_CTRL_ENABLE_REG	0x100	SDIO Error Condition Control Enable Register
SDIO_ERROR_COND_BLK_CNT	0x104	SDIO Error Condition Block Count Register
SDIO_BOOT_CONFIG_VALS_0	0x108	SDIO Boot Config Vals 0 Register
SDIO_BOOT_CONFIG_VALS_1	0x10C	SDIO Boot Config Vals 1 Register
	Fn0 Registers	
SDIO CCCR Registers	0x200 – 0x23E	CCCR Registers (0x0000 – SDIO space address)
SDIO Vendor Unique Registers	0x240 – 0x25E	Vendor Specific Registers (0x00F0)
SDIO Function1 FBR Registers	0x260 – 0x27E	Function1 FBR Registers (0x0100)
SDIO Function2 FBR Registers	0x280 – 0x29E	Function2 FBR Registers (0x0200)
SDIO Function3 FBR Registers	0x2A0 – 0x2BE	Function3 FBR Registers (0x0300)
SDIO Function4 FBR Registers	0x2C0 – 0x2DE	Function4 FBR Registers (0x0400)
SDIO Function5 FBR Registers	0x2E0 – 0x2FE	Function5 FBR Registers (0x0500)
SDIO CIS Register	0x300 – 0x4FE	CIS Registers (0x1000)

16.2.5 Register Description

16.2.5.1 Function0 Registers

OCR [7:0]

Table 16.26. SDIO Operational Conditions Register[7:0] Description

Bit	Access	Function	POR Value	Description
[7:0]	R/W	SDIO_OCR1	0x00/0x80	OCR – Operational conditions register indicates the card operating conditions which will be read by the host through CMD5

OCR[15:8]/vendor_specific_reg0

Table 16.27. SDIO Operational Conditions Register [15:8] Description

Bit	Access	Function	POR Value	Description
[7:0]	R	SDIO_OCR2	0x80/0x00	OCR [15:8] This register can be written only during bootload configuration After bootup configuration, this register will act as vendor_specific_reg0

OCR [23:16]/vendor_specific_reg1

Table 16.28. SDIO Operational Conditions Register [23:16] Description

Bit	Access	Function	POR Value	Description
[7:0]	R/W	SDIO_OCR3	0xFF/0x00	OCR [23:16] - This register can be written only during bootload configuration . After bootup configuration, this register will act as vendor_specific_reg1.

RD_NXT_DELAY1

Table 16.29. Read Next Delay Register 1 Description

Bit	Access	Function	POR Value	Description
[7:0]	R/W	SDIO_RD_NXT_DELAY1	0xC8	Minimum delay in terms of number of SDIO clocks between read blocks (SDIO_RD_NXT_DELAY [7:0])

RD_NXT_DELAY2

Table 16.30. Read Next Delay Register 2 Description

Bit	Access	Function	POR Value	Description
7	R	Reserved	0x0	Reserved for future use.
6	R/W	SDIO_PASS_ASYNC_INTR_EN	0x0	If this bit is set, asynchronous SDIO interrupt is passed

Bit	Access	Function	POR Value	Description
5	R/W	SDIO_WRPTR_INCR_32BIT	0x0	If this bit is set, FIFO write pointer is incremented by 2 once 32 bits of data is received. (Usually set in sleep mode during which SoC clock will be lower than host clock)
4	R/W	SDIO_CSA_THROUGH-Slave	0x0	This bit is used to enable CSA through AHB Secondary
3	R/W	SDIO_CRC_STATUS_DIS	0x0	SDIO Write CRC error status disable 0 – CRC status enabled 1 – CRC status disabled If the driver is exercising this option, it has to be done preferably after loading the instructions and templates.
2	R/W	SDIO_RD_NXT_DELAY_EN	0x0	Enable for introducing minimum delay between consecutive read blocks
[1:0]	R/W	SDIO_RD_NXT_DELAY2	0x0	Minimum delay in terms of number of SDIO clocks between read blocks (SDIO_RD_NXT_DELAY [9:8])

DEVICE_ID[7:0]

Table 16.31. Device ID Description

Bit	Access	Function	POR Value	Description
[7:0]	R	SDIO_DEVID	0x17	This indicates the Device ID

DEVICE_ID[15:8]

Table 16.32. Device ID Description

Bit	Access	Function	POR Value	Description
[15:8]	R	SDIO_DEVID	0x9	This indicates the Device ID

VER_NO

Table 16.33. Version Number Register Description

Bit	Access	Function	POR Value	Description
[7:0]	R	SDIO_VER_NO	0x1	This indicates the version number of the device

INTR_STATUS_REG

Table 16.34. Function 1 Interrupt Register Description

Bit	Access	Function	POR Value	Description
[7:0]	R/W	SDIO_FN1_INTR	0x00	The Function1 Interrupt status from the system

AHB_MASTER_ACC_ADDR_LSB

Table 16.35. Master Access Address Least Significant Byte Register Description

Bit	Access	Function	POR Value	Description
[7:0]	R/W	SDIO_MASTER_ADR_LSB	0x00	Bits 23 to 16 of the AHB Primary access address

AHB_MASTER_ACC_ADDR_MSB

Table 16.36. Master Access Address Most Significant Byte Register Description

Bit	Access	Function	POR Value	Description
7:0]	R/W	SDIO_MASTER_ADR_MSB	0x00	Bits 31 to 24 of the AHB Primary access address

RFIFO_START_LEVEL

Table 16.37. RFIFO Start Level Register Description

Bit	Access	Function	POR Value	Description
[7:0]	R/W	SDIO_RFIFO_START_LEV	0x04	Minimum FIFO occupancy level before which data will not be read out. This has to be specified in number of half words (16-bit).

RFIFO_AFULL_LEVEL

Table 16.38. RFIFO Almost Full Level Register Description

Bit	Access	Function	POR Value	Description
[7:0]	R/W	SDIO_RFIFO_AFULL_LEV	0x08	If the read FIFO occupancy is greater than this level, read FIFO's almost full signal will be asserted. This has to be specified in number of half words (16-bit).

WFIFO_AEMPTY_LEVEL

Table 16.39. WFIFO Almost Empty Level Register Description

Bit	Access	Function	POR Value	Description
[7:0]	R/W	SDIO_WFIFO_AEMPTY_LEV	0x08	If the write FIFO occupancy is less than this level, write FIFO's almost empty signal will be asserted. This has to be specified in number of half words (16-bit).

WAKEUP_REG

Table 16.40. Wake up Register Description

Bit	Access	Function	POR Value	Description
[7:1]	R	Reserved	0x00	Reserved
0	W	SDIO_WAKEUP_INT	0x0	Wakeup Interrupt – Writing '1' into this register will give wakup interrupt to the sleep fsm. This is a self clearing bit

SDIO_INTR_FN1_REGISTER

Table 16.41. SDIO Function1 Interrupt Status/Clear Register Description

Bit	Access	Function	POR Value	Description
[31:10]	R/W	Reserved	0x0	Reserved
9	R/W	SDIO_CSA_ACCESS	0x0	csa_window_access When set, indicates that current request is for CSA window register. This is only a status signal.
8	R/W	SDIO_WR_RDz	0x0	wr_rdz 0 – read request 1 – write request This is not an interrupt signal. This is only a status signal.
7	R	SDIO_RD_TOUT_INT	0x0	When set, indicates that Read FIFO hasn't reached minimum threshold value before read wait timeout.
6	R/W	SDIO_ABORT_ INT	0x0	When set, indicates that host issued an abort command.
5	R/W	SDIO_CRC_ ERR_INT	0x0	When set, indicates that data block is received with crc error.
4	R/W	SDIO_PWR_LEV_ INT	0x0	When set, indicates that power control register value has been changed by the host.
3	R/W	SDIO_CMD52_ INT	0x0	When set, indicates that CMD52 is received.
2	R/W	SDIO_CSA_ INT	0x0	When set, indicates that CMD53 request is received to CSA.
1	R/W	SDIO_RD_INT	0x0	When set, indicates that CMD53 read request is received.
0	R/W	SDIO_WR_INT	0x0	When set, indicates that CMD53 write request is received.

SDIO_INTR_FN1_ENABLE_REGISTER

Table 16.42. SDIO Function1 Interrupt Enable Register Description

Bit	Access	Function	POR Value	Description
[31:8]	R/W	Reserved	0x0	Reserved
7	R/W	SDIO_TOUT_INT_EN	0x0	This bit is used to enable "read FIFO wait time over" interrupt. '1'- Interrupt is enabled '0'- Interrupt is disabled
6	R/W	SDIO_ABORT_ INT_EN	0x0	This bit is used to enable abort interrupt. '1'- Interrupt is enabled '0'- Interrupt is disabled

Bit	Access	Function	POR Value	Description
5	R/W	SDIO_CRC_ ERR_INT_EN	0x0	This bit is used to enable CRC error interrupt. '1'- Interrupt is enabled '0'- Interrupt is disabled
4	R/W	SDIO_PWR_LEV_ INT_EN	0x0	This bit is used to enable power level change interrupt. '1'- Interrupt is enabled '0'- Interrupt is disabled
3	R/W	SDIO_CMD52_ INT_EN	0x0	This bit is used to enable CMD52 interrupt. '1'- Interrupt is enabled '0'- Interrupt is disabled
2	R/W	SDIO_CSA_ INT_EN	0x0	This bit is used to enable CMD53 CSA interrupt.
1	R/W	SDIO_RD_INT_EN	0x0	This bit is used to enable CMD53 read interrupt. '1'- Interrupt is enabled '0'- Interrupt is disabled
0	R/W	SDIO_WR_INT_EN	0x0	This bit is used to enable CMD53 write interrupt. '1'- Interrupt is enabled '0'- Interrupt is disabled

SDIO_INTR_FN1_MASK_REGISTER

Table 16.43. SDIO Function1 Interrupt Mask Register Description

Bit	Access	Function	POR Value	Description
[31:8]	R/W	Reserved	0x0	Reserved
7	R/W	SDIO_TOUT_MSK	0x0	This bit is used to mask "read FIFO wait time over" interrupt. Setting this bit will mask the interrupt. Clearing this bit has no effect.
6	R/W	SDIO_ABORT_MSK	0x0	This bit is used to mask abort interrupt. Setting this bit will mask the interrupt. Clearing this bit has no effect.
5	R/W	SDIO_CRC_ERR_MSK	0x0	This bit is used to mask CRC error interrupt. Setting this bit will mask the interrupt. Clearing this bit has no effect.

Bit	Access	Function	POR Value	Description
4	R/W	SDIO_PWR_LEV_MSK	0x0	This bit is used to mask power level change interrupt. Setting this bit will mask the interrupt. Clearing this bit has no effect.
3	R/W	SDIO_CMD52_MSK	0x0	This bit is used to mask CMD52 interrupt. Setting this bit will mask the interrupt. Clearing this bit has no effect.
2	R/W	SDIO_CSA_MSK	0x0	This bit is used to mask CMD53 CSA interrupt. Setting this bit will mask the interrupt. Clearing this bit has no effect.
1	R/W	SDIO_RD_INT_MSK	0x0	This bit is used to mask CMD53 read interrupt. Setting this bit will mask the interrupt. Clearing this bit has no effect.
0	R/W	SDIO_WR_INT_MSK	0x0	This bit is used to mask CMD53 write interrupt. Setting this bit will mask the interrupt. Clearing this bit has no effect.

SDIO_INTR_FN1_UNMASK_REGISTER

Table 16.44. SDIO Function1 Interrupt Unmask Register Description

Bit	Access	Function	POR Value	Description
[31:8]	R/W	Reserved	0x0	Reserved
7	R/W	SDIO_TOUT_UNMSK	0x0	This bit is used to unmask "read FIFO wait time over" interrupt. Setting this bit will unmask the interrupt. Clearing this bit has no effect.
6	R/W	SDIO_ABORT_ UNMSK	0x0	This bit is used to unmask abort interrupt. Setting this bit will unmask the interrupt. Clearing this bit has no effect.
5	R/W	SDIO_CRC_ERR_UNMSK	0x0	This bit is used to unmask CRC error interrupt. Setting this bit will unmask the interrupt. Clearing this bit has no effect.

Bit	Access	Function	POR Value	Description
4	R/W	SDIO_PWR_LEV_UNMSK	0x0	This bit is used to unmask power level change interrupt. Setting this bit will unmask the interrupt. Clearing this bit has no effect.
3	R/W	SDIO_CMD52_UNMSK	0x0	This bit is used to unmask CMD52 interrupt. Setting this bit will unmask the interrupt. Clearing this bit has no effect.
2	R/W	SDIO_CSA_UNMSK	0x0	This bit is used to unmask CMD53 CSA interrupt. Setting this bit will unmask the interrupt. Clearing this bit has no effect.
1	R/W	SDIO_RD_INT_UNMSK	0x0	This bit is used to unmask CMD53 read interrupt. Setting this bit will unmask the interrupt. Clearing this bit has no effect.
0	R/W	SDIO_WR_INT_UNMSK	0x0	This bit is used to unmask CMD53 write interrupt. Setting this bit will unmask the interrupt. Clearing this bit has no effect.

SDIO_BLK_LEN_REGISTER

Table 16.45. SDIO Block Length Register Description

Bit	Access	Function	POR Value	Description
[31:12]	R	Reserved	0x0	Reserved for future use.
[11:0]	R	SDIO_BLK_LEN	0x0	Length of each block for the last received CMD53.

SDIO_BLK_CNT_REGISTER

Table 16.46. SDIO Block Count Register Description

Bit	Access	Function	POR Value	Description
[31:9]	R	Reserved	0x0	Reserved for future use.
[8:0]	R	SDIO_BLK_CNT	0x0	Block count for the last received CMD53.

SDIO_ADDRESS_REGISTER

Table 16.47. SDIO Address Register Description

Bit	Access	Function	POR Value	Description
[31:16]	R	Reserved	0x0	Reserved
[15:0]	R	SDIO_ADDR	0x0	Lower 16-bits of the 17-bit address field in the last received CMD53.

SDIO_CMD52_RDATA_REGISTER

Table 16.48. SDIO CMD52 RDATA Register Description

Bit	Access	Function	POR Value	Description
[31:8]	R	Reserved	0x0	Reserved for future use.
[7:0]	R	RDATA	0x0	Data to be given to host for CMD52. Secondary mode access read command has to be written into this register.

SDIO_CMD52_WDATA_REGISTER

Table 16.49. SDIO CMD52 WDATA Register Description

Bit	Access	Function	POR Value	Description
[31:8]	R	Reserved	0x0	Reserved for future use.
[7:0]	R	WDATA	0xFF	Data from host in CMD52. Secondary mode access write command is available in this register.

SDIO_INTR_REGISTER

Table 16.50. SDIO Interrupt Status Register Description

Bit	Access	Function	POR Value	Description
[31:6]	R	Reserved	0x0	Reserved for future use.
5	R	SDIO_INT_FN5	0x0	Interrupt is pending for function5.
4	R	SDIO_INT_FN4	0x0	Interrupt is pending for function4.
3	R	SDIO_INT_FN3	0x0	Interrupt is pending for function3.
2	R	SDIO_INT_FN2	0x0	Interrupt is pending for function2.
1	R	SDIO_INT_FN1	0x0	Interrupt is pending for function1.
0	R	SDIO_INT_ERROR	0x0	Interrupt is pending because of error condition from any of the functions.

SDIO_INTR_FN_NUMER_REGISTER

Table 16.51. SDIO Interrupt Function Number Register Description

Bit	Access	Function	POR Value	Description
[31:3]	R	Reserved	0x0	Reserved

Bit	Access	Function	POR Value	Description
[2:0]	R	SDIO_INTR_FUN_NO	0x0	Indicates the function number to which interrupt is pending. This register is provided to enable the software to decode the interrupt register easily. Once this interrupt is cleared, this register gets the next function number to which interrupt is pending (if simultaneous interrupts are pending). 2 – function 2 3 – function 3 4 – function 4 5 – function 5 0 – there is no pending interrupt There are two interrupt lines coming out of SDIO. One line is dedicate to function 1. Remaining function interrupts will come on second line. Since function 1 has dedicate interrupt line, pending of the same is not mapped into this register

SDIO_FIFO_STATUS_REGISTER

Table 16.52. SDIO FIFO Status Register Description

Bit	Access	Function	POR Value	Description
[31:12]	R		0x0	Reserved
[11:7]	R	SDIO_BUS_CONTROL_STATE	0x1	SDIO bus control state 1– When set, indicates FSM is in idle state 2– When set, indicates FSM is in CMD52 read state 4– When set, indicates FSM is in CMD52 write state 8– When set, indicates FSM is CMD53 read state 16 – When set, indicates FSM is CMD53 write state
[6:4]	R	SDIO_CURRENT_FN_NO	0x0	Indicates the function number of the last received command.
3	R	SDIO_RFIFO_AEMPTY	0x0	When set, indicates that RFIFO is almost empty.
2	R	SDIO_RFIFO_EMPTY	0x0	When set, indicates that RFIFO is empty. RFIFO is used in SDIO writes from host for sending data from host to AHB.
1	R	SDIO_WFIFO_AFULL	0x0	When set, indicates that WFIFO is almost full.
0	R	SDIO_WFIFO_FULL	0x0	When set, indicates that WFIFO is full. WFIFO is used in SDIO reads from host for sending data from AHB to Host.

SDIO_FIFO_OCC_REGISTER

Table 16.53. SDIO FIFO Occupancy Register Description

Bit	Access	Function	POR Value	Description
[31:16]	R	Reserved	0x0	Reserved
[15:8]	R	SDIO_RFIFO_AVAIL	0x80	Indicates the available space in the read FIFO.
[7:0]	R	SDIO_WFIFO_OCC	0x0	Indicates the occupancy level of the write FIFO.

SDIO_HOST_INTR_SET_REGISTER

Table 16.54. SDIO Host Interrupt Set Register Description

Bit	Access	Function	POR Value	Description
[31:4]	R/W	Reserved	0x0	Reserved for future use
3	R/W	SDIO_INTSET_FN5	0x0	This bit is used to raise an interrupt to host for function5. Setting this bit will raise the interrupt. Clearing this bit has no effect.
2	R/W	SDIO_INTSET_FN4	0x0	This bit is used to raise an interrupt to host for function4. Setting this bit will raise the interrupt. Clearing this bit has no effect.
1	R/W	SDIO_INTSET_FN3	0x0	This bit is used to raise an interrupt to host for function3. Setting this bit will raise the interrupt. Clearing this bit has no effect.
0	R/W	SDIO_INTSET_FN2	0x0	This bit is used to raise an interrupt to host for function2. Setting this bit will raise the interrupt. Clearing this bit has no effect.

SDIO_HOST_INTR_CLEAR_REGISTER

Table 16.55. SDIO Host Interrupt Clear Register Description

Bit	Access	Function	POR Value	Description
[31:4]	R/W	Reserved	0x0	Reserved for future use.
3	R/W	SDIO_INTCLR_FN5	0x0	This bit is used to clear the interrupt to host for function5. Setting this bit will clear the interrupt. Clearing this bit has no effect.
2	R/W	SDIO_INTCLR_FN4	0x0	This bit is used to clear the interrupt to host for function4. Setting this bit will clear the interrupt. Clearing this bit has no effect.

Bit	Access	Function	POR Value	Description
1	R/W	SDIO_INTCLR_FN3	0x0	This bit is used to clear the interrupt to host from function3. Setting this bit will clear the interrupt. Clearing this bit has no effect.
0	R/W	SDIO_INTCLR_FN2	0x0	This bit is used to clear the interrupt to host for function2. Setting this bit will clear the interrupt. Clearing this bit has no effect.

SDIO_RFIFO_DATA

Table 16.56. SDIO Read FIFO Data Register Description

Bit	Access	Function	POR Value	Description
[31:0]	W	SDIO_RFIFO	0x0	Data to be written into SDIO Read FIFO has to be written in this register.

SDIO_WFIFO_DATA

Table 16.57. SDIO Write FIFO Data Register Description

Bit	Access	Function	POR Value	Description
[31:0]	R	SDIO_WFIFO	0x0	SDIO Write FIFO data can be read through this register.

SDIO_INTR_FN2_REGISTER

Table 16.58. SDIO Function2 Interrupt Status/Clear Register Description

Bit	Access	Function	POR Value	Description
[31:10]	R/W	Reserved	0x0	Reserved
9	R/W	SDIO_CSA_ACCESS	0x0	csa_window_access When set, indicates that current request is for CSA window register. This is only status signal.
8	R/W	SDIO_WR_RDz	0x0	wr_rdz 0 – read request 1 – write request This is not an interrupt signal. This is only status signal.
7	R/W	SDIO_RD_TOUT_INT	0x0	When set, indicates that Read FIFO hasn't reached minimum threshold value before read wait timeout.
6	R/W	SDIO_ABORT_ INT	0x0	When set, indicates that host issued an abort command.
5	R/W	SDIO_CRC_ ERR_INT	0x0	When set, indicates that data block is received with crc error.
4	R/W	SDIO_PWR_LEV_ INT	0x0	When set, indicates that power control register value has been changed by the host.
3	R/W	SDIO_CMD52_ INT	0x0	When set, indicates that CMD52 is received.
2	R/W	SDIO_CSA_ INT	0x0	When set, indicates that CMD53 request is received to CSA.

Bit	Access	Function	POR Value	Description
1	R/W	SDIO_RD_INT	0x0	When set, indicates that CMD53 read request is received.
0	R/W	SDIO_WR_INT	0x0	When set, indicates that CMD53 write request is received.

SDIO_INTR_FN2_ENABLE_REGISTER

Table 16.59. SDIO Function2 Interrupt Enable Register Description

Bit	Access	Function	POR Value	Description
[31:8]	R/W	Reserved	0x0	Reserved
7	R/W	SDIO_TOUT_INT_EN	0x0	This bit is used to enable "read FIFO wait time over" interrupt. '1'- Interrupt is enabled '0'- Interrupt is disabled
6	R/W	SDIO_ABORT_ INT_EN	0x0	This bit is used to enable abort interrupt. '1'- Interrupt is enabled '0'- Interrupt is disabled
5	R/W	SDIO_CRC_ ERR_INT_EN	0x0	This bit is used to enable CRC error interrupt. '1'- Interrupt is enabled '0'- Interrupt is disabled
4	R/W	SDIO_PWR_LEV_ INT_EN	0x0	This bit is used to enable power level change interrupt. '1'- Interrupt is enabled '0'- Interrupt is disabled
3	R/W	SDIO_CMD52_ INT_EN	0x0	This bit is used to enable CMD52 interrupt. '1'- Interrupt is enabled '0'- Interrupt is disabled
2	R/W	SDIO_CSA_ INT_EN	0x0	This bit is used to enable CMD53 CSA interrupt.
1	R/W	SDIO_RD_INT_EN	0x0	This bit is used to enable CMD53 read interrupt. '1'- Interrupt is enabled '0'- Interrupt is disabled
0	R/W	SDIO_WR_INT_EN	0x0	This bit is used to enable CMD53 write interrupt. '1'- Interrupt is enabled '0'- Interrupt is disabled

SDIO_INTR_FN2_MASK_REGISTER

Table 16.60. SDIO Function2 Interrupt Mask Register Description

Bit	Access	Function	POR Value	Description
[31:8]	R/W	Reserved	0x0	Reserved
7	R/W	SDIO_TOUT_MSK	0x0	This bit is used to mask "read FIFO wait time over" interrupt. Setting this bit will mask the interrupt. Clearing this bit has no effect.
6	R/W	SDIO_ABORT_MSK	0x0	This bit is used to mask abort interrupt Setting this bit will mask the interrupt. Clearing this bit has no effect.
5	R/W	SDIO_CRC_ERR_MSK	0x0	This bit is used to mask CRC error interrupt. Setting this bit will mask the interrupt. Clearing this bit has no effect.
4	R/W	SDIO_PWR_LEV_MSK	0x0	This bit is used to mask power level change interrupt. Setting this bit will mask the interrupt. Clearing this bit has no effect.
3	R/W	SDIO_CMD52_MSK	0x0	This bit is used to mask CMD52 interrupt. Setting this bit will mask the interrupt. Clearing this bit has no effect.
2	R/W	SDIO_CSA_MSK	0x0	This bit is used to mask CMD53 CSA interrupt. Setting this bit will mask the interrupt. Clearing this bit has no effect.
1	R/W	SDIO_RD_INT_MSK	0x0	This bit is used to mask CMD53 read interrupt. Setting this bit will mask the interrupt. Clearing this bit has no effect.
0	R/W	SDIO_WR_INT_MSK	0x0	This bit is used to mask CMD53 write interrupt. Setting this bit will mask the interrupt. Clearing this bit has no effect.

SDIO_INTR_FN2_UNMASK_REGISTER

Table 16.61. SDIO Function2 Interrupt Unmask Register Description

Bit	Access	Function	POR Value	Description
[31:8]	R/W	Reserved	0x0	Reserved

Bit	Access	Function	POR Value	Description
7	R/W	SDIO_TOUT_UNMSK	0x0	This bit is used to unmask "read FIFO wait time over" interrupt. Setting this bit will unmask the interrupt. Clearing this bit has no effect.
6	R/W	SDIO_ABORT_ UNMSK	0x0	This bit is used to unmask abort interrupt. Setting this bit will unmask the interrupt. Clearing this bit has no effect.
5	R/W	SDIO_CRC_ERR_UNMSK	0x0	This bit is used to unmask CRC error interrupt. Setting this bit will unmask the interrupt. Clearing this bit has no effect.
4	R/W	SDIO_PWR_LEV_UNMSK	0x0	This bit is used to unmask power level change interrupt. Setting this bit will unmask the interrupt. Clearing this bit has no effect.
3	R/W	SDIO_CMD52_UNMSK	0x0	This bit is used to unmask CMD52 interrupt. Setting this bit will unmask the interrupt. Clearing this bit has no effect.
2	R/W	SDIO_CSA_UNMSK	0x0	This bit is used to unmask CMD53 CSA interrupt. Setting this bit will unmask the interrupt. Clearing this bit has no effect.
1	R/W	SDIO_RD_INT_UNMSK	0x0	This bit is used to unmask CMD53 read interrupt. Setting this bit will unmask the interrupt. Clearing this bit has no effect.
0	R/W	SDIO_WR_INT_UNMSK	0x0	This bit is used to unmask CMD53 write interrupt. Setting this bit will unmask the interrupt. Clearing this bit has no effect.

SDIO_INTR_FN3_REGISTER

Table 16.62. SDIO Function3 Interrupt Status/Clear Register Description

Bit	Access	Function	POR Value	Description
[31:10]	R/W	Reserved	0x0	Reserved

Bit	Access	Function	POR Value	Description
9	R/W	SDIO_CSA_ACCESS	0x0	csa_window_access When set, indicates that current request is for CSA window register. This is only status signal.
8	R/W	SDIO_WR_RDz	0x0	wr_rdz 0 – read request 1 – write request This is not an interrupt signal. This is only status signal.
7	R/W	SDIO_RD_TOUT_INT	0x0	When set, indicates that Read FIFO hasn't reached minimum threshold value before read wait timeout during.
6	R/W	SDIO_ABORT_ INT	0x0	When set, indicates that host issued an abort command.
5	R/W	SDIO_CRC_ ERR_INT	0x0	When set, indicates that data block is received with crc error.
4	R/W	SDIO_PWR_LEV_ INT	0x0	When set, indicates that power control register value has been changed by the host.
3	R/W	SDIO_CMD52_ INT	0x0	When set, indicates that CMD52 is received.
2	R/W	SDIO_CSA_ INT	0x0	When set, indicates that CMD53 request is received to CSA.
1	R/W	SDIO_RD_INT	0x0	When set, indicates that CMD53 read request is received.
0	R/W	SDIO_WR_INT	0x0	When set, indicates that CMD53 write request is received.

SDIO_INTR_FN3_ENABLE_REGISTER

Table 16.63. SDIO Function3 Interrupt Enable Register Description

Bit	Access	Function	POR Value	Description
[31:8]	R/W	Reserved	0x0	Reserved
7	R/W	SDIO_TOUT_INT_EN	0x0	This bit is used to enable "read FIFO wait time over" interrupt. '1'- Interrupt is enabled '0'- Interrupt is disabled
6	R/W	SDIO_ABORT_ INT_EN	0x0	This bit is used to enable abort interrupt. '1'- Interrupt is enabled '0'- Interrupt is disabled
5	R/W	SDIO_CRC_ ERR_INT_EN	0x0	This bit is used to enable CRC error interrupt. '1'- Interrupt is enabled '0'- Interrupt is disabled
4	R/W	SDIO_PWR_LEV_ INT_EN	0x0	This bit is used to enable power level change interrupt. '1'- Interrupt is enabled '0'- Interrupt is disabled

Bit	Access	Function	POR Value	Description
3	R/W	SDIO_CMD52_ INT_EN	0x0	This bit is used to enable CMD52 interrupt. '1'- Interrupt is enabled '0'- Interrupt is disabled
2	R/W	SDIO_CSA_ INT_EN	0x0	This bit is used to enable CMD53 CSA interrupt.
1	R/W	SDIO_RD_INT_EN	0x0	This bit is used to enable CMD53 read interrupt. '1'- Interrupt is enabled '0'- Interrupt is disabled
0	R/W	SDIO_WR_INT_EN	0x0	This bit is used to enable CMD53 write interrupt. '1'- Interrupt is enabled '0'- Interrupt is disabled

SDIO_INTR_FN3_MASK_REGISTER

Table 16.64. SDIO Function3 Interrupt Mask Register Description

Bit	Access	Function	POR Value	Description
[31:8]	R/W	Reserved	0x0	Reserved
7	R/W	SDIO_TOUT_MSK	0x0	This bit is used to mask "read FIFO wait time over" interrupt. Setting this bit will mask the interrupt. Clearing this bit has no effect.
6	R/W	SDIO_ABORT_MSK	0x0	This bit is used to mask abort interrupt. Setting this bit will mask the interrupt. Clearing this bit has no effect.
5	R/W	SDIO_CRC_ERR_MSK	0x0	This bit is used to mask CRC error interrupt. Setting this bit will mask the interrupt. Clearing this bit has no effect.
4	R/W	SDIO_PWR_LEV_MSK	0x0	This bit is used to mask power level change interrupt. Setting this bit will mask the interrupt. Clearing this bit has no effect.
3	R/W	SDIO_CMD52_MSK	0x0	This bit is used to mask CMD52 interrupt. Setting this bit will mask the interrupt. Clearing this bit has no effect.

Bit	Access	Function	POR Value	Description
2	R/W	SDIO_CSA_MSK	0x0	This bit is used to mask CMD53 CSA interrupt. Setting this bit will mask the interrupt. Clearing this bit has no effect.
1	R/W	SDIO_RD_INT_MSK	0x0	This bit is used to mask CMD53 read interrupt. Setting this bit will mask the interrupt. Clearing this bit has no effect.
0	R/W	SDIO_WR_INT_MSK	0x0	This bit is used to mask CMD53 write interrupt. Setting this bit will mask the interrupt. Clearing this bit has no effect.

SDIO_INTR_FN3_UNMASK_REGISTER

Table 16.65. SDIO Function3 Interrupt Unmask Register Description

Bit	Access	Function	POR Value	Description
[31:8]	R/W	Reserved	0x0	Reserved
7	R/W	SDIO_TOUT_UNMSK	0x0	This bit is used to unmask "read FIFO wait time over" interrupt. Setting this bit will unmask the interrupt. Clearing this bit has no effect.
6	R/W	SDIO_ABORT_ UNMSK	0x0	This bit is used to unmask abort interrupt. Setting this bit will unmask the interrupt. Clearing this bit has no effect.
5	R/W	SDIO_CRC_ERR_UNMSK	0x0	This bit is used to unmask CRC error interrupt. Setting this bit will unmask the interrupt. Clearing this bit has no effect.
4	R/W	SDIO_PWR_LEV_UNMSK	0x0	This bit is used to unmask power level change interrupt. Setting this bit will unmask the interrupt. Clearing this bit has no effect.
3	R/W	SDIO_CMD52_UNMSK	0x0	This bit is used to unmask CMD52 interrupt. Setting this bit will unmask the interrupt. Clearing this bit has no effect.
2	R/W	SDIO_CSA_UNMSK	0x0	This bit is used to unmask CMD53 CSA interrupt. Setting this bit will unmask the interrupt. Clearing this bit has no effect.

Bit	Access	Function	POR Value	Description
1	R/W	SDIO_RD_INT_UNMSK	0x0	This bit is used to unmask CMD53 read interrupt. Setting this bit will unmask the interrupt. Clearing this bit has no effect.
0	R/W	SDIO_WR_INT_UNMSK	0x0	This bit is used to unmask CMD53 write interrupt. Setting this bit will unmask the interrupt. Clearing this bit has no effect.

SDIO_INTR_FN4_REGISTER

Table 16.66. SDIO Function4 Interrupt Status/Clear Register Description

Bit	Access	Function	POR Value	Description
[31:10]	R/W	Reserved	0x0	Reserved
9	R/W	SDIO_CSA_ACCESS	0x0	csa_window_access When set, indicates that current request is for CSA window register. This is only status signal.
8	R/W	SDIO_WR_RDz	0x0	wr_rdz 0 – read request 1 – write request This is not an interrupt signal. This is only status signal.
7	R/W	SDIO_RD_TOUT_INT	0x0	When set, indicates that Read FIFO hasn't reached minimum threshold value before read wait timeout.
6	R/W	SDIO_ABORT_ INT	0x0	When set, indicates that host issued an abort command.
5	R/W	SDIO_CRC_ ERR_INT	0x0	When set, indicates that data block is received with crc error.
4	R/W	SDIO_PWR_LEV_ INT	0x0	When set, indicates that power control register value has been changed by the host.
3	R/W	SDIO_CMD52_ INT	0x0	When set, indicates that CMD52 is received.
2	R/W	SDIO_CSA_ INT	0x0	When set, indicates that CMD53 request is received to CSA.
1	R/W	SDIO_RD_INT	0x0	When set, indicates that CMD53 read request is received.
0	R/W	SDIO_WR_INT	0x0	When set, indicates that CMD53 write request is received.

SDIO_INTR_FN4_ENABLE_REGISTER

Table 16.67. SDIO Function4 Interrupt Enable Register Description

Bit	Access	Function	POR Value	Description
[31:8]	R/W	Reserved	0x0	Reserved

Bit	Access	Function	POR Value	Description
7	R/W	SDIO_TOUT_INT_EN	0x0	This bit is used to enable "read FIFO wait time over" interrupt. '1'- Interrupt is enabled '0'- Interrupt is disabled
6	R/W	SDIO_ABORT_ INT_EN	0x0	This bit is used to enable abort interrupt. '1'- Interrupt is enabled '0'- Interrupt is disabled
5	R/W	SDIO_CRC_ ERR_INT_EN	0x0	This bit is used to enable CRC error interrupt. '1'- Interrupt is enabled '0'- Interrupt is disabled
4	R/W	SDIO_PWR_LEV_ INT_EN	0x0	This bit is used to enable power level change interrupt. '1'- Interrupt is enabled '0'- Interrupt is disabled
3	R/W	SDIO_CMD52_ INT_EN	0x0	This bit is used to enable CMD52 interrupt. '1'- Interrupt is enabled '0'- Interrupt is disabled
2	R/W	SDIO_CSA_ INT_EN	0x0	This bit is used to enable CMD53 CSA interrupt.
1	R/W	SDIO_RD_INT_EN	0x0	This bit is used to enable CMD53 read interrupt. '1'- Interrupt is enabled '0'- Interrupt is disabled
0	R/W	SDIO_WR_INT_EN	0x0	This bit is used to enable CMD53 write interrupt. '1'- Interrupt is enabled '0'- Interrupt is disabled

SDIO_INTR_FN4_MASK_REGISTER

Table 16.68. SDIO Function4 Interrupt Mask Register Description

Bit	Access	Function	POR Value	Description
[31:8]	R/W	Reserved	0x0	Reserved
7	R/W	SDIO_TOUT_MSK	0x0	This bit is used to mask "read FIFO wait time over" interrupt. Setting this bit will mask the interrupt. Clearing this bit has no effect.

Bit	Access	Function	POR Value	Description
6	R/W	SDIO_ABORT_MSK	0x0	This bit is used to mask abort interrupt. Setting this bit will mask the interrupt. Clearing this bit has no effect.
5	R/W	SDIO_CRC_ERR_MSK	0x0	This bit is used to mask CRC error interrupt. Setting this bit will mask the interrupt. Clearing this bit has no effect.
4	R/W	SDIO_PWR_LEV_MSK	0x0	This bit is used to mask power level change interrupt. Setting this bit will mask the interrupt. Clearing this bit has no effect.
3	R/W	SDIO_CMD52_MSK	0x0	This bit is used to mask CMD52 interrupt. Setting this bit will mask the interrupt. Clearing this bit has no effect.
2	R/W	SDIO_CSA_MSK	0x0	This bit is used to mask CMD53 CSA interrupt. Setting this bit will mask the interrupt. Clearing this bit has no effect.
1	R/W	SDIO_RD_INT_MSK	0x0	This bit is used to mask CMD53 read interrupt. Setting this bit will mask the interrupt. Clearing this bit has no effect.
0	R/W	SDIO_WR_INT_MSK	0x0	This bit is used to mask CMD53 write interrupt. Setting this bit will mask the interrupt. Clearing this bit has no effect.

SDIO_INTR_FN4_UNMASK_REGISTER

Table 16.69. SDIO Function4 Interrupt Unmask Register Description

Bit	Access	Function	POR Value	Description
[31:8]	R/W	Reserved	0x0	Reserved
7	R/W	SDIO_TOUT_UNMSK	0x0	This bit is used to unmask "read FIFO wait time over" interrupt. Setting this bit will unmask the interrupt. Clearing this bit has no effect.

Bit	Access	Function	POR Value	Description
6	R/W	SDIO_ABORT_ UNMSK	0x0	This bit is used to unmask abort interrupt. Setting this bit will unmask the interrupt. Clearing this bit has no effect.
5	R/W	SDIO_CRC_ERR_UNMSK	0x0	This bit is used to unmask CRC error interrupt. Setting this bit will unmask the interrupt. Clearing this bit has no effect.
4	R/W	SDIO_PWR_LEV_UNMSK	0x0	This bit is used to unmask power level change interrupt. Setting this bit will unmask the interrupt. Clearing this bit has no effect.
3	R/W	SDIO_CMD52_UNMSK	0x0	This bit is used to unmask CMD52 interrupt. Setting this bit will unmask the interrupt. Clearing this bit has no effect.
2	R/W	SDIO_CSA_UNMSK	0x0	This bit is used to unmask CMD53 CSA interrupt. Setting this bit will unmask the interrupt. Clearing this bit has no effect.
1	R/W	SDIO_RD_INT_UNMSK	0x0	This bit is used to unmask CMD53 read interrupt. Setting this bit will unmask the interrupt. Clearing this bit has no effect.
0	R/W	SDIO_WR_INT_UNMSK	0x0	This bit is used to unmask CMD53 write interrupt. Setting this bit will unmask the interrupt. Clearing this bit has no effect.

SDIO_INTR_FN5_REGISTER

Table 16.70. SDIO Function5 Interrupt Status/Clear Register Description

Bit	Access	Function	POR Value	Description
[31:10]	R/W	Reserved	0x0	Reserved
9	R/W	SDIO_CSA_ACCESS	0x0	csa_window_access When set, indicates that current request is for CSA window register. This is only status signal.

Bit	Access	Function	POR Value	Description
8	R/W	SDIO_WR_RDz	0x0	wr_rdz 0 – read request 1 – write request This is not an interrupt signal. This is only status signal.
7	R/W	SDIO_RD_TOUT_INT	0x0	When set, indicates that Read FIFO hasn't reached minimum threshold value before read wait timeout.
6	R/W	SDIO_ABORT_ INT	0x0	When set, indicates that host issued an abort command.
5	R/W	SDIO_CRC_ ERR_INT	0x0	When set, indicates that data block is received with crc error.
4	R/W	SDIO_PWR_LEV_ INT	0x0	When set, indicates that power control register value has been changed by the host.
3	R/W	SDIO_CMD52_ INT	0x0	When set, indicates that CMD52 is received.
2	R/W	SDIO_CSA_ INT	0x0	When set, indicates that CMD53 request is received to CSA.
1	R/W	SDIO_RD_INT	0x0	When set, indicates that CMD53 read request is received.
0	R/W	SDIO_WR_INT	0x0	When set, indicates that CMD53 write request is received.

SDIO_INTR_FN5_ENABLE_REGISTER

Table 16.71. SDIO Function5 Interrupt Enable Register Description

Bit	Access	Function	POR Value	Description
[31:8]	R/W	Reserved	0x0	Reserved
7	R/W	SDIO_TOUT_INT_EN	0x0	This bit is used to enable "read FIFO wait time over" interrupt. '1'- Interrupt is enabled '0'- Interrupt is disabled
6	R/W	SDIO_ABORT_ INT_EN	0x0	This bit is used to enable abort interrupt. '1'- Interrupt is enabled '0'- Interrupt is disabled
5	R/W	SDIO_CRC_ ERR_INT_EN	0x0	This bit is used to enable CRC error interrupt. '1'- Interrupt is enabled '0'- Interrupt is disabled
4	R/W	SDIO_PWR_LEV_ INT_EN	0x0	This bit is used to enable power level change interrupt. '1'- Interrupt is enabled '0'- Interrupt is disabled
3	R/W	SDIO_CMD52_ INT_EN	0x0	This bit is used to enable CMD52 interrupt. '1'- Interrupt is enabled '0'- Interrupt is disabled

Bit	Access	Function	POR Value	Description
2	R/W	SDIO_CSA_ INT_EN	0x0	This bit is used to enable CMD53 CSA interrupt.
1	R/W	SDIO_RD_INT_EN	0x0	This bit is used to enable CMD53 read interrupt. '1'- Interrupt is enabled '0'- Interrupt is disabled
0	R/W	SDIO_WR_INT_EN	0x0	This bit is used to enable CMD53 write interrupt. '1'- Interrupt is enabled '0'- Interrupt is disabled

SDIO_INTR_FN5_MASK_REGISTER

Table 16.72. SDIO Function5 Interrupt Mask Register Description

Bit	Access	Function	POR Value	Description
[31:8]	R/W	Reserved	0x0	Reserved
7	R/W	SDIO_TOUT_MSK	0x0	This bit is used to mask "read FIFO wait time over" interrupt. Setting this bit will mask the interrupt. Clearing this bit has no effect.
6	R/W	SDIO_ABORT_MSK	0x0	This bit is used to mask abort interrupt. Setting this bit will mask the interrupt. Clearing this bit has no effect.
5	R/W	SDIO_CRC_ERR_MSK	0x0	This bit is used to mask CRC error interrupt. Setting this bit will mask the interrupt. Clearing this bit has no effect.
4	R/W	SDIO_PWR_LEV_MSK	0x0	This bit is used to mask power level change interrupt. Setting this bit will mask the interrupt. Clearing this bit has no effect.
3	R/W	SDIO_CMD52_MSK	0x0	This bit is used to mask CMD52 interrupt. Setting this bit will mask the interrupt. Clearing this bit has no effect.
2	R/W	SDIO_CSA_MSK	0x0	This bit is used to mask CMD53 CSA interrupt. Setting this bit will mask the interrupt. Clearing this bit has no effect.

Bit	Access	Function	POR Value	Description
1	R/W	SDIO_RD_INT_MSK	0x0	This bit is used to mask CMD53 read interrupt. Setting this bit will mask the interrupt. Clearing this bit has no effect.
0	R/W	SDIO_WR_INT_MSK	0x0	This bit is used to mask CMD53 write interrupt. Setting this bit will mask the interrupt. Clearing this bit has no effect.

SDIO_INTR_FN5_UNMASK_REGISTER

Table 16.73. SDIO Function5 Interrupt Unmask Register Description

Bit	Access	Function	POR Value	Description
[31:8]	R/W	Reserved	0x0	Reserved
7	R/W	SDIO_TOUT_UNMSK	0x0	This bit is used to unmask "read FIFO wait time over" interrupt. Setting this bit will unmask the interrupt. Clearing this bit has no effect.
6	R/W	SDIO_ABORT_ UNMSK	0x0	This bit is used to unmask abort interrupt. Setting this bit will unmask the interrupt. Clearing this bit has no effect.
5	R/W	SDIO_CRC_ERR_UNMSK	0x0	This bit is used to unmask CRC error interrupt. Setting this bit will unmask the interrupt. Clearing this bit has no effect.
4	R/W	SDIO_PWR_LEV_UNMSK	0x0	This bit is used to unmask power level change interrupt. Setting this bit will unmask the interrupt. Clearing this bit has no effect.
3	R/W	SDIO_CMD52_UNMSK	0x0	This bit is used to unmask CMD52 interrupt. Setting this bit will unmask the interrupt. Clearing this bit has no effect.
2	R/W	SDIO_CSA_UNMSK	0x0	This bit is used to unmask CMD53 CSA interrupt. Setting this bit will unmask the interrupt. Clearing this bit has no effect.

Bit	Access	Function	POR Value	Description
1	R/W	SDIO_RD_INT_UNMSK	0x0	This bit is used to unmask CMD53 read interrupt. Setting this bit will unmask the interrupt. Clearing this bit has no effect.
0	R/W	SDIO_WR_INT_UNMSK	0x0	This bit is used to unmask CMD53 write interrupt. Setting this bit will unmask the interrupt. Clearing this bit has no effect.

SDIO_ERROR_COND_CTRL_ENABLE_REGISTER

Table 16.74. SDIO Error Condition Enable Register Description

Bit	Access	Function	POR Value	Description
[31:3]	R	Reserved	0x0	Reserved for future use.
2	R/W	SDIO_RD_DATA_ERROR_EN	0x0	When set, stops the DMA from doing data accesses till read data error interrupt is cleared.
1	R/W	SDIO_ABORT_EN	0x0	When set, stops the DMA from doing data accesses till ABORT interrupt is cleared.
0	R/W	SDIO_CRC_EN	0x0	When set, stops the DMA from doing data accesses till CRC error interrupt is cleared.

SDIO_ERROR_COND_BLK_CNT

Table 16.75. SDIO Error Condition State Register Description

Bit	Access	Function	POR Value	Description
[31:23]	R	Reserved	0x0	Reserved for future use.
[22:16]	R/W	SDIO_ERROR_BLK_CNT	0x0	Indicates block count when one of the error conditions occurred.
[15:12]	R	Reserved	0x0	Reserved for future use.
[11:0]	R/W	SDIO_ERROR_BYTE_CNT	0x0	Indicates byte count when one of the error conditions occurred.

SDIO Boot Config Values Register 0 Description

Table 16.76. SDIO_BOOT_CONFIG_VALS_0

Bit	Access	Function	POR Value	Description
31:8	R	csa_msbyte	0xff8000	MS byre of CSA address. Lower 24 bits of CSA will come through SDIO CSA registers. Whenever CSA access is done, 32-bit address will be prepared using these fields.
7:0	R	ocr_r	0x09	Operating conditions. The value written by bootloader can be read here.

SDIO Boot Config Values Register 1 Description

Table 16.77. SDIO_BOOT_CONFIG_VALS_1

Bit	Access	Function	POR Value	Description
31:8	NA	Reserved	0x0	Reserved
7	R	ignore_disable_hs	0x0	if ignore_disable_hs is set, sdmem_disable_high_speed_switching coming from combo mode module is ignored.
6	R	sdmem_disable_interrupt_mb_read	0x0	When set, interrupt will be not be driven during sd memory mb read transfer.
5	R	sdmem_drive_hiz_mb_read	0x0	When set, High will be driven in the second cycle of interrupt period during sd memory mb read transfer.
4	R	sdmem_ignore_sdmem_present	0x0	When set, sdmem_present signal, coming from GPIO, will be ignored.
3	R	combocard	0x0	When set, combo mode will be enabled.
2:0	R	no_of_io_functions	0x1	Indicates number functions supported. The value written by bootloader can be read here.

Non-I/O Aware Initialization

Multiple block write command shall be used rather than continuous single write command to make faster write operation.

16.3 Configurable Timers

16.3.1 General Description

Configurable timers are used for counting clocks and events, capturing events on the GPIOs in input mode and outputting modulated signals. They can be programmed to work in Pulse Width Modulation (PWM) mode in which a pulse width modulated wave is driven on the outputs according to the programmed ON time and OFF times. Configurable Timers are present in MCU HP peripherals.

16.3.2 Features

The key features of the Configurable Timers are listed below:

- Supports two 16-bit timers
- The timers can be programmed to select clocks or events as ticks.
- Supports up counters, down counters and up-down counters
- Supports 1 input and 2 outputs.
- The output signals can be used to either output modulated signals or can be toggled after a programmable duration once a trigger is generated using the inputs.
- Support for PWM signals as outputs with any cycle/pulse length and superimposing of a waveform on the PWM signal.
- Supports wide range of features like starting the counter, stopping the counter, continuing the counter from the stopped value, halt, increment the counter and capturing the events
- Support for DMA flow control
- Generates interrupt for different events

16.3.3 Functional Description

The CT contains two16-bit configurable counters. It communicates with external devices through the GPIOs. Each configuration counter contains independent register set for controlling and accessing internal timers. Four registers are provided to program the selection of the DMA flow control signals for the Output Compare Unit (OCU) mode. Another register is also provided to obtain multiplexed output event of the available output events.

The counters has the capability to run as free-running counter, start counting, halt the counting operation, continuing the counter operation, increment the counter value and also to capture the counter value respectively according to the programmed values in the respective registers. Events for respective operations can also be programmed through the registers. There are also control registers available to support OCU operation.

16.3.3.1 Programming Sequence

Events handling

Input signal provided through GPIO pins is used to decide the event type such as rising edge, falling edge, high level triggering or low level triggering. These Events are described in the table below. Appropriate registers should be programmed according to the requirement (see registers description section for more details). Respective counter values can be read using firmware and also output events are obtained through the output pins.

Table 16.78. Events Description

Event selection	Event type	Description
0	None	No event is selected.
1	Eve_0_re	Event 0 rising edge.
5	Eve_0_fe	Event 0 falling edge.
9	Eve_0_rfe	Event 0 rising or falling edge.
13	Eve_0_lev0	Event 0 Level 0.
17	Eve_0_lev1	Event 0 Level 1.

Procedure for following Configurable timer use cases:

1. Free Run Timer (FRT) up mode.
   a. Set UP mode in gen_ctrl_reg register.
   b. Write PEAK value to MATCH_REG register.
   c. Configure the CT_INTR registers for generating required interrupts.
   d. Select the start signals either from input events or software triggers.
2. FRT up-down mode.
   a. Set UP-DOWN mode in gen_ctrl_reg register.
   b. Write PEAK value to MATCH_REG register.
   c. Configure the CT_INTR registers for generating required interrupts.
   d. Select the start signals either from input events or software triggers.
3. FRT down mode.
   a. Set DOWN mode in gen_ctrl_reg register.
   b. Write PEAK value to MATCH_REG register.
   c. Configure the CT_INTR registers for generating required interrupts.
   d. Select the start signals either from input events or software triggers.
4. FRT DOWN-UP mode.
   a. Set DOWN-UP mode in gen_ctrl_reg register.
   b. Write PEAK value to MATCH_REG register.
   c. Configure the CT_INTR registers for generating required interrupts.
   d. Select the start signals either from input events or software triggers.
5. FRT up mode with MATCH_BUF register.
   a. Enable buf_reg_0/1_en for respective counter.
   b. Set UP mode in gen_ctrl_reg.
   c. Write PEAK value to MATCH_REG register.
   d. Configure the CT_INTR registers for generating required interrupts.
   e. Select the start signals either from input events or software triggers.
6. PWM/WFG outputs.
   a. Set output_is_ocu
   b. Write PEAK value to MATCH_REG register.
   c. Write cycle/period value to OCU_COMPARE_REG.
   d. Select make_output_*_high_sel and make_output_*_low_sel accordingly.
   e. For WFG, after above procedure, set wfg_tgl_cnt register for required number of clocks.

7. ICU (input capture)
    a. Select the input event on which counter needs to be started, otherwise trigger counter through software option.
    b. Select the input event(s) on which counter need to be captured.
    c. Select the interrupt generation also to the capture event so that CPU gets the interrupt when capture happen.
    d. Read the CAPTURE_REG when interrupt is seen.
8. OCU mode.

ON period of the next OCU cycle should be programmed using the “OCU_COMPARE_NEXT_REG”. To avoid the overwriting of the already programmed ON period of the next cycle before it gets executed, “fifo_full” signal should be polled. Program the out of the available “fifo_full” signals using the “MUX_SEL_REG”. Two fifo_full signals for each counter can be selected at the top level. Counter OCUs description is shown in the following table.

Table 16.79. Counter OCUs Description

No	Select	Result becomes
1	0	0
2	1	1
3	2	It becomes one, when counter matches with ocu_compare_0
4	3	It becomes one, when counter matches with ocu_compare_1
5	4	It becomes one, when counter matches with ocu_compare_0 & counter is in up direction
6	5	It becomes one, when counter matches with ocu_compare_1 & counter is in up direction
7	6	It becomes one, when counter matches with ocu_compare_0 & counter is in down direction
8	7	It becomes one, when counter matches with ocu_compare_1 & counter is in down direction

16.3.4 Register Summary

Table 16.80. List of Timer Offset Addresses

Base Address: 0x4506_0000

S.no	Timer Name	Offset value	Description
1	Timer 0	0x0000	Timer 0 offset address
3	DMA flow control	0xF000	DMA flow control multiplexing select register offset address

Table 16.81. List of Counter Offset Addresses

S.no	Counter Name	Offset value	Description
1	Counter 0	0x000	Counter 0 offset address
2	Counter 1	0x100	Counter 1 offset address

Table 16.82. List of Registers in Each Counter

Register Name	Offset	Description
CT_GEN_CTRL_SET_REG	0x0	General control set register. Holds general control signals like enables and modes. This register is used only for setting the control signals
CT_GEN_CTRL_RESET_REG	0x4	General control reset register. Holds general control signals like disables. This register is used only for resetting the control signals
CT_INTR_STS_REG	0x8	Holds the interrupt status
CT_INTR_MASK_REG	0xC	Unwanted interrupts mask. Bits can be set to not to receive the interrupt indication
CT_INTR_UNMASK_REG	0x10	Wanted interrupts mask. Bits can be set to receive the interrupt indication
CT_INTR_ACK_REG	0x14	Interrupt clear/ack register
CT_MATCH_REG	0x18	Match value register. Top value for the timer/counter
CT_MATCH_BUF_REG	0x1C	Match buffer register. This will be loaded to CT_MATCH_REG when counter is active and becomes zero
CT_CAPTURE_REG	0x20	Captures and holds the counter value at the selected event occurrence
CT_COUNTER_REG	0x24	Holds the values of counter-1 and counter-2
CT_OCU_CTRL_REG	0x28	Bits in this register are used to configure the values corresponding to OCU operation

Register Name	Offset	Description
CT_OCU_COMPARE_REG	0x2C	Holds the threshold values for counter-0 which can be compared for making output event of counter-0 high/low basing on the ocu_ctrl_reg signals for present OCU period
CT_OCU_COMPARE2_REG	0x30	Holds the threshold values for counter-1 which can be compared for making output event of counter-1 high/low basing on the ocu_ctrl_reg signals for present OCU period
CT_OCU_SYNC_REG	0x34	Used as starting value for syncing OCU counters
CT_OCU_COMPARE_NXT_REG	0x38	Holds the threshold for next OCU period which will get loaded in to CT_OCU_COMPARE_REG
CT_OCU_COMPARE2_NXT_REG	0x40	Holds the threshold for next OCU period which will get loaded in to CT_OCU_COMPARE2_REG
CT_WFG_CTRL_REG	0x3C	Holds WFG related control signals like enables and modes. Write only register. Register value is maintained but can be read
Reserved	0x40 – 0x4f	Reserved for future use
CT_START_COUNTER_EVENT_SEL_REG	0x50	Start counter event select register
CT_CONTINUE_COUNTER_EVENT_SEL_REG	0x5C	Continue counter event select register
CT_STOP_COUNTER_EVENT_SEL_REG	0x68	Stop counter event select register
CT_HALT_COUNTER_EVENT_SEL_REG	0x74	Halt counter event select register
CT_INCREMENT_COUNTER_EVENT_SEL_REG	0x80	Increment counter event select register
CT_CAPTURE_COUNTER_EVENT_SEL_REG	0x8C	Capture counter event select register
CT_OUTPUT_EVENT_SEL_REG	0x98	Output event select configuration register
CT_INTR_EVENT_SEL_REG	0xA4	Interrupt event select configuration register
CT_RE_FE_RFE_LEV0_LEV1_EVENT_ENABLE_REG	0xB0	Rising Edge Falling Edge Rising and Falling Edge level0 level1 event enable register

CT_MUX_SEL_0_REG

Table 16.83. List of Common Registers in CT

Register Name	Offset	Description
CT_MUX_SEL_0_REG	0xf000	Mux sel-0 configuration register
CT_MUX_SEL_1_REG	0xf004	Mux sel-1 configuration register
CT_MUX_SEL_2_REG	0xf008	Mux sel-2 configuration register
CT_MUX_SEL_3_REG	0xf00C	Mux sel-3 configuration register
	0xf018	Output event ADC select register

Register Name	Offset	Description
	0xf01C	Output event ADC select register

16.3.5 Register Description

Legend: R = Read-only, W = Write-only, R/W = Read/Write

16.3.5.1 CT_GEN_CTRL_SET_REG

Table 16.84. GEN_CTRL_SET_REG Register Description

Bit	Access	Function	POR Value	Description
31:24	R	Reserved	0	Reserved for future use.
23	R/W	Buf_reg_1_en	0x0	Buffer register gets enabled for MATCH REG. MATCH_BUF_REG is always available and whenever this bit is set only, gets copied to MATCH REG. If write 1 – Buffer will be enabled and in path 0 – No effect. If read 1 – Buffer is enabled and in path 0 – Buffer is not enabled and not in path
22	R/W	Counter_1_sync_trig	0x0*	This enables the counter to run/active when sync is found. If write 1 – Counter_1 will be active. 0 – No effect. If read Read should always return 0. For counter to get increment / decrement, counter should be active and hit with selected event.
21:20	R/W	Counter_1_up_down	0x0	This enables the counter to run in up/down/up-down/down-up directions If write 0 – No effect to old value. 1 – Counter_1 up direction enable 2 – Counter_1 down direction enable 3 – Both up and down directions enable. If read 0 – Counter_1 is in down-up counting mode. 1 – Counter_1 is in up counting mode. 2 – Counter_1 is in down counting mode. 3 – Counter_1 is in up-down counting mode.

Bit	Access	Function	POR Value	Description
19	R/W	Counter_1_trig_frm	0x0*	This enables the counter to run/active. If write 1 – Counter_1 will be active. 0 – No effect. If read Read should always return 0. For counter to get increment / decrement, counter should be active and hit with selected event.
18	R/W	Periodic_en_Counter_1_frm_reg	0x0	This enables the counter to re-run even after expire/saturation value hit. If write 1 – Counter_1 will be in periodic mode. 0 – No effect. If read 1 – Counter_1 is in periodic mode. 0 – Counter_1 is not in periodic mode.
17	R/W	Soft_reset_Counter_1_frm_reg	0x0*	This resets the counter on the write. If write 1 – Counter_1 will be reset. 0 – No effect. If Read Read always should return 0.
16:8	R	Reserved	0x0	Reserved for future use.
7	R/W	Buf_reg_0_en	0x0	Buffer register gets enabled for MATCH REG. MATCH_BUF_REG is always available and whenever this bit is set only, gets copied to MATCH REG. If write 1 – Buffer will be enabled and in path 0 – No effect. If read 1 – Buffer is enabled and in path 0 – Buffer is not enabled and not in path.

Bit	Access	Function	POR Val-ue	Description
6	R/W	Counter_0_sync_trig	0x0*	This enables the counter to run/active when sync is found. If write 1 – Counter_0 will be active. 0 – No effect. If read Read should always return 0. For counter to get increment / decrement, counter should be active and hit with selected event.
5:4	R/W	Counter_0_up_down	0x0	This enables the counter to run in up/down/up-down/down-up directions If write 0 – No effect to old value. 1 – Counter_0 up direction enable 2 – Counter_0 down direction enable 3 – Both up and down directions enable. If read 0 – Counter_0 is in down-up counting mode. 1 – Counter_0 is in up counting mode. 2 – Counter_0 is in down counting mode. 3 – Counter_0 is in up-down counting mode.
3	R/W	Counter_0_trig_frm_reg	0x0*	This enables the counter to run/active. If write 1 – Counter_0 will be active. 0 – No effect. If read Read should always return 0. For counter to get increment / decrement, counter should be active and hit with selected event.

Bit	Access	Function	POR Value	Description
2	R/W	Periodic_en_Counter_0_frm_reg	0x0	This enables the counter to re-run even after expire/saturation value hit. If write 1 – Counter_1 will be in periodic mode. 0 – No effect. If read 1 – Counter_1 is in periodic mode. 0 – Counter_1 is not in periodic mode. The saturation value is programmed in “match_reg”. Also note that, in periodic mode, two values less than the required period needs to be programmed in the “match_reg”. For example, if the counter needs to run from 0 to 4 (i.e period is 5 counter values), then program 3 in the match_reg in periodic mode.
1	R/W	Soft_reset_counter_0_frm_reg	0x0*	This resets the counter on the write. If write 1 – Counter_0 will be reset. 0 – No effect. If Read Read always should return 0. * - Indicates self clear.
0	R	RESERVED	0x0	RESERVED
Note: * - Indicates self clear.				

16.3.5.2 CT_GEN_CTRL_RESET_REG

Table 16.85. GEN_CTRL_RESET_REG Register Description

Bit	Access	Function	POR Value	Description
31:24	R	Reserved	0	Reserved for future use.

Bit	Access	Function	POR Value	Description
23	R/W	Buf_reg_1_en	0x0	Buffer register gets enabled for MATCH REG. MATCH_BUF_REG is always available and whenever this bit is set only, gets copied to MATCH REG. If write 1 – Buffer will be disabled and in path 0 – No effect. If read 1 – Buffer is enabled and in path 0 – Buffer is not enabled and not in path.
22	R	Counter_1_sync_trig	0x0	Self clear bit.
21:20	R/W	Counter_1_up_down	0x0	This enables the counter to run in up/down/up-down/down-up directions 0 – Counter_1 is in down-up counting mode. 1 – Counter_1 is in up counting mode. 2 – Counter_1 is in down counting mode. 3 – Counter_1 is in up-down counting mode.
19	R	Counter_1_trig_frm_reg	0x0	This is a self clear bit in set register.
18	R/W	Periodic_en_Counter_1_frm_reg	0x0	This enables the counter to re-run even after expire/saturation value hit. If write 1 – Counter_1 will be in single shot mode. 0 – No effect. If read 1 – Counter_1 is in periodic mode. 0 – Counter_1 is in single shot mode. The saturation value is programmed in “match_reg_1”. Also note that, in periodic mode, two values less than the required period needs to be programmed in the “match_reg_1”. For example, if the counter needs to run from 0 to 4 (i.e period is 5 counter values), then program 3 in the match_reg in periodic mode.
17	R	Soft_reset_Counter_1_frm_reg	0x0	This is a self clear bit in set register.
16:8	R	Reserved	0x0	Reserved for future use.

Bit	Access	Function	POR Value	Description
7	R/W	Buf_reg_0_en	0x0	Buffer register gets enabled for MATCH REG. MATCH_BUF_REG is always available and whenever this bit is set only, gets copied to MATCH REG. If write 1 – Buffer will be disabled and in path 0 – No effect. If read 1 – Buffer is enabled and in path 0 – Buffer is not enabled and not in path.
6	R	Counter_0_sync_trig	0x0	This is a self clear bit in set register.
5:4	R/W	Counter_0_up_down	0x0	This enables the counter to run in up/down/up-down/down-up directions 0 – Counter_0 is in down-up counting mode. 1 – Counter_0 is in up counting mode. 2 – Counter_0 is in down counting mode. 3 – Counter_0 is in up-down counting mode.
3	R	Counter_0_trig_frm_reg	0x0	This is a self clear bit in set register.
2	R/W	Periodic_en_Counter_0_frm_reg	0x0	This enables the counter to re-run even after expire/saturation value hit. If write 1 – Counter_0 will be in single count mode. 0 – No effect. If read 1 – Counter_0 is in periodic mode. 0 – Counter_0 is in single shot mode.
1	R	Soft_reset_counter_0_frm_reg	0x0	This is a self clear bit in set register.
0	R	RESERVED	0x0	RESERVED

16.3.5.3 CT_INTR_STS

Table 16.86. INTR_STS Register Description

Bit	Access	Function	POR Value	Description
31:20	R	Reserved	0	Reserved for future use.
19	R	Counter_1_is_peak_l	0	Counter 1 hit peak (MATCH) in active mode.
18	R	Counter_1_is_zero_l	0	Counter 1 hit zero in active mode.
17	R	fifo_1_full_l	0	Indicates the FIFO full signal of channel-1
16	R	intr_1_l	0	This is derived from the input events.

Bit	Access	Function	POR Value	Description
15:4	R	Reserved	0	Reserved for future use.
3	R	Counter_0_is_peak_l	0	Counter 0 hit peak (MATCH) in active mode.
2	R	Counter_0_is_zero_l	0	Counter 0 hit zero in active mode.
1	R	fifo_0_full_l	0	Indicates the FIFO full signal of channel-0
0	R	intr_0_l	0	This is derived from the input events.

16.3.5.4 CT_INTR_MASK

Table 16.87. INTR_MASK Register Description

Bit	Access	Function	POR Value	Description
31:20	R	Reserved	0	Reserved for future use.
19	R/W	Counter_1_is_peak_l	0x0	Interrupt mask signal. If write 1 – Interrupt will be masked. 0 – No effect. If read 1 – Interrupt is unmasked. 0 – Interrupt is masked.
18	R/W	Counter_1_is_zero_l	0x0	Interrupt mask signal. If write 1 – Interrupt will be masked. 0 – No effect. If read 1 – Interrupt is unmasked. 0 – Interrupt is masked.
17	R/W	fifo_1_full_l	0x0	Interrupt mask signal. If write 1 – Interrupt will be masked. 0 – No effect. If read 1 – Interrupt is unmasked. 0 – Interrupt is masked.

Bit	Access	Function	POR Value	Description
16	R/W	intr_1_l	0x0	Interrupt mask signal. If write 1 – Interrupt will be masked. 0 – No effect. If read 1 – Interrupt is unmasked. 0 – Interrupt is masked.
15:4	R	Reserved	0	Reserved for future use.
3	R/W	Counter_0_is_peak_l	0x0	Interrupt mask signal. If write 1 – Interrupt will be masked. 0 – No effect. If read 1 – Interrupt is unmasked. 0 – Interrupt is masked.
2	R/W	Counter_0_is_zero_l	0x0	Interrupt mask signal. If write 1 – Interrupt will be masked. 0 – No effect. If read 1 – Interrupt is unmasked. 0 – Interrupt is masked.
1	R/W	fifo_0_full_l	0x0	Interrupt mask signal. If write 1 – Interrupt will be masked. 0 – No effect. If read 1 – Interrupt is unmasked. 0 – Interrupt is masked.
0	R/W	intr_0_l	0x0	Interrupt mask signal. If write 1 – Interrupt will be masked. 0 – No effect. If read 1 – Interrupt is unmasked. 0 – Interrupt is masked.

16.3.5.5 CT_INTR_UNMASK

Table 16.88. INTR_UNMASK Register Description

Bit	Access	Function	POR Value	Description
31:20	R	Reserved	0	Reserved for future use.
19	R/W	Counter_1_is_peak_l	0x0	Interrupt mask signal. If write 1 – Interrupt will be unmasked. 0 – No effect. If read 1 – Interrupt is unmasked. 0 – Interrupt is masked.
18	R/W	Counter_1_is_zero_l	0x0	Interrupt mask signal. If write 1 – Interrupt will be unmasked. 0 – No effect. If read 1 – Interrupt is unmasked. 0 – Interrupt is masked.
17	R/W	fifo_1_full_l	0x0	Interrupt mask signal. If write 1 – Interrupt will be unmasked. 0 – No effect. If read 1 – Interrupt is unmasked. 0 – Interrupt is masked.
16	R/W	intr_1_l	0x0	Interrupt mask signal. If write 1 – Interrupt will be unmasked. 0 – No effect. If read 1 – Interrupt is unmasked. 0 – Interrupt is masked.
15:4	R	Reserved	0	Reserved for future use.

Bit	Access	Function	POR Value	Description
3	R/W	Counter_0_is_peak_l	0x0	Interrupt mask signal. If write 1 – Interrupt will be unmasked. 0 – No effect. If read 1 – Interrupt is unmasked. 0 – Interrupt is masked.
2	R/W	Counter_0_is_zero_l	0x0	Interrupt mask signal. If write 1 – Interrupt will be unmasked. 0 – No effect. If read 1 – Interrupt is unmasked. 0 – Interrupt is masked.
1	R/W	fifo_0_full_l	0x0	Interrupt mask signal. If write 1 – Interrupt will be unmasked. 0 – No effect. If read 1 – Interrupt is unmasked. 0 – Interrupt is masked.
0	R/W	intr_0_l	0x0	Interrupt mask signal. If write 1 – Interrupt will be unmasked. 0 – No effect. If read 1 – Interrupt is unmasked. 0 – Interrupt is masked.

16.3.5.6 CT_INTR_ACK

Table 16.89. INTR_ACK Register Description

Bit	Access	Function	POR Value	Description
31:20	R	Reserved	0	Reserved for future use.

Bit	Access	Function	POR Value	Description
19	R/W	Counter_1_is_peak_l	0x0*	Interrupt ack signal. If write 1 – Interrupt will be deasserted. 0 – No effect. If read 0 should be returned as this is self clear bit.
18	R/W	Counter_1_is_zero_l	0x0*	Interrupt ack signal. If write 1 – Interrupt will be deasserted. 0 – No effect. If read 0 should be returned as this is self clear bit.
17	R/W	fifo_1_full_l	0x0*	Interrupt ack signal. If write 1 – Interrupt will be deasserted. 0 – No effect. If read 0 should be returned as this is self clear bit.
16	R/W	intr_1_l	0x0*	Interrupt ack signal. If write 1 – Interrupt will be deasserted. 0 – No effect. If read 0 should be returned as this is self clear bit.
15:4	R	Reserved	0	Reserved for future use.
3	R/W	Counter_0_is_peak_l	0x0*	Interrupt ack signal. If write 1 – Interrupt will be deasserted. 0 – No effect. If read 0 should be returned as this is self clear bit.
2	R/W	Counter_0_is_zero_l	0x0*	Interrupt ack signal. If write 1 – Interrupt will be deasserted. 0 – No effect. If read 0 should be returned as this is self clear bit.
1	R/W	fifo_0_full_l	0x0*	Interrupt ack signal. If write 1 – Interrupt will be deasserted. 0 – No effect. If read 0 should be returned as this is self clear bit.

Bit	Access	Function	POR Value	Description
0	R/W	intr_0_l	0x0*	Interrupt ack signal. If write 1 – Interrupt will be deasserted. 0 – No effect. If read 0 should be returned as this is self clear bit.

16.3.5.7 CT_MATCH_REG

Table 16.90. MATCH Register Description

Bit	Access	Function	POR Value	Description
31:16	R/W	Counter_1_match	0xffff	Counter 1 match register. This will act as match register for counter-1. Whenever this is hit/matched, counter 1 generates peak interrupt nd reset the counter to 0x0
15:0	R/W	Counter_0_match	0xffff	Counter 0 match register. This will act as match register for counter-0. Whenever this is hit/matched, counter 0 generates peak interrupt nd reset the counter to 0x0

16.3.5.8 CT_MATCH_BUF_REG

Table 16.91. MATCH_BUF Register Description

Bit	Access	Function	POR Value	Description
31:16	R/W	Counter_1_match_buf	0xffff	This gets copied to MATCH register if bug_reg_1_en is set. Copying is done when counter 1 is active and hits 0.
15:0	R/W	Counter_0_match_buf	0xffff	This gets copied to MATCH register if bug_reg_0_en is set. Copying is done when counter 0 is active and hits 0.

16.3.5.9 CT_CAPTURE_REG

Table 16.92. Capture Register Description

Bit	Access	Function	POR Value	Description
31:16	R	Counter_1_capture	0x0	This is a latched value of counter upper part when the selected capture_event occurs.
15:0	R	Counter_0_capture	0x0	This is a latched value of counter lower part when the selected capture_event occurs.

16.3.5.10 CT_COUNTER_REG

Table 16.93. Counter Register Description

Bit	Access	Function	POR Value	Description
31:16	R/W	Counter 1	0x0	This holds the value of counter-1.

Bit	Access	Function	POR Value	Description
15:0	R/W	Counter 0	0x0	This holds the value of counter-0.

16.3.5.11 CT_OCU_CTRL_REG

Table 16.94. OCU_CTRL Register Description

Bit	Access	Function	POR Value	Description
31:28	R/W	Reserved	0	Reserved for future use.
27:25	R/W	Make_output_1_low_sel	0	Check counter ocus for possibilities. When this is hit output will be made low.
24:22	R/W	Make_output_1_high_sel	0	Check counter ocus for possibilities. When this is hit output will be made high.
21	R/W	ocu_1_mode_8_16_mode	0	Indicates whether entire 16 bits or only 8-bits of the channel-1 are used in OCU mode. 1 - 8 bit mode 0 - 16 bit mode
20	R/W	ocu_1_dma_mode	0	Indicates whether the OCU – DMA mode is active or not for channel-1.
19:17	R/W	sync_with_1	0	Indicates whether the other channel is in sync with this channel-1.
16	R/W	output_1_is_ocu	0	Indicates whether the output is in OCU mode or not for channel-1.
15:12	R/W	Reserved	0	Reserved for future use.
11:9	R/W	Make_output_0_low_sel	0	Check counter ocus for possibilities. When this is hit output will be made low.
8:6	R/W	Make_output_0_high_sel	0	Check counter ocus for possibilities. When this is hit output will be made high.
5	R/W	ocu_0_mode_8_16	0	Indicates whether entire 16 bits or only 8-bits of the channel-0 are used in OCU mode. 1- 8 bit mode 0 - 16 bit mode
4	R/W	ocu_0_dma_mode	0	Indicates whether the OCU – DMA mode is active or not for channel-0.
3:1	R/W	sync_with_0	0	Indicates whether the other channel is in sync with this channel-0.
0	R/W	output_is_ocu_0	0	Indicates whether the output is in OCU mode or not for channel-0.

16.3.5.12 CT_OCU_COMPARE_REG

Table 16.95. OCU_Compare Register Description

Bit	Access	Function	POR Value	Description
31:16	R/W	ocu_compare_1_reg	0	Holds the threshold value of present OCU period which denotes the number of clock cycles for which the OCU output should be considered (counter 1)
15:0	R/W	ocu_compare_0_reg	0	Holds the threshold value of present OCU period which denotes the number of clock cycles for which the OCU output should be considered (counter 0)

16.3.5.13 CT_OCU_COMPARE2_REG

Table 16.96. OCU_Compare2 Register Description

Bit	Access	Function	POR Value	Description
31:16	R/W	ocu_compare2_1_reg	0	Holds the threshold value of present OCU period2 which denotes the number of clock cycles for which the OCU output should be considered (counter 1)
15:0	R/W	ocu_compare2_0_reg	0	Holds the threshold value of present OCU period2 which denotes the number of clock cycles for which the OCU output should be considered (counter 0)

16.3.5.14 CT_OCU_SYNC_REG

Table 16.97. OCU_Sync Register Description

Bit	Access	Function	POR Value	Description
31:16	R/W	ocu_sync_reg_1	0	Holds the starting point of channel-1 for synchronization purpose.
15:0	R/W	ocu_sync_reg_0	0	Holds the starting point of channel-0 for synchronization purpose.

16.3.5.15 CT_OCU_COMPARE_NXT_REG

Table 16.98. OCU_Compare_Nxt Register Description

Bit	Access	Function	POR Value	Description
31:16	R/W	ocu_compare_nxt_reg_counter1	0	Holds the threshold value of next OCU period-1 which ultimately gets loaded in to ocu_compare_reg in dma mode for (counter 1).
15:0	R/W	ocu_compare_nxt_reg_counter0	0	Holds the threshold value of next OCU period-2 which ultimately gets loaded in to ocu_compare_reg in dma mode for (counter 0).

16.3.5.16 CT_OCU_COMPARE2_NXT_REG

Table 16.99. OCU_Compare_Nxt2 Register Description

Bit	Access	Function	POR Value	Description
31:16	R/W	ocu_compare2_nxt_reg_counter1	0	Holds the threshold value of next OCU period-1 which ultimately gets loaded in to ocu_compare2_reg in dma mode for (counter 1).
15:0	R/W	ocu_compare2_nxt_reg_counter0	0	Holds the threshold value of next OCU period-2 which ultimately gets loaded in to ocu_compare2_reg in dma mode for (counter 0).

16.3.5.17 CT_WFG_CTRL_REG

Table 16.100. WFG_Ctrl Register Description

Bit	Access	Function	POR Value	Description
31:24	W	Wfg_tgl_cnt_1_peak	0	WFG mode output toggle count clock for channel 1.
23:22	W	Reserved	0	Reserved for future use.
21:19	W	Make_output_1_tgl_1_sel	0	Check the counter ocus possibilities for description for channel 1.
18:16	W	Make_output_1_tgl_0_sel	0	Check the counter ocus possibilities for description for channel 1.
15:8	W	Wfg_tgl_cnt_0_peak	0	WFG mode output toggle count clock for channel 0.
7:6	W	Reserved	0	Reserved for future use.
5:3	W	Make_output_0_tgl_1_sel	0	Check the counter ocus possibilities for description for channel 0.
2:0	W	Make_output_0_tgl_0_sel	0	Check the counter ocus possibilities for description for channel 0.

16.3.5.18 CT_START_COUNTER_EVENT_SEL

Table 16.101. Start_Counter_Event_Sel Register Description

Bit	Access	Function	POR Value	Description
31:22	R	Reserved	0	Reserved for future use.
21:16	R/W	Start_Counter_1_event_sel	0	Event select for starting the Counter_1.
15:6	R/W	Reserved	0	Reserved for future use.
5:0	R/W	Start_Counter_0_event_sel	0	Event select for starting the Counter_0.

16.3.5.19 CT_CONTINUE_COUNTER_EVENT_SEL

Table 16.102. Continue_Counter_Event_Sel Register Description

Bit	Access	Function	POR Value	Description
31:22	R	Reserved	0	Reserved for future use.
21:16	R/W	Continue_Counter_1_event_sel	0	Event select for continuing the Counter_1.
15:6	R/W	Reserved	0	Reserved for future use.
5:0	R/W	Continue_Counter_0_event_sel	0	Event select for continuing the Counter_0.

16.3.5.20 CT_STOP_COUNTER_EVENT_SEL

Table 16.103. Stop_Counter_Event_Sel Register Description

Bit	Access	Function	POR Value	Description
31:22	R	Reserved	0	Reserved for future use.
21:16	R/W	Stop_Counter_1_event_sel	0	Event select for Stopping the Counter_1.
15:6	R/W	Reserved	0	Reserved for future use.
5:0	R/W	Stop_Counter_0_event_sel	0	Event select for Stopping the Counter_0.

16.3.5.21 CT_HALT_COUNTER_EVENT_SEL

Table 16.104. Halt_Counter_Event_Sel Register Description

Bit	Access	Function	POR Value	Description
31:23	R	Reserved	0	Reserved for future use.
22	W	Resume_from_halt_counter_1	0	When halt event for counter1 is occurred, then counter1 went into halt mode. To resume this counter from halt mode, set this bit from f/w. If the counter1 entered into halt mode, there is effect of any event on counter1 till this is asserted. It is self clear bit.
21:16	R/W	Halt_Counter_1_event_sel	0	Event select for Halting the Counter_1.
15:7	R	Reserved	0	Reserved for future use.
6	W	Resume_from_halt_counter_0	0	When halt event for counter0 is occurred, then counter0 went into halt mode. To resume this counter from halt mode, set this bit from f/w. If the counter0 entered into halt mode, there is effect of any event on counter0 till this is asserted. It is self clear bit.
5:0	R/W	Halt_Counter_0_event_sel	0	Event select for Halting the Counter_0.

16.3.5.22 CT_INCREMENT_COUNTER_EVENT_SEL

Table 16.105. Increment_Counter_Event_Sel Register Description

Bit	Access	Function	POR Value	Description
31:22	R	Reserved	0	Reserved for future use.
21:16	R/W	Increment_Counter_1_event_sel	0	Event select for incrementing the Counter_1.
15:6	R	Reserved	0	Reserved for future use.
5:0	R/W	Increment_Counter_0_event_sel	0	Event select for incrementing the Counter_0.

16.3.5.23 CT_CAPTURE_COUNTER_EVENT_SEL

Table 16.106. Capture_Counter_Event_Sel Register Description

Bit	Access	Function	POR Value	Description
31:22	R	Reserved	0	Reserved for future use.
21:16	R/W	Capture_Counter_1_event_sel	0	Event select for Capturing the Counter_1.
15:6	R	Reserved	0	Reserved for future use.
5:0	R/W	Capture_Counter_0_event_sel	0	Event select for Capturing the Counter_0.

16.3.5.24 CT_OUTPUT_EVENT_SEL

Table 16.107. Output_Event_Sel Register Description

Bit	Access	Function	POR Value	Description
31:22	R	Reserved	0	Reserved for future use.
21:16	R/W	output_event_sel_1	0	Event select for output event from counter_1.
15:6	R	Reserved	0	Reserved for future use.
5:0	R/W	output_event_sel_0	0	Event select for output event from Counter_0.

16.3.5.25 CT_INTR_EVENT_SEL

Table 16.108. Intr_Event_Sel Register Description

Bit	Access	Function	POR Value	Description
31:22	R	Reserved		Reserved
21:16	R/W	intr_event_sel_1	0	Event select for interrupt event from counter_1.
15:6	R	Reserved		Reserved
5:0	R/W	intr_event_sel_0	0	Event select for interrupt event from Counter_0.

16.3.5.26 CT_RE_FE_RFE_LEV0_LEV1_EVENT_ENABLE_REG

Table 16.109. RE_FE_RFE_Lev0_Lev1_Event_Enable Register Description

Bit	Access	Function	POR Value	Description
31:20	R	Reserved		Reserved
19:16	R/W	Input_event_lev1_enable	0xF	Input event level1 enables: 0001: input event0 is enabled
15:12	R/W	Input_event_lev0_enable	0xF	Input event level0 enables: 0001: input event0 is enabled
11:8	R/W	Input_event_rfe_enable	0xF	Input event rising and falling edge enables: 0001: input event0 is enabled
7:4	R/W	Input_event_fe_enable	0xF	Input event falling edge enables: 0001: input event0 is enabled
3:0	R/W	Input_event_re_enable	0xF	Input event rising edge enables: 0001: input event0 is enabled

16.3.5.27 CT_MUX_SEL_0_REG

Table 16.110. Mux_Sel_0_Reg Description

Bit	Access	Function	POR Value	Description
3:0	R/W	mux_sel_0	0	Select value to select first output value “fifo_0_full[0]” out of all the “fifo_0_full_muxed” signals of counter_0.

16.3.5.28 CT_MUX_SEL_1_REG

Table 16.111. Mux_Sel_1_Reg Description

Bit	Access	Function	POR Value	Description
3:0	R/W	mux_sel_1	0	Select value to select first output value “fifo_0_full[1]” out of all the “fifo_0_full_muxed” signals of counter_0.

16.3.5.29 CT_MUX_SEL_2_REG

Table 16.112. Mux_Sel_2_Reg Description

Bit	Access	Function	POR Value	Description
3:0	R/W	mux_sel_2	0	Select value to select first output value “fifo_1_full[0]” out of all the “fifo_1_full_muxed” signals of counter_1

16.3.5.30 CT_MUX_SEL_3_REG

Table 16.113. Mux_Sel_3_Reg Description

Bit	Access	Function	POR Value	Description
3:0	R/W	mux_sel_3	0	Select value to select first output value “fifo_1_full[0]” out of all the “fifo_1_full_muxed” signals of counter_1

16.4 CRC Accelerator

16.4.1 General Description

CRC (cyclic redundancy check) is used in a wide range of applications as a first data integrity check. The CRC Accelerator is present in MCU HP peripherals.

16.4.2 Features

- Supports 1-32 bit polynomials.
- Supports 1-32 bit stream-in data widths.
- Supports DMA flow control.
- Supports AHB interface.

16.4.3 Functional Description

CRC accelerator computes CRC without any cycle penalty for any continuous data provided. It has 16x37 WFIFO, which holds 16 rows with 32 bit data and 5 bit length in each row. Zero indicates 32 bits of data is valid and the rest of values directly indicates the valid bits in each row. The LFSR is initialized with polynomial and initial value. Then input packet is read from WFIFO. After computation of CRC, status is updated on registers.

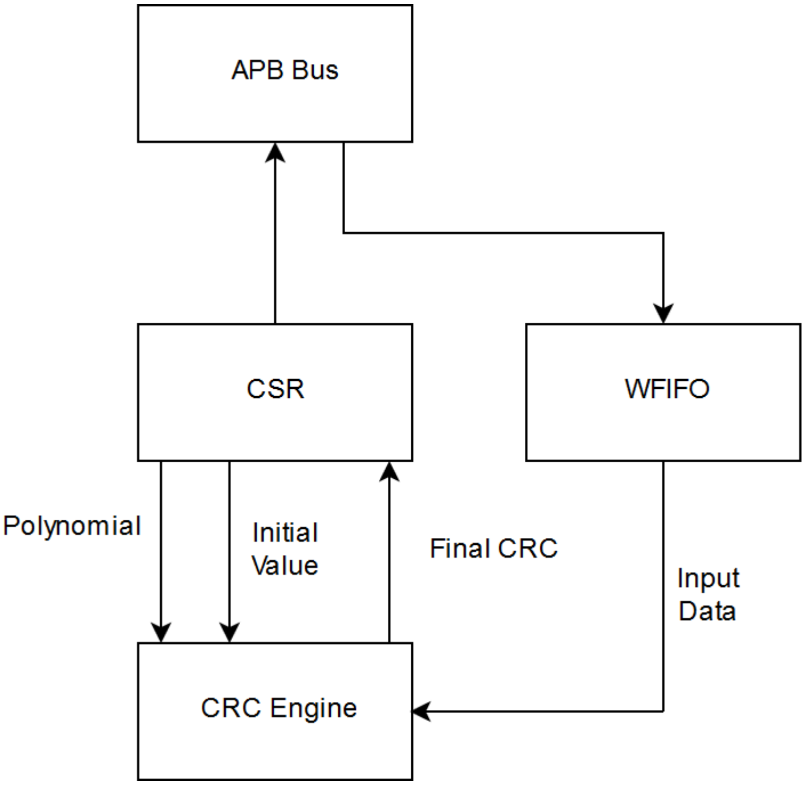


Figure 16.16. CRC accelerator block diagram

16.4.3.1 Programming Sequence for Computing CRC

1. Polynomial has to be loaded to POLYNOMIAL Register.
2. Polynomial width/length of the polynomial has to be loaded to polynomial_ctrl register. Clear the polynomial_width by writing 0x1f into polynomial_ctrl_reset register and write the actual width/length into polynomial_ctrl_set register.
3. LFSR_INIT has to be updated if required. Set LFSR_INIT_CTRL_SET [clear_lfsr] if LFSR required to be cleared. Set LFSR_INIT_CTRL_SET [init_lfsr] if LFSR_STATE required to be initialized.
4. Input data width :
   a. Width from ULI: This is the default mode with out any configuration. Set DIN_CTRL_RESET [din_width_from_reg] and DIN_CTRL_RESET [din_width_from_cnt] to get the engine into this ULI width mode. In this mode, input data will be written to FIFO along with the length computed from the ULI_BE. For example if uli_be is 0xf (hsize is 2, 32 bit), a value 0x1f is written to FIFO as length token.
   b. Width from reg: Set DIN_CTRL_SET [din_width_from_reg] and DIN_CTRL_RESET [din_width_from_cnt] to get the engine into this Width from reg mode. In this mode, input data will be written to FIFO along with the length from the DIN_CTRL [din_width]. For example if DIN_CTRL [din_width] is 0xf (hsize is 2, 32 bit), a value 0x1f is written to FIFO as length token.
   c. Width from cnt: Set DIN_CTRL_SET [din_width_from_cnt] and DIN_CTRL_RESET [din_width_from_reg] to get the engine into this Width from cnt mode. Write the number of bytes that the packet/data has in total into DIN_NUM_BYTES register. In this mode, input data will be written to FIFO along with the length from either DIN_NUM_BYTES or ULI whichever is less. For example, if DIN_NUM_BYTES is 10 and a ULI_BE 0xf write happened, FIFO will be written with 0x1f as length token and DIN_NUM_BYTES will become 6. If DIN_NUM_BYTES is 1 and a ULI_BE 0x3 write happened, FIFO will be written with 0x7 as length token and DIN_NUM_BYTES will become 0.
5. FIFO indication signals afull and aempty, can be configured through DIN_CTRL register.
6. Write the data into the DIN_FIFO register which is indirectly mapped to FIFO input. Address incremented writes are not supported.
7. Monitor DIN_STS [calc_done] to get set to know the completion of computation of final CRC. In case of din_width_from_cnt, monitor DIN_STS [din_num_bytes_done] also to get set.
8. Read LFSR_STATE register for final CRC.

16.4.4 Register Summary

Table 16.114. Register Summary

Base Address: 0x4508_0000

Register Name	Offset	Description
CRC_GEN_CTRL_SET	0x0	General control set register. Holds general control signals like soft_rst. This register is used only for setting the control signals.
CRC_GEN_CTRL_RESET	0x4	General control reset register. Holds general control signals like soft_rst. This register is used only for resetting the control signals.
CRC_GEN_STS	0x8	General status register. This holds the general status signals.
CRC_POLYNOMIAL	0xC	Polynomial register. This register holds the polynomial that will be used to compute the final CRC.
CRC_POLYNOMIAL_CTRL_SET	0x10	Polynomial control set register. This register holds the control signals for polynomial like the valid length of polynomial. This register access can only set the bits in polynomial control register.
CRC_POLYNOMIAL_CTRL_RESET	0x14	Polynomial control reset register. This register holds the control signals for polynomial like the valid length of polynomial. This register access can only reset the bits in polynomial control register.
CRC_LFSR_INIT_VAL	0x18	LFSR initialization value register. This register holds the initialization value of LFSR.
CRC_LFSR_INIT_CTRL_SET	0x1C	LFSR initialization value control set register. This register holds the control signals for LFSR initialization. This register can only set the bits in the LFSR initialization control register.
CRC_LFSR_INIT_CTRL_RESET	0x20	LFSR initialization value control reset register. This register holds the control signals for LFSR initialization. This register can only reset the bits in the LFSR initialization control register.
CRC_DIN_FIFO	0x24	Input data FIFO is mapped under this address. Anything to be written to FIFO has to be written to this register.
CRC_DIN_CTRL_SET	0x28	Input data FIFO control set register. This holds the control signals required to handle the input data. This register can only set the control signals.
CRC_DIN_CTRL_RESET	0x2C	Input data FIFO control reset register. This holds the control signals required to handle the input data. This register can only reset the control signals.
CRC_DIN_NUM_BYTES	0x30	Input data number of bytes. This holds the number of bytes that will be provided to compute the final CRC in one the input data modes.
CRC_DIN_STS	0x34	Input data status register. This holds the status indication signal like FIFO flow controls.
CRC_LFSR_STATE	0x38	LFSR state register. This holds the final CRC value.

16.4.5 Register Description

Legend:
R = Read-only, W = Write-only, RW = Read/Write, - = Reserved

16.4.5.1 CRC_GEN_CTRL_SET

Table 16.115. GEN_CTRL_SET Register Description

Bit	Access	Function	POR Value	Description
31:1	RW	Reserved	0	Reserved for future use.
0	RW	soft_rst	0*	Soft reset. This clears the FIFO and settles all the state machines to their IDLE. *- indicates self clear bit

16.4.5.2 CRC_GEN_CTRL_RESET

Table 16.116. GEN_CTRL_RESET Register Description

Bit	Access	Function	POR Value	Description
31:0	RW	Reserved	0	Reserved for future use.

16.4.5.3 CRC_GEN_STS

Table 16.117. GEN_STS Register Description

Bit	Access	Function	POR Value	Description
31:2	R	Reserved	0	Reserved for future use.
1	R	din_num_bytes_done	0	When number of bytes requested for computation of final CRC is read from fifo by internal FSM, this will get set to 1, otherwise 0.
0	R	calc_done	0	When the computation of final CRC with the data out of fifo, this will get set to 1, otherwise 0.

16.4.5.4 CRC_POLYNOMIAL

Table 16.118. POLYNOMIAL Register Description

Bit	Access	Function	POR Value	Description
31:0	RW	Polynomial	0	Polynomial register. This register holds the polynomial with which the final CRC is computed.

16.4.5.5 CRC_POLYNOMIAL_CTRL_SET

Table 16.119. POLYNOMIAL_CTRL_SET Register Description

Bit	Access	Function	POR Value	Description
31:5	RW	Reserved	0	Reserved for future use.

Bit	Access	Function	POR Value	Description
4:0	RW	polynomial_width_set	0	Polynomial width set. Number of bits/width of the polynomial has to be written here for the computation of final CRC. If a new width has to be configured, clear the existing length first by writing 0x1f in polynomial_ctrl_reset register.

16.4.5.6 CRC_POLYNOMIAL_CTRL_RESET

Table 16.120. POLYNOMIAL_CTRL_RESET Register Description

Bit	Access	Function	POR Value	Description
31:5	RW	Reserved	0	Reserved for future use.
4:0	RW	polynomial_width_set	0	Polynomial width set. Number of bits/width of the polynomial has to be written here for the computation of final CRC. If a new width has to be configured, clear the existing length first by writing 0x1f in polynomial_ctrl_reset register.

16.4.5.7 CRC_LFSR_INIT_VAL

Table 16.121. LFSR_INIT_VAL Register Description

Bit	Access	Function	POR Value	Description
31:0	RW	LFSR initialization value	0	This holds LFSR initialization value. When ever LFSR needs to be initialized, this has to be updated with the init value and trigger init_lfsr in LFSR_INIT_CTRL_SET register.

16.4.5.8 CRC_LFSR_INIT_CTRL_SET

Table 16.122. LFSR_INIT_CTRL_SET Register Description

Bit	Access	Function	POR Value	Description
31:3	RW	Reserved	0	Reserved for future use.
2	RW	use_swapped_init_val	0	Use bit swapped init value. If this is set bit swapped version of LFSR init value will be loaded / initialized to LFSR state. 1 – use_swapped_init_val will be enabled. 0 – No effect. When read 1 – use_swapped_init_val is enabled. 0 – use_swapped_init_val is disabled.
1	RW	init_lfsr	0*	Initialize LFSR state. When this is set LFSR state will be initialized with LFSR_INIT_VAL/bit swapped LFSR_INIT_VAL in the next cycle. 1 – Initialization will be done in next cycle. 0 – No effect. When read Read 0 always. *- indicates self clear bit

Bit	Access	Function	POR Value	Description
0	RW	clear_lfsr	0*	Clear LFSR state. When this is set, LFSR state is cleared to 0. *- indicates self clear bit

16.4.5.9 CRC_LFSR_INIT_CTRL_RESET

Table 16.123. LFSR_INIT_CTRL_RESET Register Description

Bit	Access	Function	POR Value	Description
31:3	RW	Reserved	0	Reserved for future use.
2	RW	use_swapped_init_val	0	Use bit swapped init value. If this is set bit swapped version of LFSR init value will be loaded / initialized to LFSR state. 1 – use_swapped_init_val will be disabled. 0 – No effect. When read 1 – use_swapped_init_val is enabled. 0 – use_swapped_init_val is disabled.
1	RW	Reserved.	0	Reserved.
0	RW	Reserved.	0	Reserved.

16.4.5.10 CRC_DIN_FIFO

Table 16.124. DIN_FIFO Register Description

Bit	Access	Function	POR Value	Description
31:0	W	DIN_FIFO	0	FIFO input port is mapped to this register. Data on which the final CRC has to be computed has to be loaded to this FIFO.

16.4.5.11 CRC_DIN_CTRL_SET

Table 16.125. DIN_CTRL_SET Register Description

Bit	Access	Function	POR Value	Description
31:28	RW	fifo_afull_threshold	0	FIFO almost full threshold value. This has to be cleared by writing 0xf0000000 into din_ctrl_reset before updating any new value. Bits which are 1s in the wdata[31:28] will be set in threshold value. When read Actual threshold value will be read.
27:24	RW	fifo_aempty_threshold	0	FIFO almost empty threshold value. This has to be cleared by writing 0x0f000000 into din_ctrl_reset before updating any new value. When write Bits which are 1s in the wdata[27:24] will be set in threshold value. When read Actual threshold value will be read.

Bit	Access	Function	POR Value	Description
23:9	RW	Reserved	0	Reserved for future use.
8	RW	reset_fifo_ptrs	0*	Reset fifo pointer. This clears the FIFO. When this is set, FIFO will be cleared. When write 1 – FIFO will be cleared in the next cycle. 0 – No effect. When read Read 0 always. *- indicates self clear bit
7	RW	use_swapped_din	0	Use bit swapped input data. If this is set, input data will be swapped and filled in to FIFO. Whatever read out from FIFO will be directly fed to LFSR engine. When write 1 – Bit swapped data will be filled in to FIFO. 0 – No effect. When read 1 – Bit swapped data is filled in to FIFO. 0 – Direct write data is filled in to FIFO.
6	RW	din_width_from_cnt	0	Valid number of bits in the input data. In default, number of valid bits in the input data is taken from ULI (uli_be). If this is set, a mix of ULI length and number of bytes remaining will form the valid bits (which ever is less that will be considered as valid bits). When write 1 – Din width will be taken from both apb and cnt value. 0 – No effect. When read 1 – Din width is from ULI and cnt value. 0 – Din width doesn't consider cnt value. This overrides the din_width_from_reg.
5	RW	din_width_from_reg	0	Valid number of bits in the input data. In default, number of valid bits in the input data is taken from ULI (uli_be). If this is set, whatever is the input size, only din_ctrl_reg[4:0] is taken as valid length/width for inout data. When write 1 – Din valid width will be taken from reg. 0 – No effect. When read 1 – Din valid width is taken from reg. 0 – Din valid width is not taken from reg.
4:0	RW	din_width_reg	0	Valid number of bits in the input data in din_width_from_reg set mode. Before writing a new value into this, din_ctrl_reset_reg has to be written with 0x1f to clear this field as these are set/clear bits.

16.4.5.12 CRC_DIN_CTRL_RESET

Table 16.126. DIN_CTRL_RESET Register Description

Bit	Access	Function	POR Value	Description
31:28	RW	fifo_afull_threshold	0	FIFO almost full threshold value. This has to be cleared by writing 0xf0000000 into din_ctrl_reset before updating any new value. When write Bits which are 1s in the wdata[31:28] will be reset in threshold value. When read Actual threshold value will be read.
27:24	RW	fifo_aempty_threshold	0	FIFO almost empty threshold value. This has to be cleared by writing 0x0f000000 into din_ctrl_reset before updating any new value. When write Bits which are 1s in the wdata[27:24] will be reset in threshold value. When read Actual threshold value will be read.
23:9	RW	Reserved	0	Reserved for future use.
8	RW	Reserved	0	Reserved.
7	RW	use_swapped_din	0	Use bit swapped input data. If this is set, input data will be swapped and filled in to FIFO. Whatever read out from FIFO will be directly fed to LFSR engine. When write 1 – Direct data will be filled in to FIFO. 0 – No effect. When read 1 – Bit swapped data is filled in to FIFO. 0 – Direct write data is filled in to FIFO.
6	RW	din_width_from_cnt	0	Valid number of bits in the input data. In default, number of valid bits in the input data is taken from ULI (uli_be). If this is set, a mix of ULI length and number of bytes remaining will form the valid bits (which ever is less that will be considered as valid bits). When write 1 – Din width won't consider cnt value. 0 – No effect. When read 1 – Din width is from ULI and cnt value. 0 – Din width doesn't consider cnt value. This overrides the din_width_from_reg.
5	RW	din_width_from_reg	0	Valid number of bits in the input data. In default, number of valid bits in the input data is taken from ULI (uli_be). If this is set, whatever is the input size, only din_ctrl_reg[4:0] is taken as valid length/width for input data. When write 1 – Din valid width will not taken from reg. 0 – No effect. When read 1 – Din valid width is taken from reg. 0 – Din valid width is not taken from reg.
4:0	RW	din_width_reg	0	Valid number of bits in the input data in din_width_from_reg set mode. Before writing a new value into this, din_ctrl_reset_reg has to be written with 0x1F to clear this field as these are set/clear bits.

16.4.5.13 CRC_DIN_NUM_BYTES

Table 16.127. DIN_NUM_BYTES Register Description

Bit	Access	Function	POR Value	Description
31:0	RW	din_num_bytes	0	inout data number of bytes. In din_width_from_cnt mode, this register has to be loaded with the number of bytes that the entire packet contains. Write on to this address will update a down-counter running in the engine. For every ULI write operation on to the FIFO in the din_width_from_cnt mode, this counter will be manipulated by engine.

16.4.5.14 CRC_DIN_STS

Table 16.128. DIN_STS Register Description

Bit	Access	Function	POR Value	Description
31:10	R	Reserved	0	Reserved for future use.
9:4	R	fifo_occ	0	FIFO occupancy.
3	R	fifo_full	0	FIFO full indication status.
2	R	fifo_afull	0	FIFO almost full indication status.
1	R	fifo_aempty	1	FIFO almost empty indication status.
0	R	fifo_empty	1	FIFO empty indication status.

16.4.5.15 CRC_LFSR_STATE

Table 16.129. LFSR_STATE Register Description

Bit	Access	Function	POR Value	Description
31:0	RW	lfsr_state	0	For write operation, if LFSR dynamic loading is required this can be used for writing the LFSR state directly. For read operation, Final CRC/LFSR current state can be read through this address.

16.5 eFuse Controller

16.5.1 General Description

The chipset provides 256 eFuse bits as a one-time programmable memory location. These bits use 32-bit addressing with each address containing 8 bits. The eFuse controller is used to program and read these bits. The 255th eFuse bit is programmed to 1'b1 and tested as part of manufacturing tests. Hence this bit has to be marked as Reserved with a default value to '1'.

16.5.2 Features

- Supports eFuse programming and read operations
- Supports memory mapped and FSM based read operation

16.5.3 Functional Description

The eFuse Controller block diagram is shown below. eFuse controller is used to write or read data from 32 bytes eFuse. The address lines A4~A0 are used to select one from 32 Bytes in eFuse. The remaining address lines A7~A5 are not used during read operation.

8 bits data is read out from eFuse at the same time during sensing. The address lines A7~A5 are used to select one of 8 bits in each byte during write operation. So address lines A7~A0 will select only one bit in eFuse during write operation.

A7~A0 need to be ready before the STROBE pulse is generated to avoid programming of the wrong data.

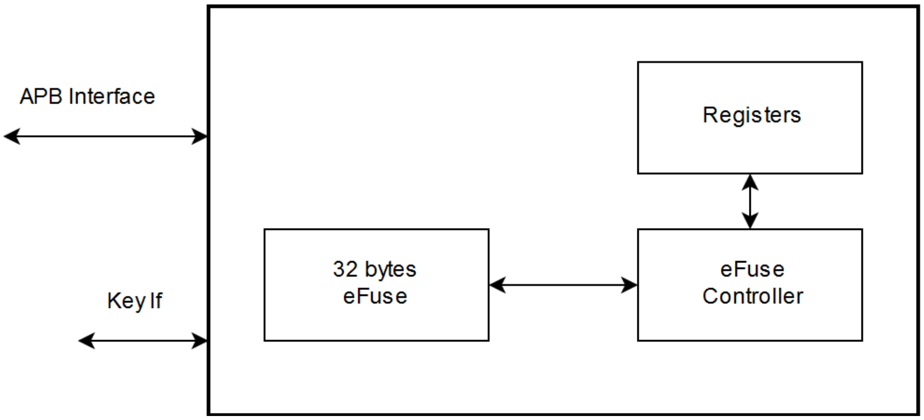


Figure 16.17. eFuse Controller Block Diagram

16.5.3.1 Programming Sequence

Write Operation

Each time, only one eFuse bit cell can be programmed by using this controller. Write operation is possible in direct access mode. The steps to be followed are specified below:

- Program ENABLE_EFUSE_WRITE and EFUSE_DA_ENABLE of EFUSE_CTRL_REG register to one.
- Program 16-bit address in EFUSE_DA_ADDR Register.
- Program CSB, PGENB, STROBE, and LOAD in EFUSE_DA_CTRL_REG with proper delays mention in below program mode timing spec table. Refer to the below example write operation for more details.
- Toggle the signals according to below timing diagram.

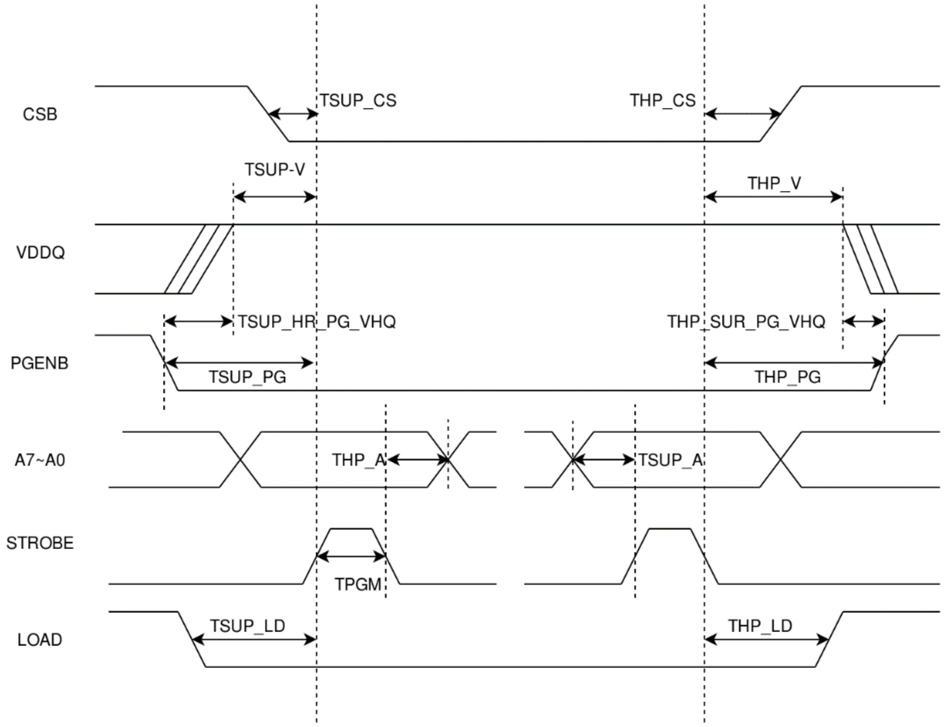


Figure 16.18. eFuse Controller Timing Diagram for Write Mode

Table 16.130. Program Mode Timing Spec

Parameter	Description	Min	Typ	Max	Units
TSUP_HR_ PG_VQ	PGENB falling edge to VDDQ rising edge timing constraint during power on	6	—	—	ns
THP_SUR_ PG_VQ	PGENB rising edge to VDDQ falling edge timing constraint during power on	6	—	—	ns
TSUP_CS	CSB to STROBE setup time into Program mode	6	—	—	ns
THP_CS	CSB to STROBE hold time out of Program mode	6	—	—	ns
TSUP_V (Note 1)	VDDQ to STROBE setup time into Program mode	10	—	—	ns
THP_V	VDDQ to STROBE hold time out of Program mode	10	—	—	ns
TSUP_PG	PGENB to STROBE setup time into Program mode	16	—	—	ns
THP_PG	PGENB to STROBE hold time into Program mode	16	—	—	ns
TPGM	Program mode STROBE pulse width high	1.8	2	2.2	us

Parameter	Description	Min	Typ	Max	Units
TSUP_A	A11~A0 to STROBE setup time into Program mode	6	—	—	ns
THP_A	A11~A0 to STROBE hold time into Program mode	6	—	—	ns
TSUP_LD	LOAD to STROBE setup time into Program mode	6	—	—	ns
THP_LD	LOAD to STROBE hold time into Program mode	6	—	—	ns

Write mode example:

- Set ENABLE_EFUSE_WRITE and EFUSE_DA_ENABLE of EFUSE_CTRL_REG register.
- Set BIT(3), BIT(1) and BIT(0) in EFUSE_DA_CTRL_CLEAR_REG for asserting LOAD,CSB, PGENB.
- Wait 6 ns minimum time.
- Load 8 bit address for M4EFUSE into EFUSE_DA_ADDR_REG.
- Set BIT(2) to assert STROBE signal in EFUSE_DA_CTRL_SET_REG.
- Load Strobe count value and strobe en bit in EFUSE_DA_CLR_STROBE_REG based on APB CLK frequency.

  Example :

  APB Clk =100MHz, Clock period 1/100 = 10 ns

  Delay for TPGM is 2 µs. Calculate no off cycles required for STROBE clear using below formula Delay/Clock period 2000/10 = 200 cycles for 100MHz.Load this value in EFUSE_DA_CLR_STROBE_REG[8:0].
- Poll for Strobe bit clear in EFUSE_STATUS_REG register.
- Set LOAD and CSB bits in EFUSE_DA_CTRL_SET_REG.
- Wait 6 ns minimum time.
- Set PGENB bit in EFUSE_DA_CTRL_SET_REG.
- Clear BIT(2) bit in EFUSE_CTRL_REG.

16.5.3.2 Read Operation

Read operation can be done in one of the following ways as mentioned below.

Indirect access in eFuse controller

- Program EFUSE_RD_TMNG_PARAM_REG according to the apb bus clk frequency.
- Program the 11-bit byte address in EFUSE_READ_ADDR_REG to read data.
- Set EFUSE_ENABLE bit in EFUSE_CTRL_REG.
- Wait for EFUSE_READ_FSM to become one in EFUSE_READ_DATA_REG.
- EFUSE_READ_DATA of EFUSE_READ_DATA_REG will have the 8-bit read data.

Direct access by registers[2]

- To program the eFuse in read mode, drive CSB to low, PGENB to high and LOAD to high. during this read operation, when the read Address A4~A0 is ready, a pulse needs to be sent on the STROBE pin.

  Then read data to be sensed on data lines Q7~Q0. Read data is invalid once STROBE goes high.
- Program EFUSE_DA_ENABLE of EFUSE_CTRL_REG to one.
- Program 11 bit byte address, in EFUSE_DA_ADDR Register (The MSB three bits should be zeros to enable byte addressing).
- Program CSB, PGENB, STROBE, and LOAD in EFUSE_DA_CTRL_REG with proper delays mention in Table 16.131 Read Mode Timing Specification on page 401 . Refer to the below example write operation for more details.
- Toggle the signals according to below shown timing diagram.
- Wait for Strobe bit clear in EFUSE_STATUS_REG[10].
- read 8 bit data from EFUSE_STATUS_REG.

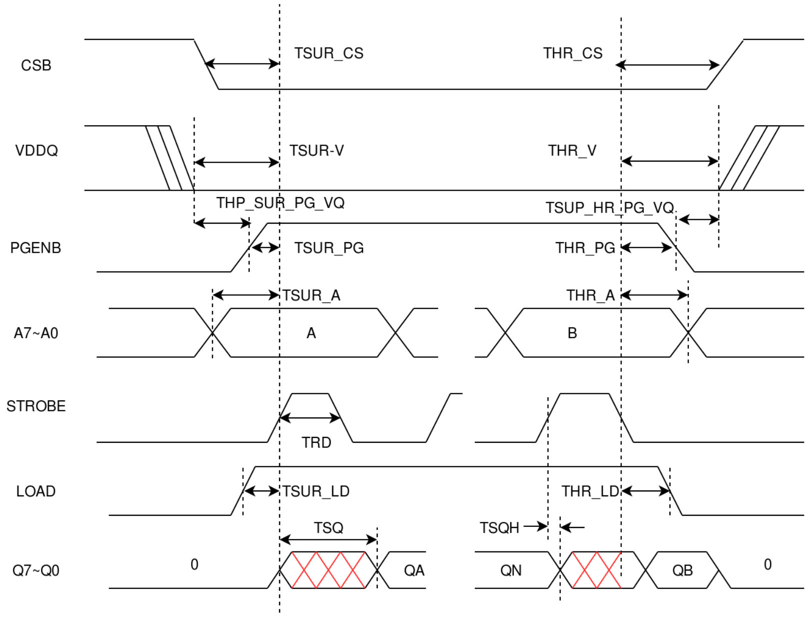


Figure 16.19. eFuse Controller Timing Diagram for Read Mode

Table 16.131. Read Mode Timing Specification

Parameter	Description	Min	Typ	Max	Units
TSUP_HR_ PG_VQ	PGENB falling edge to VDDQ rising edge timing constraint during power on	6	—	—	ns
THP_SUR_ PG_VQ	PGENB rising edge to VDDQ falling edge timing constraint during power on	6	—	—	ns
TSUR_CS	CSB to STROBE setup time into Read mode	6	—	—	ns
THR_CS	CSB to STROBE hold time out of Read mode	6	—	—	ns

Parameter	Description	Min	Typ	Max	Units
TSUR_V	VDDQ to STROBE setup time into Read mode	12	—	—	ns
THR_V	VDDQ to STROBE hold time out of Read mode	12	—	—	ns
TSUR_PG	PGENB to STROBE setup time into Read mode	6	—	—	ns
THR_PG	PGENB to STROBE hold time into Read mode	6	—	—	ns
TRD	Read mode STROBE pulse width high	42	—	—	ns
TSUR_A	A8~A0 to STROBE setup time into Read mode	6	—	—	ns
THR_A	A8~A0 to STROBE hold time into Read mode	6	—	—	ns
TSUR_LD	LOAD to STROBE setup time into Read mode	6	—	—	ns
THR_LD	LOAD to STROBE hold time into Read mode	6	—	—	ns
TSQ	Q7~Q0 access time from STROBE rising edge [with output load: 0.1pF]	—	—	42	ns
TSQH	Q7~Q0 hold time to the next STROBE	—	—	0	ns

Direct read mode example[2]

- Set effuse_ direct_access enable bit in EFUSE_CTRL_REG.
- Set BIT(1) in EFUSE_DA_CTRL_SET_REG for CSB High.
- Set BIT(3), BIT(2) and BIT(0) in EFUSE_DA_CTRL_CLEAR_REG for Lowering LOAD,STROBE, PGENB.
- Wait 6 ns minimum time.
- Set BIT(1) in EFUSE_DA_CTRL_CLEAR_REG for CSB low.
- Load 9 bit address for NWP eFuse 5bit address for M4 eFuse into EFUSE_DA_ADDR_REG.
- Set BIT(2) to set STROBE signal in EFUSE_DA_CTRL_SET_REG.
- Load Strobe count value and strobe en bit in EFUSE_DA_CLR_STROBE_REG based on APB CLK frequency.

Example :

If APB Clk =100MHz Clock period (1/100 = 10nsec)

Delay for TRD is minimum 42 ns. Calculate no off cycles required for STROBE clear using below formula Delay/Clock period (42/10 = 4.2 cycles for 100 MHz) Load result value in EFUSE_DA_CLR_STROBE_REG[8:0].

- Poll for Strobe bit clear in EFUSE_STATUS_REG.
- Read data from EFUSE_STATUS_REG[9:2] bits.
- Set LOAD and PGENB bits in EFUSE_DA_CTRL_SET_REG.
- Wait for 6 ns minimum time.
- Reset CSB in EFUSE_DA_CTRL_SET_REG
- \Memory mapped access.
- Program EFUSE_RD_TMNG_PARAM_REG according to clock frequency used.
- Set EFUSE_ENABLE bit in EFUSE_CTRL_REG.
- In Memory Mapped access, length of the transfer has to be programmed in the EFUSE_MEM_MAP_LEN register with the value either 1(16 Bit Read) or 0 (Default 8 bit Read).
- Address [13] is set then the memory mapped access is enabled else register access will be enabled.
- Address[4:0] are used as eFuse read address.
- Read data is available on APB bus.

16.5.4 Register Summary

Table 16.132. Register Summary

Base Address: 0x4600_C000

Register Name	Offset	Description
EFUSE_DA_ADDR_REG	0x00	Address Register in Direct Access
EFUSE_DA_CTRL_SET_REG	0x04	Control Set Register in Direct Access
EFUSE_DA_CTRL_CLEAR_REG	0x08	Control Clear Register in Direct Access
EFUSE_CTRL_REG	0x0C	eFuse Control Register
EFUSE_READ_ADDR_REG	0x10	Read address Register
EFUSE_READ_DATA_REG	0x14	Read Data Register
EFUSE_STATUS_REG	0x18	eFuse Status Register
EFUSE_RD_TMNG_PARAM_REG	0x1C	Read Timing Parameter Register
EFUSE_MEM_MAP_LENGTH	0x24	Memory Mapped Length Register
EFUSE_READ_BLOCK_STARTING_LOCATION	0x28	Read Block Starting Location Register
EFUSE_READ_BLOCK_END_LOCATION	0x2C	Read Block Ending Location Register
EFUSE_READ_BLOCK_ENABLE	0x30	Read Block Enable Register
EFUSE_DA_CLR_STROBE_REG	0x34	Direct Access Strobe Clearing Count Register

16.5.5 Register Description

Legend:

R = Read-only, W = Write-only, RW = Read/Write, - = Reserved

16.5.5.1 EFUSE_DA_ADDR_REG

Table 16.133. EFUSE_DA_ADDR_REG Description

Bit	Access	Function	Reset Value	Description
[31:16]	-	Reserved	0	Reserved
[15:0]	RW	ADDR_BITS	0	These bits specifies the address to write or read from eFuse macro model.

16.5.5.2 EFUSE_DA_CTRL_SET_REG

Table 16.134. EFUSE_DA_CTRL_SET_REG Description

Bit	Access	Function	Reset Value	Description
[31:4]	R	Reserved	0	Reserved bits
3	RW	Set Load enable (LOAD)	0	1 - Sets eFuse load enable (LOAD) pin when direct accessing is enabled 0 – no effect.
2	RW	STROBE	0	1 - Sets eFuse strobe enable (STROBE) pin when direct accessing is enabled 0 - no effect.
1	RW	Set Chip Enable (CSB)	1	1 - Sets eFuse Chip enable (CSB) pin when direct accessing is enabled 0 – no effect.
0	RW	Set Program enable (PGENB)	1	1 - Sets eFuse program enable (PGENB) pin when direct accessing is enabled 0 – no effect.

16.5.5.3 EFUSE_DA_CTRL_CLEAR_REG

Table 16.135. EFUSE_DA_CTRL_CLEAR_REG Description

Bit	Access	Function	Reset Value	Description
[31:4]	R	Reserved	0	Reserved bits
3	RW	Clear Load enable (LOAD)	0	1 - Clear eFuse load enable (LOAD) pin when direct accessing is enabled 0 – no effect.
2	RW	Clear Strobe enable (STROBE)	0	1 - Clear eFuse strobe enable (STROBE) pin when direct accessing is enabled 0 – no effect.
1	RW	Clear Chip Enable (CSB)	1	1 - Clear eFuse Chip enable (CSB) pin when direct accessing is enabled 0 – no effect.
0	RW	Clear Program enable (PGENB)	1	1 - Clear eFuse program enable (PGENB) pin when direct accessing is enabled 0 – no effect.

16.5.5.4 EFUSE_CTRL_REG

Table 16.136. EFUSE_CTRL_REG Description

Bit	Access	Function	Reset Value	Description
[31:3]	R	Reserved	0	Reserved bits
2	RW	enable_eFuse_write	0	Controls the switch on VDDIQ for eFuse read/write. 0 – VDDIQ is gated 1 – VDDIQ is supplied.
1	RW	eFuse direct path enable	0	This bit specifies whether the eFuse direct path is enabled or not for direct accessing of the eFuse pins 1 – eFuse direct accessing enabled 0 - eFuse direct accessing disabled.
0	RW	eFuse enable	0	This bit specifies whether the eFuse module is enabled or not 1 – eFuse module enabled 0 - eFuse module disabled.

16.5.5.5 EFUSE_READ_ADDR_REG

Table 16.137. EFUSE_READ_ADDR_REG Description

Bit	Access	Function	Reset Value	Description
[31:16]	-	Reserved	0	Reserved
15	W	do_read	1	Enables read FSM after eFuse is enabled.
[14: 12]	R	Reserved	0	Reserved
[11:0]	RW	Read address bits	0	These bits specifies the address from which read operation has to be performed.

16.5.5.6 EFUSE_READ_DATA_REG

Table 16.138. EFUSE_READ_DATA_REG Description

Bit	Access	Function	Reset Value	Description
[31:16]	-	Reserved	0	Reserved
15	R	Read FSM done	0	Indicates read FSM is done. After this read data is available in EFUSE_READ_DATA_REGISTER[7:0].
[14:8]	R	Reserved	0	Reserved bits
[7:0]	RW	Read data bits	0	These bits specifies the data bits that are read from a given address specified in the EFUSE_READ_ADDRESS_REGISTER[8:0].

16.5.5.7 EFUSE_STATUS_REG

Table 16.139. EFUSE_STATUS_REG Description

Bit	Access	Function	Reset Value	Description
[31:11]	R	Reserved	0	Reserved bits
10	R	Strobe Clear bit	0	This bit indicates STROBE signal goes low after strobe count value reached '0'.
9:2	R	eFuse_dout_sync	0	This bit specifies the 8-bit data read out from the eFuse macro. This is synchronized with pclk.
1	R	Reserved	0	Reserved bit
0	R	eFuse Enabled	0	This bit specifies whether the eFuse is enabled or not 1 - eFuse enabled 0 - eFuse not enabled.

16.5.5.8 EFUSE_RD_TMNG_PARAM_REG

Table 16.140. EFUSE_RD_TMNG_PARAM_REG Description

Bit	Access	Function	Reset Value	Description
[31:12]	R	Reserved bits	0	Reserved bits
[11:8]	RW	tHRA	5	for 32x8 macro: A4-A0 to STROBE hold time into Read mode 5122x8 macro: A8-A0 to STROBE hold time into Read mode
[7:4]	RW	tSQ	2	Q7-Q0 access time from STROBE rising edge
[3:0]	RW	tSUR_CS	1	CSB to STROBE setup time into read mode

16.5.5.9 EFUSE_MEM_MAP_LENGTH_REG

Table 16.141. EFUSE_MEM_MAP_LENGTH Description

Bit	Access	Function	Reset Value	Description
[31:1]	R	Reserved bits	0	Reserved bits
0	RW	EFUSE_MEM_MAP_LENGTH_REG	0	0: 8 bit read 1: 16 bit read

16.5.5.10 EFUSE_READ_BLOCK_STARTING_LOCATION

Table 16.142. EFUSE_READ_BLOCK_STARTING_LOCATION Description

Bit	Access	Function	Reset Value	Description
[31:16]	-	Reserved	0	Reserved
[15:0]	RW	eFuse_read_block_start_addr	0	Starting address from which the read has to be blocked. Once the end address is written, it cannot be changed till power on reset is given.

16.5.5.11 EFUSE_READ_BLOCK_END_LOCATION

Table 16.143. EFUSE_READ_BLOCK_END_LOCATION Description

Bit	Access	Function	Reset Value	Description
[31:16]	-	Reserved	0	Reserved
[15:0]	RW	eFuse_read_block_end_addr	0	End address till which the read has to be blocked. Once the end address is written , it cannot be changed till power on reset is given.

16.5.5.12 EFUSE_READ_BLOCK_ENABLE_REG

Table 16.144. EFUSE_READ_BLOCK_ENABLE_REG Description

Bit	Access	Function	Reset Value	Description
[31:1]	R	Reserved bits	0	Reserved bits
0	RW	eFuse_read_block_enable	0	Enable for blocking the read access from a programmable memory location (Start and end address register). Once the blocking is enabled , it cannot be disabled till power on reset is given.

16.5.5.13 EFUSE_DA_CLR_STROBE_REG

Table 16.145. EFUSE_DA_CLR_STROBE_REG Description

Bit	Access	Function	Reset Value	Description
[31:10]	R	Reserved bits	0	Reserved bits
9	RW	eFuse_strobe_en	0	Self clear bit. 1’b1 : Enable Strobe signal
[8:0]	RW	eFuse_strobe_clr_cnt	1	Strobe signal Clear count in direct access mode. value depends on APB clock frequency of eFuse controller.

16.6 Enhanced GPIO (EGPIO)

16.6.1 General Description

The Enhanced GPIO functionality is used to configure the functionality on GPIO pins. There is one instance in MCU HP Domain that is used to control the SoC GPIOs (GPIO_n; n=0:12, 15, 25:34, 46:57) and another in MCU ULP Domain that is used to control the ULP GPIO's (ULP_GPIO_n; n=0:2, 4:11). The features and functionality are the same for both instances except for the Register Base Address.

All GPIO pins in the MCU HP / MCU ULP Domains are grouped into multiple ports. Each port consists of a maximum 16 pins. The ports provide access to multiple GPIO pins at once. They also support set, clear and toggle features. Each pin can be programmed as an output or as an input port for various functions.

16.6.2 Features

The key features of the EGPIO are listed below:

- Supports various alternate functions like set, clear, toggle on all the pins.
- Option to program mode and direction for the each GPIO pin independently.
- Supports edge and level detection based on which interrupts will be raised.
- MCU HP EGPIOs support 8 pins and 4 group interrupts.
- MCU ULP EGPIOs support 8 pin, 2 wakeup, and 2 group interrupts.

16.6.3 Functional Description

The EGPIO encapsulates the GPIO port, interrupt, and configuration register related logic.

The interrupt block encapsulates the group interrupt, wakeup interrupt, and pin interrupt logic.

The port set, clear, toggle, masked load, masked read ,read, bit load, and word load functions are implemented in this block.

The GPIO_n (n=25:30) needs to be enabled before using them as GPIOs as per the functionality described below. These GPIOs can be enabled through the GPIO_25_30_CONFIG_REG Register.

Note that GPIO_n (n=0:5) are dedicated for the Secure Zone Processor's Flash interface. The MCU should NOT be changing any configuration related to these GPIOs under any circumstances since it may lead to the Flash content being corrupted, rendering the chip unusable. This is applicable to MCU HP EGPIO Instance.

16.6.3.1 GPIO Port

GPIO port provides access to multiple GPIO pins. This supports set, clear, toggle, read, masked load, masked read, bit load/read and word load/read functions on the port pins. Five GPIO ports are present in EGPIO. port_load_reg and mask_reg registers are present in each GPIO port. The port_load_reg holds the GPIO data to be put on the GPIO output lines. When a load happens on the APB to any of the set, clear, toggle, etc register address spaces, the bits in the port load register are changed accordingly.

Registers affecting multiple port pins at a time:

- port load
- set
- clear
- toggle
- masked load
- masked read
- port read

Registers affecting single pin at a time:

- bit load/read
- word load/read

Data to be put on GPIO output lines has to be loaded into port load register. When a write is made to the set register, the bits' positions, which are set in the write data, are set in the port load register. When a write is made to the clear register, the bits' positions which are set in the write data are cleared in the port load register. When a write is made to the toggle register, the bits' positions, which are set in the write data, are toggled in the port load register. When a write is made to the masked load register, only the bit positions that are not masked in the mask register are copied to the port load register as is. Other bits are not altered. If a bit in the mask register is set to ‘0’, it is masked. Reading a masked load register provides the status of the GPIO input lines that are not masked in mask register. Other bits appear as ‘0’.

The bit load and word load registers affect only a single pin at a time. When bit load register of a pin is written, the value in the 0th position of the bit load write data is loaded into the corresponding bit position in the port load register. When a non-zero value is written into the word load register of a pin, the corresponding pin bit in the port load register is set. When a zero is written, the corresponding bit is reset.

16.6.3.2 Programming sequence

The following is the programming sequence to use EGPIOs.

1. Clock enable:
    a. Enable clock as described in MCU HP Clock Architecture or MCU ULP Clock Architecture Sections.
2. GPIO-OEN(Direction)/MODE
    a. Program mode of all GPIOs as 0. Program 5:2 bits as 0 in all GPIO_CONFIG_REG_* registers.
    b. Program OEN (BIT(0) as 0 for all GPIO_REG_* registers.
3. GPIO set/clear/toggling
    a. Program PORT_SET_REG_n for all the 4 ports. This will set all GPIOs output as 1.
    b. Program PORT_CLEAR_REG_n for all the 4 ports as 0xFFFF. This will set all GPIOs output as 0.
4. REN programming
    a. Program REN of all the GPIO's to be 1 in Pad Configuration registers.
    b. ULP GPIO's REN has to be programmed as per ULP GPIO Pad Configuration registers.
5. GPIO Status Read:
    a. Program OEN BIT(0) as 1 for all GPIO registers. EGPIO are programmed in input mode.
    b. Poll for PORT_READ_REG_n to get the proper GPIO status.
    c. Follow the same procedure all bits for all 4 EGPIO ports to get the status.

Following is the programming sequence to use EGPIOs in open drain mode.

1. Clock enable:
    a. Enable clock as described in MCU HP Clock Architecture or MCU ULP Clock Architecture Sections.
2. GPIO-OEN(Direction)/MODE
    a. Program mode of all GPIOs as 0. Program 5:2 bits as 0 in all GPIO_CONFIG_REG_* registers
    b. Program REN of all the GPIO's to be 0 in Pad Configuration registers.
    c. Program OEN (BIT(0) as 1 for all GPIO_REG_* registers. Open drain is observed on all GPIOs output.
    d. Check the status of the pin. Open drain (if pull up is added on pin, HIGH is observed) is observed.
3. GPIO clear
    a. Program PORT_CLEAR_REG_n for all the 4 ports as 0xFFFF.
    b. Program OEN (BIT(0) as 0 for all GPIO_REG_* registers.
    c. Check the status of the pin. LOW is observed on all GPIOs output

16.6.3.3 GPIO interrupt generation

The GPIO inputs monitor and generates interrupts when the programmed pattern is detected. It generates 8 pin interrupts, 4 group interrupts and 4 wakeup interrupts.

Registers present for interrupt generation are as follows:

- pin_intr_ctrl –> Control registers for 8 pin interrupts
- pin_intr_sts –> Status registers for pin interrupts.
- group_intr_ctrl --> Control registers for 4 group and 4 wakeup interrupts
- group_intr_sts --> Status registers for 4 group and 4 wakeup interrupts

The pin interrupt control registers provide enables and mask for each pin interrupt. Any of the 8 GPIO pins can be mapped on these 8 pin interrupts. The GPIO pin to be mapped to the interrupt line has to be configured in the pin_intr_ctrl register of the respective interrupt pin. Each of these GPIOs are monitors and raises an interrupt when the programmed pattern is detected. For pin interrupts, rise edge, fall edge and level interrupt status is mapped to the respective pin’s status register.

The group and the wakeup interrupts are both generated by monitoring the status of multiple pins. The difference between the interrupts lies in the GPIO input pins considered for interrupt generation. The group interrupt generation is based on synchronized interrupt pins where as wakeup interrupt generation considers unsynchronized pins. The control information as to which GPIO lines to consider for interrupts generation is present in the group interrupt control registers for both group and wakeup interrupts. The masking logic is also present. The enables for the GPIO pins to be considered and the polarity of the pin that contributes to interrupt are provided. For group interrupts, unmasked versions of the group interrupts and synchronized version of the wakeup interrupts are mapped to status registers. The wakeup interrupt itself is generated based on unsynchronized pins (i.e., clock need not be present for the generation of this interrupt). When clock is not present, the bit in the status register will not be updated. If the wakeup condition persists on the GPIO input lines till the clock is provided, then the synchronized wakeup interrupt bit gets updated. wakeup interrupt can be used to wake up the chip from sleep state.

16.6.4 Register Summary

Table 16.146. Register Summary Table

Base Address : 0x4600_8000

Register Name	Offset	Description
MCR_HOST_CTRL_REG	0x0C	Configuration Register for GPIO_n(n=25:30)

Table 16.147. Register Summary Table

Base Address for MCU HP Instance: 0x4613_0000; Base Address for MCU ULP Instance: 0x2404_C000

Register Name	Offset	Description
GPIO_CONFIG_REG_0	0x00	GPIO Configuration Register 0
GPIO_CONFIG_REG_1	0x10	GPIO Configuration Register 1
GPIO_CONFIG_REG_2	0x20	GPIO Configuration Register 2
GPIO_CONFIG_REG_3	0x30	GPIO Configuration Register 3
GPIO_CONFIG_REG_4	0x40	GPIO Configuration Register 4
GPIO_CONFIG_REG_5	0x50	GPIO Configuration Register 5
GPIO_CONFIG_REG_6	0x60	GPIO Configuration Register 6
GPIO_CONFIG_REG_7	0x70	GPIO Configuration Register 7
BIT_LOAD_REG_0	0x4	Bit Load Register 0
BIT_LOAD_REG_1	0x14	Bit Load Register 1
BIT_LOAD_REG_2	0x24	Bit Load Register 2
BIT_LOAD_REG_3	0x34	Bit Load Register 3
BIT_LOAD_REG_4	0x44	Bit Load Register 4
BIT_LOAD_REG_5	0x54	Bit Load Register 5
BIT_LOAD_REG_6	0x64	Bit Load Register 6
BIT_LOAD_REG_7	0x74	Bit Load Register 7
WORD_LOAD_REG_0	0x8	Word Load Register 0
WORD_LOAD_REG_1	0x18	Word Load Register 1
WORD_LOAD_REG_2	0x28	Word Load Register 2
WORD_LOAD_REG_3	0x38	Word Load Register 3
WORD_LOAD_REG_4	0x48	Word Load Register 4
WORD_LOAD_REG_5	0x58	Word Load Register 5
WORD_LOAD_REG_6	0x68	Word Load Register 6
WORD_LOAD_REG_7	0x78	Word Load Register 7
PORT_LOAD_REG	0x1000	Port Load Register
PORT_SET_REG	0x1004	Port Set Register
PORT_CLEAR_REG	0x1008	Port Clear Register
PORT_MASKED_LOAD_REG	0x100c	Port Masked Load Register

Register Name	Offset	Description
PORT_TOGGLE_REG	0x1010	Port Toggle Register
PORT_READ_REG	0x1014	Port Read Register
GPIO_INTR_CTRL_0	0x1200	GPIO Interrupt Control Register 0
GPIO_INTR_CTRL_1	0x1208	GPIO Interrupt Control Register 1
GPIO_INTR_CTRL_2	0x1210	GPIO Interrupt Control Register 2
GPIO_INTR_CTRL_3	0x1218	GPIO Interrupt Control Register 3
GPIO_INTR_CTRL_4	0x1220	GPIO Interrupt Control Register 4
GPIO_INTR_CTRL_5	0x1228	GPIO Interrupt Control Register 5
GPIO_INTR_CTRL_6	0x1230	GPIO Interrupt Control Register 6
GPIO_INTR_CTRL_7	0x1238	GPIO Interrupt Control Register 7
GPIO_INTR_STAT_0	0x1204	GPIO Interrupt Status Register 0
GPIO_INTR_STAT_1	0x120C	GPIO Interrupt Status Register 1
GPIO_INTR_STAT_2	0x1214	GPIO Interrupt Status Register 2
GPIO_INTR_STAT_3	0x121C	GPIO Interrupt Status Register 3
GPIO_INTR_STAT_4	0x1224	GPIO Interrupt Status Register 4
GPIO_INTR_STAT_5	0x122C	GPIO Interrupt Status Register 5
GPIO_INTR_STAT_6	0x1230	GPIO Interrupt Status Register 6
GPIO_INTR_STAT_7	0x123C	GPIO Interrupt Status Register 7
GPIO_GRP_INTR_CTRL_0	0x1240	GPIO Group Interrupt Control Register 0
GPIO_GRP_INTR_STS_0	0x1244	GPIO Group Interrupt Status Register 0
GPIO_GRP_INTR_CTRL_1	0x1248	GPIO Group Interrupt Control Register 1
GPIO_GRP_INTR_STS_1	0x124C	GPIO Group Interrupt Status Register 1
GPIO_GRP_INTR_CTRL_2	0x1250	GPIO Group Interrupt Control Register 2
GPIO_GRP_INTR_STS_2	0x1254	GPIO Group Interrupt Status Register 2
GPIO_GRP_INTR_CTRL_3	0x1258	GPIO Group Interrupt Control Register 3
GPIO_GRP_INTR_STS_3	0x125C	GPIO Group Interrupt Status Register 3
Note: GPIOs 13,14,16-24, 35-45, 58-63 and 76-79 are not used in **9117**.		

16.6.5 Register Description

16.6.5.1 MCR_HOST_CTRL_REG

Table 16.148. MCR_HOST_CTRL_REG Description

Bit	Access	Function	Reset Value	Description
31:16	NA	NA	0x0	Reserved and read as zero
15	R/W	sdio_spi_prog_sel	0x1	SDIO/SPI registers share the same based address. Based on this bit MCU can access either SDIO register or SPI registers 1 - MCU can access SDIO registers 0 - MCU can access SPI registers
14	R/W	load_host_mode_2	0x0	Overrides the Hardware detected Host on 2nd interface. 0 - H/W based Host detection 1 - S/W based Host (Depends on host_sel_2 mentioned below)
13:12	R/W	host_sel_2	0x0	Selects the Host on 2nd Interface 0 - No HOST Mode 1 - USB is selected 2, 3 - Reserved
11	NA	NA	0x0	Reserved
10	R/W	load_host_mode	0x0	Overrides the Hardware detected Host on 1st interface. 0 - H/W based Host detection 1 - S/W based Host (Depends on host_sel mentioned below)
9:8	R/W	host_sel	0x0	Selects the Host on 1st Interface 0 - No HOST Mode 1 - Reserved 2 - Host SPI is selected 3 - SDIO secondary is selected
7:5	NA	NA	0x0	Reserved
4	R	host_spi_bus_err_in	0x0	Host SPI Bus error input. 0 means there is no error on Host SPI transmit transaction 1 means there is a error on Host SPI transmit transaction
3	R/W	host_spi_bus_err_out	0x0	Host SPI Bus error output. 0 means there is no error on Host SPI receive transaction 1 means there is error in Host SPI receive transaction
2	R/W	host_spi_bus_err_oen	0x1	Host SPI Bus error output enable. It is active low signal. 0 means SPI Bus error output is enabled. 1 means SPI Bus error output is disabled.

Bit	Access	Function	Reset Value	Description
1	R/W	host_spi_poweron_rst	0x0	This is an active high reset which is used to generate host reset. This has to be enabled only in SPI Host Mode.
0	R/W	ready_from_core	0x0	Indication to the host that bootloading is done. When the reset latch bootload_en is '0' , firmware sets this bit. When hardware bootloading is enabled , this gets set only after bootloading is done. When hardware bootloading is enabled and bootloader is programmed to release the PC from soft reset, firmware sets this bit.

16.6.5.2 GPIO_CONFIG_REG_n (Offset Address = 0x00 + n*10)

Table 16.149. GPIO_CONFIG_REG_n Description

Bit	Access	Function	Reset Value	Description
15:12	R	Reserved	0x0	Reserved
11	R/W	Group Interrupt 2 polarity	0	Decides the active value of the pin to be considered for group interrupt 2 generation when enabled'0' – group interrupt gets generated when gpio input pin status is zero'1' – grp interrupt gets generated when gpio input pin status is '1'.
10	R/W	Group Interrupt 2 enable	0	When set, the corresponding GPIO is pin is selected for group intr 2 generation.
9	R/W	Group Interrupt 1 Polarity	0	Decides the active value of the pin to be considered for group interrupt 1 generation when enabled'0' – group interrupt gets generated when gpio input pin status is zero'1' – grp interrupt gets generated when gpio input pin status is '1'.
8	R/W	Group Interrupt 1 enable	0	When set, the corresponding GPIO is pin is selected for group intr 1 generation.
7:6	R	Reserved	0x0	Reserved
5:2	R/W	Mode	0x0	GPIO Pin Mode. Ranges 0000 -> Mode 0 to 1111 -> Mode 15 Used for GPIO Pin Muxing.
1	R/W	Port Mask	0	Port mask value '1' – When set, pin is masked when written/read through PORT MASK REG.
0	R/W	Direction	1	Direction of the GPIO pin. '1' – INPUT, '0'-OUTPUT.

16.6.5.3 BIT_LOAD_REG_n (Offset Address = 0x4 + n*10)

Table 16.150. BIT_LOAD_REG_n Description

Bit	Access	Function	Reset Value	Description
0	R/W	Bit Load	N/A	Loads 0th bit on to the pin on write. And reads the value on pin on read into 0th bit.

16.6.5.4 WORD_LOAD_REG_n (Offset Address = 0x8 + n*10)

Table 16.151. WORD_LOAD_REG_n Description

Bit	Access	Function	Reset Value	Description
15:0	R/W	Word Load	N/A	Loads 1 on the pin when any of the bit in load value is 1. On read, pass the bit status into all bits.

16.6.5.5 PORT_LOAD_REG_n (Offset Address = 0x1000+(N*0x40))

Table 16.152. PORT_LOAD_REG_n Description

Bit	Access	Function	Reset Value	Description
15:0	R/W	Port Load	0x0	Loads the value on to pin on write. And reads the value of load register on read.

16.6.5.6 PORT_MASKED_LOAD_REG_n (Offset Address = 0x100C+(N*0x40))

Table 16.153. PORT_MASKED_LOAD_REG_n Description

Bit	Access	Function	Reset Value	Description
15:0	R/W	Port Masked Load	0x0	Only loads into pins which are not masked. On read, pass only status unmasked pins(Mask present in GPIO CONFIG REG).

16.6.5.7 PORT_SET_REG_n (Offset Address = 0x1004+(N*0x40))

Table 16.154. PORT_SET_REG_n Description

Bit	Access	Function	Reset Value	Description
15:0	W	Port Set	N/A	Sets the pin when corresponding bit is high. Writing zero has no effect.

16.6.5.8 PORT_CLEAR_REG_n (Offset Address = 0x1008+(N*0x40))

Table 16.155. PORT_CLEAR_REG_n Description

Bit	Access	Function	Reset Value	Description
15:0	W	Port Clear	N/A	Clears the pin when corresponding bit is high. Writing zero has no effect.

16.6.5.9 PORT_TOGGLE_REG_n (Offset Address = 0x1010+(N*0x40))

Table 16.156. PORT_TOGGLE_REG_n Description

Bit	Access	Function	Reset Value	Description
15:0	W	Port Toggle	N/A	Toggles the pin when corresponding bit is high. Writing zero has not effect.

16.6.5.10 PORT_READ_REG_n (Offset Address = 0x1014+(N*0x40))

Table 16.157. PORT_READ_REG_n Description

Bit	Access	Function	Reset Value	Description
15:0	R	Port Read (Input)	0x0	Reads the value on GPIO pins irrespective of the pin mode.

16.6.5.11 GPIO_INTR_CTRL_REG_n (Offset Address= 0x1200 + n*8)

Table 16.158. GPIO_INTR_CTRL_REG_n Description

Bit	Access	Function	Reset Value	Description
13:12	R/W	Port Number	0x0	GPIO Port to be chosen for interrupt generation.
11:8	R/W	Pin Number	0x0	GPIO Pin to be chosen for interrupt generation.
7:5	R	Reserved	0x0	Reserved
4	R/W	Mask	1	Masks the interrupt. Interrupt will still be seen in status register when enabled'1' – intr masked'0' – intr unmasked.
3	R/W	Fall Edge Enable	0	enables interrupt generation when falling edge is detected on pin'1' – intr enabled'0' – disabled.
2	R/W	Rise Edge Enable	0	enables interrupt generation when rising edge is detected on pin'1' – intr enabled '0' – disabled.
1	R/W	Level Low Enable	0	enables interrupt generation when pin level is '0''1' – intr enabled'0' – disabled.
0	R/W	Level High Enable	0	enables interrupt generation when pin level is '1''1' – intr enabled'0' – disabled.

16.6.5.12 GPIO_INTR_STATUS_REG_n (Offset Address = 0x1204+8*N)

Table 16.159. GPIO_INTR_STATUS_REG_n Description

Bit	Access	Function	Reset Value	Description
4	W	Mask Clear	0	When 1 is written mask bit gets cleared. On read, this bit should result it in 0.
3	W	Mask Set	0	When 1 is written mask bit will get set. On read, this bit should result it in 0.
2	R/W	Fall Edge Status	0	Gets set when fall edge is enabled and occurs. When 1 is written it gets cleared. Writing 0 has not effect.
1	R/W	Rise Edge Status	0	Gets set when rise edge is enabled and occurs. When 1 is written it gets cleared. Writing 0 has not effect.
0	R/W	Interrupt Status	0	Gets set when interrupt is enabled and occurs. When 1 is written it gets cleared. Also clears rise edge and fall edge status bits. Writing 0 has not effect.

16.6.5.13 GPIO_GRP_INTR_n_CTRL_REG (Offset Address = 0x1240+N*8)

Table 16.160. GPIO_GRP_INTR_n_CTRL_REG Description

Bit	Access	Function	Reset Value	Description
4	R/W	Mask	1	1-mask,0-unmask (Not used for wakeup interrupts)
3	R/W	Enable Interrupt	0	1-enable normal group interrupt, 0-disable normal group interrupt. (Wakeup interrupt is unaffected by this bit.)
2	R/W	Enable Wakeup	0	For wakeup generation, actual pin status has to be seen (before double ranking point). Set to 1 if grp interrupt has to used as wakeup source.
1	R/W	Level/Edge	0	1- Edge(cannot be used as wakeup) 0 - Level
0	R/W	AND/OR	0	0 - AND 1- Or

16.6.5.14 GPIO_GRP_INTR_n_STATUS_REG (Offset Address = 0x1244+N*8)

Table 16.161. GPIO_GRP_INTR_n_STATUS_REG Description

Bit	Access	Function	Reset Value	Description
4	W	Mask Clear	0	Clear bit version of Mask bit in PORT_GRP_INTR_n_CTRL_REG_N. Gives zero on read.
3	W	Mask Set	0	Set bit version of Mask bit in PORT_GRP_INTR_n_CTRL_REG_N. Gives zero on read.
2	N/A	Reserved	0	Reserved
1	R	Wakeup	0	Double ranked version of wakeup. Gets set when wakeup is enabled and occurs.
0	R/W	Interrupt Status	0	Interrupt status is available in this bit when interrupt is enabled and generated. When '1' is written, interrupt gets cleared.

16.7 Generic SPI Primary

16.7.1 General Description

The Generic SPI Primary is present in MCU HP peripherals. It provides an I/O interface to a wide variety of SPI compatible peripheral devices. SPI is a synchronous four-wire interface consisting of two data pins (MOSI, MISO), a device select pin (CSN), and a gated clock pin (SCLK). With the two data pins, it allows for full-duplex operation to other SPI compatible devices. Typical SPI compatible peripheral devices that can be used to interface are as follows:

- LCD displays
- A/D converters
- D/A converters
- Codecs
- Microcontrollers
- Flashes

16.7.2 Features

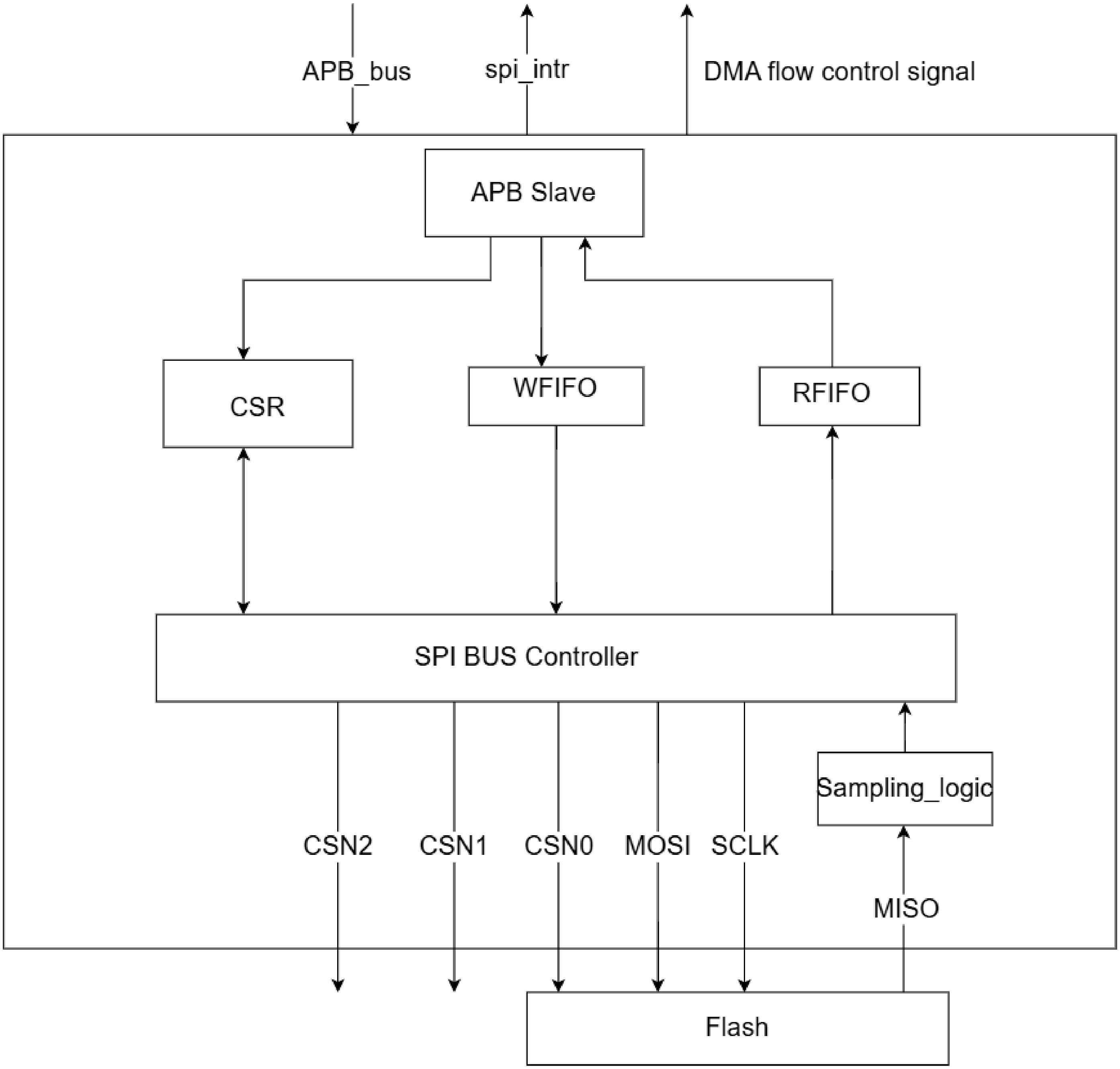
- Supports full duplex Single-bit SPI Primary mode.
- Support for Mode-0 and Mode-3 (Motorola).
- Supports both Full speed and High speed modes.
- Connect up to three SPI peripheral devices.
- SPI clock out is programmable to meet required baud rates.
- Generates interrupt for different write FIFO and read FIFO status .

  The following features of the Generic SPI Primary help reducing the load on the processor:
  - Support upto 32K bytes of read data from a SPI device in a single read operation.
  - Support for byte-wise swapping of read and write data*.
  - Programmable FIFO thresholds with maximum FIFO depth of 16 and support for DMA.

16.7.3 Functional Description

SPI has an option to select three SPI devices by using respective CSN. CSN is the active low signal. It is asserted during either write operation or read operation. The SCLK signal is a gated clock that is only active during data transfers for the duration of the transferred word. The number of active edges is equal to the number of bits driven on the data lines. The clock rate is determined by the 8-bit value of GSPI_CLK_DIV register. SCLK, CSN and MOSI are output signals and MISO is input signal. The SCLK signal is used to shift out and shift in the data driven onto the MISO and MOSI lines. The data is always shifted out on neg-edge of the clock and sampled on the either pos-edge or neg-edge of the clock depending on full-speed mode or high speed mode respectively. Full-speed and High-speed modes are configured using GSPI_BUS_MODE[0] before start of any SPI transactions. Similarly, mode0 and mode3 (clock polarity) are programmable by asserting bit in GSPI_BUS_MODE register initially. SPI controller block diagram is shown below.

Figure 16.20. General SPI Block Diagram



16.7.3.1 Programming Sequence

SPI supports read and write operations. Before starting any operation, the SPI bus busy signal has to be checked by polling SPI_STATUS[0] register bit. Then program beat size, number of bytes to read (only for read operation), select CSN to drive, Assert CSN, set read or write bits in GSPI registers.

Write Operation

- Assert respective CSN in GSPI_CONFIG1[0].
- Write command, address and write data in WRITE_FIFO register.
- Write valid number of bits into the GSPI_WRITE_DATA2 register.
- Option to enable USE_PREV_LNTH bit in GSPI_WRITE_DATA2 register to use present programmed length for further write operations also.
- Write valid number of bytes minus one into GSPI_CONFIG1 and set the bit 'TAKE_WR_SIZE_FRM_REG' to consider this byte ordered length to write into flash.
- Assert write operation bit in GSPI_CONFIG1[1].
- Write data either in DMA mode or IO mode.
- After writing complete data in WRITE_FIFO, SPI bus busy signal has to be checked by polling SPI_STATUS[0] register bit.
- Once SPI bus is idle, de-assert the respective CSN.

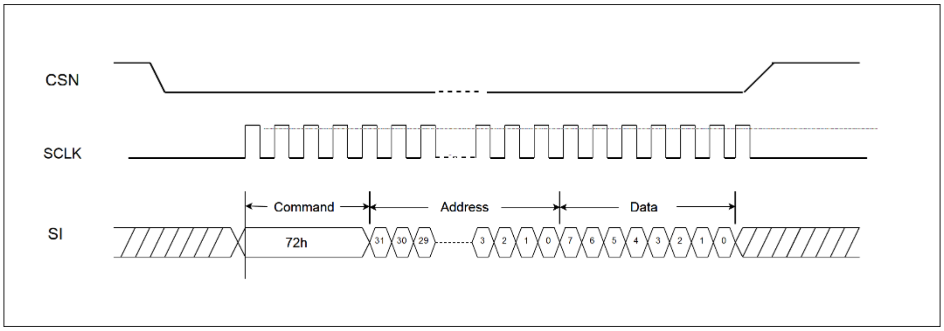


Figure 16.21. SPI Bus Write Operation Waveform

Read Operation

- Assert respective CSN in GSPI_CONFIG1[0].
- Write command and address in WRITE_FIFO register.
- Assert write operation bit in GSPI_CONFIG1[1].
- check for wfifo empty status.
- de-assert write operation bit in GSPI_CONFIG1[1].
- Write the number of bytes to read in GSPI_CONFIG1[12:3] and keep the number of bytes minus one in GSPI_CONFIG1 and trigger read operation by asserting GSPI_CONFIG1[2] bit. Poll for read fifo not empty and read from the READ_FIFO register.
- Read data either in DMA mode or IO mode.
- After reading complete data, de-assert the respective CSN.
- After completion of read/write operation, GSPI controller raises an interrupt.

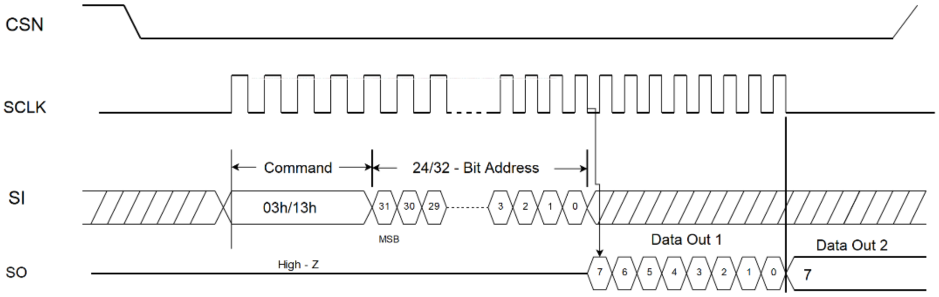


Figure 16.22. SPI Bus Read Operation Waveform

The SPI bus interface waveform is shown in the figure above. Command, address, and write data are initialed by MCU or DMA in write mode. There may be a gap between command, address writing on SPI bus and read data from SPI bus. Because these instructions are issued by MCU firmware directly. Once read is issued, there is no any overhead clock cycles for reading data from flash/device up to end of burst transfer unless read data FIFO is full. it supports up to 32K bytes of data from SPI device in a single read operation. There is no any overhead clock cycles between command, address and write data if data is available in GSPI_WRITE_FIFO before completion of current SPI operation in write mode.

16.7.4 Register Summary

Table 16.162. Register Summary Table

Base Address: 0x4503_0000

Register Name	Offset	Description
GSPI_CLK_CONFIG	0x00	GSPI Clock Configuration Register
GSPI_BUS_MODE	0x04	GSPI Bus Mode Register
GSPI_CONFIG1	0x10	GSPI Configuration 1 Register
GSPI_CONFIG2	0x14	GSPI Configuration 2 Register
GSPI_WRITE_DATA2	0x18	GSPI Write Data 2 Register
GSPI_FIFO_THRLD	0x1C	GSPI FIFO Threshold Register
GSPI_STATUS	0x20	GSPI Status Register
GSPI_INTR_MASK	0x24	GSPI Interrupt Mask Register
GSPI_INTR_UNMASK	0x28	GSPI Interrupt Unmask Register
GSPI_INTR_STS	0x2C	GSPI Interrupt Status Register
GSPI_INTR_ACK	0x30	GSPI Interrupt Acknowledge Register
GSPI_STS_MC	0x34	GSPI State Machine Monitor Register
GSPI_CLK_DIVISION_FACTOR	0x38	GSPI Clock Division Factor Register
GSPI_CONFIG3	0x3C	GSPI Configuration 3 Register
GSPI_WRITE_FIFO	0x80-0xBC	GSPI Write Data FIFO Register
GSPI_READ_FIFO	0x80-0xBC	GSPI Read Data FIFO Register

16.7.5 Register Description

Legend:

R = Read-only, W = Write-only, R/W = Read/Write, - = Reserved

16.7.5.1 GSPI_CLK_CONFIG

Table 16.163. GSPI_CLK_CONFIG Description

Bit	Access	Function	Reset Value	Description
31:2	R/W	Reserved	1	Reserved
1	R/W	GSPI_CLK_EN	0	GSPI clock enable. 0 – Dynamic clock gating is enabled in side GSPI controller. 1 – Full time clock is enabled for GSPI controller.
0	R/W	GSPI_CLK_SYNC	0	If the clock frequency to FLASH (spi_clk) and SOC clk is same. 1: SCLK clock and SOC clock are same. 0: Divided SOC clock is connected SCLK. Division value is programmable.

16.7.5.2 GSPI_BUS_MODE

Table 16.164. GSPI_BUS_MODE Description

Bit	Access	Function	Reset Value	Description
31:12	R/W	Reserved	0	Reserved
11	R/W	SPI_HIGH_PERFORMANCE_EN	0	High performance features are enabled when this bit is set to one.
10:5	R/W	GSPI_GPIO_MODE_ENABLES	0	These bits are used to map GSPI on GPIO pins. For more details go through GSPI mapping on GPIO section.
3	R/W	GSPI_CLK_MODE_CSN2	0	0 – Mode 0, GSPI_CLK is low when GSPI_CS is high for chip select2 (csn2)1 – Mode 3, GSPI_CLK is high when GSPI_CS is high for chip select2 (csn2).
2	R/W	GSPI_CLK_MODE_CSN1	0	0 – Mode 0, GSPI_CLK is low when GSPI_CS is high for chip select1 (csn1)1 – Mode 3, GSPI_CLK is high when GSPI_CS is high for chip select1 (csn1).
1	R/W	GSPI_CLK_MODE_CSN0	0	0 – Mode 0, GSPI_CLK is low when GSPI_CS is high for chip select0 (csn0)1 – Mode 3, GSPI_CLK is high when GSPI_CS is high for chip select0 (csn0).
0	R/W	GSPI_DATA_SAMPLE_EDGE	0	Samples MISO data on clock edges. This should be ZERO for mode3 clock. 0 – Posedge of loop back spi_pad_clk. 1 – Negedge of loop back spi_pad_clk.

16.7.5.3 GSPI_CONFIG1

Table 16.165. GSPI_CONFIG1 Description

Bit	Access	Function	Reset Value	Description
31:16	R/W	Reserved	0	Reserved
15	R/W	SPI_FULL_DUPLEX_EN	0	Full duplex mode enable. 0 – Full duplex mode disabled. 1 – Full duplex mode enabled. Full duplex mode means reading while writing. When this bit is enabled, while writing the data to Secondary connected to GSPI controller, reads the data from Slave selected among the Secondary connected to GSPI controller and stores in read_fifo. This fifo will get automatically flush after 16 reads and the fifo is not empty to write into.
14:13	R/W	GSPI_MANUAL_CSN_SELECT	0	Indicates which CSn is valid. Can be programmable in manual mode.
12:3	R/W	GSPI_MANUAL_RD_CNT	0	Indicates total number of bytes to be read.
2	R/W	GSPI_MANUAL_RD	0	Read enable for manual mode when CS is low.
1	R/W	GSPI_MANUAL_WR	0	Write enable for manual mode when CS is low.
0	R/W	GSPI_MANUAL_CSN	1	SPI CS in manual mode.

16.7.5.4 GSPI_CONFIG2

Table 16.166. GSPI_CONFIG2 Description

Bit	Access	Function	Reset Value	Description
31:11	R/W	Reserved	0	Reserved
10	R/W	TAKE_GSPI_MANUAL_WR_SIZE_FRM_REG	0	1 – Take write size from Manual config register1[20:19]. 0 – No action. Takes write size from fifo [19:16].
9	R/W	Reserved	0	Reserved
8	R/W	GSPI_MANUAL_SIZE_FRM_REG	1	Manual reads and manual writes (If take_manual_size_from_reg bit is 1) follow this size. 0 – 1 Byte (8 – bit mode) 1 – 2 Bytes (16 – bit mode)
7	R/W	Reserved	1	Reserved
6	R/W	GSPI_RD_DATA_SWAP_MNL_CSN2	1	Swap the read data inside the GSPI controller it-self. 0 – Manual read data swap is disabled for csn2. 1 – Manual read data swap is enabled for csn2.
5	R/W	GSPI_RD_DATA_SWAP_MNL_CSN1	1	Swap the read data inside the GSPI controller it-self. 0 – Manual read data swap is disabled for csn1. 1 – Manual read data swap is enabled for csn1.
4	R/W	GSPI_RD_DATA_SWAP_MNL_CSN0	1	Swap the read data inside the GSPI controller it-self. 0 – Manual read data swap is disabled for csn0. 1 – Manual read data swap is enabled for csn0.
3	R/W	Reserved	0	Reserved
2	R/W	GSPI_WR_DATA_SWAP_MNL_CSN2	0	Swap the write data inside the GSPI controller it-self. 0 – Manual write data swap is disabled for csn2. 1 – Manual write data swap is enabled for csn2.
1	R/W	GSPI_WR_DATA_SWAP_MNL_CSN1	0	Swap the write data inside the GSPI controller it-self. 0 – Manual write data swap is disabled for csn1. 1 – Manual write data swap is enabled for csn1.
0	R/W	GSPI_WR_DATA_SWAP_MNL_CSN0	0	Swap the write data inside the GSPI controller it-self. 0 – Manual write data swap is disabled for csn0. 1 – Manual write data swap is enabled for csn0.

16.7.5.5 GSPI_WRITE_DATA2

Table 16.167. GSPI_WRITE_DATA2 Description

Bit	Access	Function	Reset Value	Description
31:8	R/W	Reserved	0	Reserved
7	R/W	USE_PREV_LENGTH	0	Use previous length. 1 – Uses previously programmed length in [3:0] of this register for next writes. 0 – No action. Note: TAKE_WR_SIZE_FRM_REG bit should be zero to consider this register.
6:4	R/W	Reserved	0	Reserved
3:0	R/W	GSPI_MANUAL_WRITE_DATA2	0	Number of bits to be written in write mode. We can select from 1 bit to 16 bits. Update number of bits to be write along with data in write FIFO. The data is written into least 16 bits of a 16x20 FIFO and most 4 bits of FIFO contains information regarding number of bits valid. Note: TAKE_WR_SIZE_FRM_REG bit should be zero to consider these bits. 0 : 16 bits valid 1-15 : corresponding number of bits valid

16.7.5.6 GSPI_FIFO_THRLD

Table 16.168. GSPI_FIFO_THRLD Description

Bit	Access	Function	Reset Value	Description
31:10	R/W	Reserved	0	Reserved
9	R/W	RFIFO_RESET	0	Read FIFO reset
8	R/W	WFIFO_RESET	0	Write FIFO reset
7:4	R/W	FIFO_AFULL_THRLD	12	FIFO almost full threshold
3:0	R/W	FIFO_AEMPTY_THRLD	7	FIFO almost empty threshold

16.7.5.7 GSPI_STATUS

Table 16.169. GSPI_STATUS Description

Bit	Access	Function	Reset Value	Description
31:11	R/W	Reserved	0	Reserved
10	R	GSPI_MANUAL_CSN	1	Provide the status of chip select signal. 0 – Active. 1 – Inactive.
9	R	GSPI_MANUAL_RD_CNT	0	This is a result of 10 bits ORing counter 1 – Read transactions are in pending (to be done) 0 – No read transactions are in pending.
8	R	FIFO_AEMPTY_RFIFO_S	1	Aempty status indication for Rfifo in manual mode.
7	R	FIFO_EMPTY_RFIFO_S	1	Empty status indication for Rfifo in manual mode.
6	R	Reserved	0	Reserved
5	R	FIFO_FULL_RFIFO	0	Full status indication for Rfifo in manual mode.
4	R	Reserved	0	Reserved
3	R	FIFO_EMPTY_WFIFO	1	Empty status indication for Wfifo in manual mode.
2	R	FIFO_AFULL_WFIFO_S	0	Afull status indication for Wfifo in manual mode.
1	R	FIFO_FULL_WFIFO_S	0	Full status indication for Wfifo in manual mode.
0	R	GSPI_BUSY	0	State of Manual mode. 1 - A read, write or dummy cycle operation is in process in manual mode.0 – GSPI controller is IDLE in Manual mode.

16.7.5.8 GSPI_INTR_MASK

Table 16.170. GSPI_INTR_MASK Description

Bit	Access	Function	Reset Value	Description
31:7	R/W	Reserved	0	Reserved
6	R/W	FIFO_EMPTY_RFIFO_MASK	0	1 – Read fifo is empty intr mask 0 – Don't touch.
5	R/W	FIFO_FULL_WFIFO_MASK	0	1 – write fifo full intr mask. 0 – Don't touch.
4	R/W	FIFO_AFULL_WFIFO_MASK	0	1 – Write fifo almost full intr mask. 0 – Don't touch.
3	R/W	FIFO_AEMPTY_WFIFO_MASK	0	1 – write fifo almost empty intr mask. 0 – Don't touch.
2	R/W	FIFO_AFULL_RFIFO_MASK	0	1 – read fifo almost full intr mask. 0 – Don't touch.
1	R/W	FIFO_AEMPTY_RFIFO_MASK	0	1 – Read fifo almost empty intr mask. 0 – Don't touch.
0	R/W	GSPI_INTR_MASK	0	GSPI Interrupt mask bit 1 – mask the GSPI intr. 0 – Don't touch.

16.7.5.9 GSPI_INTR_UNMASK

Table 16.171. GSPI_INTR_UNMASK Description

Bit	Access	Function	Reset Value	Description
31:7	R/W	Reserved	0	Reserved
6	R/W	FIFO_EMPTY_RFIFO_UNMASK	0	1 – Read fifo is empty intr unmask 0 – Don't touch.
5	R/W	FIFO_FULL_WFIFO_UNMASK	0	1 – write fifo full intr unmask. 0 – Don't touch.
4	R/W	FIFO_AFULL_WFIFO_UNMASK	0	1 – Write fifo almost full intr unmask. 0 – Don't touch.
3	R/W	FIFO_AEMPTY_WFIFO_UNMASK	0	1 – write fifo almost empty intr unmask. 0 – Don't touch.
2	R/W	FIFO_AFULL_RFIFO_UNMASK	0	1 – read fifo almost full intr unmask. 0 – Don't touch.
1	R/W	FIFO_AEMPTY_RFIFO_UNMASK	0	1 – Read fifo almost empty intr unmask. 0 – Don't touch.
0	R/W	GSPI_INTR_UNMASK	0	GSPI Interrupt unmask bit 1 – unmask the GSPI intr. 0 – Don't touch

16.7.5.10 GSPI_INTR_STS

Table 16.172. GSPI_INTR_STS Description

Bit	Access	Function	Reset Value	Description
31:7	R	Reserved	0	Reserved
6	R	FIFO_EMPTY_RFIFO_LVL	1	1 – Read fifo is empty. 0 – Read fifo is not empty.
5	R	FIFO_FULL_WFIFO_LVL	0	1 – write fifo full 0 – write fifo not full.
4	R	FIFO_AFULL_WFIFO_LVL	0	1 – Write fifo almost full threshold 0 – Write fifo not reached almost full threshold.
3:2	R	Reserved	0	Reserved
1	R	FIFO_AEMPTY_RFIFO_LVL	1	1 – Read fifo reached almost empty threshold. 0 – Read fifo doesn't reach almost empty threshold.
0	R	GSPI_INTR_LVL	0	GSPI Interrupt Status bit 1 – GSPI raised a interrupt 0 – no interrupt.

16.7.5.11 GSPI_INTR_ACK

Table 16.173. GSPI_INTR_ACK Description

Bit	Access	Function	Reset Value	Description
31:7	W	Reserved	0	Reserved
6	W	FIFO_EMPTY_RFIFO_ACK	0*	1 – Read fifo is empty intr ack 0 – Don't touch
5	W	FIFO_FULL_WFIFO_ACK	0*	1 – write fifo full intr ack 0 – Don't touch
4	W	FIFO_AFULL_WFIFO_ACK	0*	1 – Write fifo almost full intr ack 0 – Don't touch
3:2	W	Reserved	0*	Reserved
1	W	FIFO_AEMPTY_RFIFO_ACK	0*	1 – Read fifo almost empty intr ack. 0 – Don't touch.
0	W	GSPI_INTR_ACK	0*	GSPI Interrupt ack bit 1 – GSPI intr ack. 0 – Don't touch

16.7.5.12 GSPI_STS_MC

Table 16.174. GSPI_STS_MC Description

Bit	Access	Function	Reset Value	Description
31:16	R	Reserved	0	Reserved
15:3	R	SPI_RD_CNT	0	number of pending bytes to be read by device.
2:0	R	BUS_CTRL_PSTATE	0	Provides SPI bus controller present state.

16.7.5.13 GSPI_CLK_DIV

Table 16.175. GSPI_CLK_DIV Description

Bit	Access	Function	Reset Value	Description
31:8	R	Reserved	0	Reserved
7:0	R/W	GSPI_CLK_DIV_FACTOR	0	Provides GSPI clock division factor to the clock divider, which takes SOC clock as input clock and generates required clock according to division factor.

16.7.5.14 GSPI_CONFIG3

Table 16.176. GSPI_CONFIG3 Description

Bit	Access	Function	Reset Value	Description
15	R/W	Reserved	0	Reserved
14:0	R/W	SPI_MANUAL_RD_LNTH_TO_BC	0	Bits are used to indicate the total number of bytes to read from flash during read operation. This is valid only spi_high_performance_en is enabled.

16.7.5.15 GSPI_WRITE_FIFO

Table 16.177. GSPI_WRITE_FIFO Description

Bit	Access	Function	Reset Value	Description
	W	WRITE_FIFO	0	FIFO data is written into this address space.

16.7.5.16 GSPI_READ_FIFO

Table 16.178. GSPI_READ_FIFO Description

Bit	Access	Function	Reset Value	Description
	R	READ_FIFO	0	FIFO data is read from this address space.

16.8 Hardware Random Number Generator

16.8.1 General Description

The Hardware Random Number Generator (HRNG) generates 32-bit random numbers. These random numbers are generated using either a True Random Number Generator or a Pseudo Random Number Generator. These random number generators can be selectively enabled or disabled. Typically, the output of the HRNG is used as a seed for Deterministic Random Bit Generator (DRBG) based random number generators. HRNG is present in MCU HP peripherals.

16.8.2 Features

- Supports 32-bit True Random Number Generator.
- Supports 32-bit Pseudo Random Number Generator.
- Option to selectively enable these random number generators.

16.8.3 Functional Description

True random number generator can be enabled by setting the HRNG_CTRL[0] bit. Pseudo random number generator is enabled by setting the HRNG_CTRL[1] bit. Resetting the HRNG_CTRL[1] or HRNG_CTRL[0] bit, disables the respective random number generators. Only one of them can be active at any point of time. It is recommended to disable the true random number generator when it is not in use to avoid excessive power consumption. The generated random number can be read using 32 bit HRNG_RAND_NUM register. There is a soft reset (HRNG_CTRL[2] bit) to reset the scrambled data to its default value.

16.8.4 Register Summary

Base Address: 0x4509_0000

Table 16.179. Register Summary Table

Register Name	Offset	Description
HRNG_CTRL	0x00	HRNG control register
HRNG_RAND_NUM	0x04	32-bit random number register
HRNG_LFSR_INPUT_LATCH_REG	0x08	LFSR Input Latch Register

16.8.5 Register Description

Legend:

R = Read-only, W = Write-only, R/W = Read/Write, - = Reserved

16.8.5.1 HWRNG_CTRL REG

Table 16.180. Random Number Generator Control Register Description

Bit	Access	Function	POR Value	Description
[31:3]	R	Reserved	0	Reserved for future use.
2	R/W	SOFT_RESET	0	Soft_reset to reset the scrambled data to its default value (For zeroize purposes) 1 – Reset the scrambled data 0 – Not reset.
1	R/W	HWRNG_PRBS_ST	0	This bit is used to start the pseudo random number generation. '1'- Enables pseudo random number generation '0'- Disables pseudo random number generation.
0	R/W	HWRNG_TRNG_ST	0	This bit is used to start the true number generation. '1'- Enables true random number generation '0'- Disables true random number generation.

16.8.5.2 HWRNG_RAND_NUM_REG

Table 16.181. Hardware Random Number Register Description

Bit	Access	Function	POR Value	Description
[31:0]	R	HWRNG_RAND_NUM	0x0000_0380	Generated random number can be read from this register.

16.8.5.3 HRNG_LFSR_INPUT_LATCH_REG

Table 16.182. HRNG LFSR Input Latch Register Description

Bit	Access	Function	POR Value	Description
[31:0]	R	LFSR_INPUT_LATCH_REG	0x0000_0000	32 bit LFSR input data. If tap_lfsr_input bit is set, 32 bit LFSR input data is latched for every 32 cycles. This data is valid only LFSR_32_bit_input_valid bit is set.

16.9 I2C Primary and Secondary

16.9.1 General Description

There are four I^{2}C Primary/Secondary controllers - two in the MCU HP peripherals (I2C0, I2C1), one in the NWP/security subsystem and one in the MCU ULP subsystem (ULP_I2C). The I2C interface allows the processor to serve as a Primary or Secondary on the I2C bus.

16.9.2 Features

Each of these support the following features:

- I^{2}C standard compliant bus interface with open-drain pins
- Configurable as Primary or Secondary
- Four speed modes:
  - Standard Mode (100 kbps) and Fast Mode (400 kbps) are supported by HP I2C0, I2C1 and ULP_I2C
  - Fast Mode Plus (1 Mbps) and High-Speed Mode (3.4 Mbps) are supported only by HP I2C0, I2C1
- 7 or 10-bit addressing
- 7 or 10-bit combined format transfers
- Support for Clock synchronization and Bus Clear
- Programmable SDA Hold time

The I^{2}C controllers also support additional features listed below to reduce the load on the processor:

- Integrated transmit and receive buffers with support for DMA in transmit mode only
- Bulk transmit mode in I^{2}C Secondary mode
- Interrupt based operation (polled mode also available)

The I^{2}C in the MCU ULP subsystem (ULP_I2C) supports the following additional power-save features:

- After the DMA is programmed in PS2 state for I^{2}C transfers, the MCU can switch to PS1 state (processor is shutdown) while the I^{2}C controller continues with the data transfer
- In PS1 state (ULP Peripheral mode) the I^{2}C controller completes the data transfer and, triggered by the Peripheral Interrupt, shifts either to the sleep state (without processor intervention) or the active state.

The NWP/Security subsystem I2C supports following additional feature:

- Ability to connect to external "hardware secure element" accessible through secure API interface.

Note:

- DMA is supported only in Transmit mode
- ULP_I2C Does supports only Standard Mode (100 kbps) and Fast Mode (400 kbps). Fast Mode Plus (1 Mbps) and High-Speed Mode (3.4 Mbps) are not supported

16.9.3 Functional Description

The I2C is a synchronous serial interface. The SDA line is a bidirectional signal and changes only while the SCL line is low, except for STOP, START, and RESTART conditions. The output drivers are open-drain or open-collector to perform wire-AND functions on the bus. These wires carry information between the devices connected to the bus. Each device is recognized by a unique address and can operate as either a “transmitter” or “receiver” depending on the function of the device. Devices can also be considered as primaries or secondaries when performing data transfers. A primary is a device that initiates a data transfer on the bus and generates the clock signals to permit that transfer. At that time, any device addressed is considered a secondary. The I2C module can operate in standard mode (with data rates 0 to 100 Kb/s), fast mode (with data rates less than or equal to 400 Kb/s), fast mode plus (with data rates less than or equal to 1000 Kb/s), high-speed mode (with data rates less than or equal to 3.4 Mb/s). high-speed mode and fast mode devices are downward compatible. The I2C is made up of an AMBA APB secondary interface, an I2C interface, and FIFO logic to maintain coherency between the two interfaces. A simplified block diagram of the I2C is shown below.

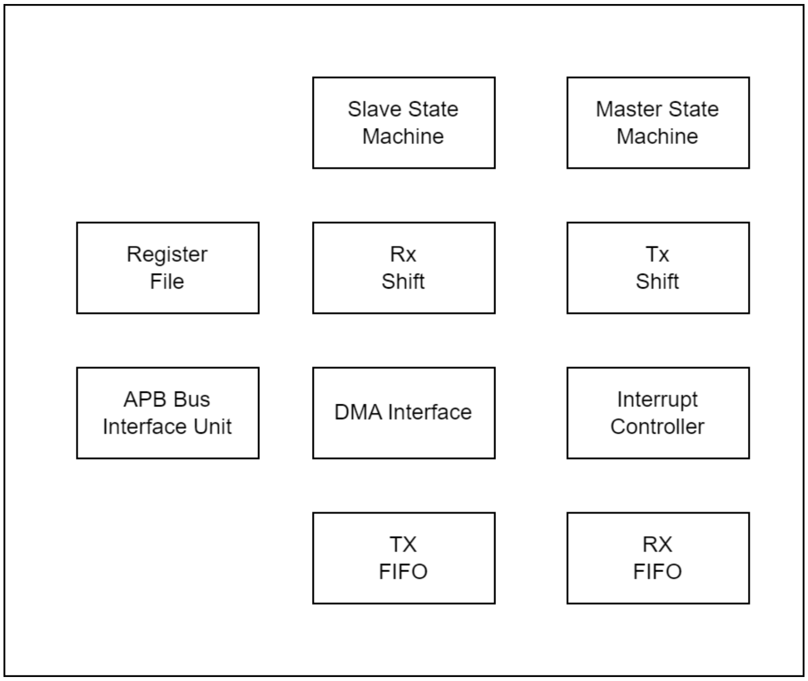


Figure 16.23. I2C Block Diagram

The I2C can be controlled via software to be either I2C Primary or I2C Secondary. The primary is responsible for generating the clock and controlling the transfer of data. The secondary is responsible for either transmitting or receiving data to/from the primary. The acknowledgement of data is sent by the device that is receiving data, which can be either a primary or a secondary.

Each secondary has a unique address that is determined by the system designer. When a primary wants to communicate with a secondary, the primary transmits a START/RESTART condition that is then followed by the secondaries address and a control bit (R/W) to determine if the primary wants to transmit data or receive data from the secondary. The secondary then sends an acknowledge (ACK) pulse after the address. If the primary (primary-transmitter) is writing to the secondary (secondary-receiver), the receiver gets one byte of data. This transaction continues until the primary terminates the transmission with a STOP condition. If the primary is reading from a secondary (primary-receiver), the secondary transmits (secondary-transmitter) a byte of data to the primary, and the primary then acknowledges the transaction with the ACK pulse. This transaction continues until the primary terminates the transmission by not acknowledging (NACK) the transaction after the last byte is received, and then the primary issues a STOP condition or addresses another Secondary after issuing a RESTART condition. This behavior is shown below.

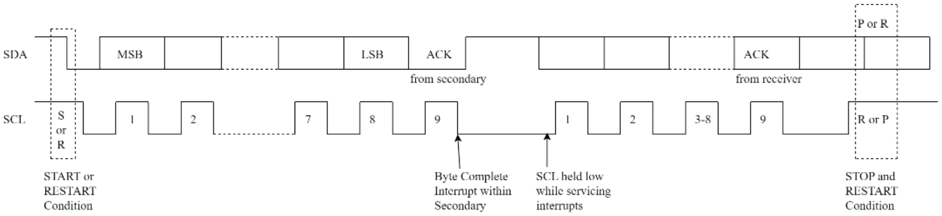

P or R
SDA
MSB
LSB
ACK
from secondary
ACK
from receiver
SCL
S or R
1
2
7
8
9
1
2
3-8
9
R or P
START or RESTART Condition
Byte Complete Interrupt within Secondary
SCL held low while servicing interrupts
STOP and RESTART Condition

Figure 16.24. Data Transfer on the I2C Bus

16.9.4 Operating Modes

16.9.4.1 Secondary Mode Operation

Initial Configuration

To use the I2C as a secondary, perform the following steps:

1. Disable the I2C by writing a ‘0’ to bit 0 of the IC_ENABLE register.
2. Write to the IC_SAR register (bits 9:0) to set the secondary address. This is the address to which the I2C responds.
3. Write to the IC_CON register to specify which type of addressing is supported (7- or 10-bit by setting bit 3). Enable the I2C in secondary-only mode by writing a ‘0’ into bit 6 (IC_SLAVE_DISABLE) and a ‘0’ to bit 0 (MASTER_MODE).
4. Enable the I2C by writing a ‘1’ in bit 0 of the IC_ENABLE register.

Secondary-Transmitter Operation for a Single Byte

When another I2C primary device on the bus addresses this I2C and requests data, this I2C acts as a secondary-transmitter and the following steps occur:

1. The other I2C Master device initiates an I2C transfer with an address that matches the Secondary address in the IC_SAR register of this I2C.
2. The I2C acknowledges the sent address and recognizes the direction of the transfer to indicate that it is acting as a Secondary-transmitter.
3. The I2C asserts the RD_REQ interrupt (bit 5 of the IC_RAW_INTR_STAT register) and holds the SCL line low. It is in a wait state until software responds. If the RD_REQ interrupt has been masked, due to IC_INTR_MASK[5] register (M_RD_REQ bit field) being set to 0, then it is recommended that a hardware and/or software timing routine be used to instruct the CPU to perform periodic reads of the IC_RAW_INTR_STAT register.
   a. Reads that indicate IC_RAW_INTR_STAT[5] (R_RD_REQ bit field) being set to 1 must be treated as the equivalent of the RD_REQ interrupt being asserted.
   b. Software must then act to satisfy the I2C transfer.
   c. The timing interval used should be in the order of 10 times the fastest SCL clock period the I2C can handle. For example, for 400 kb/s, the timing interval is 25 µs.
4. If there is any data remaining in the Tx FIFO before receiving the read request, then the I2C asserts a TX_ABRT interrupt (bit 6 of the IC_RAW_INTR_STAT register) to flush the old data from the TX FIFO. If the TX_ABRT interrupt has been masked, due to of IC_INTR_MASK[6] register (M_TX_ABRT bit field) being set to 0, then it is recommended that re-using the timing routine (described in the previous step), or a similar one, be used to read the IC_RAW_INTR_STAT register.
   a. Reads that indicate bit 6 (R_TX_ABRT) being set to 1 must be treated as the equivalent of the TX_ABRT interrupt being asserted.
   b. There is no further action required from software.
   c. The timing interval used should be similar to that described in the previous step for the IC_RAW_INTR_STAT[5] register.
5. Software writes to the IC_DATA_CMD register with the data to be written (by writing a ‘0’ in bit 8).
6. Software must clear the RD_REQ and TX_ABRT interrupts (bits 5 and 6, respectively) of the IC_RAW_INTR_STAT register before proceeding. If the RD_REQ and/or TX_ABRT interrupts have been masked, then clearing of the IC_RAW_INTR_STAT register will have already been performed when either the R_RD_REQ or R_TX_ABRT bit has been read as 1.
7. The I2C releases the SCL and transmits the byte.
8. The master may hold the I2C bus by issuing a RESTART condition or release the bus by issuing a STOP condition.

Secondary-Receiver Operation for a Single Byte

When another I2C Master device on the bus addresses this I2C and is sending data, this I2C acts as a Secondary-receiver and the following steps occur:

1. The other I2C Master device initiates an I2C transfer with an address that matches this I2C’s Secondary address in the IC_SAR register.
2. This I2C acknowledges the sent address and recognizes the direction of the transfer to indicate that the I2C is acting as a Secondary-receiver.
3. I2C receives the transmitted byte and places it in the receive buffer.
4. I2C asserts the RX_FULL interrupt (IC_RAW_INTR_STAT[2] register). If the RX_FULL interrupt has been masked, due to setting IC_INTR_MASK[2] register to 0 or setting IC_TX_TL to a value larger than 0, then it is recommended that a timing routine be implemented for periodic reads of the IC_STATUS register. Reads of the IC_STATUS register, with bit 3 (RFNE) set at 1, must then be treated by software as the equivalent of the RX_FULL interrupt being asserted.
5. Software may read the byte from the IC_DATA_CMD register (bits 7:0).
6. The other master device may hold the I2C bus by issuing a RESTART condition, or release the bus by issuing a STOP condition.

16.9.4.2 Primary Mode Operation

Initial Configuration

1. Disable the I2C by writing 0 to bit 0 of the IC_ENABLE register.
2. Write to the IC_CON register to set the maximum speed mode supported for secondary operation (bits 2:1) and to specify whether the I2C starts its transfers in 7/10 bit addressing mode when the device is a secondary (bit 3).
3. Write to the IC_TAR register the address of the I2C device to be addressed. It also indicates whether a General Call or a START BYTE command is going to be performed by I2C. The desired speed of the I2C primary-initiated transfers, either 7-bit or 10-bit addressing, is controlled by the IC_10BITADDR_MASTER bit field (bit 12).
4. Only applicable for high-speed mode transfers. Write to the IC_HS_MADDR register the desired master code for the I2C. The master code is programmer-defined.
5. Enable the I2C by writing a 1 to bit 0 of the IC_ENABLE register.
6. Now write the transfer direction and data to be sent to the IC_DATA_CMD register. If the IC_DATA_CMD register is written before the I2C is enabled, the data and commands are lost as the buffers are kept cleared when I2C is not enabled.

START and STOP Generation

When operating as an I2C Primary, putting data into the transmit FIFO causes the I2C to generate a START condition on the I2C bus. Writing a 1 to IC_DATA_CMD[9] causes the I2C to generate a STOP condition on the I2C bus; a STOP condition is not issued if this bit is not set, even if the transmit FIFO is empty.

Primary Transmit and Primary Receive

The I2C supports switching back and forth between reading and writing dynamically. To transmit data, write the data to be written to the lower byte of the I2C Rx/Tx FIFO and Command Register (IC_DATA_CMD). The CMD bit [8] should be written to 0 for I2C write operations. Subsequently, a read command may be issued by writing “don’t cares” to the lower byte of the IC_DATA_CMD register, and a 1 should be written to the CMD bit. The I2C primary continues to initiate transfers as long as there are commands present in the transmit FIFO. If the transmit FIFO becomes empty, the primary either inserts a STOP condition after completing the current transfers or it checks to see if IC_DATA_CMD[9] is set to 1.

1. If set to 1, it issues a STOP condition after completing the current transfer.
2. If set to 0, it holds SCL low until next command is written to the transmit FIFO.

16.9.4.3 Transmit with DMA

Transmitter Operation with DMA

This additional programming sequence required for Transmit with DMA mode operation

1. Software fills the DMA source buffer with 16-bit data words containing command in bits [11:8] and data byte in [7:0]
2. Software writes the DMA transfer size with twice the data bytes needs to be transferred
3. DMA writes to the IC_DATA_CMD register with the data to be written with above data words from source buffer.
4. I2C transmits the data from IC_DATA_CMD[7:0]

16.9.5 Register Summary

Table 16.183. Register Summary Table

I2C0 Base Address: 0x4401_0000; I2C1 Base Address: 0x4704_0000; ULP_I2C Base Address: 0x2404_0000

Register Name	Offset	Description
IC_CON	0x00	Control Register
IC_TAR	0x04	Target Address Register
IC_SAR	0x08	SecondaryAddress Register
IC_HS_MADDR	0x0C	High Speed Primary Mode Code Address Register
IC_DATA_CMD	0x10	Rx/Tx Data Buffer and Command Register
IC_SS_SCL_HCNT	0x14	Standard Speed I2C Clock SCL High Count Register
IC_SS_SCL_LCNT	0x18	Standard Speed I2C Clock SCL Low Count Register
IC_FS_SCL_HCNT	0x1C	Fast Mode or Fast Mode Plus I2C Clock SCL High Count Register
IC_FS_SCL_LCNT	0x20	Fast Mode or Fast Mode Plus I2C Clock SCL Low Count Register
IC_HS_SCL_HCNT	0x24	High Speed I2C Clock SCL High Count Register
IC_HS_SCL_LCNT	0x28	High Speed I2C Clock SCL Low Count Register
IC_INTR_STAT	0x2C	Interrupt Status Register
IC_INTR_MASK	0x30	Interrupt Mask Register
IC_RAW_INTR_STAT	0x34	Raw Interrupt Status Register
IC_RX_TL	0x38	Receive FIFO Threshold Register
IC_TX_TL	0x3C	Transmit FIFO Threshold Register
IC_CLR_INTR	0x40	Clear Combined and Individual Interrupt Register
IC_CLR_RX_UNDER	0x44	Clear RX_UNDER Interrupt Register
IC_CLR_RX_OVER	0x48	Clear RX_OVER Interrupt Register
IC_CLR_TX_OVER	0x4C	Clear TX_OVER Interrupt Register
IC_CLR_RD_REQ	0x50	Clear RD_REQ Interrupt Register
IC_CLR_TX_ABRT	0x54	Clear TX_ABRT Interrupt Register
IC_CLR_RX_DONE	0x58	Clear RX_DONE Interrupt Register
IC_CLR_ACTIVITY	0x5c	Clear ACTIVITY Interrupt Register
IC_CLR_STOP_DET	0x60	Clear STOP_DET Interrupt Register
IC_CLR_START_DET	0x64	Clear START_DET Interrupt Register
IC_CLR_GEN_CALL	0x68	Clear GEN_CALL Interrupt Register
IC_ENABLE	0x6C	Enable Register
IC_STATUS	0x70	Status Register
IC_TXFLR	0x74	Transmit FIFO Level Register
IC_RXFLR	0x78	Receive FIFO Level Register
IC_SDA_HOLD	0x7C	SDA Hold Time Length Register

Register Name	Offset	Description
IC_TX_ABRT_SOURCE	0x80	Transmit Abort Source Register
IC_SLV_DATA_NACK_ONLY	0x84	Generate SecondaryData NACK Register
IC_DMA_CR	0x88	DMA Control Register
IC_DMA_TDLR	0x8c	DMA Transmit Data Level Register
IC_DMA_RDLR	0x90	Receive Data Level Register
IC_SDA_SETUP	0x94	SDA Setup Register
IC_ACK_GENERAL_CALL	0x98	ACK General Call Register
IC_ENABLE_STATUS	0x9C	Enable Status Register
IC_FS_SPKLEN	0xA0	SS and FS Spike Suppression Limit Register
IC_HS_SPKLEN	0xA4	HS Spike Suppression Limit Register
IC_CLR_RESTART_DET	0xA8	Clear RESTART_DET Interrupt Register
IC_COMP_PARAM_1	0xF4	Component Parameter Register 1
IC_COMP_VERSION	0xF8	Component Version Register
IC_COMP_TYPE	0xFC	Component Type Register
IC_SCL_STUCK_AT_LOW_TIMEOUT	0xAC	SCL Stuck at Low Timeout
IC_SDA_STUCK_AT_LOW_TIMEOUT	0xB0	SDA Stuck at Low Timeout
IC_CLR_SCL_STUCK_DET	0xB4	Clear SCL Stuck at Low Detect Interrupt Register
IC_DEVICE_ID	0xB8	Device ID
IC_OPTIONAL_SAR	0xD8	Optional SecondaryAddress Register

16.9.6 Register Description

Legend:
R = Read-only, W = Write-only, R/W = Read/Write, N/A = Reserved

16.9.6.1 IC_CON

This register can be written only when the i2c is disabled, which corresponds to IC_ENABLE[0] being set to 0. Writes at other times have no effect.

Bit	Access	Function	POR Value	Description
31:12	N/A	Reserved		Reserved
11	R/W	BUS_CLEAR_FEATURE_CTRL	0	In Master Mode: • 1'b1: Bus Clear Feature is enabled • 1'b0: Bus Clear Feature is disabled In Slave Mode, this register bit is not applicable.
10	R/W	STOP_DET_IF_MASTER_ACTIVE	0	In Master mode • 1’b1: Issues the STOP_DET interrupt only when the master is active • 1’b0: Issues the STOP_DET irrespective of whether the master is active
9	N/A	Reserved	0	Reserved
8	R/W	TX_EMPTY_CTRL	0	This bit controls the generation of the TX_EMPTY interrupt, as described in the IC_RAW_INTR_STAT register.
7	R/W	STOP_DET_IFADDRESSED	0	In Slave mode: 1’b1 – issues the STOP_DET interrupt only when it is addressed. 1’b0 – issues the STOP_DET irrespective of whether it’s addressed or not. Dependencies: This register bit value is applicable in the Slave mode only (MASTER_MODE = 1’b0) NOTE: During a general call address, this Slave does not issue the STOP_DET interrupt if STOP_DET_IF_ADDRESSED = 1’b1, even if the Slave responds to the general call address by generating ACK. The STOP_DET interrupt is generated only when the transmitted address matches the Slave address (SAR).
6	R/W	IC_SLAVE_DISABLE	0x1	If this bit is set (Slave is disabled), I2C functions only as a master and does not perform any action that requires a Slave. 0: Slave is enabled 1: Slave is disabled NOTE: Software should ensure that if this bit is written with ‘0,’ then bit 0 should also be written with a ‘0’.

Bit	Access	Function	POR Value	Description
5	R/W	IC_RESTART_EN	0x1	Determines whether RESTART conditions may be sent when acting as a master. Some older Slaves do not support handling RESTART conditions; however, RESTART conditions are used in several I2C operations. 0: disable 1: enable When the RESTART is disabled, the I2C Master is incapable of performing the following functions: • Sending a START BYTE • Performing any high-speed mode operation • Performing direction changes in combined format mode • Performing a read operation with a 10-bit address By replacing RESTART condition followed by a STOP and a subsequent START condition, split operations are broken down into multiple I2C transfers. If the above operations are performed, it will result in setting bit 6 (TX_ABRT) of the IC_RAW_INTR_STAT register.
4	R	IC_10BITADDR_MASTER_RD_ONLY	0x1	the function of this bit is handled by bit 12 of IC_TAR register, and becomes a read-only copy called IC_10BITADDR_MASTER_rd_only. 0: 7-bit addressing 1: 10-bit addressing
3	R/W	IC_10BITADDR_SLAVE	0x1	When acting as a Slave, this bit controls whether the I2C responds to 7- or 10-bit addresses. 0: 7-bit addressing. The I2C ignores transactions that involve 10-bit addressing; for 7-bit addressing, only the lower 7 bits of the IC_SAR register are compared. 1: 10-bit addressing. The I2C responds to only 10-bit addressing transfers that match the full 10 bits of the IC_SAR register.
2:1	R/W	SPEED	0x3	These bits control at which speed the I2C operates. Hardware protects against illegal values being programmed by software. register These bits must be programmed appropriately for Slave mode also, as it is used to capture correct value of spike filter as per the speed mode. This register should be programmed only with a value in the range of 1 to IC_MAX_SPEED_MODE; otherwise, hardware updates this register with the value of IC_MAX_SPEED_MODE. • 1: standard mode (0 to 100 Kb/s) • 2: fast mode (≤ 400 Kb/s) or fast mode plus (≤ 1000 Kb/s) • 3: high speed mode (≤ 3.4 Mb/s)
0	R/W	MASTER_MODE	0x1	This bit controls whether the I2C Master is enabled. 0: Master disabled 1: Master enabled NOTE: Software should ensure that if this bit is written with ‘1,’ then bit 6 should also be written with a ‘1’.

IC_CON Description

16.9.6.2 IC_TAR

Bit	Access	Function	POR Value	Description
31:14	N/A	Reserved		Reserved
13	R/W	Device_ID	0	If bit 11 (SPECIAL) is set to 1, then this bit indicates whether a Device-ID of a particular Slave mentioned in IC_TAR[6:0] is to be performed by the I2C Master. • 0: Device-ID is not performed and checks ic_tar[10] to perform either general call or START byte command. • 1: Device-ID transfer is performed and bytes based on the number of read commands in the Tx-FIFO are received from the targeted Slave and put in the Rx-FIFO.
12	R/W	IC_10BITADDR_MASTER	1	This bit controls whether the I2C starts its transfers in 7- or 10-bit addressing mode when acting as a master. • 0: 7-bit addressing • 1: 10-bit addressing
11	R/W	SPECIAL	0	This bit indicates whether software performs a Device-ID, General Call or START BYTE command. • 0: ignore bit 10 GC_OR_START and use IC_TAR normally • 1: perform special I2C command as specified in Device-ID or GC_OR_START bit
10	R/W	GC_OR_START	0	If bit 11 (SPECIAL) is set to 1 and bit 13 (Device-ID) is set to 0, then this bit indicates whether a General Call or START byte command is to be performed by the I2C. • 0: General Call Address – after issuing a General Call, only writes may be performed. Attempting to issue a read command results in setting bit 6 (TX_ABRT) of the IC_RAW_INTR_STAT register. The I2C remains in General Call mode until the SPECIAL bit value (bit 11) is cleared. • 1: START BYTE
9:0	R/W	IC_TAR	0x0A0	This is the target address for any master transaction. When transmitting a General Call, these bits are ignored. To generate a START BYTE, the CPU needs to write only once into these bits. If the IC_TAR and IC_SAR are the same, loopback exists but the FIFOs are shared between master and Slave, so full loopback is not feasible. Only one direction loopback mode is supported (simplex), not duplex. A master cannot transmit to itself; it can transmit to only a Slave.

IC_TAR Description

16.9.6.3 IC_SAR

Bit	Access	Function	POR Value	Description
31:10	N/A	Reserved		Reserved
9:0	R/W	IC_SAR	0x055	The IC_SAR holds the Slave address when the I2C is operating as a Slave. For 7-bit addressing, only IC_SAR[6:0] is used. This register can be written only when the I2C interface is disabled, which corresponds to IC_ENABLE[0] being set to 0. Writes at other times have no effect. NOTE: The default values cannot be any of the reserved address locations: that is, 0x00 to 0x07, or 0x78 to 0x7f. The correct operation of the device is not guaranteed the IC_SAR or IC_TAR is programmed to a reserved value.

IC_SAR Description

16.9.6.4 IC_HS_MADDR

Bit	Access	Function	POR Value	Description
31:3	N/A	Reserved		Reserved
2:0	R/W	IC_HS_MAR	0x1	This bit field holds the value of the I2C HS mode master code. HS-mode master codes are reserved 8-bit codes (00001xxx) that are not used for Slave addressing or other purposes. Each master has its unique master code; up to eight high speed mode masters can be present on the same I2C bus system. Valid values are from 0 to 7. This register goes away and becomes read-only returning 0’s if the IC_MAX_SPEED_MODE configuration parameter is set to either Standard(1) or Fast (2). This register can be written only when the I2C interface is disabled, which corresponds to IC_ENABLE[0] being set to 0. Writes at other times have no effect.

IC_HS_MADDR Description

16.9.6.5 IC_DATA_CMD

This is the register the CPU writes to when filling the TX FIFO and the CPU reads from when retrieving bytes from RX FIFO.

In order for the i2c to continue acknowledging reads, a read command should be written for every byte that is to be received; otherwise i2c will stop acknowledging.

Bit	Access	Function	POR Value	Description
31:12	N/A	Reserved		Reserved
11	R	FIRST_DATA_BYTE	0	Indicates the first data byte received after the address phase for receive transfer in master receiver or Slave receiver mode.
10	W	RESTART	0	This bit controls whether a RESTART is issued before the byte is sent or received. • 1 – If IC_RESTART_EN is 1, a RESTART is issued before the data is sent/received (according to the value of CMD), regardless of whether or not the transfer direction is changing from the previous command; if IC_RESTART_EN is 0, a STOP followed by a START is issued instead. • 0 – If IC_RESTART_EN is 1, a RESTART is issued only if the transfer direction is changing from the previous command; if IC_RESTART_EN is 0, a STOP followed by a START is issued instead.
9	W	STOP	0	This bit controls whether a STOP is issued after the byte is sent or received. • 1 – STOP is issued after this byte, regardless of whether or not the Tx FIFO is empty. If the Tx FIFO is not empty, the master immediately tries to start a new transfer by issuing a START and arbitrating for the bus. • 0 – STOP is not issued after this byte, regardless of whether or not the Tx FIFO is empty. If the Tx FIFO is not empty, the master continues the current transfer by sending/receiving data bytes according to the value of the CMD bit. If the Tx FIFO is empty, the master holds the SCL line low and stalls the bus until a new command is available in the Tx FIFO.
8	W	CMD	0	This bit controls whether a read or a write is performed. This bit does not control the direction when the I2C acts as a Slave. It controls only the direction when it acts as a master. • 1 – Read • 0 – Write When a command is entered in the TX FIFO, this bit distinguishes the write and read commands. In Slave-receiver mode, this bit is a “don’t care” because writes to this register are not required. In Slave-transmitter mode, a “0” indicates that the data in IC_DATA_CMD is to be transmitted. When programming this bit, the following should be remembered: attempting to perform a read operation after a General Call command has been sent results in a TX_ABRT interrupt (bit 6 of the IC_RAW_INTR_STAT register), unless bit 11 (SPECIAL) in the IC_TAR register has been cleared. If a “1” is written to this bit after receiving a RD_REQ interrupt, then a TX_ABRT interrupt occurs.

Bit	Access	Function	POR Value	Description
7:0	R/W	DAT	0	This register contains the data to be transmitted or received on the I2C bus. If this register is being written and a read is to be performed, bits 7:0 (DAT) are ignored by the I2C. However, when this register is read, these bits return the value of data received on the I2C interface.

IC_DATA_CMD Description

16.9.6.6 IC_SS_SCL_HCNT

Bit	Access	Function	POR Value	Description
31:16	N/A	Reserved		Reserved
15:0	R/W	IC_SS_SCL_HCNT	0x01F4	This register must be set before any I2C bus transaction can take place to ensure proper I/O timing. This register sets the SCL clock high-period count for standard speed. This register can be written only when the I2C interface is disabled which corresponds to IC_ENABLE[0] being set to 0. Writes at other times have no effect. The minimum valid value is 6; hardware prevents values less than this being written, and if attempted results in 6 being set. NOTE: This register must not be programmed to a value higher than 65525, because I2C uses a 16-bit counter to flag an I2C bus idle condition when this counter reaches a value of IC_SS_SCL_HCNT + 10.

IC_SS_SCL_HCNT Description

16.9.6.7 IC_SS_SCL_LCNT

Bit	Access	Function	POR Value	Description
31:16	N/A	Reserved		Reserved
15:0	R/W	IC_SS_SCL_LCNT	0x024C	This register must be set before any I2C bus transaction can take place to ensure proper I/O timing. This register sets the SCL clock low period count for standard speed. This register can be written only when the I2C interface is disabled which corresponds to IC_ENABLE[0] being set to 0. Writes at other times have no effect. The minimum valid value is 8; hardware prevents values less than this being written, and if attempted, results in 8 being set.

IC_SS_SCL_LCNT Description

16.9.6.8 IC_FS_SCL_HCNT

Bit	Access	Function	POR Value	Description
31:16	N/A	Reserved		Reserved
15:0	R/W	IC_FS_SCL_HCNT	0x004B	This register must be set before any I2C bus transaction can take place to ensure proper I/O timing. This register sets the SCL clock high-period count for fast mode or fast mode plus. It is used in high-speed mode to send the master Code and START BYTE or General CALL. This register goes away and becomes read-only returning 0s if IC_MAX_SPEED_MODE = standard. This register can be written only when the I2C interface is disabled, which corresponds to IC_ENABLE[0] being set to 0. Writes at other times have no effect. The minimum valid value is 6; hardware prevents values less than this being written, and if attempted results in 6 being set.

IC_FS_SCL_HCNT Description

16.9.6.9 IC_FS_SCL_LCNT

Bit	Access	Function	POR Value	Description
31:16	N/A	Reserved		Reserved
15:0	R/W	IC_FS_SCL_LCNT	0x00A3	This register must be set before any I2C bus transaction can take place to ensure proper I/O timing. This register sets the SCL clock low period count for fast mode or fast mode plus. It is used in high-speed mode to send the master Code and START BYTE or General CALL. This register goes away and becomes read-only returning 0s if IC_MAX_SPEED_MODE = standard. This register can be written only when the I2C interface is disabled, which corresponds to IC_ENABLE[0] being set to 0. Writes at other times have no effect. The minimum valid value is 8; hardware prevents values less than this being written, and if attempted results in 8 being set.

IC_FS_SCL_LCNT Description

16.9.6.10 IC_HS_SCL_HCNT

Bit	Access	Function	POR Value	Description
31:16	N/A	Reserved		Reserved
15:0	R/W	IC_HS_SCL_HCNT	0x000F	This register must be set before any I2C bus transaction can take place to ensure proper I/O timing. This register sets the SCL clock high period count for high speed. The SCL High time depends on the loading of the bus. For 100pF loading, the SCL High time is 60ns; for 400pF loading, the SCL High time is 120ns. This register goes away and becomes read-only returning 0s if IC_MAX_SPEED_MODE != high. This register can be written only when the I2C interface is disabled, which corresponds to IC_ENABLE[0] being set to 0. Writes at other times have no effect. The minimum valid value is 6; hardware prevents values less than this being written, and if attempted results in 6 being set.

IC_HS_SCL_HCNT Description

16.9.6.11 IC_HS_SCL_LCNT

Bit	Access	Function	POR Value	Description
31:16	N/A	Reserved		Reserved
15:0	R/W	IC_HS_SCL_LCNT	0x0028	This register must be set before any I2C bus transaction can take place to ensure proper I/O timing. This register sets the SCL clock low period count for high speed. The SCL low time depends on the loading of the bus. For 100pF loading, the SCL low time is 160ns; for 400pF loading, the SCL low time is 320ns. This register goes away and becomes read-only returning 0s if IC_MAX_SPEED_MODE != high. This register can be written only when the I2C interface is disabled, which corresponds to IC_ENABLE[0] being set to 0. Writes at other times have no effect. The minimum valid value is 8; hardware prevents values less than this being written, and if attempted results in 8 being set.

IC_HS_SCL_LCNT Description

16.9.6.12 IC_INTR_STAT

Bit	Access	Function	POR Value	Description
31:15	N/A	Reserved		Reserved
14	R or R/W	M_SCL_STUCK_AT_LOW	0	See IC_RAW_INTR_STAT for a detailed description of this bit.
13	R or R/W	R_MST_ON_HOLD	0	See “IC_RAW_INTR_STAT” for a detailed description of this bit.
12 11 10 9 8 7 6 5 4 3 2 1 0	R	R_RESTART_DET R_GEN_CALL R_START_DET R_STOP_DET R_ACTIVITY R_RX_DONE R_TX_ABRT R_RD_REQ R_TX_EMPTY R_TX_OVER R_RX_FULL R_RX_OVER R_RX_UNDER	0	See “IC_RAW_INTR_STAT” for a detailed description of these bits.

IC_INTR_STAT Description

16.9.6.13 IC_INTR_MASK

Bit	Access	Function	POR Value	Description
31:15	N/A	Reserved		Reserved
14	R or R/W	R_SCL_STUCK_AT_LOW	1	This bit masks the R_SCL_STUCK_AT_LOW interrupt bit in the IC_INTR_STAT register
13	R/W	M_MST_ON_HOLD	0	This bit masks the R_MST_ON_HOLD interrupt bit in the IC_INTR_STAT register.
12	R/W	M_RESTART_DET		This bit masks the R_RESTART_DET interrupt status bit in the IC_INTR_STAT register.
11 10 9 8 7 6 5 4 3 2 1 0	R or R/W	M_GEN_CALL M_START_DET M_STOP_DET M_ACTIVITY M_RX_DONE M_TX_ABRT M_RD_REQ M_TX_EMPTY M_TX_OVER M_RX_FULL M_RX_OVER M_RX_UNDER	14’h8ff	These bits mask their corresponding interrupt status bits in the IC_INTR_STAT register.

IC_INTR_MASK Description

16.9.6.14 IC_RAW_INTR_STAT

Bit	Access	Function	POR Value	Description
31:15	N/A	Reserved		Reserved
14	R	SCL_STUCK_AT_LOW	0	Indicates whether the SCL Line is stuck at low for the IC_SCL_STUCK_LOW_TIMOUT number of ic_clk periods.
13	R	MST_ON_HOLD	0	Indicates whether a master is holding the bus and the Tx FIFO is empty.
12	R	RESTART_DET	0	Indicates whether a RESTART condition has occurred on the I2C interface when I2C is operating in Slave mode and the Slave is the addressed Slave. NOTE: However, in high-speed mode or during a START BYTE transfer, the RESTART comes before the address field as per the I2C protocol. In this case, the Slave is not the addressed Slave when the RESTART is issued, therefore I2C does not generate the RESTART_DET interrupt.
11	R	GEN_CALL	0	Set only when a General Call address is received and it is acknowledged. It stays set until it is cleared either by disabling I2C or when the CPU reads bit 0 of the IC_CLR_GEN_CALL register. I2C stores the received data in the Rx buffer.
10	R	START_DET	0	Indicates whether a START or RESTART condition has occurred on the I2C interface regardless of whether I2C is operating in Slave or master mode.
9	R	STOP_DET	0	Indicates whether a STOP condition has occurred on the I2C interface regardless of whether I2C is operating in Slave or master mode. In Slave Mode: • If IC_CON[7]=1'b1 (STOP_DET_IFADDRESSED), the STOP_DET interrupt is generated only if the Slave is addressed. Note: During a general call address, this Slave does not issue a STOP_DET interrupt if STOP_DET_IF_ADDRESSED=1'b1, even if the Slave responds to the general call address by generating ACK. The STOP_DET interrupt is generated only when the transmitted address matches the Slave address (SAR). • If IC_CON[7]=1'b0 (STOP_DET_IFADDRESSED), the STOP_DET interrupt is issued irrespective of whether it is being addressed. In master Mode: • If IC_CON[10]=1’b1 (STOP_DET_IF_MASTER_ACTIVE), the STOP_DET interrupt is issued only if the master is active. • If IC_CON[10]=1’b0 (STOP_DET_IFADDRESSED), the STOP_DET interrupt is issued irrespective of whether the master is active.

Bit	Access	Function	POR Value	Description
8	R	ACTIVITY	0	This bit captures I2C activity and stays set until it is cleared. There are four ways to clear it: • Disabling the I2C • Reading the IC_CLR_ACTIVITY register • Reading the IC_CLR_INTR register • System reset Once this bit is set, it stays set unless one of the four methods is used to clear it. Even if the I2C module is idle, this bit remains set until cleared, indicating that there was activity on the bus.
7	R	RX_DONE	0	When the I2C is acting as a Slave-transmitter, this bit is set to 1 if the master does not acknowledge a transmitted byte. This occurs on the last byte of the transmission, indicating that the transmission is done.
6	R	TX_ABRT	0	This bit indicates if I2C, as an I2C transmitter, is unable to complete the intended actions on the contents of the transmit FIFO. This situation can occur both as an I2C Master or an I2C Slave and is referred to as a “transmit abort”. When this bit is set to 1, the IC_TX_ABRT_SOURCE register indicates the reason why the transmit abort takes places. NOTE: The I2C flushes/resets/empties only the TX_FIFO whenever there is a transmit abort caused by any of the events tracked by the IC_TX_ABRT_SOURCE register. The Tx FIFO remains in this flushed state until the register IC_CLR_TX_ABRT is read. Once this read is performed, the Tx FIFO is then ready to accept more data bytes from the APB interface.
5	R	RD_REQ	0	This bit is set to 1 when I2C is acting as a Slave and another I2C Master is attempting to read data from I2C. The I2C holds the I2C bus in a wait state (SCL=0) until this interrupt is serviced, which means that the Slave has been addressed by a remote master that is asking for data to be transferred. The processor must respond to this interrupt and then write the requested data to the IC_DATA_CMD register. This bit is set to 0 just after the processor reads the IC_CLR_RD_REQ register.

Bit	Access	Function	POR Value	Description
4	R	TX_EMPTY	0	The behavior of the TX_EMPTY interrupt status differs based on the TX_EMPTY_CTRL selection in the IC_CON register. • When TX_EMPTY_CTRL = 0: This bit is set to 1 when the transmit buffer is at or below the threshold value set in the IC_TX_TL register. • When TX_EMPTY_CTRL = 1: This bit is set to 1 when the transmit buffer is at or below the threshold value set in the IC_TX_TL register and the transmission of the address/data from the internal shift register for the most recently popped command is completed. It is automatically cleared by hardware when the buffer level goes above the threshold. When IC_ENABLE[0] is set to 0, the TX FIFO is flushed and held in reset. There the TX FIFO looks like it has no data within it, so this bit is set to 1, provided there is activity in the master or Slave state machines. When there is no longer any activity, then with ic_en=0, this bit is set to 0.
3	R	TX_OVER	0	Set during transmit if the transmit buffer is filled to IC_TX_BUFFER_DEPTH and the processor attempts to issue another I2C command by writing to the IC_DATA_CMD register. When the module is disabled, this bit keeps its level until the master or Slave state machines go into idle, and when ic_en goes to 0, this interrupt is cleared.
2	R	RX_FULL	0	Set when the receive buffer reaches or goes above the RX_TL threshold in the IC_RX_TL register. It is automatically cleared by hardware when buffer level goes below the threshold. If the module is disabled (IC_ENABLE[0]=0), the RX FIFO is flushed and held in reset; therefore the RX FIFO is not full. So this bit is cleared once IC_ENABLE[0] is set to 0, regardless of the activity that continues.
1	R	RX_OVER	0	Set if the receive buffer is completely filled to IC_RX_BUFFER_DEPTH and an additional byte is received from an external I2C device. The I2C acknowledges this, but any data bytes received after the FIFO is full are lost. If the module is disabled (IC_ENABLE[0]=0), this bit keeps its level until the master or Slave state machines go into idle, and when ic_en goes to 0, this interrupt is cleared.
0	R	RX_UNDER	0	Set if the processor attempts to read the receive buffer when it is empty by reading from the IC_DATA_CMD register. If the module is disabled (IC_ENABLE[0]=0), this bit keeps its level until the master or Slave state machines go into idle, and when ic_en goes to 0, this interrupt is cleared

IC_RAW_INTR_STAT Description

16.9.6.15 IC_RX_TL

Bit	Access	Function	POR Value	Description
31:8	N/A	Reserved		Reserved
7:0	R/W	RX_TL	0x06	Receive FIFO Threshold Level Controls the level of entries (or above) that triggers the RX_FULL interrupt (bit 2 in IC_RAW_INTR_STAT register). The valid range is 0-255, with the additional restriction that hardware does not allow this value to be set to a value larger than the depth of the buffer. If an attempt is made to do that, the actual value set will be the maximum depth of the buffer. A value of 0 sets the threshold for 1 entry, and a value of 255 sets the threshold for 256 entries.

IC_RX_TL Description

16.9.6.16 IC_TX_TL

Bit	Access	Function	POR Value	Description
31:8	N/A	Reserved		Reserved
7:0	R/W	TX_TL	0x02	Receive FIFO Threshold Level Controls the level of entries (or above) that triggers the RX_FULL interrupt (bit 2 in IC_RAW_INTR_STAT register). The valid range is 0-255, with the additional restriction that hardware does not allow this value to be set to a value larger than the depth of the buffer. If an attempt is made to do that, the actual value set will be the maximum depth of the buffer. A value of 0 sets the threshold for 1 entry, and a value of 255 sets the threshold for 256 entries.

IC_TX_TL Description

16.9.6.17 IC_CLR_INTR

Bit	Access	Function	POR Value	Description
31:1	N/A	Reserved		Reserved
0	R	CLR_INTR	0	Read this register to clear the combined interrupt, all individual interrupts, and theIC_TX_ABRT_SOURCE register. This bit does not clear hardware clear-able interrupts but software clear-able interrupts. Refer to Bit 9 of the IC_TX_ABRT_SOURCE register for an exception to clearing IC_TX_ABRT_SOURCE.

IC_CLR_INTR Description

16.9.6.18 IC_CLR_RX_UNDER

Bit	Access	Function	POR Value	Description
31:1	N/A	Reserved		Reserved
0	R	CLR_RX_UNDER	0	Read this register to clear the RX_UNDER interrupt (bit 0) of the IC_RAW_INTR_STAT register.

IC_CLR_RX_UNDER Description

16.9.6.19 IC_CLR_RX_OVER

Bit	Access	Function	POR Value	Description
31:1	N/A	Reserved		Reserved
0	R	CLR_RX_OVER	0	Read this register to clear the RX_OVER interrupt (bit 1) of the IC_RAW_INTR_STAT register.

IC_CLR_RX_OVER Description

16.9.6.20 IC_CLR_TX_OVER

Bit	Access	Function	POR Value	Description
31:1	N/A	Reserved		Reserved
0	R	CLR_TX_OVER	0	Read this register to clear the TX_OVER interrupt (bit 3) of the IC_RAW_INTR_STAT register.

IC_CLR_TX_OVER Description

16.9.6.21 IC_CLR_RD_REQ

Bit	Access	Function	POR Value	Description
31:1	N/A	Reserved		Reserved
0	R	CLR_RD_REQ	0	Read this register to clear the RD_REQ interrupt (bit 5) of the IC_RAW_INTR_STAT register.

IC_CLR_RD_REQ Description

16.9.6.22 IC_CLR_TX_ABRT

Bit	Access	Function	POR Value	Description
31:1	N/A	Reserved		Reserved
0	R	CLR_TX_ABRT	0	Read this register to clear the TX_ABRT interrupt (bit 6) of the IC_RAW_INTR_STAT register, and the IC_TX_ABRT_SOURCE register. This also releases the Tx FIFO from the flushed/reset state, allowing more writes to the Tx FIFO. Refer to Bit 9 of the IC_TX_ABRT_SOURCE register for an exception to clearing IC_TX_ABRT_SOURCE.

IC_CLR_TX_ABRT Description

16.9.6.23 IC_CLR_RX_DONE

Bit	Access	Function	POR Value	Description
31:1	N/A	Reserved		Reserved
0	R	CLR_RX_DONE	0	Read this register to clear the RX_DONE interrupt (bit 7) of the IC_RAW_INTR_STAT register.

IC_CLR_RX_DONE Description

16.9.6.24 IC_CLR_ACTIVITY

Bit	Access	Function	POR Value	Description
31:1	N/A	Reserved		Reserved
0	R	CLR_ACTIVITY	0	Reading this register clears the ACTIVITY interrupt if the I2C is not active anymore. If the I2C module is still active on the bus, the ACTIVITY interrupt bit continues to be set. It is automatically cleared by hardware if the module is disabled and if there is no further activity on the bus. The value read from this register to get status of the ACTIVITY interrupt (bit 8) of the IC_RAW_INTR_STAT register.

IC_CLR_ACTIVITY Description

16.9.6.25 IC_CLR_STOP_DET

Bit	Access	Function	POR Value	Description
31:1	N/A	Reserved		Reserved
0	R	CLR_STOP_DET	0	Read this register to clear the STOP_DET interrupt (bit 9) of the IC_RAW_INTR_STAT register.

IC_CLR_STOP_DET Description

16.9.6.26 IC_CLR_START_DET

Bit	Access	Function	POR Value	Description
31:1	N/A	Reserved		Reserved
0	R	CLR_START_DET	0	Read this register to clear the START_DET interrupt (bit 10) of the IC_RAW_INTR_STAT register.

IC_CLR_START_DET Description

16.9.6.27 IC_CLR_GEN_CALL

Bit	Access	Function	POR Value	Description
31:1	N/A	Reserved		Reserved
0	R	CLR_GEN_CALL	0	Read this register to clear the GEN_CALL interrupt (bit 11) of the IC_RAW_INTR_STAT register.

IC_CLR_START_DET Description

16.9.6.28 IC_ENABLE

Bit	Access	Function	POR Value	Description
31:16	N/A	Reserved		Reserved
15:4	N/A	Reserved		Reserved
3	R/W	SDA_STUCK_RECOVERY_ENABLE	0	If SDA is stuck at low indicated through the TX_ABORT interrupt (IC_TX_ABRT_SOURCE[17]), then this bit is used as a control knob to initiate the SDA Recovery Mechanism (that is, send at most 9 SCL clocks and STOP to release the SDA line) and then this bit gets auto clear.
2	R/W	TX_CMD_BLOCK	0	In Master mode • 1’b1: Blocks the transmission of data on I2C bus even if Tx FIFO has data to transmit. • 1’b0: The transmission of data starts on I2C bus automatically, as soon as the first data is available in the Tx FIFO. Note: To block the execution of master commands, set the TX_CMD_BLOCK bit only when Tx FIFO is empty (IC_STATUS[2]=1) and the master is in the Idle state (IC_STATUS[5] == 0). Any further commands put in the Tx FIFO are not executed until TX_CMD_BLOCK bit is unset.
1	R/W	ABORT	0	When set, the controller initiates the transfer abort. • 0: ABORT not initiated or ABORT done • 1: ABORT operation in progress The software can abort the I2C transfer in master mode by setting this bit. The software can set this bit only when ENABLE is already set; otherwise, the controller ignores any write to ABORT bit. The software cannot clear the ABORT bit once set. In response to an ABORT, the controller issues a STOP and flushes the Tx FIFO after completing the current transfer, then sets the TX_ABORT interrupt after the abort operation. The ABORT bit is cleared automatically after the abort operation.

Bit	Access	Function	POR Value	Description
0	R/W	EN	0	Controls whether the I2C is enabled. • 0: Disables I2C (TX and RX FIFOs are held in an erased state) • 1: Enables I2C Software can disable I2C while it is active. However, it is important that care be taken to ensure that I2C is disabled properly. When I2C is disabled, the following occurs: • The TX FIFO and RX FIFO get flushed. • Status bits in the IC_INTR_STAT register are still active until I2C goes into IDLE state. If the module is transmitting, it stops as well as deletes the contents of the transmit buffer after the current transfer is complete. If the module is receiving, the I2C stops the current transfer at the end of the current byte and does not acknowledge the transfer. In systems with asynchronous pclk and ic_clk when IC_CLK_TYPE parameter set to asynchronous (1), there is a two ic_clk delay when enabling or disabling the I2C.

IC_ENABLE Description

16.9.6.29 IC_STATUS

Bit	Access	Function	POR Value	Description
31:12	N/A	Reserved		Reserved
11	R	SDA_STUCK_NOT_RECOVERED	0	This bit indicates that an SDA stuck at low is not recovered after the recovery mechanism.
10	R	SLV_HOLD_RX_FIFO_FULL	0	This bit indicates the BUS Hold in Slave mode due to the Rx FIFO being Full and an additional byte being received.
9	R	SLV_HOLD_TX_FIFO_EMPTY	0	This bit indicates the BUS Hold in Slave mode for the Read request when the Tx FIFO is empty. The Bus is in hold until the Tx FIFO has data to Transmit for the read request.
8	R	MST_HOLD_RX_FIFO_FULL	0	This bit indicates the BUS Hold in master mode due to Rx FIFO is Full and additional byte has been received.
7	R	MST_HOLD_TX_FIFO_EMPTY	0	The I2C Master stalls the write transfer when Tx FIFO is empty, and the the last byte does not have the Stop bit set. This bit indicates the BUS hold when the master holds the bus because of the Tx FIFO being empty, and the the previous transferred command does not have the Stop bit set.
6	R	SLV_ACTIVITY	0	Slave FSM Activity Status. When the Slave Finite State Machine(FSM) is not in the IDLE state, this bit is set. • 0: Slave FSM is in IDLE state so the Slave part of I2C is not Active • 1: Slave FSM is not in IDLE state so the Slave part of I2C is Active
5	R	MST_ACTIVITY	0	Master FSM Activity Status. When the Master Finite State Machine(FSM) is not in the IDLE state, this bit is set. • 0: Master FSM is in IDLE state so the master part of I2C is not Active • 1: Master FSM is not in IDLE state so the master part of I2C is Active NOTE: IC_STATUS[0]—that is, ACTIVITY bit—is the OR of SLV_ACTIVITY and MST_ACTIVITY bits.
4	R	RFF	0	Receive FIFO Completely Full. When the receive FIFO is completely full, this bit is set. When the receive FIFO contains one or more empty location, this bit is cleared. • 0: Receive FIFO is not full • 1: Receive FIFO is full
3	R	RFNE	0	Receive FIFO Not Empty. This bit is set when the receive FIFO contains one or more entries; it is cleared when the receive FIFO is empty. • 0: Receive FIFO is empty • 1: Receive FIFO is not empty

Bit	Access	Function	POR Value	Description
2	R	TFE	1	Transmit FIFO Completely Empty. When the transmit FIFO is completely empty, this bit is set. When it contains one or more valid entries, this bit is cleared. This bit field does not request an interrupt. • 0: Transmit FIFO is not empty • 1: Transmit FIFO is empty
1	R	TFNF	1	Transmit FIFO Not Full. Set when the transmit FIFO contains one or more empty locations, and is cleared when the FIFO is full. • 0: Transmit FIFO is full • 1: Transmit FIFO is not full
0	R	ACTIVITY	0	I2C Activity Status

IC_STATUS Description

16.9.6.30 IC_TXFLR

Bit	Access	Function	POR Value	Description
31:4	N/A	Reserved		Reserved
3:0	R	TXFLR	0	Transmit FIFO Level Contains the number of valid data entries in the transmit FIFO.

IC_TXFLR Description

16.9.6.31 IC_RXFLR

Bit	Access	Function	POR Value	Description
31:4	N/A	Reserved		Reserved
3:0	R	RXFLR	0	Receive FIFO Level Contains the number of valid data entries in the receive FIFO.

IC_RXFLR Description

16.9.6.32 IC_SDA_HOLD

Bit	Access	Function	POR Value	Description
31:24	N/A	Reserved		Reserved
23:16	R/W	IC_SDA_RX_HOLD	0	Sets the required SDA hold time in units of ic_clk period, when I2C acts as a reciever.
15:0	R/W	IC_SDA_TX_HOLD	1	Sets the required SDA hold time in units of ic_clk period, when I2C acts as a transmitter.

IC_SDA_HOLD Description

16.9.6.33 IC_TX_ABRT_SOURCE

Bit	Access	Function	POR Value	Description
31:23	R	TX_FLUSH_CNT	0	This field indicates the number of Tx FIFO data commands that are flushed due to TX_ABRT interrupt. It is cleared whenever I2C is disabled. I2C can be in either Master-Transmitter or Slave-Transmitter mode.
22:21	R	Reserved		These bits are reserved.
20	R	ABRT_DEVICE_WRITE	0	This is a Master-mode-only bit. Master is initiating the DEVICE_ID transfer and the Tx- FIFO consists of write commands. I2C is in Master mode.
19	R	ABRT_DEVICE_SLVADDR_NOACK	0	This is a Master-mode-only bit. Master is initiating the DEVICE_ID transfer and the Slave address sent was not acknowledged by any Slave. I2C is in Master mode.
18	R	ABRT_DEVICE_NOACK	0	This is a Master-mode-only bit. Master initiates the DEVICE_ID transfer and the device ID sent is not acknowledged by any Slave. I2C is in Master mode.
17	R	ABRT_SDA_STUCK_AT_LOW	0	This is a Master-mode-only bit. Master detects the SDA is Stuck at low for the IC_SDA_STUCK_AT_LOW_TIMEOUT value of ic_clks. I2C is in Master mode.
16	R	ABRT_USER_ABRT	0	This is a Master-mode-only bit. Master has detected the transfer abort (IC_ENABLE[1]). I2C is in Master mode.
15	R	ABRT_SLVRD_INTX	0	1: When the processor side responds to a Slave mode request for data to be transmitted to a remote Master and user writes a 1 in CMD (bit 8) of IC_DATA_CMD register. I2C is in Master-Transmitter mode.
14	R	ABRT_SLV_ARBLOST	0	1: Slave lost the bus while transmitting data to a remote Master. IC_TX_ABRT_SOURCE[12] is set at the same time. NOTE: Even though the Slave never “owns” the bus, something could go wrong on the bus. This is a fail safe check. For instance, during a data transmission at the low-to-high transition of SCL, if what is on the data bus is not what is supposed to be transmitted, then I2C no longer own the bus. I2C is in Slave-Transmitter mode.
13	R	ABRT_SLVFLUSH_TXFIFO	0	1: Slave has received a read command and some data exists in the TX FIFO so the Slave issues a TX_ABRT interrupt to flush old data in TX FIFO. I2C is in Slave-Transmitter mode.

Bit	Access	Function	POR Value	Description
12	R	ARB_LOST	0	1: Master has lost arbitration, or if IC_TX_ABRT_SOURCE[14] is also set, then the Slave transmitter has lost arbitration. I2C can be either Master-Transmitter or Slave-Transmitter mode.
11	R	ABRT_MASTER_DIS	0	1: User tries to initiate a Master operation with the Master mode disabled. I2C can be either Master-Transmitter or Master-Receiver mode.
10	R	ABRT_10B_RD_NORSTRT	0	1: The restart is disabled (IC_RESTART_EN bit (IC_CON[5]) = 0) and the Master sends a read command in 10-bit addressing mode. I2C is in Master-Receiver mode.
9	R	ABRT_SBYTE_NORSTRT	0	To clear Bit 9, the source of the ABRT_SBYTE_NORSTRT must be fixed first; restart must be enabled (IC_CON[5]=1), the SPECIAL bit must be cleared (IC_TAR[11]), or the GC_OR_START bit must be cleared (IC_TAR[10]). Once the source of the ABRT_SBYTE_NORSTRT is fixed, then this bit can be cleared in the same manner as other bits in this register. If the source of the ABRT_SBYTE_NORSTRT is not fixed before attempting to clear this bit, bit 9 clears for one cycle and then gets re-asserted. 1: The restart is disabled (IC_RESTART_EN bit (IC_CON[5]) = 0) and the user is trying to send a START Byte. I2C is in Master mode.
8	R	ABRT_HS_NORSTRT	0	1: The restart is disabled (IC_RESTART_EN bit (IC_CON[5]) = 0) and the user is trying to use the Master to transfer data in High Speed mode. I2C can be either Master-Transmitter or Master-Receiver mode.
7	R	ABRT_SBYTE_ACKDET	0	1: Master has sent a START Byte and the START Byte was acknowledged (wrong behavior). I2C is in Master mode.
6	R	ABRT_HS_ACKDET	0	1: Master is in High Speed mode and the High Speed Master code was acknowledged (wrong behavior). I2C is in Master mode.
5	R	ABRT_GCALL_READ	0	1: I2C in Master mode sent a General Call but the user programmed the byte following the General Call to be a read from the bus (IC_DATA_CMD[9] is set to 1). I2C is in Master-Transmitter mode.

Bit	Access	Function	POR Value	Description
4	R	ABRT_GCALL_NOACK	0	1: I2C in Master mode sent a General Call and no Slave on the bus acknowledged the General Call. I2C is in Master-Transmitter mode.
3	R	ABRT_TXDATA_NOACK	0	1: This is a Master-mode only bit. Master has received an acknowledgement for the address, but when it sent data byte(s) following the address, it did not receive an acknowledge from the remote Slave(s). I2C is in Master-Transmitter mode.
2	R	ABRT_10ADDR2_NOACK	0	1: Master is in 10-bit address mode and the second address byte of the 10-bit address was not acknowledged by any Slave. I2C can be either Master-Transmitter or Slave-Receiver mode.
1	R	ABRT_10ADDR1_NOACK	0	1: Master is in 10-bit address mode and the first 10-bit address byte was not acknowledged by any Slave. I2C can be either Master-Transmitter or Master-Receiver mode.
0	R	ABRT_7B_ADDR_NOACK	0	1: Master is in 7-bit addressing mode and the address sent was not acknowledged by any Slave. I2C can be in either Master-Transmitter or Master-Receiver mode

IC_TS_ABRT_SOURCE Description

16.9.6.34 IC_SLV_DATA_NACK_ONLY

Bit	Access	Function	POR Value	Description
31:1	N/A	Reserved		Reserved
0	R/W	NACK	0	Generate NACK. This NACK generation only occurs when I2C is a Slavereceiver. If this register is set to a value of 1, it can only generate a NACK after a data byte is received; hence, the data transfer is aborted and the data received is not pushed to the receive buffer. When the register is set to a value of 0, it generates NACK/ACK, depending on normal criteria. • 1 - generate NACK after data byte received • 0 - generate NACK/ACK normally

IC_SLV_DATA_NACK_ONLY Description

16.9.6.35 IC_DMA_CR

Bit	Access	Function	POR Value	Description
31:2	N/A	Reserved		Reserved
1	R/W	TDMAE	0	Transmit DMA Enable. This bit enables/disables the transmit FIFO DMA channel. • 0 - Transmit DMA disabled • 1 - Transmit DMA enabled
0	R/W	RDMAE	0	Receive DMA Enable. This bit enables/disables the receive FIFO DMA channel. • 0 - Receive DMA disabled • 1 - Receive DMA enabled

IC_DMA_CR Description

16.9.6.36 IC_DMA_TDLR

Bit	Access	Function	POR Value	Description
31:4	N/A	Reserved		Reserved
3:0	R/W	DMATDL	0	Transmit Data Level. This bit field controls the level at which a DMA request is made by the transmit logic. It is equal to the watermark level; that is, the dma_tx_req signal is generated when the number of valid data entries in the transmit FIFO is equal to or below this field value, and TDMAE = 1.

IC_DMA_TDLR Description

16.9.6.37 IC_DMA_RDLR

Bit	Access	Function	POR Value	Description
31:4	N/A	Reserved		Reserved
3:0	R/W	DMARDL	0	Receive Data Level. This bit field controls the level at which a DMA request is made by the receive logic. The watermark level = DMARDL+1; that is, dma_rx_req is generated when the number of valid data entries in the receive FIFO is equal to or more than this field value + 1, and RDMAE = 1. For instance, when DMARDL is 0, then dma_rx_req is asserted when 1 or more data entries are present in the receive FIFO.

IC_DMA_RDLR Description

16.9.6.38 IC_SDA_SETUP

Bit	Access	Function	POR Value	Description
31:8	N/A	Reserved		Reserved
7:0	R/W	SDA_SETUP	0x64	SDA Setup. It is recommended that if the required delay is 1000ns, then for an ic_clk frequency of 10 MHz, IC_SDA_SETUP should be programmed to a value of 11. IC_SDA_SETUP must be programmed with a minimum value of 2.

IC_SDA_SETUP Description

16.9.6.39 IC_ACK_GENERAL_CALL

Bit	Access	Function	POR Value	Description
31:1	N/A	Reserved		Reserved
0	R/W	ACK_GEN_CALL	0x1	ACK General Call. When set to 1, I2C responds with a ACK (by asserting ic_data_oe) when it receives a General Call. When set to 0, the I2C does not generate General Call interrupts.

IC_ACK_GENERAL_CALL Description

16.9.6.40 IC_ENABLE_STATUS

Bit	Access	Function	POR Value	Description
31:3	N/A	Reserved		Reserved
2	R	SLV_RX_DATA_LOST	0	Slave Received Data Lost. This bit indicates if a Slave-Receiver operation has been aborted with at least one data byte received from an I2C transfer due to setting IC_ENABLE[0] from 1 to 0. When read as 1, I2C is deemed to have been actively engaged in an aborted I2C transfer (with matching address) and the data phase of the I2C transfer has been entered, even though a data byte has been responded with a NACK. NOTE: If the remote I2C Master terminates the transfer with a STOP condition before the I2C has a chance to NACK a transfer, and IC_ENABLE[0] has been set to 0, then this bit is also set to 1. When read as 0, I2C is deemed to have been disabled without being actively involved in the data phase of a Slave- Receiver transfer. NOTE: The CPU can safely read this bit when IC_EN (bit 0) is read as 0.
1	R	SLV_DISABLED_WHILE_BUSY	0	Slave Disabled While Busy (Transmit, Receive). This bit indicates if a potential or active Slave operation has been aborted due to setting bit 0 of the IC_ENABLE register from 1 to 0. This bit is set when the CPU writes a 0 to bit 0 of IC_ENABLE while: (a) I2C is receiving the address byte of the Slave-Transmitter operation from a remote Master; OR, (b) address and data bytes of the Slave-Receiver operation from a remote Master. When read as 1, I2C is deemed to have forced a NACK during any part of an I2C transfer, irrespective of whether the I2C address matches the Slave address set in I2C (IC_SAR register) OR if the transfer is completed before bit 0 of IC_ENABLE is set to 0, but has not taken effect. NOTE: If the remote I2C Master terminates the transfer with a STOP condition before the i2c has a chance to NACK a transfer and bit 0 of IC_ENABLE has been set to 0, then this bit will also be set to 1. When read as 0, I2C is deemed to have been disabled when there is Master activity, or when the I2C bus is idle. NOTE: The CPU can safely read this bit when IC_EN (bit 0) is read as 0

Bit	Access	Function	POR Value	Description
0	R	IC_EN	0x0	ic_en Status. This bit always reflects the value driven on the output port ic_en. ■ When read as 1, i2c is deemed to be in an enabled state. ■ When read as 0, i2c is deemed completely inactive. .

IC_ENABLE_STATUS Description

16.9.6.41 IC_FS_SPKLEN

Bit	Access	Function	POR Value	Description
31:8	N/A	Reserved		Reserved
7:0	R/W	IC_FS_SPKLEN	0x07	This register must be set before any I2C bus transaction can take place to ensure stable operation. This register sets the duration, measured in ic_clk cycles, of the longest spike in the SCL or SDA lines that are filtered out by the spike suppression logic. This register can be written only when the I2C interface is disabled, which corresponds to IC_ENABLE[0] being set to 0. Writes at other times have no effect. The minimum valid value is 1; hardware prevents values less than this being written, and if attempted, results in 1 being set.

IC_FS_SPKLEN Description

16.9.6.42 IC_HS_SPKLEN

Bit	Access	Function	POR Value	Description
31:8	N/A	Reserved		Reserved
7:0	R/W	IC_HS_SPKLEN	0x02	This register must be set before any I2C bus transaction can take place to ensure stable operation. This register sets the duration, measured in ic_clk cycles, of the longest spike in the SCL or SDA lines that are filtered out by the spike suppression logic. This register can be written only when the I2C interface is disabled, which corresponds to IC_ENABLE[0] being set to 0. Writes at other times have no effect. The minimum valid value is 1; hardware prevents values less than this being written, and if attempted, results in 1 being set. This register is implemented only if the component is configured to support HS mode; that is, if the IC_MAX_SPEED_MODE parameter is set to 3.

IC_HS_SPKLEN Description

16.9.6.43 IC_CLR_RESTART_DET

Bit	Access	Function	POR Value	Description
31:1	N/A	Reserved		Reserved
0	R	CLR_RESTART_DET	0	Read this register to clear the RESTART_DET interrupt (bit 12) of the IC_RAW_INTR_STAT register.

IC_CLR_RESTART_DET Description

16.9.6.44 IC_COMP_PARAM_1

Bit	Access	Function	POR Value	Description
31:24	N/A	Reserved		Reserved
23:16	R	TX_BUFFER_DEPTH	0x08	• 0x00 = Reserved • 0x01 = 2 • 0x02 = 3 … • 0xFF = 256
15:8	R	RX_BUFFER_DEPTH	0x08	• 0x00 = Reserved • 0x01 = 2 • 0x02 = 3 … • 0xFF = 256
7	R	ADD_ENCODED_PARAMS	0x1	Reading 1 in this bit means that the capability of reading these encoded parameters via software has been included. Otherwise, the entire register is 0 regardless of the setting of any other parameters that are encoded in the bits.
6	R	HAS_DMA	0x1	Configures the inclusion of DMA handshaking interface signals.
5	R	INTR_IO	0x1	Controls which interrupt outputs are present. • 0: Individual - each interrupt source has its own output • 1: Combined - all interrupt sources are combined in to a single output
4	R	HC_COUNT_VALUES	0x0	Setting this parameter to 1 will cause the CNT registers to read only. Setting this parameter to 0 will allow the CNT registers to be writeable. Regardless of the setting, the CNT registers are always readable and have reset values from the corresponding configuration parameters, which may be user defined or else derived.
3:2	R	MAX_SPEED_MODE	0x3	• 0x0 = Reserved • 0x1 = Standard • 0x2 = Fast • 0x3 = High
1:0	R	CLR_RESTART_DET	0	Read this register to clear the RESTART_DET interrupt (bit 12) of the IC_RAW_INTR_STAT register.

IC_COMP_PARAM_1 Description

16.9.6.45 IC_COMP_VERSION

Bit	Access	Function	POR Value	Description
31:0	R	IC_COMP_VERSION	0x3230302a	Specific I2C component version number

IC_COMP_VERSION Description

16.9.6.46 IC_COMP_TYPE

Bit	Access	Function	POR Value	Description
31:0	R	IC_COMP_TYPE	0x0	This assigned unique hex value is constant.

IC_COMP_TYPE Description

16.9.6.47 IC_SCL_STUCK_AT_LOW_TIMEOUT

This register is used to store the duration, measured in ic_clk cycles, used to generate an Interrupt (SCL_STUCK_AT_LOW) if SCL is held low for the IC_SCL_STUCK_LOW_TIMEOUT duration.

Bit	Access	Function	POR Value	Description
31:0	R/W	IC_SCL_STUCK_LOW_TIMEOUT	0xFFFFFFFF	I2C generates the interrupt to indicate SCL stuck at low if it detects the SCL stuck at low for the IC_SCL_STUCK_LOW_TIMEOUT in units of ic_clk period.

IC_SCL_STUCK_AT_LOW_TIMEOUTDescription

16.9.6.48 IC_SDA_STUCK_AT_LOW_TIMEOUT

This register is used to store the duration, measured in ic_clk cycles, used to recover the Data (SDA) line through sending SCL pulses if SDA is held low for the mentioned duration.

Bit	Access	Function	POR Value	Description
31:0	R/W	IC_SDA_STUCK_LOW_TIMEOUT	0xFFFFFFFF	I2C initiates the recovery of SDA line through enabling the SDA_STUCK_RECOVERY_EN (IC_ENABLE[3]) register bit, if it detects the SDA stuck at low for the IC_SDA_STUCK_LOW_TIMEOUT in units of ic_clk period.

IC_SDA_STUCK_AT_LOW_TIMEOUT Description

16.9.6.49 IC_CLR_SCL_STUCK_DET

Bit	Access	Function	POR Value	Description
31:1	N/A	Reserved		Reserved
0	R	CLR_SCL_STUCK	0	Read this register to clear the SCL_STUCK_DET interrupt (bit 14) of the IC_RAW_INTR_STAT register.

IC_SDA_STUCK_AT_LOW_TIMEOUT Description

16.9.6.50 IC_DEVICE_ID

This register contains the Device-ID of the component, which includes 12 bits of manufacturer name, 9 bits of part identification and 3 bits of die-version.

Bit	Access	Function	POR Value	Description
31:1	N/A	Reserved		Reserved
0	R	DEVICE-ID	0x1	Contains the Device-ID of the component.

IC_DEVICE_ID Description

16.9.6.51 IC_OPTIONAL_SAR

Bit	Access	Function	POR Value	Description
15:0	N/A	Reserved		Reserved

IC_OPTIONAL_SAR Description

16.10 I2S/PCM Primary and Secondary

16.10.1 General Description

There are three I^{2}S controllers - one in the MCU HP peripherals (I2S0), one in the MCU ULP subsystem (ULP_I2S) and one in the security/NWP subsystem. Each I^{2}S controller supports PCM mode of operation also.

I2S is a protocol used for digital stereo audio. It is used in systems that process digital audio signals, such as:

- A/D and D/A converters
- digital signal processors
- error correction for compact disc and digital recording
- digital filters
- digital input/output interfaces

16.10.2 Features

I2S

The each I^{2}S controllers support the following features:

- The I2S0 supports two stereo channels while the ULP_I2S and the NWP/Security subsystem I^{2}S support one stereo channel
- Programmable Audio data resolutions of 16, 24 and 32 bits
- Supported audio sampling rates are 8, 11.025, 16, 22.05, 24, 32, 44.1, 48, 88.2, 96 and 192 kHz
- Support for Master and Slave modes
- Full duplex communication due to the independence of transmitter and receiver
- The PCM mode of operation supports the following features
  - Supports only mono audio data
  - Supports only short frame sync
  - Supports two modes for data transmission with respect to the Frame Synchronization signal – the MS bit is transmitted in the same clock cycle as the Frame Synchronization signal is asserted or one clock cycle after the Frame Synchronization signal is asserted
- Programmable FIFO thresholds with maximum FIFO depth of 8 and support for DMA
- Supports generation of interrupts for different events

  The I^{2}S in the MCU ULP subsystem supports the following additional power-save features:

- After the DMA is programmed in PS2 state for I^{2}S transfers, the MCU can switch to PS1 state (processor is shutdown) while the I^{2}S controller continues with the data transfer
- In PS1 state (ULP Peripheral mode) the I^{2}S controller completes the data transfer and, triggered by the Peripheral Interrupt, shifts either to the sleep state (without processor intervention) or the active state

  The NWP/Security subsystem's I^{2}S controller allows direct bridging between audio data and the wireless link without the MCU's intervention.

PCM

- The PCM interface works in Slave mode which can transmit and receive serial data to and from an off-chip PCM Master.
- Supports full-duplex data transfer with a PCM Master.
- Supports only mono audio data
- Supports only short frame sync
- Supports 16, 24 and 32-bit frame sizes.
- Data is driven at the rising edge of clock and sampled at the falling edge of clock.
- Maximum operating frequency is 24 MHPCM Master/Slave mode can be configured (can be configured through the register named i2s_master_slave_mode 1-Master, 0-Slave (present in misc config registers))

16.10.3 Functional Description

The figure below illustrates a block diagram of the I2S.

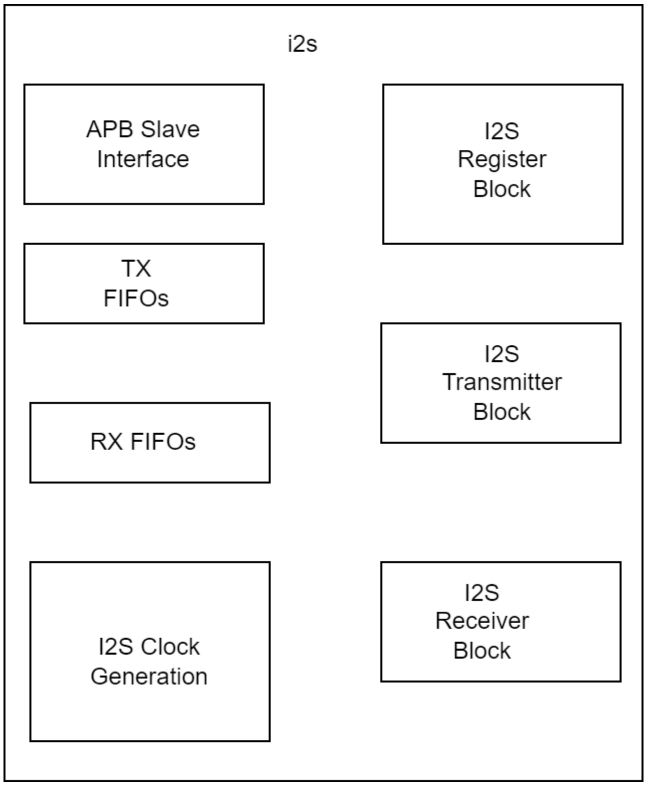


Figure 16.25. I2S Block diagram

The bus consists of a serial data line (sd), a word select line (ws), and a serial clock (sclk). The serial data line is time multiplexed to allow the transfer of two data streams (such as, left and right stereo data). It supports up to two data channels for both transmit and receive operations. The figure below illustrates simple system configurations for the I2S component.

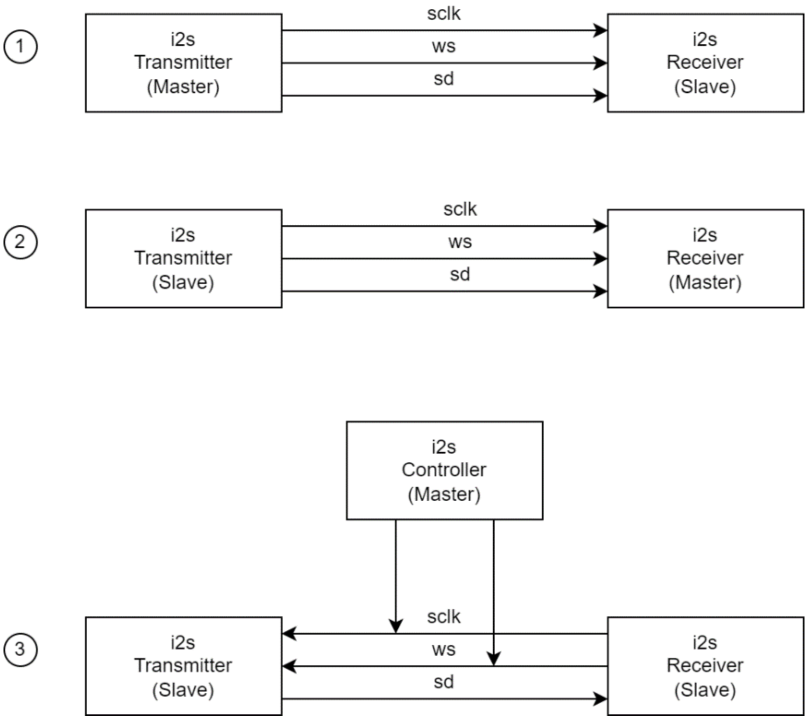


Figure 16.26. Simple System Configurations for I2S

The Master is responsible for generating the shared sclk and ws clocking signals. In complex systems where there may be several transmitters and receivers, a separate system primary can be used.

As illustrated in Figure 16.26 Simple System Configurations for I2S on page 468, this system primary can also be combined with one of the transmitters or receivers inthe system. The “controller” in this example is enabled and disabled by configuring the component to act as a primary and by programming the clock enable and clock configuration registers. The serial data is transmitted in two’s complement format with the most significant bit (MSB) first. This means that the transmitter and receiver can have different word lengths, and neither the transmitter nor receiver needs to know what size words the other can handle. If the word being transferred is too large for the receiver, the least significant bits (LSB) are truncated. Similarly, if the word size is less than what the receiver can handle, the data is zero padded. The word select line is used to time the multiplexed data streams. For instance, when ws is low, the word being transferred is left stereo data; when ws is high, the word being transferred is right stereo data. This format is illustrated in the figure below. For standard I2S formats, the MSB of a word is sent one sclk cycle after a ws change. Serial data sent by the transmitter can be synchronized with either the negative edge or positive edge of the sclk signal. However, the receiver must latch the serial data on the rising edge of sclk.

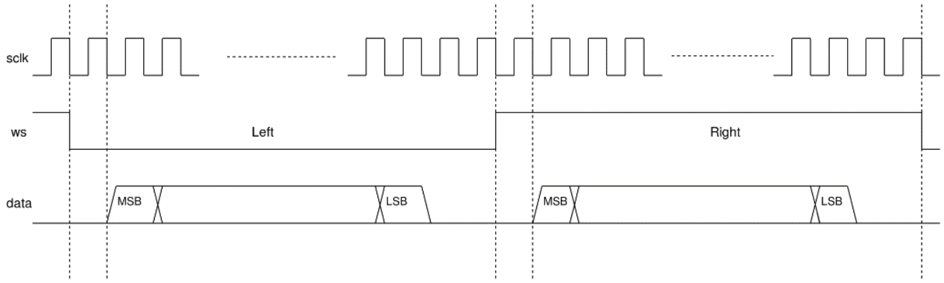


Figure 16.27. I2S Stereo Frame Format

The I2S component can be configured to support up to two stereo I2S transmit (TX) channels. These channels can operate in either Master or Slave mode. By default, I2S is configured in Slave mode. Stereo data pairs (such as, left and right audio data) written to a TX channel via the APB bus are shifted out serially on the appropriate serial data out line (sdo0, sdo1, sdo2, sdo3). The shifting is timed with respect to the serial clock (sclk) and the word select line (ws). The instantiation of the I2S transmitter block and the number of TX channels is determined by the two configuration parameters: Transmitter Block Enabled (I2S_TRANSMITTER_BLOCK) and Number of Transmit Channels (I2S_TX_CHANNELS), respectively.

Each TX channel is initially configured with a maximum audio data resolution as set by the Maximum Audio Resolution parameter (I2S_TX_WORDSIZE_x, where x is the channel number). A TX channel can be reprogrammed during operation to any supported audio data resolution that is less than I2S_TX_WORDSIZE_x.
I2S can be configured to support up to four stereo I2S receive (RX) channels. These channels can operate in either Master or Slave mode. By default, I2S is configured in Slave mode. Stereo data pairs (such as, left and right audio data) are received serially from a data input line (sdi0, sdi1, sdi2, sdi3). These data words are stored in RX FIFOs until they are read via the APB bus. The receiving is timed with respect to the serial clock (sclk) and the word select line (ws).
I2S Master PLL must be able to output following clock frequencies. The clock frequency is related to the sampling frequency of I2S module according to the relation given below.
Clock frequency = 2*bit_width*Sampling_freq.

Table 16.184. I2S Clock Frequencies

Sampling freq(kHz)	Clock Frequency (MHz)			
	16 Bits	32 Bits	48 Bits	64 bits
8	0.256	0.512	0.768	1.024
16	0.512	1.024	1.536	2.048
32	1.024	2.048	4.2336	5.6448
44.1	1.4112	2.8224	4.2336	5.6448
48	1.536	3.072	4.608	6.144
88.2	2.8224	5.6448	8.4672	11.2896
96	3.072	6.144	9.216	12.288

Sampling freq(kHz)	Clock Frequency (MHz)			
192	6.144	12.288	18.432	24.576

16.10.4 Programming Sequence of I2S

16.10.4.1 I2S Enabling

Enable the I2S component before any data can be received or transmitted into the FIFOs. To enable the component, set the I2S Enable (IEN) bit of the I2S Enable Register (IER) to 1. When you disable the device, it acts as a global disable. To disable I2S, set IER[0] to 0. After disable, the following events occur:

- TX and RX FIFOs are cleared, and read/write pointers are reset;
- Any data in the process of being transmitted or received is lost;
- All other programmable enables (such as transmitter/receiver block enables and individual TX/RX channel enables) in the component are overridden;
- Generation of Master mode clock signals sclk_en, ws_out and sclk_gate are disabled

When I2S is enabled and configured as a master, the device always starts in the left stereo data cycle (ws = 0), and one sclk cycle later transitions to the right stereo data cycle (ws = 1). This allows for half a frame of sclks to write data to the TX FIFOs and to ensure that any connected Slave receivers do not miss the start of the data frame (for instance, the ws 1-to-0 transition) once the sclk restarts.

16.10.4.2 I2S as Transmitter

The I2S component can be configured to support up to two stereo I2S transmit (TX) channels. These channels can operate in either Master or Slave mode. By default, I2S is configured in Slave mode. Stereo data pairs (such as, left and right audio data) written to a TX channel via the APB bus are shifted out serially on the appropriate serial data out line (sdo0, sdo1, sdo2, sdo3). The shifting is timed with respect to the serial clock (sclk) and the word select line (ws).

The figure below illustrates the basic usage flow for I2S when it acts as a transmitter.

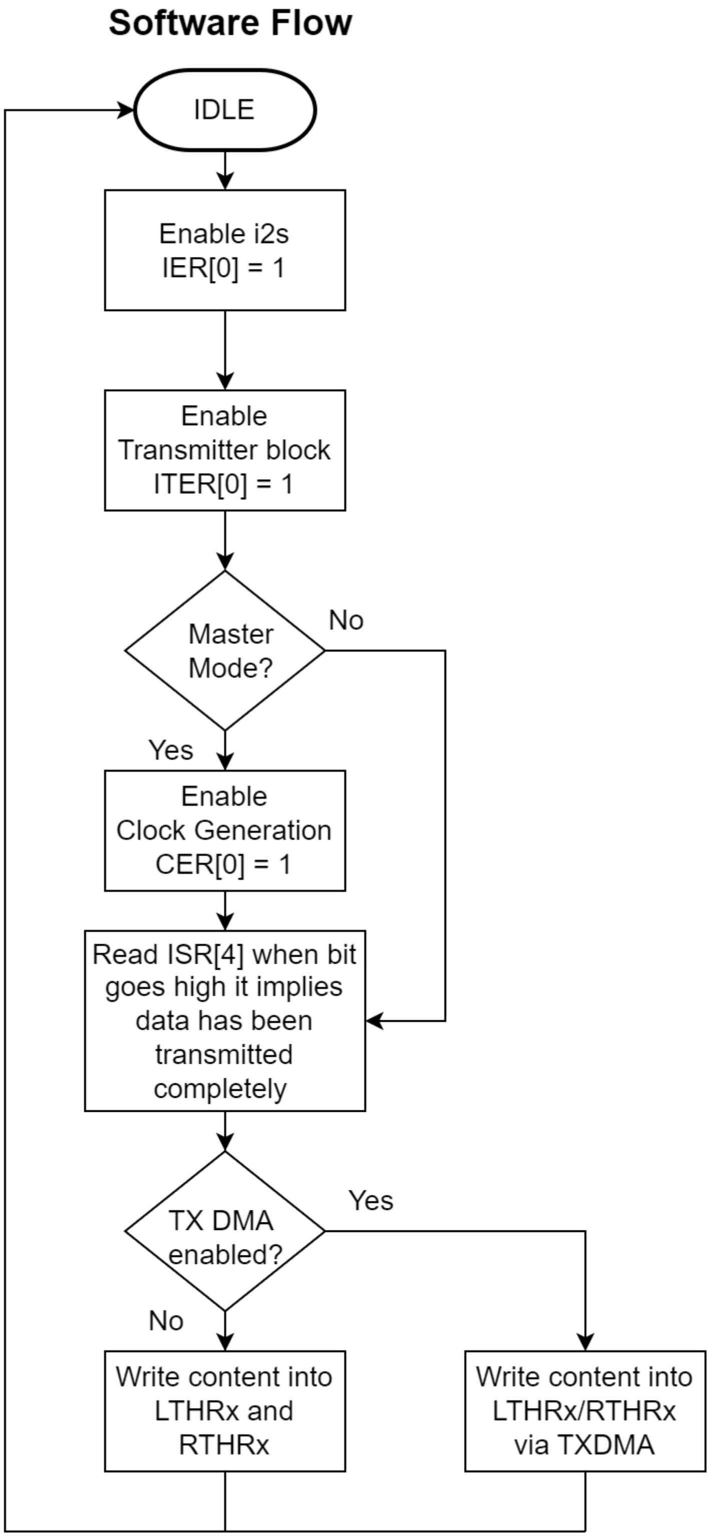

Software Flow
IDLE
Enable i2s
IER[0] = 1
Enable
Transmitter block
ITER[0] = 1
Master
Mode?
No
Yes
Enable
Clock Generation
CER[0] = 1
Read ISR[4] when bit goes high it implies data has been transmitted completely
TX DMA
enabled?
Yes
No
Write content into LTHRx and RTHRx
Write content into LTHRx/RTHRx via TXDMA

Figure 16.28. Transmitter Block Enable

The Transmitter Block Enable (TXEN) bit of the I2S Transmitter Enable Register (ITER) globally turns on and off all of the configured TX channels. To enable the transmitter block, set ITER[0] to 1. To disable the block, set ITER[0] to 0. When the transmitter block is disabled, the following events occur:

- Outgoing data is lost and the channel outputs are held low;
- Data in the TX FIFOs are preserved and the FIFOs can be written to;
- Any previous programming (like changes in word size, threshold levels, and so on) of the TX channels is preserved.
- Any individual TX channel enables are overridden.

When the transmitter block is enabled, if there is data in the TX FIFOs, the channel resumes transmission on the next left stereo data cycle.

When the block is disabled, perform any of the following procedures:

- Program (or further program) TX channel registers
- Flush the TX FIFOs by programming the Transmitter FIFOs Reset bit of the Transmitter FIFO Flush Register (TXFFR[0] = 1)
- Flush an individual channel's TX FIFO by programming the Transmit Channel FIFO Reset (TXCHFR) bit of the Transmit FIFO Flush Register (TFFx[0] = 1, where x is the channel number)

Transmit Channel Enable

Each transmit channel has its own enable/disable that can be set independently of the other channels to allow the reprogramming of a channel and to flush the channel's TX FIFOs while other TX channels are transmitting. This enable/disable is controlled by bit 0 of the Transmitter Enable Register (TERx, where x is the channel number). For example, to enable TX Channel 1, write a 1 to TER1[0]. To disable this channel, write a 0 to TER1[0]. When a TX channel is disabled, the following occurs:

- Outgoing stereo data is lost;
- Channel output is held low;
- Data in the TX FIFO is preserved, and the FIFO can be written to; and
- Any previous programming of the TX channel's registers is preserved, and the registers can be further reprogrammed.

When a TX channel is disabled, flush the channel's TX FIFO by programming the Transmit Channel FIFO Reset (TXCHFR) bit of the Transmit FIFO Flush (TFFx[0] = 1, where x is the channel number). When the TX channel is enabled, if there is data in the TX FIFO, the channel resumes transmission on the next left stereo data cycle (such as, when the ws line goes low)

Transmit Channel Audio Data Resolution

Each TX channel is initially configured with a maximum audio data resolution as set by the Maximum Audio Resolution parameter. A TX channel can be reprogrammed during operation to any supported audio data resolution that is less than Maximum Audio Resolution. Changes to the resolution are programmed via the Word Length (WLEN) bits of the Transmitter Configuration Registers (TCRx[2:0], where x is the channel number). The channel must be disabled prior to any resolution changes. On reset or if an invalid resolution is selected, the TX channel's audio data resolution defaults back to the initial value.

Transmit Channel Interrupts

All interrupts in I2S are active interrupts. Each TX channel generates two interrupts: TX FIFO Empty and Data Overrun.

- TX FIFO Empty interrupt – This interrupt is asserted when the empty trigger threshold level for the TX FIFO is reached. When this interrupt is included on the I/O, it appears on the outputs tx_emp_x_intr (where x is the channel number). A TX FIFO Empty interrupt is cleared by writing data to the TX FIFO to bring its level above the empty trigger threshold level for the channel.
- Data Overrun interrupt –This interrupt is asserted when an attempt is made to write to a full TX FIFO (any data being written is lost while data in the FIFO is preserved). When this interrupt is included on the I/O, it appears on the outputs tx_or_x_intr (where x is the channel number). A Data Overrun interrupt is cleared by reading the Transmit Channel Overrun (TXCHO) bit [0] of the Transmit Overrun Register (TORx, where x is the channel number).

The interrupt status of any TX channel can be determined by polling the Interrupt Status Register (ISRx where x is the channel number). The TXFE bit [4] indicates the status of the TX FIFO Empty interrupt, while the TXFO bit [5] indicates the status of the Data Overrun interrupt. Both the TX FIFO Empty and Data Overrun interrupts can be masked off by writing a 1 in the Transmit Empty Mask (TXFEM) and Transmit Overrun Mask (TXFOM) bits of the Interrupt Mask Register (IMRx where x is the channel number), respectively. This prevents the interrupts from driving their output lines, however, the ISRx always shows the current status of the interrupts regardless of any masking.

Writing to a Transmit Channel

The stereo data pairs to be transmitted by a TX channel are written to the TX FIFOs via the Left Transmit Holding Register (LTHRx, where x is the channel number) and the Right Transmit Holding Register (RTHRx, where x is the channel number). All stereo data pairs must be written using the following two- stage process:

1. Write left stereo data to LTHRx

2. Write right stereo data to RTHRx.

All enabled TX channels starting from the lowest-numbered enabled channel. After a stereo data pair is transmitted, the component will point to the next enabled channel. When I2S is enabled, if the TX FIFO is empty and data is not written to the FIFOs before the next left cycle, the channel outputs zeros for a full frame (left and right cycle). Transmission only commences if there is data in the TX FIFO prior to the transition to the left data cycle. In other words, if the start of the frame is missed, the channel output idles until the next available frame.

16.10.4.3 I2S as Receiver

I2S supports up to two stereo I2S receive (RX) channels. These channels can operate in either Primary or Secondary mode. By default, I2S is configured in secondary mode. Stereo data pairs (such as, left and right audio data) are received serially from a data input line (sdi0, sdi1, sdi2, sdi3). These data words are stored in RX FIFOs until they are read via the APB bus. The receiving is timed with respect to the serial clock (sclk) and the word select line (ws). The figure below illustrates the basic usage flow for I2S when it acts as a receiver.

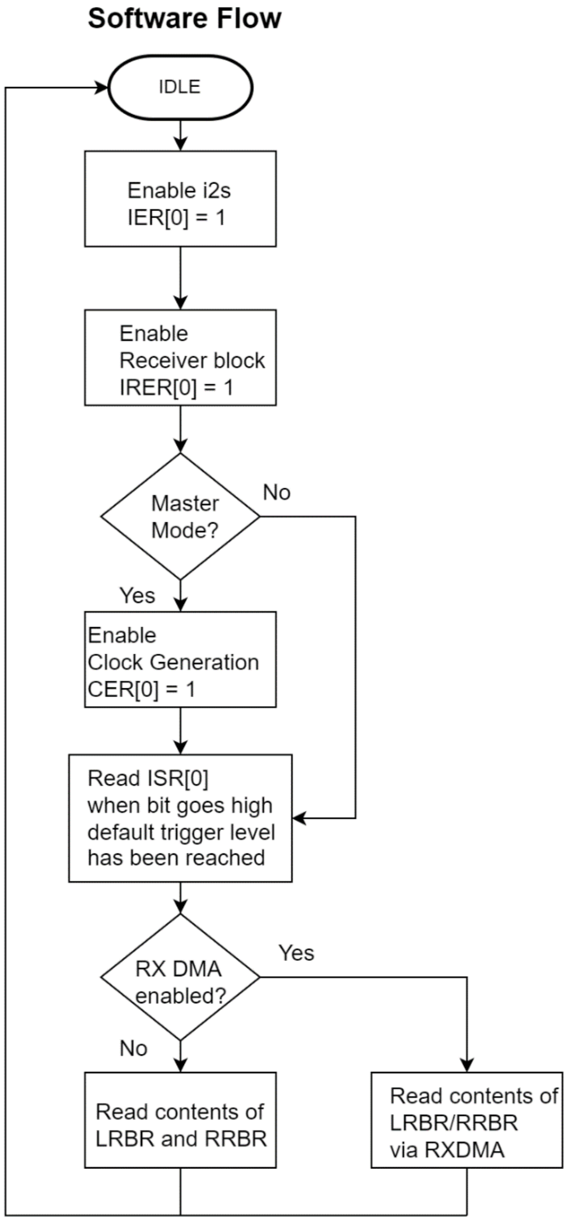


Figure 16.29. Basic Usage Flow for I2S as Receiver

Receiver Block Enable

The Receiver Block Enable (RXEN) bit of the I2S Receiver Enable Register (IRER) enables/disables all configured RX channels. To enable the receiver block, set IRER[0] to ‘1.’ To disable the block, set this bit to ‘0.’ When the receiver block is disabled, the following events occur:

- Incoming data is lost
- Data in the RX FIFOs is preserved and the FIFOs can be read.
- Any previous programming (such as changes in word size, threshold levels, and so on) of the RX channels is preserved

- Any individual RX channel enable is overridden. Enabling the channel resumes receiving on the next left stereo data cycle (for instance, when ws goes low).

  When the block is disabled, you can perform any of the following procedures:

- Program (or further program) the RX channel registers;
- Flush the RX FIFOs by programming the Receiver FIFOs Reset (RXFR) bit of the Receiver FIFO Flush Register (RXFFR[0] = 1).
- Flush an individual channel's RX FIFO by programming the Receive Channel FIFO Reset (RXCHFR) bit of the Receive FIFO Flush Register (RFFx [0] = 1, where x is the channel number).

Receive Channel Enable

Each RX channel has its own enable/disable that can be set independently of the other channels to allow programming of the channel and to clear the channel's RX FIFO while other RX channels are still receiving data. This enable/disable is controlled by bit 0 of the Receiver Enable Register (RERx[0], where x is the channel number). For example, to enable RX Channel 1, write a 1 to RER1[0]. To disable this channel, write a 0 to RER1[0]. When the RX channel is disabled, the following occurs:

- Incoming data is lost
- Data in the RX FIFO is preserved
- FIFO can be read
- Previous programming of the RX channel is preserved.
- RX channel can be further programmed.

When the RX channel or block is disabled, flush the channel's RX FIFO by writing 1 in bit 0 of the Receive FIFO Flush Register (RFFx, where x is the channel number). When the channel is enabled, it resumes receiving on the next left stereo data cycle (for instance. when ws line goes low).

Receive Channel Audio Data Resolution

Each RX channel is initially configured with a maximum audio data resolution. A RX channel can be reprogrammed during operation to any supported audio data resolution that is less than Maximum Audio Resolution. Changes to the resolution are programmed via the Word Length (WLEN) bits of the Transmitter Configuration Registers (RCRx[2:0], where x is the channel number). The channel must be disabled prior to any resolution changes. On reset or if an invalid resolution is selected, the RX channel's audio data resolution defaults back to the initial value.

Receive Channel Interrupts

Each RX channel generates two interrupts: RX FIFO Data Available and Data Overrun.

- RX FIFO Data Available interrupt – This interrupt is asserted when the trigger level for the RX FIFO is reached. When this interrupt is included on the I/O, it appears on the outputs rx_da_x_intr (where x is the channel number). This interrupt is cleared by reading data from the RX FIFO until its level drops below the data available trigger level for the channel.
- Data Overrun interrupt – This interrupt is asserted when an attempt is made to write received data to a full RX FIFO (any data being written is lost while data in the FIFO is preserved). When this interrupt is included on the I/O, it appears on the outputs rx_or_x_intr (where x is the channel number). This interrupt is cleared by reading the Receive Channel Overrun (RXCHO) bit [0] of the Receive Overrun Register (RORx, where x is the channel number).

The interrupt status of any RX channel can be determined by polling the Interrupt Status Register (ISRx where x is the channel number). The RXDA bit [0] indicates the status of the RX FIFO Data Available interrupt. the RXFO bit [1] indicates the status of the RX FIFO Data Overrun interrupt. Both the Receive Empty Threshold and Data Overrun interrupts can be masked by writing a 1 in the Receive Empty Threshold Mask (RDM) and Receive Overrun Mask (ROM) bits of the Interrupt Mask Register (IMRx, where x is the channel number), respectively. This prevents the interrupts from driving their output lines, however, the ISRx always shows the current status of the interrupts regardless of any masking.

Reading from a Receive Channel

The stereo data pairs received by a RX channel are written to the left and right RX FIFOs. These FIFOs can be read via the Left Receive Buffer Register (LRBRx, where x is the channel number) and the Right Receive Buffer Register (RRBRx, where x is the channel number). All stereo data pairs must be read using the following two-stage process:

1. Read the left stereo data from LRBRx

2. Read the right stereo data from RRBRx

16.10.5 PCM

The PCM interface can transmit and receive serial data to and from an off-chip PCM Master/Slave. It uses the I2S Master/Slave engine for data transmission and reception.

As a result of this, there is a delay of one frame while transmitting data. The following are the features supported by the PCM interface.

- Supports full-duplex data transfer with a PCM Master/Slave.
- Supports two modes for data transmission with respect to the Frame Synchronization signal – the MS bit is transmitted in the same clock cycle as the Frame Synchronization signal is asserted or one clock after the Frame Synchronization signal is asserted.
- Supports 16, 24 and 32-bit frame sizes.
- Data is driven at the rising edge of clock and sampled at the falling edge of clock.

The two figures below show the timing diagrams of the PCM interface module. The diagrams are shown for the 12-bit frame size. The same diagrams apply for 12, 16, 20, 24 and 32-bit frame sizes. If they are an extra clock cycles between each pcm_fsync signal assertion, the data during those clock cycles is ignored. The PCM_FSYNC signal may be asserted for 1 clock (short FSYNC) or more clocks (long FSYNC) but should be deasserted before the last bit is transmitted.

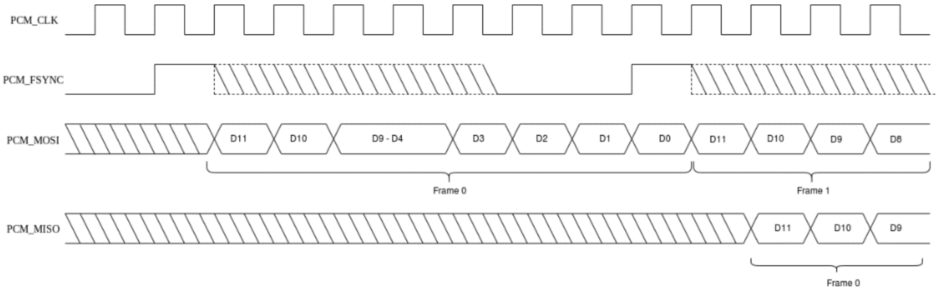


Figure 16.30. Timing Diagram for 12-bit Frame Size and FSYNC Asserted with LS bit

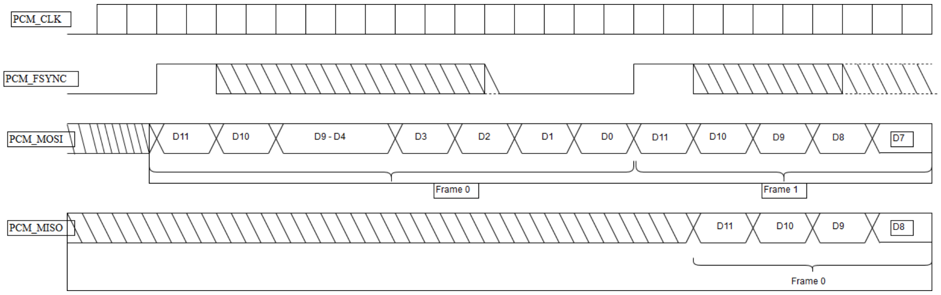


Figure 16.31. Timing Diagram for 12-bit Frame Size and FSYNC Asserted with MS bit

16.10.6 Register Summary

Base Address: 0x4705_0000

ULP_I2S Base Address: 0x2404_0400

Table 16.185. Register Summary Table

Register Name	Offset	Description
16.10.7.1 I2S_IER	0x000	i2s Enable Register
16.10.7.2 I2S_IRER	0x004	I2S Receiver Block Enable Register
16.10.7.3 I2S_ITER	0x008	I2S Transmitter Block Enable Register
16.10.7.4 I2S_CER	0x0C	Clock Enable Register
16.10.7.5 I2S_CCR	0x010	Clock Configuration Register
16.10.7.6 I2S_RXFFR	0x014	Receiver Block FIFO Register
16.10.7.7 I2S_TXFFR	0x018	Transmitter Block FIFO Register
16.10.7.8 I2S_LRBR_n_	0x020	Left Receive Buffer Register for Channel 0
16.10.7.9 I2S_LTHR_n_	0x020	Left Transmit Holding Register for Channel 0
16.10.7.10 I2S_RRBR_n_	0x024	Right Receive Buffer Register for Channel 0
16.10.7.11 I2S_RTHR_n_	0x024	Right Transmit Holding Register for Channel 0
16.10.7.12 I2S_RER_n_	0x028	Receive Enable Register for Channel 0
16.10.7.13 I2S_TER_n_	0x02C	Transmit Enable Register for Channel 0
16.10.7.14 I2S_RCR_n_	0x030	Receive Configuration Register for Channel 0
16.10.7.15 I2S_TCR_n_	0x034	Transmit Configuration Register for Channel 0
16.10.7.16 I2S_ISR_n_	0x038	Interrupt Status Register for Channel 0
16.10.7.17 I2S_IMR_n_	0x03C	Interrupt Mask Register for Channel 0
16.10.7.18 I2S_ROR_n_	0x040	Receive Overrun Register for Channel 0
16.10.7.19 I2S_TOR_n_	0x044	Transmit Overrun Register for Channel 0
16.10.7.20 I2S_RFCR_n_	0x048	Receive FIFO Configuration Register for Channel 0
16.10.7.21 I2S_TXFCR_n_	0x04C	Transmit FIFO Configuration Register for Channel 0
16.10.7.22 I2S_RFF_n_	0x050	Receive FIFO Flush Register for Channel 0
16.10.7.23 I2S_TFF_n_	0x054	Transmit FIFO Flush Register for Channel 0
16.10.7.8 I2S_LRBR_n_	0x060	Left Receive Buffer Register for Channel 1
16.10.7.9 I2S_LTHR_n_	0x060	Left Transmit Holding Register for Channel 1
16.10.7.10 I2S_RRBR_n_	0x064	Right Receive Buffer Register for Channel 1
16.10.7.11 I2S_RTHR_n_	0x064	Right Transmit Holding Register for Channel 1
16.10.7.12 I2S_RER_n_	0x068	Receive Enable Register for Channel 1
16.10.7.13 I2S_TER_n_	0x06C	Transmit Enable Register for Channel 1
16.10.7.14 I2S_RCR_n_	0x070	Receive Configuration Register for Channel 1
16.10.7.15 I2S_TCR_n_	0x074	Transmit Configuration Register for Channel 1
16.10.7.16 I2S_ISR_n_	0x078	Interrupt Status Register for Channel 1

Register Name	Offset	Description
16.10.7.17 I2S_IMR_n_	0x07C	Interrupt Mask Register for Channel 1
16.10.7.18 I2S_ROR_n_	0x080	Receive Overrun Register for Channel 1
16.10.7.19 I2S_TOR_n_	0x084	Transmit Overrun Register for Channel 1
16.10.7.20 I2S_RFCR_n_	0x088	Receive FIFO Configuration Register for Channel 1
16.10.7.21 I2S_TXFCR_n_	0x08C	Transmit FIFO Configuration Register for Channel 1
16.10.7.22 I2S_RFF_n_	0x090	Receive FIFO Flush Register for Channel 1
16.10.7.23 I2S_TFF_n_	0x094	Transmit FIFO Flush Register for Channel 1
16.10.7.24 I2S_RXDMA	0x1C0	Receiver Block DMA Register
16.10.7.25 I2S_RRXDMA	0x1C4	Reset Receiver Block DMA Register
16.10.7.26 I2S_TXDMA	0x1C8	Transmitter Block DMA Register
16.10.7.27 I2S_RTXDMA	0x1CC	Reset Transmitter Block DMA Register
16.10.7.28 I2S_COMP_PARAM_2	0x1F0	Component Parameter 2 Register
16.10.7.29 I2S_COMP_PARAM_1	0x1F4	Component Parameter 1 Register
16.10.7.30 I2S_COMP_VERSION_REG	0x1F8	Component Version ID
16.10.7.31 I2S_COMP_TYPE_REG	0x1FC	DesignWare Component Type

16.10.7 Register Description

Legend:

R = Read-only, W = Write-only, R/W = Read/Write, N/A = Reserved

16.10.7.1 I2S_IER

Table 16.186. I2S_IER Description

Bit	Access	Function	Reset Value	Description
31:1	N/A	Reserved	0	Reserved and read as zero.
0	R/W	IEN	0	i2s enable. A disable on this bit overrides any other block or channel enables and flushes all FIFOs. 1: enable i2s 0: disable i2s

16.10.7.2 I2S_IRER

Table 16.187. I2S_IRER Description

Bit	Access	Function	Reset Value	Description
31:1	N/A	Reserved	0	Reserved and read as zero.

Bit	Access	Function	Reset Value	Description
0	R/W	RXEN	0	Receiver block enable. A disable on this bit overrides any individual receive channel enables. 1: enable receiver 0: disable receiver

16.10.7.3 I2S_ITER

Table 16.188. I2S_ITER Description

Bit	Access	Function	Reset Value	Description
31:1	N/A	Reserved	0	Reserved and read as zero.
0	R/W	TXEN	0	Transmitter block enable. A disable on this bit overrides any individual transmit channel enables. 1: enable transmitter 0: disable transmitter

16.10.7.4 I2S_CER

Table 16.189. I2S_CER Description

Bit	Access	Function	Reset Value	Description
31:1	N/A	Reserved	0	Reserved and read as zero.
0	R/W	CLKEN	0	Clock generation enable/disable. This bit enables/disables the clock generation signals when i2s is a Primary : sclk_en, ws_out, and sclk_gate. 1: enable 0: disable

16.10.7.5 I2S_CCR

Table 16.190. I2S_CCR Description

Bit	Access	Function	Reset Value	Description
31:5	N/A	Reserved	0	Reserved and read as zero.
4:3	R/W	WSS	0x01	These bits are used to program the number of sclk cycles for which the word select line (ws_out) stays in the left or right sample mode: 0: 16 clock cycles 1: 24 clock cycles 2: 32 clock cycles The I2S Clock Generation block must be disabled (CER[0] = 0) prior to any changes in this value.

Bit	Access	Function	Reset Value	Description
2:0	R/W	SCLKG	0x04	These bits are used to program the gating of sclk: 0: No clock gating 2: Gate after 16 clock cycles 4: Gate after 24 clock cycles 5: Gate after 32 clock cycles The programmed gating value should be greater than or equal to the largest programmed audio resolution to prevent the truncating of RX/TX data. The I2S Clock Generation block must be disabled (CER[0] = 0) prior to any changes in this value.

16.10.7.6 I2S_RXFFR

Table 16.191. Receiver Block FIFO Reset Register Description

Bit	Access	Function	Reset Value	Description
31:1	N/A	Reserved	0	Reserved and read as zero.
0	W	RXFFR	0	Receiver FIFO Reset. Writing a 1 to this register flushes all the RX FIFOs (this is a self clearing bit). Receiver Block must be disabled prior to writing this bit.

16.10.7.7 I2S_TXFFR

Table 16.192. Transmitter Block FIFO Reset Register Description

Bit	Access	Function	Reset Value	Description
31:1	N/A	Reserved	0	Reserved and read as zero.
0	W	TXFFR	0	Transmitter FIFO Reset. Writing a 1 to this register flushes all the TX FIFOs (this is a self clearing bit). Transmitter Block must be disabled prior to writing this bit.

16.10.7.8 I2S_LRBR_n_

x is the channel number ; x = 0,1

Table 16.193. Left Receive Buffer Register Description

Bit	Access	Function	Reset Value	Description
31:24	N/A	Reserved	0	Reserved and read as zero.
23:0	R	LRBR	0x0	The left stereo data received serially from the receive channel input (sdix) is read through this register. If the RX FIFO is full and the two-stage read operation (for instance, a read from LRBRx followed by a read from RRBRx) is not performed before the start of the next stereo pair, then the new data is lost and an overrun interrupt occurs. (Data already in the RX FIFO is preserved.) NOTE: Before reading this register again, the right stereo data MUST be read from RRBRx, or the status/interrupts will not be valid.

16.10.7.9 I2S_LTHR_n_

x is the channel number ; x = 0,1

Table 16.194. Left Transmit Holding Register Description

Bit	Access	Function	Reset Value	Description
31:24	N/A	Reserved	0	Reserved and read as zero.
23:0	W	LTHR	0x0	The left stereo data to be transmitted serially through the transmit channel output (sdox) is written through this register. Writing is a two-stage process: (1) A write to this register passes the left stereo sample to the transmitter. (2) This MUST be followed by writing the right stereo sample to the RTHRx register. Data should only be written to the FIFO when it is not full. Any attempt to write to a full FIFO results in that data being lost and an overrun interrupt being generated.

16.10.7.10 I2S_RRBR_n_

x is the channel number ; x = 0,1

Table 16.195. Right Receive Buffer Register Description

Bit	Access	Function	Reset Value	Description
31:24	N/A	Reserved	0	Reserved and read as zero.
23:0	R	RRBR	0x0	The right stereo data received serially from the receive channel input (sdix) is read through this register. If the RX FIFO is full and the two-stage read operation (for instance, read from LRBRx followed by a read from RRBRx) is not performed before the start of the next stereo pair, then the new data is lost and an overrun interrupt occurs. (Data already in the RX FIFO is preserved.) NOTE: Prior to reading this register, the left stereo data MUST be read from LRBRx, or the status/interrupts will not be valid.

16.10.7.11 I2S_RTHR_n_

x is the channel number ; x = 0,1

Table 16.196. Right Transmit Holding Register Description

Bit	Access	Function	Reset Value	Description
31:24	N/A	Reserved	0	Reserved and read as zero.
23:0	W	RTHR	0x0	The right stereo data to be transmitted serially through the transmit channel output (sdox) is written through this register. Writing is a two-stage process: (1) A left stereo sample MUST first be written to the LTHRx register. (2) A write to this register passes the right stereo sample to the transmitter. Data should only be written to the FIFO when it is not full. Any attempt to write to a full FIFO results in that data being lost and an overrun interrupt being generated.

16.10.7.12 I2S_RER_n_

x is the channel number ; x = 0,1

Table 16.197. Receive Enable Register Description

Bit	Access	Function	Reset Value	Description
31:1	N/A	Reserved	0	Reserved and read as zero.
0	R/W	RXCHEN	1	Receive channel enable. This bit enables/disables a receive channel, independently of all other channels. On enable, the channel begins receiving on the next left stereo cycle. A global disable of i2s (IER[0] = 0) or the Receiver block (IRER[0] = 0) overrides this value. 1: Enable 0: Disable

16.10.7.13 I2S_TER_n_

x is the channel number ; x = 0,1

Table 16.198. Transmit Enable Register Description

Bit	Access	Function	Reset Value	Description
31:1	N/A	Reserved	0	Reserved and read as zero.
0	R/W	TXCHEN	1	Transmit channel enable. This bit enables/disables a transmit channel, independently of all other channels. On enable, the channel begins transmitting on the next left stereo cycle. A global disable of i2s (IER[0] = 0) or Transmitter block (ITER[0] = 0) overrides this value. 0: Disable 1: Enable

16.10.7.14 I2S_RCR_n_

x is the channel number ; x = 0,1

Table 16.199. Receive Configuration Register Description

Bit	Access	Function	Reset Value	Description
31:3	N/A	Reserved	0	Reserved and read as zero.
2:0	R/W	WLEN	0x4	These bits are used to program the desired data resolution of the receiver and enables the LSB of the incoming left (or right) word to be placed in the LSB of the LRBRx (or RRBRx) register. 100 implies 24 bit resolution Programmed data resolution must be less than or equal to 24 bits. If the selected resolution is greater than the 24 bits, the receive channel defaults back to 24 bits. The channel must be disabled prior to any changes in this value (RERx[0] = 0).

16.10.7.15 I2S_TCR_n_

x is the channel number ; x = 0,1

Table 16.200. Transmit Configuration Register Description

Bit	Access	Function	Reset Value	Description
31:3	N/A	Reserved	0	Reserved and read as zero.
2:0	R/W	WLEN	0x4	These bits are used to program the data resolution of the transmitter and ensures the MSB of the data is transmitted first. 100 implies 24 bit resolution Programmed resolution must be less than or equal to 24 bits. If the selected resolution is greater than 24 bits, the transmit channel defaults back to 24 bits. The channel must be disabled prior to any changes in this value (TERx[0] = 0).

16.10.7.16 I2S_ISR_n_

x is the channel number ; x = 0,1

Table 16.201. Interrupt Status Register Description

Bit	Access	Function	Reset Value	Description
31:6	N/A	Reserved	0	Reserved and read as zero.
5	R	TXFO	0	Status of Data Overrun interrupt for the TX channel. Attempt to write to full TX FIFO. 0: TX FIFO write valid 1: TX FIFO write overrun
4	R	TXFE	1	Status of Transmit Empty Trigger interrupt. TX FIFO is empty. 1: trigger level reached 0: trigger level not reached
3:2	N/A	Reserved		Reserved and read as zero.
1	R	RXFO	0	Status of Data Overrun interrupt for the RX channel. Incoming data lost due to a full RX FIFO. 0: RX FIFO write valid 1: RX FIFO write overrun
0	R	RXDA	0	Status of Receive Data Available interrupt. RX FIFO data available. 1: trigger level reached 0: trigger level not reached

16.10.7.17 I2S_IMR_n_

x is the channel number ; x = 0,1

Table 16.202. Interrupt Mask Register Description

Bit	Access	Function	Reset Value	Description
31:6	N/A	Reserved	0	Reserved and read as zero.
5	R/W	TXFOM	1	Masks TX FIFO Overrun interrupt. 1: masks interrupt 0: unmasks interrupt
4	R/W	TXFEM	1	Masks TX FIFO Empty interrupt. 1: masks interrupt 0: unmasks interrupt
3:2	N/A	Reserved		Reserved and read as zero.
1	R/W	RXFOM	1	Masks RX FIFO Overrun interrupt. 1: masks interrupt 0: unmasks interrupt
0	R/W	RXDAM	1	Masks RX FIFO Data Available interrupt. 1: masks interrupt 0: unmasks interrupt

16.10.7.18 I2S_ROR_n_

x is the channel number ; x = 0,1

Table 16.203. Receive Overrun Register Description

Bit	Access	Function	Reset Value	Description
31:1	N/A	Reserved	0	Reserved and read as zero.
0	R	RXCHO	0	Read this bit to clear the RX FIFO Data Overrun interrupt. 0: RX FIFO write valid 1: RX FIFO write overrun

16.10.7.19 I2S_TOR_n_

x is the channel number ; x = 0,1

Table 16.204. Trasmit Overrun Register Description

Bit	Access	Function	Reset Value	Description
31:1	N/A	Reserved	0	Reserved and read as zero.
0	R	TXCHO	0	Read this bit to clear the TX FIFO Data Overrun interrupt. 0: TX FIFO write valid 1: TX FIFO write overrun

16.10.7.20 I2S_RFCR_n_

x is the channel number ; x = 0,1

Table 16.205. Receive FIFO Configuration Register Description

Bit	Access	Function	Reset Value	Description
31:4	N/A	Reserved	0	Reserved and read as zero.
3:0	R/W	RXCHDT	0x3	These bits program the trigger level in the RX FIFO at which the Received Data Available interrupt is generated. Trigger Level = Programmed Value + 1 . Valid RXCHDT values: 0 to 7 If an illegal value is programmed, these bits saturate to 7. The channel must be disabled prior to any changes in this value (that is, RERx[0] = 0).

16.10.7.21 I2S_TXFCR_n_

x is the channel number ; x = 0,1

Table 16.206. Transmit FIFO Configuration Register Description

Bit	Access	Function	Reset Value	Description
31:4	N/A	Reserved	0	Reserved and read as zero.
3:0	R/W	TXCHET	0x3	Transmit Channel Empty Trigger. These bits program the trigger level in the TX FIFO at which the Empty Threshold Reached Interrupt is generated. Trigger Level = TXCHET. TXCHET values: 0 to 7 If an illegal value is programmed, these bits saturate to 7. The channel must be disabled prior to any changes in this value (that is, TERx[0] = 0).

16.10.7.22 I2S_RFF_n_

x is the channel number ; x = 0,1

Table 16.207. Receive FIFO Flush Register Description

Bit	Access	Function	Reset Value	Description
31:1	N/A	Reserved	0	Reserved and read as zero.
0	W	RXCHFR	0	Receive Channel FIFO Reset. Writing a 1 to this register flushes an individual RX FIFO. (This is a self clearing bit.) RX channel or block must be disabled prior to writing to this bit.

16.10.7.23 I2S_TFF_n_

x is the channel number ; x = 0,1

Table 16.208. Transmit FIFO Flush Register Description

Bit	Access	Function	Reset Value	Description
31:1	N/A	Reserved	0	Reserved and read as zero.
0	W	TXCHFR	0	Transmit Channel FIFO Reset. Writing a 1 to this register flushes channel's TX FIFO. (This is a self clearing bit.) TX channel or block must be disabled prior to writing to this bit.

16.10.7.24 I2S_RXDMA

Table 16.209. Receiver Block DMA Register Description

Bit	Access	Function	Reset Value	Description
31:0	R	RXDMA	0x00	Receiver Block DMA Register. Used to cycle repeatedly through the enabled receive channels (from lowest numbered to highest), reading stereo data pairs.

16.10.7.25 I2S_RRXDMA

Table 16.210. Reset Receiver Block DMA Register Description

Bit	Access	Function	Reset Value	Description
31:1	N/A	Reserved	N/A	Reserved
0	W	RRXDMA	0x0	Reset Receiver Block DMA Register. Writing a 1 to this self-clearing register resets the RXDMA register mid-cycle to point to the lowest enabled Receive channel. Note: Writing to this register has no effect if the component is performing a stereo pair read (such as, when left stereo data has been read but not right stereo data).

16.10.7.26 I2S_TXDMA

Table 16.211. Transmitter Block DMA Register Description

Bit	Access	Function	Reset Value	Description
31:0	W	TXDMA	0x00	Transmitter Block DMA Register. This register can be used to cycle repeatedly through the enabled Transmit channels (from lowest numbered to highest) to allow writing of stereo data pairs.

16.10.7.27 I2S_RTXDMA

Table 16.212. Reset Transmitter Block DMA Register Description

Bit	Access	Function	Reset Value	Description
31:1	N/A	Reserved	N/A	Reserved
0	W	RTXDMA	0x0	Reset Transmitter Block DMA Register. Writing a 1 to this self-clearing register resets the TXDMA register mid-cycle to point to the lowest enabled Transmit channel. Note: This register has no effect in the middle of a stereo pair write (such as, when left stereo data has been written but not right stereo data).

16.10.7.28 I2S_COMP_PARAM_2

Bit	Access	Function	Reset Value	Description
31:6	N/A	Reserved		Reserved
5:3	R	I2S_RX_WORDSIZE_1	0x3	Sets the maximum audio data resolution (word size) of the left and right data for Receive Channel 1. 0x3 = 24 bit resolution
2:0	R	I2S_RX_WORDSIZE_0	0x3	Sets the maximum audio data resolution (word size) of the left and right data for Receive Channel 0. 0x3 = 24 bit resolution

Component Parameter Register 2 Description

16.10.7.29 I2S_COMP_PARAM_1

Component Parameter Register 1 Description

Bit	Access	Function	Reset Value	Description
31:22	N/A	Reserved		Reserved
21:19	R	I2S_TX_WORDSIZE_1	0x3	Sets the maximum audio data resolution (word size) of the left and right data for Transmit Channel 1. 0x3 implies 24 bit resolution
18:16	R	I2S_TX_WORDSIZE_0	0x3	Sets the maximum audio data resolution (word size) of the left and right data for Transmit Channel 0. 0x3 - 24 bit resolution
15:11	N/A	Reserved		Reserved
10:9	R	I2S_TX_CHANNELS	0x2	Controls the number of transmit channels for this i2s component. The range of values is 1 to 2 and is enabled only if parameter I2S_TRANSMITTER_BLOCK is also enabled. 0x2 implies 2 channels are present.
8:7	R	I2S_RX_CHANNELS	0x2	Controls the number of receive channels for this i2s component. The range of values is 1 to 2 and is enabled only if parameter I2S_RECEIVER_BLOCK is also enabled. 0x2 implies 2 channels are present.
6	R	I2S_RECEIVER_BLOCK	0x1	Controls whether the i2s component has I2S receiver block(s) or not. This must be enabled to be able to set the number of RX channels (I2S_RX_CHANNELS).
5	R	I2S_TRANSMITTER_BLOCK	0x1	Controls whether the i2s component has I2S transmitter block(s) or not. This must be enabled to be able to set the number of TX channels (I2S_TX_CHANNELS).
4	R	I2S_MODE_EN	0x1	Determines whether the component acts as the I2S bus Master or Slave. 0x1 = TRUE
3:2	R	I2S_FIFO_DEPTH_GLOBAL	0x2	0x2 implies/ I2S_FIFO_DEPTH_GLOBAL = 8
1:0	R	APB_DATA_WIDTH	0x2	32 bit width

16.10.7.30 I2S_COMP_VERSION_REG

Table 16.213. I2S Component Version Register Description

Bit	Access	Function	Reset Value	Description
31:0	R	I2S_COMP_VERSION	0x3130392a	Specific I2S component version number.

16.10.7.31 I2S_COMP_TYPE_REG

Table 16.214. I2S Component Type Register Description

Bit	Access	Function	Reset Value	Description
31:0	R	I2S_COMP_TYPE	0x445701a0	Specific I2S component Type.This assigned unique hex value is constant.

16.11 MCU Configuration Registers

16.11.1 General Description

This block provides programming support for configuration of MCU blocks. Various features in the chip are enabled using this.

16.11.2 Features

Here is a list of features controlled by this block.

- Supports programming of crystal ON/OFF and clock cleaner controls for 40MHz crystal.
- Provides a scratch pad register for software.
- Supports invalid AHB access detection and provides acknowledgment.
- Provides package select information.
- Provides information related to trap detection and supports trap handling for 64K,128K and NWP memories.
- Provides information about device and version of the chip.
- Supports host SPI interrupt handling.
- Provides option to select required host interfaces.
- Indicates host detection to firmware.
- Provides enable for registering AHB data path between MCU and NWP.
- Provides SDIO secondary status to firmware.
- Supports I2S and PCM interface control information.
- Provides enable for registering data path from MCU ROM and NWP RAM.
- Provides enable for Cortex M4 I-port prefetching from NWP memory.
- Provides Cortex M4 control and status information like sleep and deepsleep.
- Provides programmable option to connect MCU HP peripherals to either UDMA or GPDMA.
- Provides enables for memory low power option.
- Provides registers for inter-processor communication between MCU and NWP.
- Provides registers to wakeup NWP from sleep and indicates active/sleep status of MCU and NWP.
- Provides enables for MCU HP peripheral interrupts connected to NWP.
- Supports programming of SPI Flash Controller DLLs in test mode.

16.11.3 Functional Description

- Provides firmware controls to turn ON/OFF the 40 MHz crystal and the respective clock cleaners by using MCR_XTAL_ON_CTRL register.

  Provides a scratch pad register (MCR_SW_SCRATCHPAD_SET_REG , MCR_SW_SCRATCHPAD_CLEAR_REG) for software to be used as storage space. There are set and clear registers.

  Supports invalid AHB access detection using MCR_AHB_DUMMY_SLAVE_SELECTED_MASTER and MCR_AHB_ERROR_PER_MASTER_STATUS registers.

  Provides package select information on 10th and 12th bits of MCR_RST_LATCH_STATUS register.

  Provides information related to trap detection and handling. For more information on this, refer to "Trap Generation and handling" in Memory Architecture.

  Device and version information of the chip are present in MCR_CHIP_DEVICE_ID and MCR_CHIP_VER_NO registers respectively.

  Provides I2S interface control information on 23rd bit of MCR_GENERIC_CTRL_1_REG, 14th bit of MCR_GENERIC_CTRL_1_REG.
- Provides PCM interface control information using MCR_PCM_CTRL_SET/ MCR_PCM_CTRL_CLEAR registers.
- Supports Host SPI interrupt handling using MCR_HOST_SPI_INTR_MASK, MCR_HOST_SPI_INTR_SET, MCR_HOST_SPI_INTR_CLR registers.
- Provides option to select required host interfaces using MCR_HOST_CTRL_REG and MCR_RST_LATCH_STATUS_REG.
- Firmware can read the host detection status from MCR_RST_LATCH_STATUS register.
- Provides SDIO slave status to firmware via MCR_SDIO_STATE and MCR_SDIO_STATE_CTRL registers.
- Programmable option to connect MCU HP peripherals to either UDMA or GPDMA is present in MCR_PERIPHERAL_UDMA_DMA_SEL register.
- Light sleep for MCU memories and FIFOs can be enabled using MCR_MEM_LS_ENABLE register. Memory RME and RM can be asserted by using MCR_MEM_RM_RME register.
- Inter processor communication from MCU to NWP can be done through MCR_MCU_P2P_INTR_SET, MCR_MCU_P2P_INTR_CLR registers.
- Inter processor communication from NWP to MCU can be done through MCR_NWP_P2P_INTR_MASK_SET, MCR_NWP_P2P_INTR_MASK_CLR, MCR_NWP_P2P_INTR_CLR registers.
- Active/sleep status of MCU and NWP is present inMCR_MCU_P2P_COMM_STATUS_REG .
- For registers supporting handling of MCU HP peripheral interrupts to NWP refer to Table 16.281 MCU HP Peripheral Interrupts to NWP Description on page 525.
- Miscellaneous functions provided by MCR_GENERIC_CTRL register:
  - Enables AHB invalid access trap when 6th bit of the register is set.
  - Enables daisy chaining in JTAG when 10th bit is set.
  - Supports CCI programming on bits 15, 16 and 17.
- Provides enable for registering AHB data path between MCU and NWP using 4th bit of MCR_AHB_BRIDGE_CTRL register.
- Provides enable for registering data path from MCU ROM using 4th bit of MCR_GENERIC_CTRL_1_REG
- Cortex M4 can be kept under soft reset using 0th bit of MCR_CM_CTRL register.
- Sleep and Deepsleep status of MCU can be obtained from MCR_CM_STATUS register.

16.11.4 Register Summary

Base Address: 0x4600_8000

Table 16.215. Register Summary

Register Name	Offset	Description
MCR_HOST_SPI_INTR_MASK_REG	0x00	Host SPI interface interrupt mask register
MCR_HOST_SPI_INTR_SET_REG	0x04	Host SPI interface interrupt set register
MCR_HOST_SPI_INTR_CLR_REG	0x08	Host SPI interface interrupt clear register
MCR_HOST_CTRL_REG	0x0C	Host Control Register
MCR_RST_LATCH_STATUS_REG	0x10	Reset Latch Status Register
MCR_GENERIC_CTRL_REG	0x14	MCU Generic Control Register
MCR_AHB_BRIDGE_CTRL_REG	0x18	AHB bridge Control Register
MCR_SDIO_STATE_CTRL_REG	0x20	SDIO State Control Register
MCR_SDIO_STATE_REG	0x24	SDIO State status Register
MCR_XTAL_ON_CTRL_REG	0x28	Crystal Control Register
MCR_SW_SCRATCHPAD_SET_REG	0x2C	Software Scratch Pad Set Register
MCR_SW_SCRATCHPAD_CLEAR_REG	0x30	Software Scratch Pad Clear Register
MCR_PCM_CTRL_SET_REG	0x34	PCM Interface Control Set Register
MCR_PCM_CTRL_CLEAR_REG	0x38	PCM Interface Control Clear Register
MCR_GENERIC_CTRL_1_REG	0x44	MCU Generic Control Register1
MCR_CM_CTRL_REG	0x48	Cortex M4 Control Register
MCR_CM_STATUS_REG	0x4C	Cortex M4 Status Register
MCR_CHIP_DEVICE_ID_REG	0x50	Chip Device ID Register
MCR_CHIP_VER_NO_REG	0x54	Chip Version ID Register
MCR_PERIPHERAL_UDMA_DMA_SEL_REG	0x58	DMA Peripheral Selection Register
MCR_AHB_DUMMY_SLAVE_SELECTED_MASTER_REG	0x5C	AHB Dummy slave selection register
MCR_AHB_ERROR_PER_MASTER_STATUS_REG	0x60	AHB Master Error status register

Register Name	Offset	Description
MCR_I2S_LOOP_BACK_REG	0x68	MCU I2S Loopback Enable Register
MCR_RESET_TO_CORE_CNT	0x6C	Reset to Core Register
MCR_ENABLE_TRAP	0x70	Enable Trap Register
MCR_SPARE_REG	0x74	M4SS Spare Register
MCR_SOC_ICM_CTRL_REG	0x7C	SOC ICM control Register
MCR_MEM_LS_ENABLE_REG	0x8C	Memory Light Sleep Enable Register
MCR_DM_TRAP_ENABLE_REG_160K	0x90	DM Trap Enable Register for memory set 160K
MCR_DMA_WR_TRAP_ENABLE_REG_160K	0x94	DMA Write Trap Enable Register for memory set 160K
MCR_DMA_RD_TRAP_ENABLE_REG_160K	0x98	DMA Read Trap Enable Register for memory set 160K
MCR_AHB_MASTER_TRAP_ENABLE_REG_160K	0x9C	AHB Master Trap Enable Register for memory set 160K
MCR_AHB_MASTER_TRAP_ENABLE_REG_64K0	0xA0	AHB Master Trap Enable Register for memory set 64K0
MCR_ASYNC_TRAP_DETECTED_160K	0xA4	Async Trap Detected Register for memory set 160K
MCR_DM_TRAP_STATUS_160K	0xA8	DM Trap Status Register for memory set 160K
MCR_DM_TRAP_STATUS_64K0	0xB4	DM Trap Status Register for memory set 64K0
MCR_DMA0_TRAP_STATUS_160K	0xB8	DMA0 Trap Status Register for memory set 160K
MCR_DMA0_TRAP_STATUS_64K0	0xC4	DMA0 Trap Status Register for memory set 64K0
MCR_DMA1_TRAP_STATUS_160K	0xC8	DMA1 Trap Status Register for memory set 160K
MCR_DMA1_TRAP_STATUS_64K0	0xD4	DMA1 Trap Status Register for memory set 64K0
MCR_DMA2_TRAP_STATUS_160K	0xD8	DMA2 Trap Status Register for memory set 160K
MCR_MVP_PSRAM_TRAP_STATUS	0xDC	MVP PSRAM Trap Status register
MCR_MVP_PSRAM_TRAP_CLEAR	0xE0	MVP PSRAM Trap Clear register
MCR_DMA2_TRAP_STATUS_64K0	0xE4	DMA2 Trap Status Register for memory set 64K0
MCR_DMA_DEVICE_SEL_REG	0x108	DMA Device Select Register

Register Name	Offset	Description
MCR_DM_TRAP_ENABLE_REG_64K0	0x118	DM Trap Enable Register for memory set 64K0
MCR_DMA_RD_TRAP_ENABLE_REG_64K0	0x124	DMA Read Trap Enable Register for memory set 64K0
MCR_DMA_WR_TRAP_ENABLE_REG_64K0	0x130	DMA Write Trap Enable Register for memory set 64K0
MCR_ASYNC_TRAP_STATUS_160K	0x13C	Async Trap Status Register for memory set 160K
MCR_ASYNC_TRAP_STATUS_64K0	0x148	Async Trap Status Register for memory set 64K0
MCR_ASYNC_TRAP_CLEAR_160K	0x150	Async Trap Clear Register for memory set 160K
MCR_ASYNC_TRAP_CLEAR_64K0	0x15C	Async Trap Clear Register for memory set 64K0
MCR_ASYNC_TRAP_DETECTED_64K0	0x164	Async Trap Detected Register for memory set 64K0
MCR_MCU_P2P_INTR_SET_REG	0x16C	MCU to NWP P2P Interrupt Set Register
MCR_MCU_P2P_INTR_CLR_REG	0x170	MCU to NWP P2P Interrupt Clear Register
MCR_MCU_P2P_COMM_STATUS_REG	0x174	MCU to NWP P2P Communication Status Register
MCR_NWP_P2P_INTR_MASK_SET_REG	0x178	NWP to MCU P2P Interrupt Mask Register
MCR_NWP_P2P_INTR_MASK_CLR_REG	0x17C	NWP to MCU P2P Interrupt Unmask Register
MCR_NWP_P2P_INTR_CLR_REG	0x180	NWP to MCU P2P Interrupt Clear Register
MCR_PERI_INTR_MASK_SET_TH0_REG	0x184	Mask Register for MCU HP Peripheral Interrupts going to NWP on Thread 0
MCR_PERI_INTR_MASK_CLR_TH0_REG	0x188	Unmask Register for MCU HP Peripheral Interrupts going to NWP on Thread 0
MCR_PERI_INTR_MASK_SET_TH1_REG	0x18C	Mask Register for MCU HP Peripheral Interrupts going to NWP on Thread 1
MCR_PERI_INTR_MASK_CLR_TH1_REG	0x190	Unmask Register for MCU HP Peripheral Interrupts going to NWP on Thread 1

Register Name	Offset	Description
MCR_PERI_INTR_MASK_SET_TH2_REG	0x194	Mask Register for MCU HP Peripheral Interrupts going to NWP on Thread 2
MCR_PERI_INTR_MASK_CLR_TH2_REG	0x198	Unmask Register for MCU HP Peripheral Interrupts going to NWP on Thread 2
MCR_PERI_INTR_MASK_SET_TH3_REG	0x19C	Mask Register for MCU HP Peripheral Interrupts going to NWP on Thread 3
MCR_PERI_INTR_MASK_CLR_TH3_REG	0x1A0	Unmask Register for MCU HP Peripheral Interrupts going to NWP on Thread 3
MCR_PERI_INTR_STS_TH0_REG	0x1A4	Status Register for MCU HP Peripheral Interrupts going to NWP on Thread 0
MCR_PERI_INTR_STS_TH1_REG	0x1A8	Status Register for MCU HP Peripheral Interrupts going to NWP on Thread 1
MCR_PERI_INTR_STS_TH2_REG	0x1AC	Status Register for MCU HP Peripheral Interrupts going to NWP on Thread 2
MCR_PERI_INTR_STS_TH3_REG	0x1B0	Status Register for MCU HP Peripheral Interrupts going to NWP on Thread 3
MCR_MEM_RM_RME_REG	0x1B8	Memory RM and RME Control Register
MCR_ULP_AHB_BRIDGE_CLK_ENABLE_REG	0x1FC	ULP AHB-AHB bridge static clock enable register
MCR_AHB_MASTER_TRAP_ENABLE_64K1	0x1C8	AHB Master Trap Enable Register for memory set 64K1
MCR_DM_TRAP_STATUS_64K1	0x1D0	DM Trap Status Register for memory set 64K1
MCR_DMA0_TRAP_STATUS_64K1	0x1D4	DMA0 Trap Status Register for memory set 64K1
MCR_DMA1_TRAP_STATUS_64K1	0x1D8	DMA1 Trap Status Register for memory set 64K1
MCR_DMA2_TRAP_STATUS_64K1	0x1DC	DMA2 Trap Status Register for memory set 64K1
MCR_DM_TRAP_ENABLE_REG_64K1	0x1E0	DM Trap Enable Register for memory set 64K1

Register Name	Offset	Description
MCR_DMA_RD_TRAP_ENABLE_REG_64K1	0x1E4	DMA Read Trap Enable Register for memory set 64K1
MCR_DMA_WR_TRAP_ENABLE_REG_64K1	0X1E8	DMA Write Trap Enable Register for memory set 64K1
MCR_ASYNC_TRAP_STATUS_64K1	0X1EC	Async Trap Status Register for memory set 64K1
MCR_ASYNC_TRAP_CLEAR_64K1	0x1F0	Async Trap Clear Register for memory set 64K1
MCR_ASYNC_TRAP_DETECTED_64K1	0x1F4	Async Trap Detected Register for memory set 64K1
MCR_DCACHE_CTRL_AND_STATUS_REG	0x1F8	Dcache Control and Status Register

16.11.5 Register Description

Legend:

R = Read-only, W = Write-only, R/W = Read/Write, NA = Reserved

16.11.5.1 MCR_HOST_SPI_INTR_MASK_REG

Table 16.216. MCR_HOST_SPI_INTR_MASK_REG Description

Bit	Access	Function	Reset Value	Description
31:10	NA	NA	0x0	Reserved and read as zero.
9	R/W	HOST_SPI_INTR_ACTIVE_LOW_MODE	0x0	Writing '1' to this bit configures the host SPI interrupt in active low mode. By default, it will be active high.
8	R/W	HOST_SPI_INTR_OPEN_DRAIN_MODE	0x1	Writing '1' to this bit configures the host SPI interrupt in open drain mode. When open drain mode is enabled and interrupt is configured in active high mode, external PULL DOWN has to be used on the board.
7:0	R/W	HOST_SPI_INTR_MSK	0x0	Writing '1' in any bit masks the corresponding interrupt in HOST_SPI_INTR_STATUS

16.11.5.2 MCR_HOST_SPI_INTR_SET_REG

Table 16.217. MCR_HOST_SPI_INTR_SET_REG Description

Bit	Access	Function	Reset Value	Description
31:8	NA	NA	0x0	Reserved and read as zero
7:0	R/W	HOST_SPI_INTR_STATUS	0x0	Writing '1' to any bit raises an interrupt to SPI host. Writing '1' at the corresponding bit position in 16.11.5.3 MCR_HOST_SPI_INTR_CLR_REG clears the interrupt. Performing read gives HOST_SPI_INTR_STATUS always. Note: Interrupt will be raised only if the corresponding interrupt is unmasked in the 16.11.5.1 MCR_HOST_SPI_INTR_MASK_REG register.

16.11.5.3 MCR_HOST_SPI_INTR_CLR_REG

Table 16.218. MCR_HOST_SPI_INTR_CLR_REG Description

Bit	Access	Function	Reset Value	Description
31:8	NA	NA	0x0	Reserved and read as zero
7:0	R/WC	HOST_SPI_INTR_CLEAR	0x0	Writing '1' to this bit clears the 16.11.5.2 MCR_HOST_SPI_INTR_SET_REG. This register gets cleared in the next clock cycle. Performing read gives HOST_SPI_INTR_STATUS always

16.11.5.4 MCR_HOST_CTRL_REG

16.6.5.1 MCR_HOST_CTRL_REG

16.11.5.5 MCR_RST_LATCH_STATUS_REG

Table 16.219. MCR_RST_LATCH_STATUS_REG Description

Bit	Access	Function	Reset Value	Description
31:16	NA	NA	0x0	Reserved and read as zero
15:10	R	Reserved	0x0	
9	R	Reserved	0x0	Reserved
8	R	ulp_wakeup	0x0	This bit differentiates between normal power up and ULP wakeup state '1' – ULP based wakeup '0' – Not ULP based wakeup (first power up)
7	R	mcu_first_powerup_por	0x0	This bit indicates MCU first power up status
6	R	ram_retention_status	0x0	This bit indicates whether RAM is retained or not. '0' – Ram not retained '1' – Ram retained
5	-	Reserved	-	
4	R	spi_sel	0x0	This bit indicates the SPI secondary host select '0' – SPI is not selected '1' – SPI is selected
3	R	sdio_sel	0x1	This bit indicates the SDIO secondary host select '0' – SDIO is not selected '1' – SDIO is selected
2:1	R	boot_mode	0x3	Reserved
0	R	Boot_mode_en	0x0	This bit is used to indicate if boot mode is enabled.

16.11.5.6 MCR_GENERIC_CTRL_REG

Table 16.220. MCR_GENERIC_CTRL_REG Description

Bit	Access	Function	Reset Value	Description
31:15	R	Reserved	0x0	Reserved
14	R/W	jtag_daisy_chain_en	0x0	This bit has to be set to enable daisy chaining in JTAG
13:10	R/W	Reserved	0x0	Reserved
9	R/W	hspi_ssi_sel	0x0	When this bit is set, it indicates that the host SPI interface pins are intended to be connected to the SSI slave and not to Host SPI. Host SPI chip select is made inactive in this case. '1' – Host SPI pins used for SSI slave. Host SPI inactive. '0' – Host SPI active.
8	R/W	i2s_ssi_gpio_mode_sel	0x1	When set, GPIO 11,12,13 & 14 are used for SSI in GPIO mode 2. '1' – GPIO 11,12,13,14 are used for SSI when configured in GPIO mode 2. '0' - GPIO 11,12,13,14 are used for I2S when configured in GPIO mode 2.
7	R/W	Reserved	0x0	Reserved
6	R/W	AHB invalid access trap enable	0x0	When this bit is set, trap will be generated to processor in the case an invalid access is done on AHB.
5:0	R/W	Reserved	0x0	Reserved

16.11.5.7 MCR_AHB_BRIDGE_CTRL_REG

Table 16.221. MCR_AHB_BRIDGE_CTRL_REG Description

Bit	Access	Function	Reset Value	Description
31:5	R	Reserved	0x0	Reserved
4	R/W	Bypass_registering_for_AHB_bridge	0x0	When this bit is set, bypass the AHB bus registering in AHB bridge, which is present in between MCU and NWP subsystems. It should be asserted when MCU clock is less than 100MHz.
3:0	R/W	Reserved	0x0	Reserved

16.11.5.8 MCR_SDIO_STATE_CTRL_REG

Table 16.222. MCR_SDIO_STATE_CTRL_REG Description

Bit	Access	Function	Reset Value	Description
31:2	NA	Reserved	0x0	Reserved
1	W	Load_sdio_state	0x0	When this bit is set, hardware loads the value in MCR_SDIO_STATE register to SDIO block. The state to be restored has to be loaded to MCR_SDIO_STATE register before setting this. This is done by firmware after coming out of sleep. It is used for ULP Mode State Retention.
0	W	latch_sdio_state	0x0	When this bit is set, hardware latches the SDIO state into MCR_SDIO_STATE register. This is done by firmware while going to sleep. It is used for ULP Mode State Retention.

16.11.5.9 MCR_SDIO_STATE_REG

Table 16.223. MCR_SDIO_STATE_REG Description

Bit	Access	Function	Reset Value	Description
31:16	R/W	sdio_state_upper	0x0	Upper 16-bits of the SDIO state that is to be loaded into the SDIO block. Upon reading, it represents the upper 16-bits of the state read from SDIO block.
15:0	R/W	sdio_state_lower	0x0	Lower 16-bits of the SDIO state that is to be loaded into the SDIO block. Upon reading, it represents the lower 16-bits of the state read from SDIO block.

16.11.5.10 MCR_XTAL_ON_CTRL_REG

Table 16.224. MCR_XTAL_ON_CTRL_REG Description

Bit	Access	Function	Reset Value	Description
31:4	R	Reserved	0x009	Reserved
3	R/W	40Mhz_xtal_on_fw	0x0	This bit drives the 40MHz crystal ON indication. This bit is considered only if the firmware based driving is enabled using BIT(0) of this register. '1' – crystal ON is set '0' – crystal ON is reset
2:1	R/W	Reserved	0x1	Reserved
0	R/W	40Mhz_xtal_on_fw_sel	0x0	This bit determines the source of 40MHz CRYSTAL ON indication. '1' –CRYSTAL ON is controlled by firmware through BIT(3) of this register. '0' – CRYSTAL ON is controlled by the sleep state machine and the XTAL_ON_IN coming to the chip. It is asserted whenever there is an indication from either of the clients.

16.11.5.11 MCR_SW_SCRATCHPAD_SET_REG

Table 16.225. MCR_SW_SCRATCHPAD_SET_REG Description

Bit	Access	Function	Reset Value	Description
31:0	R/W	Software_scratchpad_set	0x0	This register is used by software for storing information. It does not affect anything in the hardware. Writing a '1' to any of the bits sets the corresponding bit in the soft register. Writing '0' has no effect on this register.

16.11.5.12 MCR_SW_SCRATCHPAD_CLEAR_REG

Table 16.226. MCR_SW_SCRATCHPAD_CLEAR_REG Description

Bit	Access	Function	Reset Value	Description
31:0	R/W	Software_scratchpad_clear	0x0	This register is used by software for storing information. It does not affect anything in the hardware. Writing a '1' to any of the bits clears the corresponding bit in the soft register. Writing '0' has no effect on this register.

16.11.5.13 MCR_PCM_CTRL_SET_REG

Table 16.227. MCR_PCM_CTRL_SET_REG Description

Bit	Access	Function	Reset Value	Description
31:5	R/W	Reserved	0x0	Reserved
4:2	R/W	pcm_bit_res	0x0	The bit-resolution of the data on PCM. 3'b000 - 8-bit 3'b001 - 12-bit 3'b010 - 16-bit, 3'b011 - 24-bit 3'b1xx - 32-bit Writing a '1' to any of the bits sets the corresponding bit in the soft register. Writing '0' has no effect on this register.
1	R/W	pcm_fsync_start_m	0x0	This bit has to be programmed according to when the MS bit of the PCM data is driven w.r.t. the fsync signal of PCM. '1' - The MS bit of data is driven in the same clock cycle as fsync going high. '0' - The MS bit of data is driven one clock cycle after fsync goes high. Writing a '1' to any of the bits sets the corresponding bit in the soft register. Writing '0' has no effect on this register.
0	R/W	pcm_enable_m	0x0	Enable/disable PCM mode of I2S interface. When PCM is enabled, I2S is disabled and vice versa '1' - PCM mode is enabled and I2S mode is disabled. This programming is valid only when the GPIO signals are programmed for I2S mode. '0' - PCM mode is disabled and I2S mode is enabled. This programming is in addition to the other GPIO level programming to enable I2S mode. Writing a '1' to any of the bits sets the corresponding bit in the soft register. Writing '0' has no effect on this register.

16.11.5.14 MCR_PCM_CTRL_CLEAR_REG

Table 16.228. MCR_PCM_CTRL_CLEAR_REG Description

Bit	Access	Function	Reset Value	Description
31:5	R/W	Reserved	0x0	Reserved
4:2	R/W	pcm_bit_res	0x0	Setting the bits clear the pcm_bit_res. Writing '0' has no effect on this register.
1	R/W	pcm_fsync_start_m	0x0	Setting the bit 1 clear the pcm_fsync_start. Writing '0' has no effect on this register.
0	R/W	pcm_enable_m	0x0	Setting the bit clears the pcm_enable_m. Writing '0' has no effect on this register.

16.11.5.15 MCR_GENERIC_CTRL_1_REG

Table 16.229. MCR_GENERIC_CTRL_1_REG Description

Bit	Access	Function	Reset Value	Description
31:24	R/W	Reserved	0x0	Reserved
23	R	i2s_master_slave_mode	0x0	0 – I2S/I2S PCM act as slave 1 -- I2S/I2S PCM act as master
22	R	Reserved	0x0	Reserved
21	R/W	enable_icache_seq_access_ps_mode	0x0	When this bit is set, power save is enabled and a icache read that is sequential to the previous cache read of the same line is saved in local buffer and accessed
20	R/W	icache_dram_power_save_mode	0x0	When this bit is set, only half performance is valid with 1x clock. (Data two cycles after clock). Full performance is valid with 2x clock to icache dram
19	R/W	Reserved	0x0	Reserved
18:14	R/W	host_pads_gpio_mode	0x0	Control bit for 5 pins to use it either as host pin or gpio pin. Pins from 30 to 26 are controlled by these bits respectively. 0 – HOST mode 1 – GPIO mode
13:8	R	Reserved	0x0	Reserved
7	R	Provide_soc_clk_2x_to_icache_dram	0x0	When set, twice the frequency of soc clk will be provided to icache dram,. When zero, normal soc_clk is given.
6:5	R	Reserved	0x0	Reserved
4	R/W	register_rom_output	0x0	When set, the ready and read data from ROM will be registered. It should be asserted when MCU clock is greater than 100MHz.
3:0	R/W	Reserved	0x0	Reserved

16.11.5.16 MCR_CM_CTRL_REG

Table 16.230. MCR_CM_CTRL_REG Description

Bit	Access	Function	Reset Value	Description
31:2	R/W	Reserved	0x0	Reserved
1	R	cm_reset_por	0x0	Status of cm_reset_por can be seen on this bit.
0	R/W	cm_reset	0x0	When 1'b1 is written in this bit, both por and non regions of M4 will be under reset. When 1'b0 is written in this bit, por and non regions of M4 will be out of reset. For JTAG and host resets, cm_reset_por will not get reset and only cm_reset gets reset.

16.11.5.17 MCR_CM_STATUS_REG

Table 16.231. MCR_CM_STATUS_REG Description

Bit	Access	Function	Reset Value	Description
31:3	R	Reserved	0x0	Reserved
2	R	SLEEPING	0x0	When high, indicates the Cortex M4 is in sleeping state
1	R	SLEEPDEEP	0x0	When high, indicates the Cortex M4 is in deep sleep state
0	R	cm_lockup	0x0	When high, indicates the Cortex M4 is in LOCKUP state

16.11.5.18 MCR_CHIP_DEVICE_ID_REG

Table 16.232. MCR_CHIP_DEVICE_ID_REG Description

Bit	Access	Function	Reset Value	Description
31:16	R	Reserved	0x0	Reserved
15:0	R	device_id	0x917	It gives the device ID of the chip. Device ID is 917.

16.11.5.19 MCR_CHIP_VER_NO_REG

Table 16.233. MCR_CHIP_VER_NO_REG Description

Bit	Access	Function	Reset Value	Description
31:8	R	Reserved	0x0	Reserved
7:0	R	ver_no	0x1	Indicates the revision number of the chip.

16.11.5.20 MCR_PERIPHERAL_UDMA_DMA_SEL_REG

Table 16.234. MCR_PERIPHERAL_UDMA_DMA_SEL_REG Description

Bit	Access	Function	Reset Value	Description
31:8	R/W	Reserved	0x0	Reserved
7	R/W	I2C	0x0	When set, ack from udma will go to the I2C, else ack from GPDMA will go.
6	R/W	Reserved	0x0	Reserved
5	R/W	SSI master	0x0	When set, ack from udma will go to the SSI master, else ack from GPDMA will go.
4	R/W	SSI slave	0x0	When set, ack from udma will go to the SSI, else ack from GPDMA will go.
3:2	R/W	Reserved	0x0	Reserved
1	R/W	UART1	0x0	When set, ack from udma will go to the uart1, else ack from GPDMA will go.
0	R/W	USART0	0x0	When set, ack from udma will go to the uart0/usart0, else ack from GPDMA will go.

16.11.5.21 MCR_AHB_DUMMY_SLAVE_SELECTED_MASTER_REG

Table 16.235. MCR_AHB_DUMMY_SLAVE_SELECTED_MASTER_REG Description

Bit	Access	Function	Reset Value	Description
31:16	R/W	Reserved	0x0	Reserved
15	R/W	NWP AHB-AHB master	0x0	Hardware set this bit, When the NWP AHB-AHB bridge master is trying to access the wrong slave address. Firmware reset this bit by writing zero in this register.
14	-	Reserved	-	
13	R/W	GPDMA M2	0x0	Hardware set this bit, When the GPDMA AHB master2 is trying to access the wrong slave address. Firmware reset this bit by writing zero in this register.
12	R/W	Reserved	0x0	Reserved
11	-	Reserved	-	
10	R/W	NWP AHB-AHB master	0x0	Hardware set this bit, when NWP AHB is trying to access the wrong slave address. Firmware reset this bit by writing zero in this register.
9	R/W	RPDMA 2	0x0	Hardware set this bit, when RPDMA M2 is trying to access the wrong slave address. Firmware reset this bit by writing zero in this register.
8	R/W	ULP AHB bridge	0x0	Hardware set this bit, when ULP AHB is trying to access the wrong slave address. Firmware reset this bit by writing zero in this register.
7	-	Reserved	-	
6	R/W	UDMA	0x0	Hardware set this bit, When the uDMA AHB master is trying to access the wrong slave address. Firmware reset this bit by writing zero in this register.
5	R/W	Icache	0x0	Hardware set this bit, When the MCU Icache AHB master is trying to access the wrong slave address. Firmware reset this bit by writing zero in this register.
4	R/W	M4 S port	0x0	Hardware set this bit, When the Cortex M4 S-port AHB master is trying to access the wrong slave address. Firmware reset this bit by writing zero in this register.
3	R/W	M4 D port	0x0	Hardware set this bit, When the Cortex M4 D-port AHB master is trying to access the wrong slave address. Firmware reset this bit by writing zero in this register.
2	R/W	M4 I port	0x0	Hardware set this bit, When the Cortex M4 I-port AHB master is trying to access the wrong slave address. Firmware reset this bit by writing zero in this register.
1	R/W	HIF	0x0	Hardware set this bit, When the HIF AHB master is trying to access the wrong slave address. Firmware reset this bit by writing zero in this register.
0	R/W	GPDMA M1	0x0	Hardware set this bit, When the GPDMA AHB master1 is trying to access the wrong slave address. Firmware reset this bit by writing zero in this register.

16.11.5.22 MCR_AHB_ERROR_PER_MASTER_STATUS_REG

Table 16.236. MCR_AHB_ERROR_PER_MASTER_STATUS_REG Description

Bit	Access	Function	Reset Value	Description
31:16	R/W	Reserved	0x0	Reserved
15	R/W	NWP AHB-AHB master	0x0	Hardware set this bit, When the NWP AHB-AHB bridge master is getting error response. Firmware reset this bit by writing zero in this register.
14	-	Reserved	-	
13	R/W	GPDMA M2	0x0	Hardware set this bit, When the GPDMA AHB master2 is getting error response. Firmware reset this bit by writing zero in this register.
12	R/W	Reserved	0x0	Reserved
11	-	Reserved	-	
10	R/W	ULP AHB bridge	0x0	Hardware set this bit, When the ULP AHB-AHB bridge master is getting error response. Firmware reset this bit by writing zero in this register.
9	-	Reserved	-	
8	R/W	Reserved	0x0	Reserved
7	-	Reserved	-	
6	R/W	UDMA	0x0	Hardware set this bit, When the uDMA AHB master is getting error response. Firmware reset this bit by writing zero in this register.
5	R/W	Icache	0x0	Hardware set this bit, When the MCU Icache AHB master is getting error response. Firmware reset this bit by writing zero in this register.
4	R/W	M4 S port	0x0	Hardware set this bit, When the Cortex M4 S-port AHB master is getting error response. Firmware reset this bit by writing zero in this register.
3	R/W	M4 D port	0x0	Hardware set this bit, When the Cortex M4 D-port AHB master is getting error response. Firmware reset this bit by writing zero in this register.
2	R/W	M4 I port	0x0	Hardware set this bit, When the Cortex M4 I-port AHB master is getting error response. Firmware reset this bit by writing zero in this register.
1	R/W	HIF	0x0	Hardware set this bit, When the HIF AHB master is getting error response. Firmware reset this bit by writing zero in this register.
0	R/W	GPDMA M1	0x0	Hardware set this bit, When the GPDMA AHB master1 is getting error response. Firmware reset this bit by writing zero in this register.

16.11.5.23 MCR_I2S_LOOP_BACK_REG

Table 16.237. MCR_I2S_LOOP_BACK_REG Description

Bit	Access	Function	Reset Value	Description
31:15	R/W	Reserved	0x0	Reserved
14	R/W	i2s_loop_back_mode	0x0	Enables MCU I2S loop back mode
13:8	R/W	Reserved	0x0	Reserved
7:0	R/W	Reserved	0x0	Reserved

16.11.5.24 MCR_RESET_TO_CORE_CNT

Table 16.238. MCR_RESET_TO_CORE_CNT Description

Bit	Access	Function	Default Value	Description
31:9	R	Reserved		
8	R/W	reset_to_core_sel	0x0	1: host/debugger reset will be synchronized to sleep clock and used 0: host/debugger reset will be synchronized to soc clock and used
7:0	R/W	reset_to_core_cnt	8'd1	This fields hold the reset active duration in number of clocks.

16.11.5.25 MCR_ENABLE_TRAP

Table 16.239. MCR_ENABLE_TRAP Description

Bit	Access	Function	Default Value	Description
15:10	R/W	Reserved	0x0	Reserved
9	R/W	ahb error trap enable	0x0	When set, ahb error trap is enabled
8	R/W	apb_dummy_slave_selected	0x0	When set, any of the master is trying to access the wrong peripheral address.
7:6	R/W	Reserved	0x0	
5:2	R/W	Reserved	0xF	
1	R/W	trap_enable_register_cortex	0x0	Enable the trap for cortex. • Trap will not be raised to cortex • Trap will be raised to cortex
0	R	AHB_DUMMY slave selected	0x0	AHB dummy slave is selected in cortex

16.11.5.26 MCR_SPARE_REG

Table 16.240. M4SS_SPARE_REG Description

Bit	Access	Function	Default Value	Description
31:16	R	Reserved	0x0	
15:0	R/W	Reserved	16'h0	M4SS spare register

16.11.5.27 MCR_SOC_ICM_CTRL_REG

Table 16.241. MCR_SOC_ICM_CTRL_REG Description

Bit	Access	Function	Default Value	Description
31:5	R	Reserved	0x0	

Bit	Access	Function	Default Value	Description
12	R	remap	0x0	Internal logic signal for (remap_valid && remap[0]) This bit is going to AHB ICM input
11:5	R/W	Reserved	0x0	Reserved
4	R/W	remap_ valid	0x0	remap valid bit If remap feature is required, along with remap[0], this bit also has to set.
3:1	R/W	Reserved	0x0	Reserved
0	R/W	remap[0]	0x0	When remap[0] == 1, the address space for rom will be reduced from 0x003F_FC00 to 0x003F_FFFF and qspi address space will add become 0x0800_0000 to 0x0BFF_FFFF and 0x0030_0000 to 0x003F_FBFF

16.11.5.28 MCR_MEM_LS_ENABLE_REG

Table 16.242. MCR_MEM_LS_ENABLE_REG Description

Bit	Access	Function	Reset Value	Description
31:4	R	Reserved	0x0	Reserved
3	R/W	gen_spi_master_mem_lightsleep_enable	0x0	Light sleep enable signal for GSPI master FIFOs. This bit has to be set to make GSPI FIFO memories enter into light sleep. Prior to this, Bit(0) of this register has to be set to enable light sleep for entire MCU.
2:1	R/W	Reserved	0x0	Reserved
0	R/W	MCU_mem_lightsleep_enable	0x0	Light sleep enable signal for all MCU memories.This bit must be set to enable Light sleep mode for any memory in MCU.

16.11.5.29 MCR_DM_TRAP_ENABLE_REG_160K

Table 16.243. MCR_DM_TRAP_ENABLE_REG_160K Description

Bit	Access	Function	Default Value	Description
31:12	NA	Reserved	0x0	
11:0	R/W	Enable bits per bank of memory set 1	0x0	LSB corresponds to 0th bank in the particular set. A '1' in any bit position indicates that if through that port that bank access , a trap will be generated

16.11.5.30 MCR_DMA_WR_TRAP_ENABLE_REG_160K

Table 16.244. MCR_DMA_WR_TRAP_ENABLE_REG_160K Description

Bit	Access	Function	Default Value	Description
31: 12	NA	Reserved	0x0	
11:0	R/W	Enable bits per bank of memory set 1	0x0	LSB corresponds to oth bank in the particular set. A '1' in any bit position indicates that if through that port that bank access , a trap will be generated

16.11.5.31 MCR_DMA_RD_TRAP_ENABLE_REG_160K

Table 16.245. MCR_DMA_RD_TRAP_ENABLE_REG_160K Description

Bit	Access	Function	Default Value	Description
31: 12	NA	Reserved	0x0	
11:0	R/W	Enable bits per bank of memory set 1	0x0	LSB corresponds to oth bank in the particular set. A '1' in any bit position indicates that if through that port that bank access , a trap will be generated

16.11.5.32 MCR_AHB_MASTER_TRAP_ENABLE_REG_160K

Table 16.246. MCR_AHB_MASTER_TRAP_ENABLE_REG_160K Description

Bit	Access	Function	Default Value	Description
31:16	R	Reserved		
15:0	R/W	Trap enable bits per master for memory set 160K	0x0	'1' on a particular bit position indicates that access by that master to a trap enable bank will generate trap otherwise if it is '0' it won't generate a trap if it is accessing a trap enable bank. The masking is for UM1 transactions indicated in DMA_WR_TRAP_ENABLE_REG_160K , DMA_WR_TRAP_ENABLE_REG_160K .

16.11.5.33 MCR_AHB_MASTER_TRAP_ENABLE_REG_64K0

Table 16.247. MCR_AHB_MASTER_TRAP_ENABLE_REG_64K0 Description

Bit	Access	Function	Default Value	Description
31:16	R/W	Trap enable bits per master for memory set 64K0	0x0	'1' on a particular bit position indicates that access by that master to a trap enable bank will generate trap otherwise if it is '0' it won't generate a trap if it is accessing a trap enable bank. The masking is for UM1 transactions indicated in DMA_WR_TRAP_ENABLE_REG_64K0, DMA_WR_TRAP_ENABLE_REG_64K0
15:0	R	Reserved		

16.11.5.34 MCR_ASYNC_TRAP_DETECTED_160K /64K0/64K1

Offset Address: 0xA4/0x164/0x1F4

Table 16.248. MCR_ASYNC_TRAP_DETECTED_160K /64K0/64K1 Description

Bit	Access	Function	Default Value	Description
15:1	R/W	Reserved	0x0	Reserved
0	R/W	Async_trap_detected_cortex	0x0	This will give you the indication that trap has been detected by cortex. Firmware has to write this bit as '1' so that the asyn_trap can be cleared.

16.11.5.35 MCR_DM_TRAP_STATUS_160K /64K0

Offset Address: 0xA8+(n-1)*4

Table 16.249. MCR_DM_TRAP_STATUS_160K /64K0 Description

Bit	Access	Function	Default Value	Description
24:6	R	Address	0x0	The address for which the read or write request came on dm port.
5:2	R	master_number	0x0	The number of the master which requested the transaction.
1	R	Write/read	0x0	1=write request 0=read request
0	R	Gnt	0x0	The grant signal for pm port of unified memory.

16.11.5.36 MCR_DMA0_TRAP_STATUS_160K /64K0

Offset Address: 0xB8 + (n-1)*4

Table 16.250. MCR_DMA0_TRAP_STATUS_160K /64K0 Description

Bit	Access	Function	Default Value	Description
24:6	R	Address	0x0	The address for which the read or write request came on dma0 port.
5:2	R	master_number	0x0	The number of the master which requested the transaction.
1	R	Write/read	0x0	1 = write request 0 = read request
0	R	Gnt	0x0	The grant signal for dma0 port of unified memory.

16.11.5.37 MCR_DMA1_TRAP_STATUS_160K /64K0

Offset Address: 0xC8 + (n-1)*4

Table 16.251. MCR_DMA1_TRAP_STATUS_160K /64K0 Description

Bit	Access	Function	Default Value	Description
24:6	R	Address	0x0	The address for which the read or write request came on dma1 port.
5:2	R	master_number	0x0	The number of the master which requested the transaction.
1	R	Write/read	0x0	1=write request 0=read request
0	R	Gnt	0x0	The grant signal for dma1 port of unified memory.

16.11.5.38 MCR_DMA2_TRAP_STATUS_160K /64K0

Offset Address: 0xD8 + (n-1)*4

Table 16.252. MCR_DMA2_TRAP_STATUS_160K /64K0 Description

Bit	Access	Function	Default Value	Description
24:6	R	Address	0x0	The address for which the read or write request came on dma2 port.
5:2	R	master_number	0x0	The number of the master which requested the transaction.
1	R	Write/read	0x0	1=write request 0=read request
0	R	Gnt	0x0	The grant signal for dma2 port of unified memory.

16.11.5.39 MCR_MVP_PSRAM_TRAP_STATUS

Table 16.253. MCR_MVP_PSRAM_TRAP_STATUS Description

Bit	Access	Function	Default Value	Description
31:2	R	Reserved	0x0	The address for which the read or write request came on dma2 port.
1	R	Status	0x0	Tells the status of unauthorized access to PSRAM
0	R	Status	0x0	Tells the status of unauthorized access to MVP

16.11.5.40 MCR_MVP_PSRAM_TRAP_CLEAR

Table 16.254. MCR_MVP_PSRAM_TRAP_CLEAR Description

Bit	Access	Function	Default Value	Description
31:2	R	Reserved	0x0	The address for which the read or write request came on dma2 port.
1	R/W	Read/Write	0x0	Clears the interrupt for PSRAM trap
0	R/W	Read/Write	0x0	Clears the interrupt for MVP trap

16.11.5.41 MCR_DMA_DEVICE_SEL_REG

Table 16.255. MCR_DMA_DEVICE_SEL_REG Description

Bit	Access	Function	Default Value	Description
31:7	R/W	Reserved	0x0	Reserved
6	R/W	qspi_crc_dma_sel	0x0	To select between qspi and crc slave 2 flow control signals, 0 – CRC 1 – QSPI(q2)
5	R/W	ct0_mvp_dma_sel	0x0	To select between sct and mvp flow control signals 0 – ct 1 – mvp
4	R/W	ssi_slv2_dma_sel (Not used)	0x0	To select ssi slave 2 flow control signals 0 – SSI Slave 2 1 – Reserved
3	R/W	uart1_i2s_dma_sel (Not used)	0x0	To select between I2S and UART1 flow control signals 0 – UART 1 1 – I2S
2	R/W	sdio_multi_fn_ssi_dma_sel (Not used)	0x0	To select between SSI master and sdio flow control signals 0 – sdio multifunctional 1 – SSI master
1	R/W	i2s_rf_spi_dma_sel (Not used)	0x0	To select between i2s and rf spi dma flow control signals
0	R/W	Reserved	0x0	Reserved

16.11.5.42 MCR_DM_TRAP_ENABLE_REG_64K0

Table 16.256. MCR_DM_TRAP_ENABLE_REG_64K0 Description

Bit	Access	Function	Default Value	Description
31: 4	NA	Reserved	0x0	
3:0	R/W	Enable bits per bank of memory set 4	0x0	LSB corresponds to 0th bank in the particular set. A '1' in any bit position indicates that if through that port that bank access , a trap will be generated

16.11.5.43 MCR_DMA_WR_TRAP_ENABLE_REG_64K0/64K1

Offset Address: 0x124/0x1E8

Table 16.257. MCR_DMA_WR_TRAP_ENABLE_REG_64K0/64K1 Description

Bit	Access	Function	Default Value	Description
31: 8	NA	Reserved	0x0	
7:0	R/W	Enable bits per bank of memory set 4	0x0	LSB corresponds to 0th bank in the particular set. A '1' in any bit position indicates that if through that port that bank access , a trap will be generated

16.11.5.44 MCR_DMA_RD_TRAP_ENABLE_REG_64K0/64K1

Offset Address: 0x130/0x1E4

Table 16.258. MCR_DMA_RD_TRAP_ENABLE_REG_64K0/64K1 Description

Bit	Access	Function	Default Value	Description
31: 8	NA	Reserved	0x0	
7:0	R/W	Enable bits per bank of memory set 4	0x0	LSB corresponds to 0th bank in the particular set. A '1' in any bit position indicates that if through that port that bank access , a trap will be generated

16.11.5.45 MCR_ASYNC_TRAP_STATUS_160K

Table 16.259. MCR_ASYNC_TRAP_STATUS_160K Description

Bit	Access	Function	Default Value	Description
31:5	R	Reserved	0x0	
3	R	AHB write trap	0x0	Async write trap is detected from other AHB ports
2	R	AHB Read trap	0x0	Async read is detected from other AHB ports
1	R	DM trap	0x0	Async trap is detected from cortex IM port
0	R	Reserved	0x0	

16.11.5.46 MCR_ASYNC_TRAP_STATUS_64K0/64K1

Offset Address: 0x148/0x1EC

Table 16.260. MCR_ASYNC_TRAP_STATUS_64K0/64K1 Description

Bit	Access	Function	Default Value	Description
31:5	R	Reserved	0x0	
3	R	AHB write trap	0x0	Async write trap is detected from other AHB ports
2	R	AHB Read trap	0x0	Async read is detected from other AHB ports
1	R	DM trap	0x0	Async trap is detected from cortex IM port
0	R	Reserved	0x0	

16.11.5.47 MCR_ASYNC_TRAP_CLEAR_160K

Table 16.261. MCR_ASYNC_TRAP_CLEAR_160K Description

Bit	Access	Function	Default Value	Description
31:5	R	Reserved	0x0	
3	R	AHB write trap	0x0	when set to 1, Async write trap is cleared from other AHB ports
2	R	AHB Read trap	0x0	when set to 1, Async read is cleared from other AHB ports
1	R	DM trap	0x0	when set to 1, Async trap is cleared from cortex IM port
0	R	Reserved	0x0	

16.11.5.48 MCR_ASYNC_TRAP_CLEAR_64K0/64K1

Offset Address: 0x15C/0x1F0

Table 16.262. MCR_ASYNC_TRAP_CLEAR_64K0/64K1 Description

Bit	Access	Function	Default Value	Description
31:5	R	Reserved	0x0	
3	R	AHB write trap	0x0	when set to 1, Async write trap is cleared from other AHB ports
2	R	AHB Read trap	0x0	when set to 1, Async read is cleared from other AHB ports
1	R	DM trap	0x0	when set to 1, Async trap is cleared from cortex IM port
0	R	Reserved	0x0	

16.11.5.49 MCR_MCU_P2P_INTR_SET_REG

Table 16.263. MCR_MCU_P2P_INTR_SET_REG Description

Bit	Access	Function	Reset Value	Description
31:16	R	Reserved	0x0	Reserved for future use.
15:0	R/W	MCU_P2P_INTR_SET	0x0	There are 16 interrupts for P2P(processor to processor) communication from MCU to NWP. Each bit is used to raise the interrupt to NWP For write operation, '1'- Raises the Interrupt '0'- Writing a zero into this has no effect. For read operation, '1' – Interrupt is raised '0' – Not raised

16.11.5.50 MCR_MCU_P2P_INTR_CLR_REG

Table 16.264. MCR_MCU_P2P_INTR_CLR_REG Description

Bit	Access	Function	Reset Value	Description
31:16	R	Reserved	0x0	Reserved for future use.
15:0	R/W	MCU_P2P_INTR_CLR	0x0	There are 16 interrupts for P2P(processor to processor) communication from MCU to NWP. Each bit is used to clear the interrupt to NWP For write operation, '1'- Clears the Interrupt '0'- Writing a zero into this has no effect. For read operation, '1' – Interrupt is raised '0' – Not raised

16.11.5.51 MCR_MCU_P2P_COMM_STATUS_REG

Table 16.265. MCR_MCU_P2P_COMM_STATUS_REG Description

Bit	Access	Function	Reset Value	Description
31:4	R	Reserved	0x0	Reserved.
3	R	NWP active status	0x0	This bit is used to indicate MCU that NWP is active '1'- NWP is active '0'- NWP is sleeping
2	R	NWP wakeup MCU	0x0	This bit is used to indicate MCU that it should wakeup from sleep '1'- NWP wakes up MCU '0'- No operation
1	R/W	MCU active status	0x0	This bit is used to indicate NWP that MCU is active '1'- MCU is active '0'- MCU is sleeping
0	R/W	MCU wakeup NWP	0x0	This bit is used to wakeup NWP from sleep. '1'- MCU wakes up NWP '0'- No operation

16.11.5.52 MCR_NWP_P2P_INTR_MASK_SET_REG

Table 16.266. MCR_NWP_P2P_INTR_MASK_SET_REG Description

Bit	Access	Function	Reset Value	Description
31:16	R	Reserved	0x0	Reserved for future use.
15:0	R/W	NWP_P2P_INTR_MASK	0xFFFF	There are 16 interrupts from NWP to MCU for P2P communication. Each bit is used to mask the NWP P2P interrupt For write operation, '1'- masks the NWP P2P Interrupt '0'- Writing a zero into this has no effect. For read operation, '1' – Interrupt is masked '0' – Not raised

16.11.5.53 MCR_NWP_P2P_INTR_MASK_CLR_REG

Table 16.267. MCR_NWP_P2P_INTR_MASK_CLR_REG Description

Bit	Access	Function	Reset Value	Description
31:16	R	Reserved	0x0	Reserved for future use.
15:0	R/W	NWP_P2P_INTR_UNMASK	0xFFFF	There are 16 interrupts from NWP to MCU for P2P communication. Each bit is used to unmask the NWP P2P interrupt For write operation, '1'- unmask the Interrupt '0'- Writing a zero into this has no effect. For read operation, '1' – Interrupt is masked '0' – Not raised

16.11.5.54 MCR_NWP_P2P_INTR_CLR_REG

Table 16.268. MCR_NWP_P2P_INTR_CLR_REG Description

Bit	Access	Function	Reset Value	Description
31:16	R	Reserved	0x0	Reserved for future use.
15:0	R/W	NWP_P2P_INTR_CLR	0x0	There are 16 interrupts from NWP to MCU for P2P communication. Each bit is used to clear the interrupt For write operation, '1'- Clears the Interrupt for MCU instantly '0'- Writing a zero into this has no effect. For read operation, '1' – Interrupt is raised '0' – Not raised

16.11.5.55 MCR_PERI_INTR_MASK_SET_TH0/1/2/3

Offset Address: (0x184 + (0x8 * n)) ; n = 0,1,2,3; n indicates the thread number

Table 16.269. MCR_PERI_INTR_MASK_TH0/1/2/3 Description

Bit	Access	Function	Reset Value	Description
31:29	R	Reserved	0x0	Reserved
28:0	R/W	MCU_PERI_MSK_SET	0x1FFF_FFFF	There are 29 MCU HP peripheral interrupts connected to NWP. These interrupts are shown in Table 16.281 MCU HP Peripheral Interrupts to NWP Description on page 525 . Each bit is used to mask the respective MCU HP peripheral interrupt. For write operation, '1'- Mask Interrupt '0'- Writing a zero into this has no effect. For read operation, '1' – Interrupt masked '0' – Not masked

16.11.5.56 MCR_PERI_INTR_MASK_CLR_TH0/1/2/3

Offset Address: (0x188 + (0x8 * n)) ; n = 0,1,2,3; n indicates the thread number

Table 16.270. MCR_PERI_INTR_MASK_CLR_TH0/1/2/3 Description

Bit	Access	Function	Reset Value	Description
31:29	R	Reserved	0x0	Reserved.
28:0	R/W	MCU_PERI_MSK_CLR	0x1FFF_FFFF	There are 29 MCU HP peripheral interrupts connected to NWP. These interrupts are shown in Table 16.281 MCU HP Peripheral Interrupts to NWP Description on page 525. Each bit is used to unmask the respective MCU HP peripheral interrupt. For write operation, '1'- Unmask Interrupt '0'- Writing a zero into this has no effect. For read operation, '1' – Interrupt masked '0' – Not masked

16.11.5.57 MCR_PERI_INTR_STS_TH0/1/2/3

Offset Address: (0x1A4 + (0x8 * n)) ; n = 0,1,2,3; n indicates the thread number

Table 16.271. MCR_PERI_INTR_STS_TH0/1/2/3 Description

Bit	Access	Function	Reset Value	Description
31:29	R	Reserved	0x0	Reserved for future use.
28:0	R	MCU_PERI_INTR_STATUS	0x0	There are 29 MCU HP peripheral interrupts connected to NWP. These interrupts are shown in Table 16.281 MCU HP Peripheral Interrupts to NWP Description on page 525 . If bit m of (28:0) is 1, then the mth MCU HP peripheral interrupt is not masked and is been raised in thread n(0/1/2/3).

16.11.5.58 MCR_MEM_RM_RME_REG

Table 16.272. MCR_MEM_RM_RME_REG Description

Bit	Access	Function	Reset Value	Description
31:19	R	Reserved	0x0	Reserved
18:16	R/W	MCU_fifo_rm_rme	0x2	MCU_fifo_rm_rme[2:1] bits are used as RM ports for fifo memories which are internal to peripherals. MCU_fifo_rm_rme[0] bit is used as RM enable (RME) for fifo memories which are internal to peripherals.
15:3	R	Reserved	0x0	Reserved
2:0	R/W	MCU_ram_rom_rm_rme	0x2	MCU_ram_rom_rm_rme[2:1] bits are used as RM ports for SRAM memories. MCU_ram_rom_rm_rme[0] bit is used as RM enable (RME) for SRAM memories.

16.11.5.59 MCR_ULP_AHB_BRIDGE_CLK_ENABLE_REG

Table 16.273. MCR_ULP_AHB_BRIDGE_CLK_ENABLE_REG Description

Bit	Access	Function	Reset Value	Description
31:1	R	Reserved	0x0	Reserved
0	R/W	ULP_AHB_bridge_clk_enable	0x1	Used to enable static clock gating ULP AHB-AHB bridge. Only 32-bit write is allowed into this register.

16.11.5.60 MCR_AHB_MASTER_TRAP_ENABLE_64K1

Table 16.274. MCR_AHB_MASTER_TRAP_ENABLE_64K1 Description

Bit	Access	Function	Default Value	Description
31:16	R/W	Trap enable bits per master for memory set 64K1	0x0	'1' on a particular bit position indicates that access by that master to a trap enable bank will generate trap otherwise if it is '0' it won't generate a trap if it is accessing a trap enable bank. The masking is for UM1 transactions indicated in DMA_WR_TRAP_ENABLE_REG_64K1, DMA_WR_TRAP_ENABLE_REG_64K1
15:0	R	Reserved		

16.11.5.61 MCR_DM_TRAP_STATUS_64K1

Table 16.275. MCR_DM_TRAP_STATUS_64K1 Description

Bit	Access	Function	Default Value	Description
24:6	R	Address	0x0	The address for which the read or write request came on dm port.
5:2	R	master_number	0x0	The number of the primary which requested the transaction.
1	R	Write/read	0x0	1 = write request 0 = read request
0	R	Gnt	0x0	The grant signal for pm port of unified memory.

16.11.5.62 MCR_DMA0_TRAP_STATUS_64K1

Table 16.276. MCR_DMA0_TRAP_STATUS_64K1 Description

Bit	Access	Function	Default Value	Description
24:6	R	Address	0x0	The address for which the read or write request came on dma0 port.
5:2	R	master_number	0x0	The number of the master which requested the transaction.
1	R	Write/read	0x0	1=write request 0=read request
0	R	Gnt	0x0	The grant signal for dma0 port of unified memory.

16.11.5.63 MCR_DMA1_TRAP_STATUS_64K1

Table 16.277. MCR_DMA1_TRAP_STATUS_64K1 Description

Bit	Access	Function	Default Value	Description
24:6	R	Address	0x0	The address for which the read or write request came on dma1 port.
5:2	R	master_number	0x0	The number of the master which requested the transaction.
1	R	Write/read	0x0	1=write request 0=read request

Bit	Access	Function	Default Value	Description
0	R	Gnt	0x0	The grant signal for dma1 port of unified memory.

16.11.5.64 MCR_DMA2_TRAP_STATUS_64K1

Table 16.278. MCR_DMA2_TRAP_STATUS_64K1 Description

Bit	Access	Function	Default Value	Description
24:6	R	Address	0x0	The address for which the read or write request came on dma2 port.
5:2	R	master_number	0x0	The number of the master which requested the transaction.
1	R	Write/read	0x0	1=write request 0=read request
0	R	Gnt	0x0	The grant signal for dma2 port of unified memory.

16.11.5.65 MCR_DM_TRAP_ENABLE_REG_64K1

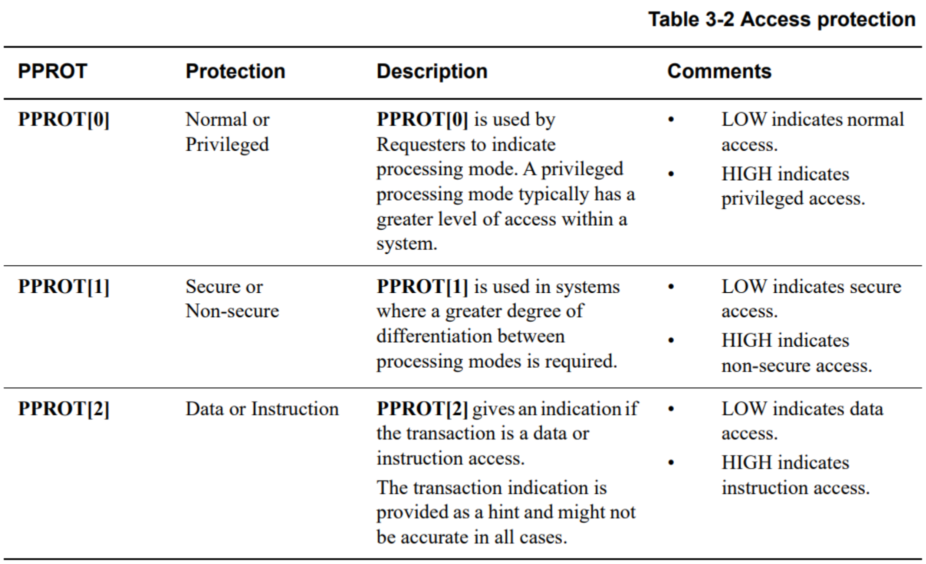
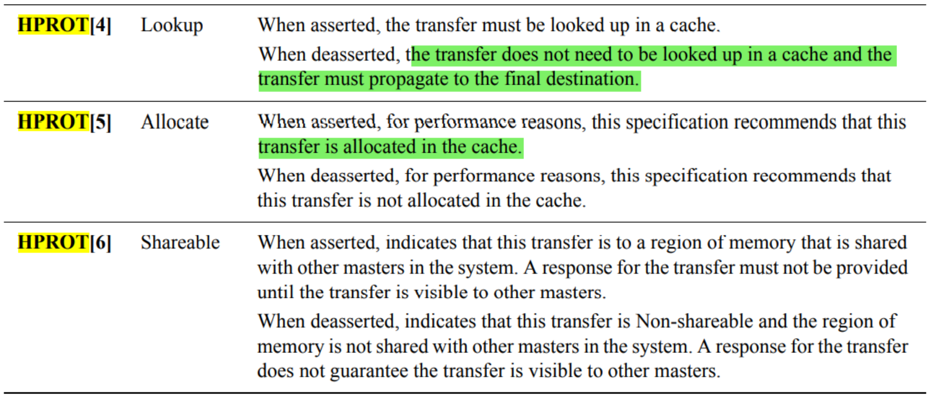
Table 16.279. MCR_DM_TRAP_ENABLE_REG_64K1 Description

Bit	Access	Function	Default Value	Description
31: 4	NA	Reserved	0x0	
3:0	R/W	Enable bits per bank of memory set 4	0x0	LSB corresponds to 0th bank in the particular set. A '1' in any bit position indicates that if through that port that bank access , a trap will be generated

16.11.5.66 MCR_DCACHE_CTRL_AND_STATUS_REG

Table 16.280. MCR_DCACHE_CTRL_AND_STATUS_REG Description

Bit	Access	Function	Default Value	Description
31	R	Reserved	2'd0	Reserved
30	R/W	mvp_qreqn	1'b1	This input is for the MVP q channel interface. (MVP is not using qchannel active low signal)
29:26	R/W	hprot_old	4'b1000	This signal represents the ahb3 hprot description
25	R/W	dis_cache_dis_maint	1'b0	This signal turns off cache disable maintenance.
24	R/W	dis_cache_en_maint	1'b0	This signal turns off cache enable maintenance.
23	R/W	dis_pwr_down_maint	1'b0	This signal turns off powerdown maintenance.
22	R/W	power_on_enable	1'b0	This signal enables the cache automatically after powerup.
21	R/W	apb_violation_resp	1'b0	If the apb_violation_resp is set to HIGH, the AHB Cache responds with errors to failed APB accesses by asserting pslverr

<table>
<tr><th>Bit</th><th>Access</th><th>Function</th><th>Default Value</th><th>Description</th></tr>
<tr><td>20</td><td>R/W</td><td>pmsnapshotreq</td><td>1'b0</td><td>A trigger signal which initiates the capture of the current value of the statistics counters. Must be a synchronous pulse.</td></tr>
<tr><td>19</td><td>R</td><td>pwr_maintenance</td><td>1'b0</td><td>The cache indicates it has finished preparing for powerdown by deasserting the pwr_maintenance signal.
The pwr_maintenance output status port is asserted while the powerdown maintenance is ongoing</td></tr>
<tr><td>18</td><td>R/W</td><td>pwakeup</td><td>1'b1</td><td>Wake up signal. This signal is used to indicate that there is ongoing activity that is associated with the APB interface</td></tr>
<tr><td>17</td><td>R/W</td><td>pclken</td><td>1'b1</td><td>The clock enable signal. This signal allows the APB to run on a divided frequency.</td></tr>
<tr><td>16:14</td><td>R/W</td><td>pprot[2:0]</td><td>3'b001</td><td>
<b>Table 3-2 Access protection</b>
<table>
<tr><th>PPROT</th><th>Protection</th><th>Description</th><th>Comments</th></tr>
<tr><td><b>PPROT[0]</b></td><td>Normal or Privileged</td><td><b>PPROT[0]</b> is used by Requesters to indicate processing mode. A privileged processing mode typically has a greater level of access within a system.</td><td>• LOW indicates normal access.
• HIGH indicates privileged access.</td></tr>
<tr><td><b>PPROT[1]</b></td><td>Secure or Non-secure</td><td><b>PPROT[1]</b> is used in systems where a greater degree of differentiation between processing modes is required.</td><td>• LOW indicates secure access.
• HIGH indicates non-secure access.</td></tr>
<tr><td><b>PPROT[2]</b></td><td>Data or Instruction</td><td><b>PPROT[2]</b> gives an indication if the transaction is a data or instruction access.
The transaction indication is provided as a hint and might not be accurate in all cases.</td><td>• LOW indicates data access.
• HIGH indicates instruction access.</td></tr>
</table></td></tr>
<tr><td>13</td><td>R</td><td>pslverr</td><td>1'b0</td><td>This signal indicates a transfer failure. APB peripherals are not required to support the PSLVERR pin. This is true for both existing and new APB peripheral designs. Where a peripheral does not include this pin then the appropriate input to the APB bridge is tied LOW.</td></tr>
<tr><td>12</td><td>R</td><td>hnonsec_m</td><td>1'b0</td><td>Non-secure transfer indicator
This signal is asserted for a Non-secure transfer and deasserted for a Secure transfer</td></tr>
<tr><td>11:9</td><td>R/W</td><td>hprot[6:4]</td><td>3'b111</td><td>Since Dcache is AHB 5 compatible, The remaining signals are included in this register as configurable.

<table>
<tr><td><b>HPROT[4]</b></td><td>Lookup</td><td>When asserted, the transfer must be looked up in a cache.
When deasserted, the transfer does not need to be looked up in a cache and the transfer must propagate to the final destination.</td></tr>
<tr><td><b>HPROT[5]</b></td><td>Allocate</td><td>When asserted, for performance reasons, this specification recommends that this transfer is allocated in the cache.
When deasserted, for performance reasons, this specification recommends that this transfer is not allocated in the cache.</td></tr>
<tr><td><b>HPROT[6]</b></td><td>Shareable</td><td>When asserted, indicates that this transfer is to a region of memory that is shared with other masters in the system. A response for the transfer must not be provided until the transfer is visible to other masters.
When deasserted, indicates that this transfer is Non-shareable and the region of memory is not shared with other masters in the system. A response for the transfer does not guarantee the transfer is visible to other masters.</td></tr>
</table>
Remaining hprot[3:0] are the same as the previous version of AHB</td></tr>
</table>

Bit	Access	Function	Default Value	Description
8	R/W	hnonsec_s	1'b0	Non-secure transfer indicator This signal is asserted for a Non-secure transfer and deasserted for a Secure transfer
7	R	clk_qactive	1'b1	This signal, when HIGH, indicates to the controller that the AHB Cache requires the clock. When the signal is driven LOW, the AHB Cache might accept a quiescence request.
6	R	clk_qdeny	1'b0	When HIGH, this signal indicates that the AHB Cache denies the quiescence request from the clock controller.
5	R	clk_qacceptn	1'b1	When LOW, this signal indicates that the AHB Cache accepts the quiescence request from the clock controller.
4	R/W	clk_qreqn	1'b1	Active-LOW quiescence request signal driven by the clock controller.
3	R	pwr_qactive	1'b1	This signal, when HIGH, indicates to the controller that the AHB Cache needs power. When the signal is driven LOW, the AHB Cache might accept a quiescence request
2	R	pwr_qdeny	1'b0	When HIGH, this signal indicates that the AHB Cache denies the quiescence request from the power controller
1	R	pwr_qacceptn	1'b1	When LOW, this signal indicates that the AHB Cache accepts the quiescence request from the power controller
0	R/W	pwr_qreqn	1'b1	Active-LOW quiescence request signal driven by the power controller.

16.11.6 MCU HP Peripheral Interrupt Handling

16.11.6.1 MCU HP Peripheral Interrupts to NWP

Table 16.281. MCU HP Peripheral Interrupts to NWP Description

Bit	Peripheral
0	Reserved
1	MCU GPDMA Interrupts
2	Reserved
3	MCU UDMA Interrupts
4	MCU Configurable Timers Interrupts
5	Reserved
6	USART 1 Interrupt
7	USART 2 Interrupt
8	Reserved
9	I2C Interrupt
10	SSI Slave Interrupt
11	Reserved
12	GSPI Master 1 Interrupt
13	SSI Master Interrupt
14	MCPWM Interrupt
15	Quadrature Encoder Interrupt
16	GPIO Group Interrupt 0
17	GPIO Group Interrupt 1
18	GPIO Pin Interrupts
19	SPI Flash Controller Interrupt
22	I2S master Interrupt
23	Reserved
26	PLL Clock Indication Interrupt
27	Reserved

Interrupts 0-4 are multi-channel interrupts. They give single interrupt to NWP based on MCU multi-channel interrupt selection registers in **Interrupts**.

MCU HP Peripheral Interrupts to NWP Description

16.11.6.2 Programming Sequence

- All interrupts are masked by default on 4 threads.
- To unmask any interrupt for thread 'n', set the corresponding bit in the Section 16.11.5.56 MCR_PERI_INTR_MASK_CLR_TH0/1/2/3 register.
- To mask the interrupts for thread 'n', set the corresponding bit in the Section 16.11.5.55 MCR_PERI_INTR_MASK_SET_TH0/1/2/3 register.
- To mask multi-channel interrupts(0-4), use MCU multi-channel interrupt selection registers in Section 8. Interrupts.
- After clearing the interrupt at source of MCU HP peripheral, a dummy read has to be made to the source from the ISR to make sure that the interrupt is cleared.
- Masked status of all the interrupts are available in Section 16.11.5.57 MCR_PERI_INTR_STS_TH0/1/2/3 for each thread.

16.11.7 Programming Sequence for P2P Interrupt Handling

- Raising interrupt from Cortex M4 to NWP
  - One of the 16 P2P Interrupts can be raised from Cortex M4 by setting the corresponding interrupt bit in MCR_MCU_P2P_INTR_SET_REG.
  - P2P Interrupts can be cleared from Cortex M4 by setting the corresponding interrupt bit in MCR_MCU_P2P_INTR_CLR_REG.
- Clearing interrupt raised from NWP to Cortex M4
  - One of the 16 P2P Interrupts can be masked and cleared from M4 by setting the corresponding interrupt bit in MCR_NWP_P2P_INTR_MASK_SET_REG and MCR_NWP_P2P_INTR_CLR_REG in MCU.
  - Then, unmask the respective interrupt bit in MCR_NWP_P2P_INTR_MASK_CLR_REG.
  - P2P Interrupts can be cleared from M4 by setting the corresponding interrupt bit in NWP_P2P_INTR_CLR_REG(0x4105_0090) in NWP.
  - The interrupt is cleared in two steps to reduce interrupt latency

16.12 Motor Control PWM

16.12.1 General Description

The Motor Control PWM (MCPWM) controller is used to generate a periodic pulse waveform, which is useful in motor control and power control applications. The MCPWM controller acts as a timer to count up to a period count value. The time period and the duty cycle of the pulses are both programmable. It is present as part of the MCU HP peripherals.

16.12.2 Features

- Supports up to eight PWM outputs with four duty cycle generators
- Complementary and Independent output modes are supported
- Dead time insertion in Complementary mode
- Manual override option for PWM output pins. Output pin polarity is programmable
- Supports generation of interrupt for different events
- Supports two hardware fault input pins
- Special event trigger for synchronizing analog-to-digital conversions

16.12.3 Functional Description

The block diagram of the MCPWM controller is shown in figure 1. CSR, 16-bit Time base counter, PWM period generator, dead time generator and PWM override logic are shown in this diagram. This is capable of generating up to four output PWM signals, with the same period but different duty cycles. The Duty cycle generator is followed by the dead time generator. The outputs from this module go through the PWM Override Logic and finally to the output pins. The output pins are grouped in pairs, to facilitate driving the low side and high side of a power half-bridge. The safety fault feature is directly connected to the output stages in order to have the smallest possible response time. A/D Conversion trigger block generates event trigger for synchronizing A/D conversion with time base counter.

The divided clock is input to the time base period counter. This counter output depending on the time period match, zero match depending on time base mode selection. If a match occurs, a period match signal is generated. The counter direction is controlled by counter direction control bit. If the direction bit value is zero, timer is counting upward, and if it is one, timer is counting downward. The time base postscaler is useful when the PWM duty cycles need not be updated every PWM cycle. The interrupt control logic decides when to generate a PWM interrupt, depending on the postscale value and operation mode. The interrupt generation for each of the operating modes is described below:

Free running mode: An interrupt event is generated when the time base counter is reset to zero due to a match with the time base period register. The postscaler selection bits can be used in free running mode to reduce the frequency of the interrupt events.

Single event mode: An interrupt event is generated when the time base counter is reset to zero due to a match with the time base period register. The PWM time base counter enable bit is also cleared to inhibit further time base counter increments. The postscaler selection bits have no effect in single event mode

Up/Down counting mode: An interrupt event is generated each time the value of the time base counter is equal to zero and the time base counter begins to count upward. The postscale selection can be used to reduce the frequency of interrupt events in up/down counting mode.

Up/down counting mode with double update of duty cycle: An interrupt event is generated each time the time base counter is equal to zero and each time a period match occurs. The postscale selection has no effect in up/down counting mode with double update of duty cycle. This mode allows the control loop bandwidth to be doubled because the PWM duty cycles can be updated twice per period. Every rising and falling edge of the PWM signal can be controlled using the double update mode.

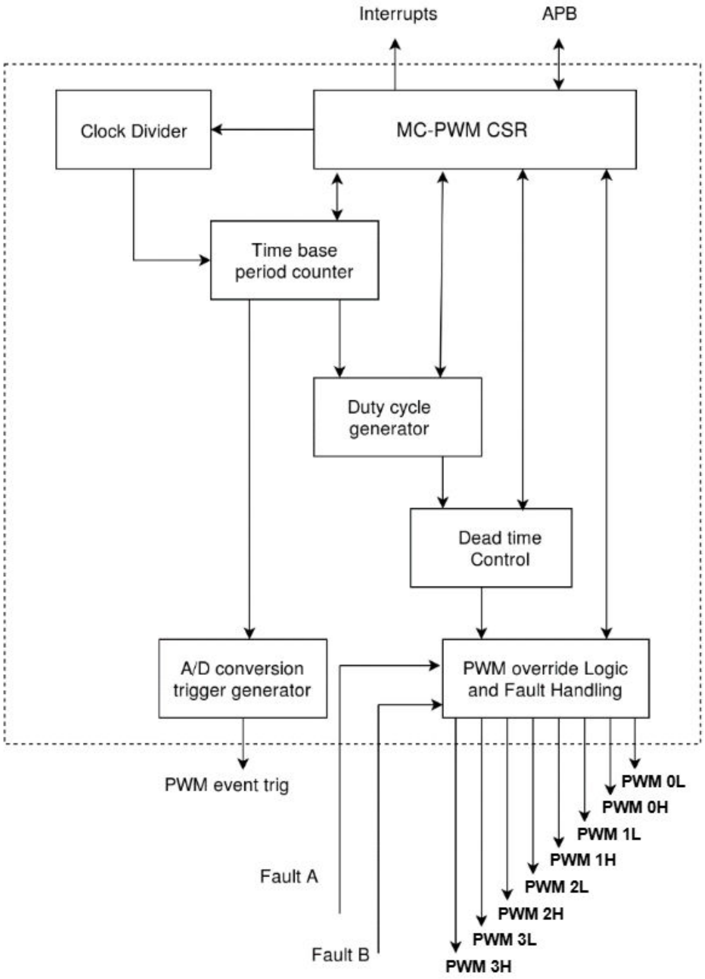


Figure 16.32. Motor Control PWM Block Diagram

The PWM duty cycle generator has four duty cycle generators for eight PWM outputs. It generates duty cycle by comparing duty cycle register value with time base period counter. There are multiple modes for generating duty cycle. They are single event mode, Tail Edge-aligned mode, Lead Edge-aligned mode, Center-aligned mode and Center-aligned mode with double updates. Each complementary output pair from the PWM duty cycle generator has a dead time counter that is used to insert the dead time on both rising and falling edges. Deadtime applies only to PWM output pair , which are in complementary mode. This dead time insertion is shown in the figure below.

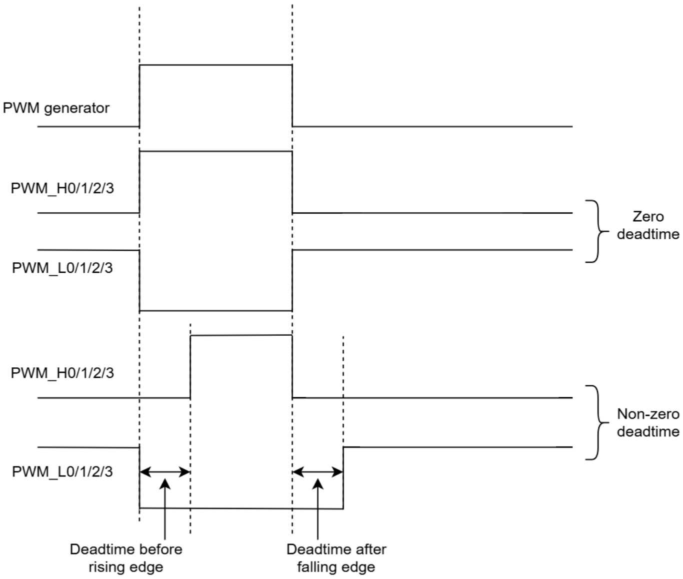


Figure 16.33. Dead Time Insertion Diagram

There is a overdrive control bit for each PWM output pin. If this bit is cleared, the pin will output a PWM signal after dead time insertion. If this bit is set, the pin will be controlled manually. There is a manual out bit for each PWM pin that sets the state of the pin when the output is manually controlled. There are two fault pins, FLT_A and FLT_B, associated with the PWM. When it is asserted, these pins can optionally drive each of the PWM I/O pins to a defined state. Each fault pin is readable using its corresponding status register. Each fault pin has its own interrupt vector, interrupt flag bit, interrupt ack bit and interrupt mask bits. The function of the FLT_A pin is controlled by the fault_A configuration register, and the function of the FLT_B pin is controlled by the fault_B configuration register. The PWM module has a special trigger that allows the A/D converter to be synchronized to the PWM time base. This trigger will start the conversion phase of the A/D block, in order to minimize the delay between getting the A/D conversion result and updating the duty cycle value. The Special Event Compare Register is loaded with a number which is compared to the content of the time base timer, while this one is counting up or down. The counting direction is set by the Special Event Direction bit. As soon as there is a match between the two registers, the trigger is generated and the A/D conversion is started.

16.12.4 Programing Sequence

1. Initializing and configuring MCPWM registers.
    a. Reset deadtime counter for each channel by setting the corresponding bit in deadtime_disable_frm_reg[11:8] in PWM_DEADTIME_CTRL_SET_REG
    b. Set deadtime counter for each channel by setting the corresponding bit in deadtime_disable_frm_reg[11:8] in PWM_DEADTIME_CTRL_RESET_REG.
    c. Set the deadtime select bits for PWM going inactive by setting deadtime_select_inactive[7:4] in PWM_DEADTIME_CTRL_SET_REG . 0 implies counter A and 1 implies counter B.
    d. Set the deadtime select bits for PWM going active by setting deadtime_select_active[3:0] in PWM_DEADTIME_CTRL_SET_REG. 0 implies counter A and 1 implies counter B.
    e. If bits in deadtime_select_active are set to zero, then program the deadtime_prescale_select_A in PWM_DEADTIME_PRESCALE_SELECT_A corresponding to each channel and deadtime_A for the specific channel.
    f. If bits in deadtime_select_active are set to one, then program the deadtime_prescale_select_B in PWM_DEADTIME_PRESCALE_SELECT_B corresponding to each channel and deadtime_B for the specific channel.
    g. Reset deadtime counter for each channel by setting the corresponding bit in deadtime_disable_frm_reg[11:8] in PWM_DEADTIME_CTRL_SET_REG
2. Initializing interrupt status structure.
3. a. Set BIT(0) in PWM_TIME_PRD_COMMON_REG to zero to enable individual timers for each channel.
    b. Set bits[2:1] in PWM_TIME_PRD_COMMON_REG to generate a special event.
    c. Set BIT(3) in PWM_TIME_PRD_COMMON_REG to enable use of external trigger for base time counter increment or decrement.
    d. Set the corresponding bits in PWM_FLT_OVERRIDE_CTRL_SET_REG[15:12] for the enable PWM I/O pin pair is in the complementary output mode for the specific channels.
    e. Set the output polarity for low side and high side signals by setting BIT(3) and BIT(2) in PWM_FLT_OVERRIDE_CTRL_SET_REG
    f. For each channel n (n = 0,1,2,3) , program the pulse width.
    g. To program the pulse width
        i. Program tmr_operating_mode i.e., bits[2:0] in PWM_TIME_PRD_PARAM_REG_CHn to one for a single event mode. Configure pre-scalar_value and post_scalar_value by programming bits[6:4],[11:8] in PWM_TIME_PRD_PARAM_REG_CHn (n = 0,1,2,3) for each channel.
        ii. Program the initial value of base timer in PWM_TIME_PRD_CNTR_WR_REG_CH0[15:0] and base timer period in PWM_TIME_PRD_WR_REG_CHn[15:0]
    h. Configure the dutycycle for each channel by disabling corresponding bit in [7:4] and enabling corresponding bit in [3:0] in PWM_DUTYCYCLE_CTRL_SET_REG and programming duty cycle value in PWM_DUTYCYCLE_REG_WR_VALUE_n
    i. Enable PWM timer for each channel by setting BIT(1) in PWM_TIME_PRD_CTRL_REG_CH_n.
    j. Wait until an interrupt is raised by reading BIT(8) , BIT(6), BIT(4), BIT(0) in PWM_INTR_STS.
    k. Clear all interrupts by setting all the bits in PWM_INTR_ACK

16.12.5 Register Summary

Base Address: 0x4707_0000

Table 16.282. Register Summary Table

Register Name	Offset	Description
Section 16.12.6.1 PWM_INTR_STS	0x00	Interrupt Status Register
Section 16.12.6.2 PWM_INTR_UNMASK	0x04	Interrupt Unmask Register
Section 16.12.6.3 PWM_INTR_MASK	0x08	Interrupt Mask Register
Section 16.12.6.4 PWM_INTR_ACK	0x0C	Interrupt Acknowledgment Register
Section 16.12.6.5 PWM_TIME_PRD_WR_REG_CH0	0x28	Base timer period register of channel 0
Section 16.12.6.6 PWM_TIME_PRD_CNTR_WR_REG_CH0	0x2C	Base time counter initial value register for channel 0
Section 16.12.6.7 PWM_TIME_PRD_PARAM_REG_CH0	0x30	Base time period config parameter's register for channel0
Section 16.12.6.8 PWM_TIME_PRD_CTRL_REG_CH0	0x34	Base time period control register for channel0
Section 16.12.6.9 PWM_TIME_PRD_STS_REG_CH0	0x38	Base time period status register for channel0
Section 16.12.6.10 PWM_TIME_PRD_CNTR_VALUE_CH0	0x3C	Base Time period counter current value register for channel0
Section 16.12.6.11 PWM_DUTYCYCLE_CTRL_SET_REG	0x50	Duty cycle Control Set Register
Section 16.12.6.12 PWM_DUTYCYCLE_CTRL_RESET_REG	0x54	Duty cycle Control Reset Register
Section 16.12.6.13 PWM_DUTYCYCLE_REG_WR_VALUE_0	0x58	Duty cycle Value Register for Channel 0
Section 16.12.6.14 PWM_DUTYCYCLE_REG_WR_VALUE_1	0x5C	Duty cycle Value Register for Channel 1
Section 16.12.6.15 PWM_DUTYCYCLE_REG_WR_VALUE_2	0x60	Duty cycle Value Register for Channel 2
Section 16.12.6.16 PWM_DUTYCYCLE_REG_WR_VALUE_3	0x64	Duty cycle Value Register for Channel 3
Section 16.12.6.17 PWM_DEADTIME_CTRL_SET_REG	0x78	Dead time Control Set Register
Section 16.12.6.18 PWM_DEADTIME_CTRL_RESET_REG	0x7C	Dead time Control Reset Register
Section 16.12.6.19 PWM_DEADTIME_PRESCALE_SELECT_A	0x80	Dead time Prescale Select Register for A
Section 16.12.6.20 PWM_DEADTIME_PRESCALE_SELECT_B	0x84	Dead time Prescale Select Register for B
Section 16.12.6.21 PWM_DEADTIME_A_0	0x88	Dead time A for Channel 0 Register
Section 16.12.6.22 PWM_DEADTIME_B_0	0x8C	Dead time B for Channel 0 Register
Section 16.12.6.23 PWM_DEADTIME_A_1	0x90	Dead time A for Channel 1 Register
Section 16.12.6.24 PWM_DEADTIME_B_1	0x94	Dead time B for Channel 1 Register
Section 16.12.6.25 PWM_DEADTIME_A_2	0x98	Dead time A for Channel 2 Register
Section 16.12.6.26 PWM_DEADTIME_B_2	0x9C	Dead time B for Channel 2 Register
Section 16.12.6.27 PWM_DEADTIME_A_3	0xA0	Dead time A for Channel 3 Register
Section 16.12.6.28 PWM_DEADTIME_B_3	0xA4	Dead time B for Channel 3 Register
Section 16.12.6.29 PWM_OP_OVERRIDE_CTRL_SET_REG	0xC8	output override control set register
Section 16.12.6.30 PWM_OP_OVERRIDE_CTRL_RESET_REG	0xCC	output override control reset register
Section 16.12.6.31 PWM_OP_OVERRIDE_ENABLE_SET_REG	0xD0	output override enable set register

Register Name	Offset	Description
Section 16.12.6.32 PWM_OP_OVERRIDE_ENABLE_RESET_REG	0xD4	output override enable reset register
Section 16.12.6.33 PWM_OP_OVERRIDE_VALUE_SET_REG	0xD8	output override value set register
Section 16.12.6.34 PWM_OP_OVERRIDE_VALUE_RESET_REG	0xDC	output override enable reset register
Section 16.12.6.35 PWM_FLT_OVERRIDE_CTRL_SET_REG	0xE0	fault override control set register
Section 16.12.6.36 PWM_FLT_OVERRIDE_CTRL_RESET_REG	0xE4	fault override control reset register
Section 16.12.6.37 PWM_FLT_A_OVERRIDE_VALUE_REG	0xE8	fault A override value register
Section 16.12.6.38 PWM_FLT_B_OVERRIDE_VALUE_REG	0xEC	fault B override value register
Section 16.12.6.39 PWM_SVT_CTRL_SET_REG	0xF0	Special event control set register
Section 16.12.6.40 PWM_SVT_CTRL_RESET_REG	0xF4	Special event control reset register
Section 16.12.6.41 PWM_SVT_PARAM_REG	0xF8	Special event parameter register
Section 16.12.6.42 PWM_SVT_COMPARE_VALUE_REG	0xFC	Special event compare value register
Section 16.12.6.43 PWM_TIME_PRD_WR_REG_CH1	0x100	Base timer period register of channel1
Section 16.12.6.44 PWM_TIME_PRD_CNTR_WR_REG_CH1	0x104	Base time counter initial value register for channel1
Section 16.12.6.45 PWM_TIME_PRD_PARAM_REG_CH1	0x108	Base time period config parameter's register for channel1
Section 16.12.6.46 PWM_TIME_PRD_CTRL_REG_CH1	0x10C	Base time period control register for channel1
Section 16.12.6.47 PWM_TIME_PRD_STS_REG_CH1	0x110	Base time period status register for channel1
Section 16.12.6.48 PWM_TIME_PRD_CNTR_VALUE_CH1	0x114	Time period counter current value for channel1
Section 16.12.6.49 PWM_TIME_PRD_WR_REG_CH2	0x118	Base timer period register of channel2
Section 16.12.6.50 PWM_TIME_PRD_CNTR_WR_REG_CH2	0x11C	Base time counter initial value register for channal2
Section 16.12.6.51 PWM_TIME_PRD_PARAM_REG_CH2	0x120	Base time period config parameter's register for channel2
Section 16.12.6.52 PWM_TIME_PRD_CTRL_REG_CH2	0x124	Base time period control register for channel2
Section 16.12.6.53 PWM_TIME_PRD_STS_REG_CH2	0x128	Base time period status register for channel2
Section 16.12.6.54 PWM_TIME_PRD_CNTR_VALUE_CH2	0x12C	Time period counter current value register for channel2
Section 16.12.6.55 PWM_TIME_PRD_WR_REG_CH3	0x130	Base timer period register of channel3
Section 16.12.6.56 PWM_TIME_PRD_CNTR_WR_REG_CH3	0x134	Base time counter initial value register for channel3
Section 16.12.6.57 PWM_TIME_PRD_PARAM_REG_CH3	0x138	Base time period config parameter's register for channel3
Section 16.12.6.58 PWM_TIME_PRD_CTRL_REG_CH3	0x13C	Base time period control register for channel3
Section 16.12.6.59 PWM_TIME_PRD_STS_REG_CH3	0x140	Base time period status register for channel3
Section 16.12.6.60 PWM_TIME_PRD_CNTR_VALUE_CH3	0x144	Time period counter current value register for channel3
Section 16.12.6.61 PWM_TIME_PRD_COMMON_REG	0x148	Time period common register

16.12.6 Register Description

Legend:
R = Read-only, W = Write-only, R/W = Read/Write, - = Reserved

16.12.6.1 PWM_INTR_STS

Table 16.283. PWM_INTR_STS Description

Bit	Access	Function	POR Value	Description
15:10	R	Reserved	0	Reserved.
9	R	pwm_time_prd_match_intr_ch3	0	This time base interrupt for 3rd channel, which considers postscaler value.
8	R	rise_pwm_time_period_match_intr_ch3	0	This time base interrupt for 3rd channel without considering postscaler value.
7	R	pwm_time_prd_match_intr_ch2	0	This time base interrupt for 2nd channel, which considers postscaler value.
6	R	rise_pwm_time_period_match_intr_ch2	0	This time base interrupt for 2nd channel without considering postscaler value.
5	R	pwm_time_prd_match_intr_ch1		This time base interrupt for 1st channel, which considers postscaler value.
4	R	rise_pwm_time_period_match_intr_ch1	0	This time base interrupt for 1st channel without considering postscaler value.
3	R	flt_B_intr	0	When the fault B pin is driven low, this interrupt is raised. The PWM outputs remain in this state until the fault B pin is driven high and this interrupt flag has been cleared in software.
2	R	flt_A_intr	0	When the fault A pin is driven low, this interrupt is raised. The PWM outputs remain in this state until the fault A pin is driven high and this interrupt flag has been cleared in software.
1	R	pwm_time_prd_match_intr_ch0	0	This time base interrupt for 0th channel, which considers postscaler value. The generation of PWM interrupts depends on the mode of operation selected by the PWM Time Base Mode Select bits of the PWM Time Base Control register and. There is no effect of time base output postscaler.
0	R	rise_pwm_time_period_match_intr_ch0	0	This time base interrupt for 0th channel without considering postscaler. The generation of PWM interrupts depends on the mode of operation selected by the PWM Time Base Mode Select bits of the PWM Time Base Control register and the time base output postscaler selected using the PWM Time Base Output Postscale Select bits of the register.

16.12.6.2 PWM_INTR_UNMASK

Table 16.284. PWM_INTR_UNMASK Description

Bit	Access	Function	POR Value	Description
15:0	R/W	pwm_intr_unmask	0	Interrupt Unmask

16.12.6.3 PWM_INTR_MASK

Table 16.285. PWM_INTR_MASK Description

Bit	Access	Function	POR Value	Description
15:0	R/W	pwm_intr_mask	0	Interrupt Mask

16.12.6.4 PWM_INTR_ACK

Table 16.286. PWM_INTR_ACK Description

Bit	Access	Function	POR Value	Description
15:10	R/W	Reserved	0	reserved
9	W	pwm_time_prd_match_intr_ch3_ack	0	1 – pwm time period match interrupt for 3rd channel will be cleared. 0 – No effect.
8	W	rise_pwm_time_period_match_ch3_ack	0	1 – pwm time period match interrupt for 3rd channel will be cleared. 0 – No effect.
7	W	pwm_time_prd_match_intr_ch2_ack	0	1 – pwm time period match interrupt for 2nd channel will be cleared. 0 – No effect.
6	W	rise_pwm_time_period_match_ch2_ack	0	1 – pwm time period match interrupt for 2nd channel will be cleared. 0 – No effect.
5	W	pwm_time_prd_match_intr_ch1_ack	0	1 – pwm time period match interrupt for 1st channel will be cleared. 0 – No effect.
4	W	rise_pwm_time_period_match_ch1_ack	0	1 – pwm time period match interrupt for 1st channel will be cleared. 0 – No effect.
3	W	flt_B_intr_ack	0	1 – pwm fault B interrupt will be cleared. 0 – No effect.
2	W	flt_A_intr_ack	0	1 – pwm fault A interrupt will be cleared. 0 – No effect.
1	W	pwm_time_prd_match_intr_ch0_ack	0	1 – pwm time period match interrupt for 0th channel will be cleared. 0 – No effect.
0	W	rise_pwm_time_period_match_ch0_ack	0	1 – pwm time period match interrupt for 0th channel will be cleared. 0 – No effect.

16.12.6.5 PWM_TIME_PRD_WR_REG_CH0

Table 16.287. PWM_TIME_PRD_WR_REG_CH0 Description

Bit	Access	Function	POR Value	Description
15:0	R/W	pwm_time_prd_reg_wr_value_ch0	0	Value to update the base timer period register of channel 0

16.12.6.6 PWM_TIME_PRD_CNTR_WR_REG_CH0

Table 16.288. PWM_TIME_PRD_CNTR_WR_REG_CH0 Description

Bit	Access	Function	POR Value	Description
15:0	R/W	pwm_time_prd_cntr_wr_reg_ch0	0	To update the base time counter initial value for channel 0

16.12.6.7 PWM_TIME_PRD_PARAM_REG_CH0

Table 16.289. PWM_TIME_PRD_PARAM_REG_CH0 Description

Bit	Access	Function	POR Value	Description
15:12	R/W	Reserved	0	Reserved
11:8	R/W	pwm_time_prd_post_scalar_value_ch0	0	Time base output postscale bits for channel0 0000 – 1:1 postscale 0001 – 1:2 ……. 1111 – 1:16
7	R/W	Reserved	0	reserved
6 :4	R/W	Pwm_time_prd_pre_scalar_value_ch0	0	Base timer input clock prescale select value for channel0. 000 - 1x input clock period 001 - 2x input clock period 010 - 4x input clock period 011 - 8x input clock period 100 - 16x input clock period 101 - 32x input clock period 110 - 64x input clock period 111 is reserved
3	R/W	Reserved	0	reserved
2:0	R/W	tmr_operating_mode_ch0	0	Base timer operating mode for channel0. 000 - free running mode 001 - single event mode 010 - down count mode 100 – up/down mode 101 – up/down mode with interrupts for double PWM updates 011,110 & 111 are reserved

16.12.6.8 PWM_TIME_PRD_CTRL_REG_CH0

Table 16.290. PWM_TIME_PRD_CTRL_REG_CH0 Description

Bit	Access	Function	POR Value	Description
15:3	R/W	Reserved	0	Reserved
2	R/W	pwm_sft_rst	0	MC PWM soft reset. It is level signal. 1 means soft reset is enabled. 0 means it is out of soft reset.
1	R/W	pwm_time_base_en_frm_reg_ch0	0	Base timer enable for channnel0 1 – timer is enabled 0 – timer is disabled
0	R/W	pwm_time_prd_cntr_rst_frm_reg	0	Time period counter soft reset

16.12.6.9 PWM_TIME_PRD_STS_REG_CH0

Table 16.291. PWM_TIME_PRD_STS_REG_CH0 Description

Bit	Access	Function	POR Value	Description
15:1	R	Reserved	0	reserved
0	R	pwm_time_prd_dir_sts_ch0	0	Time period counter direction status for channel0. 1 – upward 0 - downward

16.12.6.10 PWM_TIME_PRD_CNTR_VALUE_CH0

Table 16.292. PWM_TIME_PRD_CNTR_VALUE_CH0 Description

Bit	Access	Function	POR Value	Description
15:0	R	pwm_time_prd_cntr_value_ch0	0	Time period counter current value for channel0

16.12.6.11 PWM_DUTYCYCLE_CTRL_SET_REG

Table 16.293. PWM_DUTYCYCLE_CTRL_SET_REG Description

Bit	Access	Function	POR Value	Description
15:8	R/W	Reserved	0	Reserved
7:4	R/W	dutycycle_update_disable	0	Duty cycle register updation disable. There is a separate bit for each channel. It is set register only.
3:0	R/W	imdt_dutycycle_update_en	15	Enable to update the duty cycle immediately (without syncing with PWM time base). There is a separate bit for each channel It is set register only.

16.12.6.12 PWM_DUTYCYCLE_CTRL_RESET_REG

Table 16.294. PWM_DUTYCYCLE_CTRL_RESET_REG Description

Bit	Access	Function	POR Value	Description
15:8	R/W	reserved	0	Reserved
7:4	R/W	dutycycle_update_disable	0	Duty cycle register updation disable. There is a separate bit for each channel.
3:0	R/W	imdt_dutycycle_update_en	15	Enable to update the duty cycle immediately (without syncing with PWM time base). There is a separate bit for each channel.

16.12.6.13 PWM_DUTYCYCLE_REG_WR_VALUE_0

Table 16.295. PWM_DUTYCYCLE_REG_WR_VALUE_0 Description

Bit	Access	Function	POR Value	Description
15:0	R/W	pwm_dutycycle_reg_wr_value_ch0	0	Duty cycle value for channel0

16.12.6.14 PWM_DUTYCYCLE_REG_WR_VALUE_1

Table 16.296. PWM_DUTYCYCLE_REG_WR_VALUE_1 Description

Bit	Access	Function	POR Value	Description
15:0	R/W	pwm_dutycycle_reg_wr_value_ch1	0	Duty cycle value for channel1

16.12.6.15 PWM_DUTYCYCLE_REG_WR_VALUE_2

Table 16.297. PWM_DUTYCYCLE_REG_WR_VALUE_2 Description

Bit	Access	Function	POR Value	Description
15:0	R/W	pwm_dutycycle_reg_wr_value_ch2	0	Duty cycle value for channel2

16.12.6.16 PWM_DUTYCYCLE_REG_WR_VALUE_3

Table 16.298. PWM_DUTYCYCLE_REG_WR_VALUE_3 Description

Bit	Access	Function	POR Value	Description
15:0	R/W	pwm_dutycycle_reg_wr_value_ch3	0	Duty cycle value for channel3

16.12.6.17 PWM_DEADTIME_CTRL_SET_REG

Table 16.299. PWM_DEADTIME_CTRL_SET_REG Description

Bit	Access	Function	POR Value	Description
15:12	R/W	Reserved	0	Reserved
11:8	R/W	deadtime_disable_frm_reg	0	Dead time counter soft reset for each channel. It is set register only.
7:4	R/W	deadtime_select_inactive	0	Deadtime select bits for PWM going inactive. 0 means use counter A 1 means use counter B There is a separate bit for each channel
3:0	R/W	deadtime_select_active	0	Deadtime select bits for PWM going active. 0 means use counter A 1 means use counter B There is a separate bit for each channel

16.12.6.18 PWM_DEADTIME_CTRL_RESET_REG

Table 16.300. PWM_DEADTIME_CTRL_RESET_REG Description

Bit	Access	Function	POR Value	Description
15:12	R/W	Reserved	0	Reserved
11:8	R/W	deadtime_disable_frm_reg	0	Dead time counter soft reset for each channel.
7:4	R/W	deadtime_select_inactive	0	Deadtime select bits for PWM going inactive. 0 means use counter A 1 means use counter B There is a separate bit for each channel
3:0	R/W	deadtime_select_active	0	Deadtime select bits for PWM going active. 0 means use counter A 1 means use counter B There is a separate bit for each channel

16.12.6.19 PWM_DEADTIME_PRESCALE_SELECT_A

Table 16.301. PWM_DEADTIME_PRESCALE_SELECT_A Description

Bit	Access	Function	POR Value	Description
15:8	R/W	Reserved	0	Reserved
7:0	R/W	deadtime_prescale_select_A	0	Dead time prescale selection bits for unit A. Used 2 bits for each channel. 00 means clock period for dead time unit A is 1x input clock period 01 means clock period for dead time unit A is 2x input clock period 10 means clock period for dead time unit A is 4x input clock period 11 means clock period for dead time unit A is 8x input clock period

16.12.6.20 PWM_DEADTIME_PRESCALE_SELECT_B

Table 16.302. PWM_DEADTIME_PRESCALE_SELECT_B Description

Bit	Access	Function	POR Value	Description
15:8	R/W	Reserved	0	Reserved
7:0	R/W	deadtime_prescale_select_B	0	Dead time prescale selection bits for unit B. Used 2 bits for each channel. 00 means clock period for dead time unit B is 1x input clock period 01 means clock period for dead time unit B is 2x input clock period 10 means clock period for dead time unit B is 4x input clock period 11 means clock period for dead time unit B is 8x input clock period

16.12.6.21 PWM_DEADTIME_A_0

Table 16.303. PWM_DEADTIME_A_0 Description

Bit	Access	Function	POR Value	Description
15:6	R/W	Reserved	0	Reserved
5:0	R/W	deadtime_A_ch0	0	Dead time A value to load into deadtime counter A of channel0

16.12.6.22 PWM_DEADTIME_B_0

Table 16.304. PWM_DEADTIME_B_0 Description

Bit	Access	Function	POR Value	Description
15:6	R/W	Reserved	0	Reserved
5:0	R/W	deadtime_B_ch0	0	Dead time B value to load into deadtime counter B of channel0

16.12.6.23 PWM_DEADTIME_A_1

Table 16.305. PWM_DEADTIME_A_1 Description

Bit	Access	Function	POR Value	Description
15:6	R/W	Reserved	0	Reserved
5:0	R/W	deadtime_A_ch1	0	Dead time A value to load into deadtime counter A of channel1

16.12.6.24 PWM_DEADTIME_B_1

Table 16.306. PWM_DEADTIME_B_1 Description

Bit	Access	Function	POR Value	Description
15:6	R/W	Reserved	0	Reserved
5:0	R/W	deadtime_B_ch1	0	Dead time B value to load into deadtime counter B of channel1

16.12.6.25 PWM_DEADTIME_A_2

Table 16.307. PWM_DEADTIME_A_2 Description

Bit	Access	Function	POR Value	Description
15:6	R/W	Reserved	0	Reserved
5:0	R/W	deadtime_A_ch2	0	Dead time A value to load into deadtime counter A of channel2

16.12.6.26 PWM_DEADTIME_B_2

Table 16.308. PWM_DEADTIME_B_2 Description

Bit	Access	Function	POR Value	Description
15:6	R/W	Reserved	0	Reserved
5:0	R/W	deadtime_B_ch2	0	Dead time B value to load into deadtime counter B of channel2

16.12.6.27 PWM_DEADTIME_A_3

Table 16.309. PWM_DEADTIME_A_3 Description

Bit	Access	Function	POR Value	Description
15:6	R/W	Reserved	0	Reserved
5:0	R/W	deadtime_A_ch3	0	Dead time A value to load into deadtime counter A of channel3

16.12.6.28 PWM_DEADTIME_B_3

Table 16.310. PWM_DEADTIME_B_3 Description

Bit	Access	Function	POR Value	Description
15:6	R/W	Reserved	0	Reserved
5:0	R/W	deadtime_B_ch3	0	Dead time B value to load into deadtime counter B of channel3

16.12.6.29 PWM_OP_OVERRIDE_CTRL_SET_REG

Table 16.311. PWM_OP_OVERRIDE_CTRL_SET_REG Description

Bit	Access	Function	POR Value	Description
15:1	R/W	Reserved	0	Reserved
0	R/W	op_override_sync	0	Output override is synced with pwm time period depending on operating mode. It is set register only.

16.12.6.30 PWM_OP_OVERRIDE_CTRL_RESET_REG

Table 16.312. PWM_OP_OVERRIDE_CTRL_RESET_REG Description

Bit	Access	Function	POR Value	Description
15:1	R/W	Reserved	0	Reserved
0	R/W	op_override_sync	0	Output override is synced with pwm time period depending on operating mode. It is reset register only.

16.12.6.31 PWM_OP_OVERRIDE_ENABLE_SET_REG

Table 16.313. PWM_OP_OVERRIDE_ENABLE_SET_REG Description

Bit	Access	Function	POR Value	Description
15:8	R/W	Reserved	0	Reserved
7:0	R/W	pwm_op_override_enable_reg	0	pwm output over ride enable. 0 bit for L0 1 bit for L1 2 bit for L2 3 bit for L3 4 bit for H0 5 bit for H1 6 bit for H2 7 bit for H3 It is set register only.

16.12.6.32 PWM_OP_OVERRIDE_ENABLE_RESET_REG

Table 16.314. PWM_OP_OVERRIDE_ENABLE_RESET_REG Description

Bit	Access	Function	POR Value	Description
15:8	R/W	Reserved	0	reserved
7:0	R/W	pwm_op_override_enable_reg	0	pwm output over ride enable. 0 bit for L0 1 bit for L1 2 bit for L2 3 bit for L3 4 bit for H0 5 bit for H1 6 bit for H2 7 bit for H3 It is reset register only.

16.12.6.33 PWM_OP_OVERRIDE_VALUE_SET_REG

Table 16.315. PWM_OP_OVERRIDE_VALUE_SET_REG Description

Bit	Access	Function	POR Value	Description
15:8	R/W	Reserved	0	Reserved
7:0	R/W	op_override_value	0	pwm output over ride value. 0 bit for L0 1 bit for L1 2 bit for L2 3 bit for L3 4 bit for H0 5 bit for H1 6 bit for H2 7 bit for H3 It is set register only.

16.12.6.34 PWM_OP_OVERRIDE_VALUE_RESET_REG

Table 16.316. PWM_OP_OVERRIDE_VALUE_RESET_REG Description

Bit	Access	Function	POR Value	Description
15:8	R/W	Reserved	0	Reserved
7:0	R/W	op_override_value	0	Pwm output over ride value. 0 bit for L0 1 bit for L1 2 bit for L2 3 bit for L3 4 bit for H0 5 bit for H1 6 bit for H2 7 bit for H3 It is reset register only.

16.12.6.35 PWM_FLT_OVERRIDE_CTRL_SET_REG

Table 16.317. PWM_FLT_OVERRIDE_CTRL_SET_REG Description

Bit	Access	Function	POR Value	Description
15:12	R/W	complementary_mode	0	PWM I/O pair mode. 1 – PWM I/O pin pair is in the complementary output mode 0 - PWM I/O pin pair is in the independent output mode Separate enable bit is present for channel It is set register only.
11:8	R/W	flt_B_enable	0	Fault B enable. Separate enable bit is present for channel It is set register only.
7:4	R/W	Flt_A_enable	0	Fault A enable. Separate enable bit is present for channel It is set register only.
3	R/W	op_polarity_L	0	Ouput polarity for low (L3, L2, L1, L0) side signals. 0 means active low mode 1 means active high mode It is set register only.
2	R/W	op_polarity_H	0	Ouput polarity for high (H3, H2, H1, H0) side signals. 0 means active low mode 1 means active high mode It is set register only.
1	R/W	flt_B_mode	0	Fault B mode 1 – cycle by cycle by mode 0 – latched mode It is set register only.
0	R/W	flt_A_mode	0	Fault A mode 1 – cycle by cycle by mode 0 – latched mode It is set register only.

16.12.6.36 PWM_FLT_OVERRIDE_CTRL_RESET_REG

Table 16.318. PWM_FLT_OVERRIDE_CTRL_RESET_REG Description

Bit	Access	Function	POR Value	Description
15:12	R/W	complementary_mode	0	PWM I/O pair mode. 1 – PWM I/O pin pair is in the complementary output mode 0 - PWM I/O pin pair is in the independent output mode Separate enable bit is present for channel It is reset register only.
11:8	R/W	flt_B_enable	0	Fault B enable. Separate enable bit is present for channel It is reset register only.
7:4	R/W	Flt_A_enable	0	Fault A enable. Separate enable bit is present for channel It is reset register only.
3	R/W	op_polarity_L	0	Output polarity for low (L3, L2, L1, L0) side signals. 0 means active low mode 1 means active high mode It is reset register only.
2	R/W	op_polarity_H	0	Output polarity for high (H3, H2, H1, H0) side signals. 0 means active low mode 1 means active high mode It is reset register only.
1	R/W	flt_B_mode	0	Fault B mode 1 – cycle by cycle by mode 0 – latched mode It is reset register only.
0	R/W	flt_A_mode	0	Fault A mode 1 – cycle by cycle by mode 0 – latched mode It is reset register only.

16.12.6.37 PWM_FLT_A_OVERRIDE_VALUE_REG

Table 16.319. PWM_FLT_A_OVERRIDE_VALUE_REG Description

Bit	Access	Function	POR Value	Description
15:0	R/W	Reserved	0	Reserved
7:0	R/W	pwm_flt_A_override_value_reg	0	Fault input A PWM override value. 1 means PWM output pin is driven active on an external fault input A event. 0 means PWM output pin is driven inactive on an external fault input A event. 0 bit for L0 1 bit for L1 2 bit for L2 3 bit for L3 4 bit for H0 5 bit for H1 6 bit for H2 7 bit for H3 Fault A has higher priority than fault B when both are enabled.

16.12.6.38 PWM_FLT_B_OVERRIDE_VALUE_REG

Table 16.320. PWM_FLT_B_OVERRIDE_VALUE_REG Description

Bit	Access	Function	POR Value	Description
15:8	R/W	Reserved	0	reserved
7:0	R/W	pwm_flt_B_override_value_reg	0	Fault input B PWM override value. 1 means PWM output pin is driven active on an external fault input B event. 0 means PWM output pin is driven inactive on an external fault input B event. 0 bit for L0 1 bit for L1 2 bit for L2 3 bit for L3 4 bit for H0 5 bit for H1 6 bit for H2 7 bit for H3 Fault A has higher priority than fault B when both are enabled.

16.12.6.39 PWM_SVT_CTRL_SET_REG

Table 16.321. PWM_SVT_CTRL_SET_REG Description

Bit	Access	Function	POR Value	Description
15:2	R/W	Reserved	0	Reserved
1	R/W	svt_direction_frm_reg	0	Special event trigger for time base direction 1 – A special event trigger will occur when PWM time base is counting down 0 – A special event trigger will occur when PWM time base is counting up It is set register only.
0	R/W	svt_enable_frm_reg	0	Special event trigger enable. This is used to enable generation special event trigger It is set register only.

16.12.6.40 PWM_SVT_CTRL_RESET_REG

Table 16.322. PWM_SVT_CTRL_RESET_REG Description

Bit	Access	Function	POR Value	Description
15:2	R/W	Reserved	0	Reserved
1	R/W	svt_direction_frm_reg	0	Special event trigger for time base direction 1 – A special event trigger will occur when PWM time base is counting down 0 – A special event trigger will occur when PWM time base is counting up It is reset register only.
0	R/W	svt_enable_frm_reg	0	Special event trigger enable. This is used to enable generation special event trigger It is reset register only.

16.12.6.41 PWM_SVT_PARAM_REG

Table 16.323. PWM_SVT_PARAM_REG Description

Bit	Access	Function	POR Value	Description
15:4	R/W	Reserved	0	Reserved
3:0	R/W	svt_postscaler_select	0	PWM special event trigger output postscale select bits 0000 means 1:1 post scale ……. 1111 means 1:16 post scale

16.12.6.42 PWM_SVT_COMPARE_VALUE_REG

Table 16.324. PWM_SVT_COMPARE_VALUE_REG Description

Bit	Access	Function	POR Value	Description
15:0	R/W	pwm_svt_compare_value	0	Special event compare value. This is used to compare with pwm time period counter to generate special event trigger.

16.12.6.43 PWM_TIME_PRD_WR_REG_CH1

Table 16.325. PWM_TIME_PRD_WR_REG_CH1 Description

Bit	Access	Function	POR Value	Description
15:0	R/W	pwm_time_prd_reg_wr_value_ch1	0	Value to update the base timer period register of channel 1

16.12.6.44 PWM_TIME_PRD_CNTR_WR_REG_CH1

Table 16.326. PWM_TIME_PRD_CNTR_WR_REG_CH1 Description

Bit	Access	Function	POR Value	Description
15:0	R/W	pwm_time_prd_cntr_wr_reg_ch1	0	To update the base time counter initial value for channel 1

16.12.6.45 PWM_TIME_PRD_PARAM_REG_CH1

Table 16.327. PWM_TIME_PRD_RARAM_REG_CH1 Description

Bit	Access	Function	POR Value	Description
15:12	R/W	Reserved	0	Reserved
11:8	R/W	pwm_time_prd_post_scalar_value_ch1	0	Time base output postscale bits for channel1 0000 – 1:1 postscale 0001 – 1:2 ……. 1111 – 1:16
7	R/W	Reserved	0	reserved
6 :4	R/W	Pwm_time_prd_pre_scalar_value_ch1	0	Base timer input clock prescale select value for channel1. 000 - 1x input clock period 001 - 2x input clock period 010 - 4x input clock period 011 - 8x input clock period 100 - 16x input clock period 101 - 32x input clock period 110 - 64x input clock period 111 is reserved
3	R/W	Reserved	0	reserved
2:0	R/W	tmr_operating_mode_ch1	0	Base timer operating mode for channel1. 000 - free running mode 001 - single event mode 010 - down count mode 100 – up/down mode 101 – up/down mode with interrupts for double PWM updates 011,110 & 111 are reserved

16.12.6.46 PWM_TIME_PRD_CTRL_REG_CH1

Table 16.328. PWM_TIME_PRD_CTRL_REG_CH1 Description

Bit	Access	Function	POR Value	Description
15:3	R/W	Reserved	0	Reserved
2	R/W	pwm_sft_rst	0	MC PWM soft reset
1	R/W	pwm_time_base_en_frm_reg_ch1	0	Base timer enable for channel1 1 – timer is enabled 0 – timer is disabled
0	R/W	pwm_time_prd_cntr_rst_frm_reg	0	Time period counter soft reset

16.12.6.47 PWM_TIME_PRD_STS_REG_CH1

Table 16.329. PWM_TIME_PRD_STS_REG_CH1 Description

Bit	Access	Function	POR Value	Description
15:1	R	Reserved	0	reserved
0	R	pwm_time_prd_dir_sts_ch1	0	Time period counter direction status for channel1. 1 – upward 0 - downward

16.12.6.48 PWM_TIME_PRD_CNTR_VALUE_CH1

Table 16.330. PWM_TIME_PRD_CNTR_VALUE_CH1 Description

Bit	Access	Function	POR Value	Description
15:0	R	pwm_time_prd_cntr_value_ch1	0	Time period counter current value for channel1

16.12.6.49 PWM_TIME_PRD_WR_REG_CH2

Table 16.331. PWM_TIME_PRD_WR_REG_CH2 Description

Bit	Access	Function	POR Value	Description
15:0	R/W	pwm_time_prd_reg_wr_value_ch2	0	Value to update the base timer period register of channel 2

16.12.6.50 PWM_TIME_PRD_CNTR_WR_REG_CH2

Table 16.332. PWM_TIME_PRD_CNTR_WR_REG_CH2 Description

Bit	Access	Function	POR Value	Description
15:0	R/W	pwm_time_prd_cntr_wr_reg_ch2	0	To update the base time counter initial value for channel 2

16.12.6.51 PWM_TIME_PRD_PARAM_REG_CH2

Table 16.333. PWM_TIME_PRD_RARAM_REG_CH2 Description

Bit	Access	Function	POR Value	Description
15:12	R/W	Reserved	0	Reserved
11:8	R/W	pwm_time_prd_post_scalar_value_ch2	0	Time base output postscale bits for channel2 0000 – 1:1 postscale 0001 – 1:2 ……. 1111 – 1:16
7	R/W	Reserved	0	reserved
6 :4	R/W	Pwm_time_prd_pre_scalar_value_ch2	0	Base timer input clock prescale select value for channel2. 000 - 1x input clock period 001 - 2x input clock period 010 - 4x input clock period 011 - 8x input clock period 100 - 16x input clock period 101 - 32x input clock period 110 - 64x input clock period 111 is reserved
3	R/W	Reserved	0	reserved
2:0	R/W	tmr_operating_mode_ch2	0	Base timer operating mode for channel2. 000 - free running mode 001 - single event mode 010 - down count mode 100 – up/down mode 101 – up/down mode with interrupts for double PWM updates 011,110 & 111 are reserved

16.12.6.52 PWM_TIME_PRD_CTRL_REG_CH2

Table 16.334. PWM_TIME_PRD_CTRL_REG_CH2 Description

Bit	Access	Function	POR Value	Description
15:3	R/W	Reserved	0	Reserved
2	R/W	pwm_sft_rst	0	MC PWM soft reset
1	R/W	pwm_time_base_en_frm_reg_ch2	0	Base timer enable for channel2 1 – timer is enabled 0 – timer is disabled
0	R/W	pwm_time_prd_cntr_rst_frm_reg	0	Time period counter soft reset

16.12.6.53 PWM_TIME_PRD_STS_REG_CH2

Table 16.335. PWM_TIME_PRD_STS_REG_CH2 Description

Bit	Access	Function	POR Value	Description
15:1	R	Reserved	0	Reserved
0	R	pwm_time_prd_dir_sts_ch2	0	Time period counter direction status for channel2. 1 – upward 0 - downward

16.12.6.54 PWM_TIME_PRD_CNTR_VALUE_CH2

Table 16.336. PWM_TIME_PRD_CNTR_VALUE_CH2 Description

Bit	Access	Function	POR Value	Description
15:0	R	pwm_time_prd_cntr_value_ch2	0	Time period counter current value for channel2

16.12.6.55 PWM_TIME_PRD_WR_REG_CH3

Table 16.337. PWM_TIME_PRD_WR_REG_CH3 Description

Bit	Access	Function	POR Value	Description
15:0	R/W	pwm_time_prd_reg_wr_value_ch3	0	Value to update the base timer period register of channel 3

16.12.6.56 PWM_TIME_PRD_CNTR_WR_REG_CH3

Table 16.338. PWM_TIME_PRD_CNTR_WR_REG_CH3 Description

Bit	Access	Function	POR Value	Description
15:0	R/W	pwm_time_prd_cntr_wr_reg_ch3	0	To update the base time counter initial value for channel 3

16.12.6.57 PWM_TIME_PRD_PARAM_REG_CH3

Table 16.339. PWM_TIME_PRD_RARAM_REG_CH3 Description

Bit	Access	Function	POR Value	Description
15:12	R/W	Reserved	0	Reserved
11:8	R/W	pwm_time_prd_post_scalar_value_ch3	0	Time base output postscale bits for channel3 0000 – 1:1 postscale 0001 – 1:2 ……. 1111 – 1:16
7	R/W	Reserved	0	reserved
6 :4	R/W	Pwm_time_prd_pre_scalar_value_ch3	0	Base timer input clock prescale select value for channel3. 000 - 1x input clock period 001 - 2x input clock period 010 - 4x input clock period 011 - 8x input clock period 100 - 16x input clock period 101 - 32x input clock period 110 - 64x input clock period 111 is reserved
3	R/W	Reserved	0	reserved
2:0	R/W	tmr_operating_mode_ch3	0	Base timer operating mode for channel3. 000 - free running mode 001 - single event mode 010 - down count mode 100 – up/down mode 101 – up/down mode with interrupts for double PWM updates 011,110 & 111 are reserved

16.12.6.58 PWM_TIME_PRD_CTRL_REG_CH3

Table 16.340. PWM_TIME_PRD_CTRL_REG_CH3 Description

Bit	Access	Function	POR Value	Description
15:3	R/W	Reserved	0	Reserved
2	R/W	pwm_sft_rst	0	MC PWM soft reset
1	R/W	pwm_time_base_en_frm_reg_ch3	0	Base timer enable for channel3 1 – timer is enabled 0 – timer is disabled
0	R/W	pwm_time_prd_cntr_rst_frm_reg	0	Time period counter soft reset

16.12.6.59 PWM_TIME_PRD_STS_REG_CH3

Table 16.341. PWM_TIME_PRD_STS_REG_CH3 Description

Bit	Access	Function	POR Value	Description
15:1	R	Reserved	0	Reserved
0	R	pwm_time_prd_dir_sts_ch3	0	Time period counter direction status for channel3. 1 – upward 0 - downward

16.12.6.60 PWM_TIME_PRD_CNTR_VALUE_CH3

Table 16.342. PWM_TIME_PRD_CNTR_VALUE_CH3 Description

Bit	Access	Function	POR Value	Description
15:0	R	pwm_time_prd_cntr_value_ch3	0	Time period counter current value for channel3

16.12.6.61 PWM_TIME_PRD_COMMON_REG

Table 16.343. PWM_TIME_PRD_COMMON_REG Description

Bit	Access	Function	POR Value	Description
15:3	R/W	Reserved	0	Reserved
3	R/W	use_ext_timer_trig_frm_reg	0	Enable to use external trigger for base time counter increment or decrement.
2:1	R.W	pwm_time_prd_common_timer_value	0	Base timers select to generate special event trigger. Out of four channel, special event can be generated for one channel (if only 0th timer is used, bit is not set)
0	R/W	pwm_time_prd_use_0th_timer_only	1	Instead of use four base timers for four channels, use only one base timer for all channels. 0 – one base timer for each channel 1 – only one base timer for all channels

16.13 Quadrature Encoder

16.13.1 General Description

A quadrature encoder (QE), also known as a 2-channel incremental encoder, converts angular displacement into two pulse signals. These two pulses are positioned 90 degrees out of phase. By monitoring both the number of pulses and the relative phase of the two signals, the user code can track the position, direction of rotation, and velocity. In addition, index signal, can be used to reset the position counter. The quadrature encoder decodes the digital pulses from a quadrature encoder wheel to integrate position over time and determine direction of rotation. In addition, the QE can capture the velocity of the encoder wheel. The QEI is present in MCU HP peripherals.

16.13.2 Features

- Tracks encoder wheel position
- Programmable for 1x, 2x, or 4x position counting. Increments/decrements depending on direction
- Index counter for revolution counting
- Velocity capture using built-in timer.
- Supports position counter reset for rollover/underflow or Index pulse
- Position, Index and Velocity compare registers with interrupts
- Supports logically swapping the A and B inputs
- Accepts decoded signal inputs (clock and direction) in timer mode

16.13.3 Functional Description

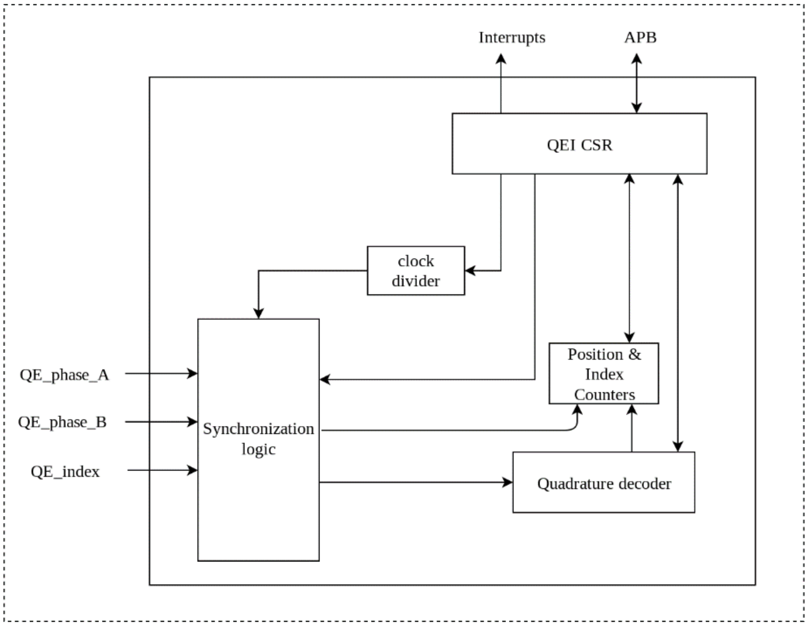
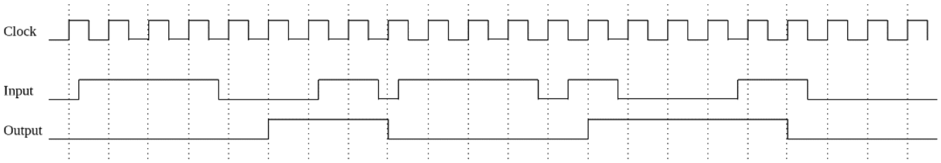


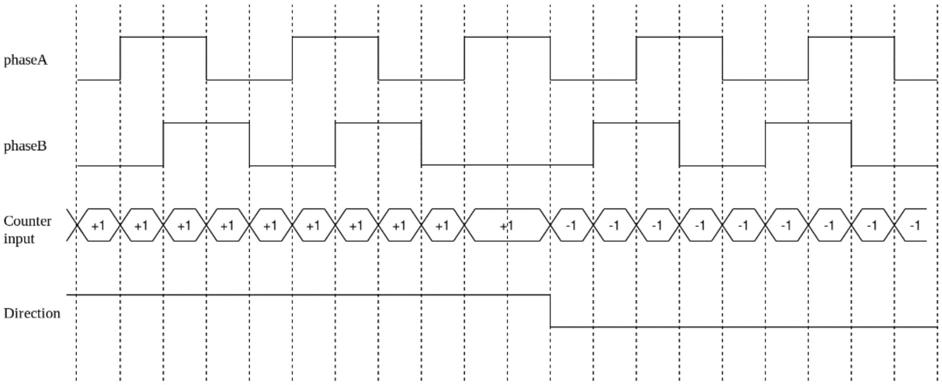
Figure 16.34. Quadrature Encoder Block Diagram

Synchronization logic is responsible for rejecting noise on the incoming index pulse and quadrature phase signals. This reject low-level noise and short duration noise spikes that typically occur in motor systems. The divided clock, which is programmable, is used to sample the input signals. The synchronization logic output signals can change only after an input level has the same value for three consecutive rising clock edges. The result is that short duration noise spikes between rising clock edges are ignored, and pulses shorter than three clock periods are rejected. It is shown in the figure below. QE uses inputs directly without filtering also, which is programmable.



Synchronization Logic Input and Output Waveforms

Quadrature decoder converts the incoming synchronized signals into count information. This multiplies the resolution of the input signals by a factor of two or four (2x or 4x decoding). When the 4x measurement mode is selected, the QE logic clocks the position counter on both edges of the Phase A and Phase B input signals (i.e increment/decrement the count by 4 for every cycle of phase A/B). The figure below illustrates the 4x measurement mode that provides finer resolution data (more position counts) to determine the encoder position.



Quadrature Decoder Signals in 4x Mode

When 2x measurement mode is selected, the QE logic looks only at the rising and falling edge of the Phase A input for the position counter increment rate. The Phase B signal is still used to determine the counter direction. When 1x measurement mode is selected, the QE logic looks only at the rising or falling edge of the Phase A input for the position counter increment/decrement. The Phase B signal is still used to determine the counter direction. When phase A leads phase B, the increment occurs on the rising edge of phase A. When phase B lead phase A, the decrement occurs on the falling edge of phase A.

The 16-bit Up/Down Position Counter counts up or down on every count pulse generated by the quadrature decoder logic. The counter acts as an integrator and its count value is proportional to the position. The direction of the count is determined by the quadrature decoder. The M4 firmware can examine the contents of the count by reading this register. Generate the interrupt value matches with position max count value. The M4 firmware can also write to the position count register to initialize a count. It also maintains Index counter. There are two modes for resetting position counter. They are reset on rollover/underflow and reset with Index pulse. If the encoder is traveling in the forward direction (position A leads position B), and the value in the position counter register matches the value in the position max count register, position counter resets to ‘0’ on the next occurring quadrature pulse edge that increments position counter. An interrupt event is generated on this rollover event. If the encoder is traveling in the reverse direction (position B leads position A), and the value in the position counter register counts down to ‘0’, the position counter register is loaded with the value in the position max count register on the next occurring quadrature pulse edge that decrements position counter. An interrupt event is generated on this underflow event. The position count can also reset, each time an index pulse is received on the Index pin. If the encoder is traveling in the forward direction (position A leads position B), position counter is reset to ‘0’. If the encoder is traveling in the reverse direction (position B leads position A), the value in the position max count register is loaded into position counter.

Compute the velocity with following procedure:

- Program the usec timer with operating frequency.
- Program the delta time counter (in usec)
- Set the start velocity counter register bit
- Wait for velocity computation over interrupt (5th bit in interrupt register) or velocity less than interrupt (3rd bit in interrupt register).
- Read the latched velocity counter value for given delta time.

16.13.4 Register Summary

Base Address: 0x4706_0000

Table 16.344. Register Summary Table

Register Name	Offset	Description
Section 16.13.5.1 QEI_STATUS_REG	0x00	Quadrature Encoder status register
Section 16.13.5.2 QEI_CTRL_REG_SET	0x04	Quadrature Encoder control set register
Section 16.13.5.3 QEI_CTRL_REG_RESET	0x08	Quadrature Encoder control reset register
Section 16.13.5.4 QEI_CNTLR_INIT_REG	0x0C	Quadrature Encoder initialization register
Section 16.13.5.5 QEI_INDEX_CNT_REG	0x10	Quadrature Encoder index counter register
Section 16.13.5.6 QEI_INDEX_MAX_CNT_REG	0x14	Quadrature Encoder maximum index counter value register
Section 16.13.5.7 QEI_POSITION_CNT_REG	0x18	Quadrature Encoder position counter register
Section 16.13.5.8 QEI_POSITION_MAX_CNT_LSW_REG	0x20	Quadrature Encoder maximum position counter value register
Section 16.13.5.9 QEI_INTR_STS_REG	0x28	Quadrature Encoder interrupt status register
Section 16.13.5.10 QEI_INTR_ACK_REG	0x2C	Quadrature Encoder interrupt acknowledge register
Section 16.13.5.11 QEI_INTR_MASK_REG	0x30	Quadrature Encoder interrupt mask register
Section 16.13.5.12 QEI_INTR_UNMASK_REG	0x34	Quadrature Encoder interrupt unmask register
Section 16.13.5.13 QEI_CLK_FREQ_REG	0x38	Quadrature Encoder clock frequency register
Section 16.13.5.14 QEI_DELTA_TIME_REG	0x3C	Quadrature Delta time register
Section 16.13.5.15 QEI_VELOCITY_REG	0x44	Quadrature velocity register
Section 16.13.5.16 QEI_POSITION_MATCH_REG	0x4C	Quadrature position match register

16.13.5 Register Description

Legend:

R = Read-only, W = Write-only, R/W = Read/Write, - = Reserved, *- dynamic value

16.13.5.1 QEI_STATUS_REG

Table 16.345. QEI_STATUS_REG Register Description

Bit	Access	Function	POR Value	Description
31:5	R	Reserved	0	Reserved.
4	R	Position_cntr_direction	0	Position Counter Direction Status bit 1 = Position counter direction is positive (+) 0 = Position counter direction is negative (-)
3	R	Position_cntr_err	0	Count Error Status Flag bit 1 = Position count error has occurred 0 = Position count error has not occurred

Bit	Access	Function	POR Value	Description
2	R	Qei_position_A	0*	This is a direct value from the position signal generator (Ex: motor wheel). If filter bypass is enable the direct value from the generator can be seen in this bit, if not filtered version of value from the generator can be seen. Value refers to the signal Position_A from the generator. * Indicates dynamic value.
1	R	Qei_position_B	0*	This is a direct value from the position signal generator (Ex: motor wheel). If filter bypass is enable the direct value from the generator can be seen in this bit, if not filtered version of value from the generator can be seen. Value refers to the signal Position_B from the generator. * Indicates dynamic value.
0	R	Qei_index	0*	This is a direct value from the position signal generator (Ex: motor wheel). If filter bypass is enable the direct value from the generator can be seen in this bit, if not filtered version of value from the generator can be seen. Value refers to the signal Index from the generator. * Indicates dynamic value.

16.13.5.2 QEI_CTRL_REG_SET

Table 16.346. QEI_CTRL_REG_SET Register Description

Bit	Access	Function	POR Value	Description
31:16	R	Reserved	0	-
15	R/W	Index_cnt_rst	0	1 - index counter is going to reset.
14	R/W	Pos_cnt_rst	0	1 - position counter is going to reset.
13	R/W	Qei_pos_cnt_16_bit_mode	0	Qei position counter 16 bit mode enable. While writing : 1 - QEI position status counter will be working as a 16 bit counter. 0 - No effect. QEI position status counter is working as 32 bit counter While reading : 1 - QEI position status counter is working as 16 bit counter. 0 - QEI position status counter is working as 32 bit counter.
12	R/W	Qei_stop_in_idle	0	1 means stop qei ctrl in idle state. 0 means continue from idle state.
11	R/W	Start_velocity_cntr	0*	Starting the velocity counter. It is self reset bit.

Bit	Access	Function	POR Value	Description
10	R/W	Timer_mode	0	1 - timer mode. In this mode, decoded timer pulse is from position A pin and direction to increment/ decrement is from position B pin. 0 - Quadrature encoder mode.
9	R/W	Digital_filter_bypass	0	1 - digital filter is bypassed for all input signals (position A, position B and Index) 0 - digital filter is in-path for all input signals
8	R/W	Index_cnt_rst_en	0	1 - index counter is going to reset after reaching max count, which is mentioned in qei_index_max_cnt register.
7	R/W	reserved	0	reserved
6	R/W	reserved	0	reserved
5	R/W	Pos_cnt_dir_frm_reg	0	Position Counter Direction indication from user 1 - Position counter direction is positive (+) 0 - Position counter direction is negative (-)
4	R/W	Pos_cnt_direction_ctrl	0	0 - position B pin defines the direction of position counter. 1 - pos_cnt_dir_frm_reg defines the position counter direction.
3	R/W	reserved	0	reserved
2	R/W	Pos_cnt_rst_with_index_en	0	1 - position counter is getting reset for every index pulse 0 - position counter is getting reset after reaching max count, which is mentioned in position_max_cnt
1	R/W	Qei_swap_phase_AB	0	Phase A and Phase B Input Swap Select bit 1 - Phase A and Phase B inputs are swapped 0 - Phase A and Phase B inputs are not swapped
0	R	Qei_sft_rst	0*	Quadrature encoder soft reset. It is self reset signal.

16.13.5.3 QEI_CTRL_REG_RESET

Table 16.347. QEI_CTRL_REG_RESET Register Description

Bit	Access	Function	POR Value	Description
31:16	R	Reserved	0	-
15	R/W	Index_cnt_rst	0	1 - index counter is going to reset.
14	R/W	Pos_cnt_rst	0	1 - position counter is going to reset.

Bit	Access	Function	POR Value	Description
13	R/W	Qei_pos_cnt_16_bit_mode	0	Qei position counter 16 bit mode enable. While writing : 1 – QEI position status counter will be working as a 32 bit counter. 0 – No effect. While reading : 1 – QEI position status counter is working as 16 bit counter. 0 – QEI position status counter is working as 32 bit counter.
12	R/W	Qei_stop_in_idle	0	1 means stop qei ctrl in idle state. 0 means continue from idle state.
11	R/W	Start_velocity_cntr	0	Starting the velocity counter. It is self reset bit.
10	R/W	Timer_mode	0	1 - timer mode. In this mode, decoded timer pulse and direction are taken from position A and position B pins respectively. 0 - Quadrature encoder mode.
9	R/W	Digital_filter_bypass	0	1 - digital filter is bypassed for all input signals (position A, position B and Index) 0 - digital filter is in-path for all input signals
8	R/W	Index_cnt_rst_en	0	1 - index counter is going to reset after reaching max count, which is mentioned in qei_index_max_cnt register.
7	R/W	Reserved	0	Reserved
6	R/W	Reserved	0	Reserved
5	R/W	Pos_cnt_dir_frm_reg	0	Position Counter Direction indication from user 1 - Position counter direction is positive (+) 0 - Position counter direction is negative (-)
4	R/W	Pos_cnt_direction_ctrl	0	0 - position B pin defines the direction of position counter. 1 - pos_cnt_dir_frm_reg defines the position counter direction.
3	R/W	Reserved	0	Reserved
2	R/W	Pos_cnt_rst_with_index_en	0	1 - position counter is getting reset for every index pulse 0 - position counter is getting reset after reaching max count, which is mentioned in position_max_cnt
1	R/W	Qei_swap_phase_AB	0	Phase A and Phase B Input Swap Select bit 1 - Phase A and Phase B inputs are swapped 0 - Phase A and Phase B inputs are not swapped
0	R	Qei_sft_rst	0*	Quadrature encoder soft reset. It is self reset signal.

16.13.5.4 QEI_CNTLR_INIT_REG

Table 16.348. QEI_CNTLR_INIT Register Description

Bit	Access	Function	POR Value	Description
31:13	R/W	Reserved	0	Reserved
12	R/W	Index_cnt_init	0	Index counter initial value in unidirectional index enable mode.
11	R/W	Unidirectional_Index	0	Uni directional index enable. 1 means direction change in position counter resets index counter
10	R/W	Unidirectional_velocity	0	Uni directional velocity enable. 1 means direction change in position counter resets velocity counter
9:6	R/W	Df_clk_divide_slt	0	Digital Filter Clock Divide Select bits 1010 = 1:1024 Clock divide for Index, position A & B 1001 = 1:512 Clock divide for Index, position A & B 1000 = 1:256 Clock divide for Index, position A & B 0111 = 1:128 Clock divide for Index, position A & B 0110 = 1:64 Clock divide for Index, position A & B 0101 = 1:32 Clock divide for Index, position A & B 0100 = 1:16 Clock divide for Index, position A & B 0011 = 1:8 Clock divide for Index, position A & B 0010 = 1:4 Clock divide for Index, position A & B 0001 = 1:2 Clock divide for Index, position A & B 0000 = 1:1 Clock divide for Index, position A & B
5:4	R/W	Index_match_value	0	These bits allow user to specify the state of position A & B during index pulse generation.
3:2	R/W	Reserved	0	Reserved
1:0	R/W	Qei_encoding_mode	0	00 = 1x mode 01 = 2x mode 10 = 4x mode

16.13.5.5 QEI_INDEX_CNT_REG

Table 16.349. QEI_INDEX_CNT Register Description

Bit	Access	Function	POR Value	Description
31:16	R/W	Qei_index_cnt_wr_value	0	User can initialize/change the index counter using this register.

Bit	Access	Function	POR Value	Description
15:0	R/W	Qei_index_cnt	0*	Index counter value. User can initialize/change the index counter using this register. When read, this provides the current index counter value. Note : This value should be less than or equal to the value in QEI_INDEX_MAX_CNT register

16.13.5.6 QEI_INDEX_MAX_CNT_REG

Table 16.350. QEI_INDEX_MAX_CNT Register Description

Bit	Access	Function	POR Value	Description
31:16	R	Reserved	-	-
15:0	R/W	Qei_index_max_cnt	0xFFFF	Qei index maximum count. This is a maximum count value that is allowed to increment in the index counter. If index counter reaches this value, will get reset to zero. Index_cnt_rst_en, qei_ctrl_reg[8] should be set (1) for resetting the index counter when reaches max value.

16.13.5.7 QEI_POSITION_CNT_REG

Table 16.351. QEI_POSITION_CNT Register Description

Bit	Access	Function	POR Value	Description
31:16	R/W	Qei_position_cnt_wr_value	0	This is used to program/change the value of position counter status[31:16]. Position counter is a 32 bit counter and can be programmed only when both registers (LSW and MSW) are programmed one after the other. Order of programming has no concern.
15:0	R/W	Qei_position_cnt_wr_value	0	This is used to program/change the value of position counter status[15:0]. Position counter is a 32 bit counter and can be programmed only when both registers (LSW and MSW) are programmed one after the other. Order of programming has no concern.

16.13.5.8 QEI_POSITION_MAX_CNT_LSW_REG

Table 16.352. QEI_POSITION_MAX_CNT_LSW Register Description

Bit	Access	Function	POR Value	Description
31:0	R/W	Qei_position_max_cnt	0x0000_FFFF	This is a maximum count value that is allowed to increment in the position counter. If position counter reaches this value in positive direction, will get set to zero and if reaches zero in negative direction, will get set to this max count. Pos_cnt_rst_with_index_en, qei_ctrl_reg[2] should be reset (0) for setting the position counter when reaches max/zero value. In 16-bit mode, only the lower 16 bits are used.

16.13.5.9 QEI_INTR_STS_REG

Table 16.353. QEI_INTR_STS Register Description

Bit	Access	Function	POR Value	Description
31:6	R	Reserved	0	Reserved
5	R	Qei_velocity_computation_over_intr_lev	0	When velocity count is computed for given delta time, than interrupt is raised.
4	R	Qei_position_cnt_match_intr_lev	0	This is raised when the position counter reaches position match value, which is programmable.
3	R	Velocity_less_than_intr_lev	0	When velocity count is less than the value given in velocity_value_to_compare register, interrupt is raised.
2	R	Position_cntr_err_intr_lev	0	Whenever number of possible positions are mismatched with actual positions are received between two index pulses this will raised.
1	R	Qei_index_cnt_match_intr_lev	0	This is raised when index counter reaches max value loaded in to index_max_cnt register.
0	R	Qei_position_cnt_reset_intr_lev	0	This is raised when the position counter reaches it’s extremes. In position direction position_max_cnt is the extreme value and in negative direction zero is the extreme value.

16.13.5.10 QEI_INTR_ACK_REG

Table 16.354. QEI_INTR_ACK Register Description

Bit	Access	Function	POR Value	Description
31:6	R/W	Reserved	0	Reserved

Bit	Access	Function	POR Value	Description
5	R/W	Velocity_computation_over_intr_lev	0	Velocity_computation_over_intr_ack If you write 1 – Velocity computation is over intr will be cleared. 0 – No effect. Zero when read
4	R/W	Qei_position_cnt_match_intr_lev	0	Qei_position_cnt_match_intr_ack If you write 1 – Qei position cnt match intr will be cleared. 0 – No effect. Zero when read
3	R/W	Velocity_less_than_intr_lev	0	Velocity_less_than_intr_ack If you write 1 – Velocity less than intr will be cleared. 0 – No effect. Zero when read
2	R/W	Position_cntr_err_intr_lev	0	Position_cntr_err_intr_ack If you write 1 – Position cntr err intr will be cleared. 0 – No effect. Zero when read
1	R/W	Qei_index_cnt_match_intr_lev	0	Position_cntr_err_intr_ack If you write 1 – Qei index cnt match intr will be cleared. 0 – No effect. Zero when read
0	R/W	Qei_position_cnt_reset_intr_lev	0	Qei_position_cnt_reset_intr_ack If you write 1 – Qei position cnt reset intr will be cleared. 0 – No effect. Zero when read

16.13.5.11 QEI_INTR_MASK_REG

Table 16.355. QEI_INTR_MASK Register Description

Bit	Access	Function	POR Value	Description
31:6	R/W	Reserved	0	Reserved.
5	R/W	Velocity_computation_over_intr_mask	0	Velocity_computation_over_intr_mask If you write 1 – Velocity computation over intr will not be given on qei_intr pin. 0 – No effect. If you read 1 – Velocity less than intr is given on qei_intr pin. 0 – Velocity less than intr is not given on qei_intr pin.
4	R/W	Qei_position_cnt_match_intr_mask	0	Qei_position_cnt_match_intr_mask If you write 1 – Qei position cnt match intr will not be given on qei_intr pin. 0 – No effect. If you read 1 – Qei position cnt match intr is given on qei_intr pin. 0 – Qei position cnt match intr is not given on qei_intr pin.
3	R/W	Velocity_less_than_intr_mask	0	Velocity_less_than_intr_mask If you write 1 – Velocity less than intr will not be given on qei_intr pin. 0 – No effect. If you read 1 – Velocity less than intr is given on qei_intr pin. 0 – Velocity less than intr is not given on qei_intr pin.

Bit	Access	Function	POR Value	Description
2	R/W	Position_cntr_err_intr_mask	0	Position_cntr_err_intr_mask If you write 1 – Position cntr err intr will not be given on qei_intr pin. 0 – No effect. If you read 1 – Position cntr err intr is given on qei_intr pin. 0 – Position cntr err intr is not given on qei_intr pin.
1	R/W	Qei_index_cnt_match_intr_mask	0	Position_cntr_err_intr_mask If you write 1 – Qei index cnt match intr will not be given on qei_intr pin. 0 – No effect. If you read 1 – Qei index cnt match intr is given on qei_intr pin. 0 – Qei index cnt match intr is not given on qei_intr pin.
0	R/W	Qei_position_cnt_reset_intr_mask	0	Qei_position_cnt_reset_intr_mask If you write 1 – Qei position cnt reset intr will not be given on qei_intr pin. 0 – No effect. If you read 1 – Qei position cnt reset intr is given on qei_intr pin. 0 – Qei position cnt reset intr is not given on qei_intr pin.

16.13.5.12 QEI_INTR_UNMASK_REG

Table 16.356. QEI_INTR_UNMASK Register Description

Bit	Access	Function	POR Value	Description
31:6	R/W	Reserved	0	Reserved.

Bit	Access	Function	POR Value	Description
5	R/W	velocity_computation_over_intr_unmask	0	Velocity_computation_over_intr_unmask If you write 1 – Velocity computation over intr will not be given on qei_intr pin. 0 – No effect. If you read 1 – Velocity less than intr is given on qei_intr pin. 0 – Velocity less than intr is not given on qei_intr pin.
4	R/W	Qei_index_cnt_match_intr_unmask	0	Qei_position_cnt_match_intr_unmask If you write 1 – Qei position cnt match intr will be given on qei_intr pin. 0 – No effect. If you read 1 – Qei position cnt match intr is given on qei_intr pin. 0 – Qei position cnt match intr is not given on qei_intr pin.
3	R/W	Velocity_less_than_intr_unmask	0	Velocity_less_than_intr_unmask If you write 1 – Velocity less than intr will be given on qei_intr pin. 0 – No effect. If you read 1 – Velocity less than intr is given on qei_intr pin. 0 – Velocity less than intr is not given on qei_intr pin.
2	R/W	Position_cntr_err_intr_unmask	0	Position_cntr_err_intr_unmask If you write 1 – Position cntr err intr will be given on qei_intr pin. 0 – No effect. If you read 1 – Position cntr err intr is given on qei_intr pin. 0 – Position cntr err intr is not given on qei_intr pin.

Bit	Access	Function	POR Value	Description
1	R/W	Qei_index_cnt_match_intr_unmask	0	Position_cntr_err_intr_unmask If you write 1 – Qei index cnt match intr will be given on qei_intr pin. 0 – No effect. If you read 1 – Qei index cnt match intr is given on qei_intr pin. 1 – Qei index cnt match intr is not given on qei_intr pin.
0	R/W	Qei_position_cnt_reset_intr_unmask	0	Qei_position_cnt_err_intr_unmask If you write 1 – Qei position cnt reset intr will be given on qei_intr pin. 0 – No effect. If you read 1 – Qei position cnt reset intr is given on qei_intr pin. 0 – Qei position cnt reset intr is not given on qei_intr pin.

16.13.5.13 QEI_CLK_FREQ_REG

Table 16.357. QEI_CLK_FREQ Register Description

Bit	Access	Function	POR Value	Description
31:9	R/W	Reserved	0	Reserved.
8:0	R/W	Qei_clk_freq	39	Indication of clock frequency on which QEI controller is running. This should be loaded with running clk freq – 1. For generating real time micro sec QEI uses this value. For ex : If QEI is running on 40 MHz this should be loaded with 39.

16.13.5.14 QEI_DELTA_TIME_REG

Table 16.358. QEI_DELTA_TIME Register Description

Bit	Access	Function	POR Value	Description
31:20	R/W	Reserved	0	Reserved.
19:0	R/W	Delta_time_for_velocity	999	Delta time to compute velocity.

16.13.5.15 QEI_VELOCITY_REG

Table 16.359. QEI_VELOCITY Register Description

Bit	Access	Function	POR Value	Description
31:0	R/W	Velocity_value_to_compare	0*	For write operation : It is the velocity value to compare with velocity count. If velocity count is less than the value given in this register, interrupt is raised. For read operation : It is the velocity count to compare using NWP firmware. This is number of position pulses for given delta time in delta time register.

16.13.5.16 QEI_POSITION_MATCH_REG

Table 16.360. QEI_POSITION_MATCH Register Description

Bit	Access	Function	POR Value	Description
31:0	R/W	position_match_value	0	Position match value to compare the position counter. When it is matched with position counter, interrupt is raised.

16.14 SSI Primary

16.14.1 General Description

There are two Synchronous Serial Interface (SSI) Primaries- one in the MCU HP peripherals (SSI_MST) and one in the MCU ULP subsystem (ULP_SSI_MST). The SSI Primaries are programmable controllers that can be configured to support different full-duplex Primary synchronous serial interface protocols.

The SSI Primary can connect to any serial-secondary peripheral device using one of the following interfaces:

- Motorola Serial Peripheral Interface (SPI)
- Texas Instruments Serial Protocol (SSP)
- National Semiconductor Microwire.

16.14.2 Features

Each of these SSI Primaries supports the following features:

- Support for Motorola SPI, TI SSP, and National Semiconductors Microwire protocols
- The SSI_MST in MCU HP peripherals provides an option to connect upto four secondaries* and supports Single, Dual*, and Quad* modes.
- The ULP_SSI_MST in the MCU ULP peripherals supports single-bit mode and can be connected to only one secondary
- Programmable receive sampling delay

In addition to the above features, the SSI Primary reduces the load on the processor by supporting the features below:

- Programmable FIFO thresholds with maximum FIFO depth of 16 and support for DMA
- Supports generation of interrupt for FIFO status and Multi-Primary Contention.
- Programmable division factor for generating SSI clock out.

The ULP_SSI_MST supports the following additional power-save features:

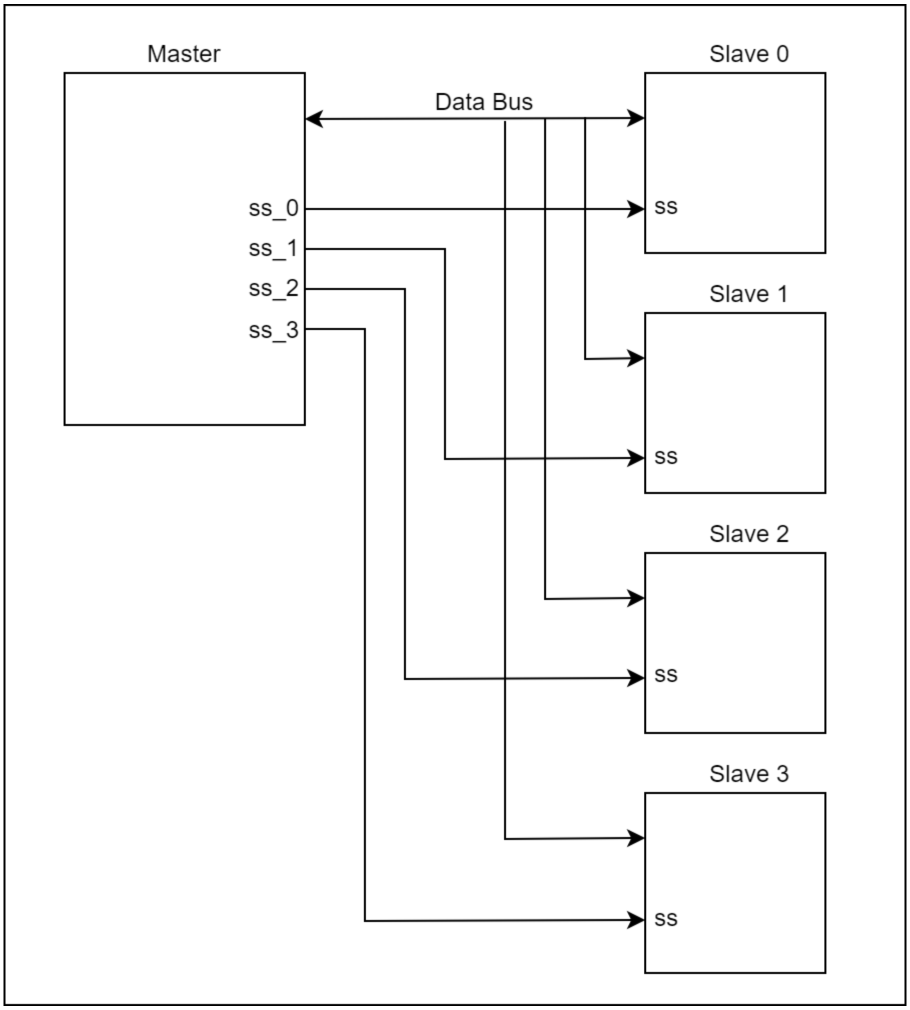
- After the DMA is programmed in PS2 state for SSI transfers, the MCU can switch to PS1 state (processor is shutdown) while the SSI Primary continues with the data transfer
- In PS1 state (ULP Peripheral mode) the SSI Primary completes the data transfer and, triggered by the Peripheral Interrupt, shifts either to the sleep state (without processor intervention) or the active state

16.14.3 Functional Description

The MCU accesses data, control, and status information on the SSI Primary through the APB interface. This may also interface with a DMA Controller using an optional set of DMA signals. The SSI Primary can connect to any serial-secondary peripheral device using one of the following interfaces:

- Motorola Serial Peripheral Interface (SPI): A four-wire, full-duplex, serial protocol from Motorola. There are four possible combinations for the serial clock phase and polarity. The clock phase (SCPH) determines whether the serial transfer begins with the falling edge of the Secondary select signal or the first edge of the serial clock. The Secondary select line is held high when the SSI Primary is idle or disabled.
- Texas Instruments Serial Protocol (SSP): A four-wire, full-duplex, serial protocol. The Secondary select line is used for SPI and Microwire protocols doubles as the frame indicator for the SSP protocol.
- National Semiconductor Microwire: A half-duplex, serial protocol, which uses a control word transmitted from the serial Primary to the target serial Secondary.

FRF (frame format) bit field in the Control Register 0 (CTRLR0) is used to select which protocol is used. Four serial Secondaries* can be connected using this SSI Primary, as shown in the figure below. The serial-Primary device asserts the select line of the target serial Secondary before data transfer begins. If there are multiple serial Primaries in the system, the Secondaries select output from all the Primaries, can be logically ANDed to generate a single Secondary select input for all serial Secondary devices.



The frequency of the SSI Primary input clock (ssi_clk) must be less than or equal to the frequency of the APB bus clock (pclk), which guarantees that control signals from the ssi_clk domain are synchronized to the pclk domain. The maximum frequency of the SSI Primary bit-rate clock (sclk_out) is one-half the frequency of ssi_clk. This allows the shift control logic to capture data on one clock edge of sclk_out and propagate data on the opposite edge. The sclk_out line toggles only when an active transfer is in progress. At all other times it is held in an inactive state, as defined by the serial protocol under which it operates. The frequency of sclk_out can be derived from the following equation:

Fssi_clk

Fsclk_out = --------------------

SCKDV

SCKDV is a bit field in the programmable register BAUDR, holding any even value in the range 0 to 65,534. If SCKDV is 0, then sclk_out is disabled.

16.14.4 SSI Interrupts

The SSI Primary supports combined and individual interrupt requests, each of which can be masked. The combined interrupt request is the ORed result of all other interrupts after masking. All SSI Primary interrupts are level interrupts and have the same active polarity level. This polarity level can be configured as active-high or active-low. The SSI Primary interrupts are described as follows:

- Transmit FIFO Empty Interrupt (ssi_txe_intr) – Set when the transmit FIFO is equal to or below its threshold value and requires service to prevent an under-run. The threshold value, set through a software-programmable register, determines the level of transmit FIFO entries at which an interrupt is generated. This interrupt is cleared by hardware when data are written into the transmit FIFO buffer, bringing it over the threshold level.
- Transmit FIFO Overflow Interrupt (ssi_txo_intr) – Set when an APB access attempts to write into the transmit FIFO after it has been completely filled. When set, data written from the APB is discarded. This interrupt remains set until reading of the transmit FIFO overflow interrupt clear register (TXOICR).
- Receive FIFO Full Interrupt (ssi_rxf_intr) – Set when the receive FIFO is equal to or above its threshold value plus 1 and requires service to prevent an overflow. The threshold value, set through a software-programmable register, determines the level of receive FIFO entries at which an interrupt is generated. This interrupt is cleared by hardware when data are read from the receive FIFO buffer, bringing it below the threshold level.
- Receive FIFO Overflow Interrupt (ssi_rxo_intr) – Set when the receive logic attempts to place data into the receive FIFO after it has been completely filled. When set, newly received data are discarded. This interrupt remains set until reading of the receive FIFO overflow interrupt clear register (RXOICR).
- Receive FIFO Underflow Interrupt (ssi_rxu_intr) – Set when an APB access attempts to read from the receive FIFO when it is empty. When set, zeros are read back from the receive FIFO. This interrupt remains set until reading of the receive FIFO underflow interrupt clear register (RXUICR).
- Multi-Primary Contention Interrupt (ssi_mst_intr) –The interrupt is set when another serial Primary on the serial bus selects the SSI Primary as a serial-Secondary device and is actively transferring data. This informs the processor of possible contention on the serial bus. This interrupt remains set until reading of the multi-Primary interrupt clear register (MSTICR).
- Combined Interrupt Request (ssi_intr) – ORed result of all the above interrupt requests after masking. To mask this interrupt signal, mask all other SSI interrupt requests.

16.14.5 Programming Sequence

Data transfers are started by the serial-Primary device. When the SSI Primary is enabled (SSI_EN=1), at least one valid data entry is present in the transmit FIFO and a serial Secondary device is selected. When actively transferring data, the busy flag (BUSY) in the status register (SR) is set. Wait until the busy flag is cleared before attempting a new serial transfer.

16.14.5.1 Primary SPI and SSP Serial Transfers

A typical software flow for completing an SPI or SSP serial transfer from the SSI Primary serial Primary is outlined as follows:

1. If the SSI Primary is enabled, disable it by writing 0 to the SSI Enable register (SSIENR).

2. Set up the SSI Primary control registers for the transfer; these registers can be set in any order.

- Write Control Register 0 (CTRLR0). For SPI transfers, the serial clock polarity and serial clock phase parameters must be set identical to target Secondary device.
- If the transfer mode is receive only, write CTRLR1 (Control Register 1) with the number of frames in the transfer minus 1;
- Write the Baud Rate Select Register (BAUDR) to set the baud rate for the transfer.
- Write the Transmit and Receive FIFO Threshold Level registers (TXFTLR and RXFTLR, respectively) to set FIFO threshold levels.
- Write the IMR register to set up interrupt masks.
- The Secondary Enable Register (SER) register can be written here to enable the target Secondary for selection. If a Secondary is enabled here, the transfer begins as soon as one valid data entry is present in the transmit FIFO. If no Secondaries are enabled prior to writing to the Data Register (DR), the transfer does not begin until a Secondary is enabled.

3. Enable the SSI Primary by writing 1 to the SSIENR register.

4. Write data for transmission to the target Secondary into the transmit FIFO (write DR). If no Secondaries were enabled in the SER register at this point, enable it now to begin the transfer.

5. Poll the BUSY status to wait for completion of the transfer. The BUSY status cannot be polled immediately. If a transmit FIFO empty interrupt request is made, write the transmit FIFO (write DR). If a receive FIFO full interrupt request is made, read the receive FIFO (read DR).

6. The transfer is stopped by the shift control logic when the transmit FIFO is empty. If the transfer mode is receive only (TMOD = 2'b10), the transfer is stopped by the shift control logic when the specified number of frames have been received. When the transfer is done, the BUSY status is reset to 0.

7. If the transfer mode is not transmit only (TMOD != 01), read the receive FIFO until it is empty.

8. Disable the SSI Primary by writing 0 to SSIENR.

The figure below shows a typical software flow for starting SPI/SSP serial transfer.

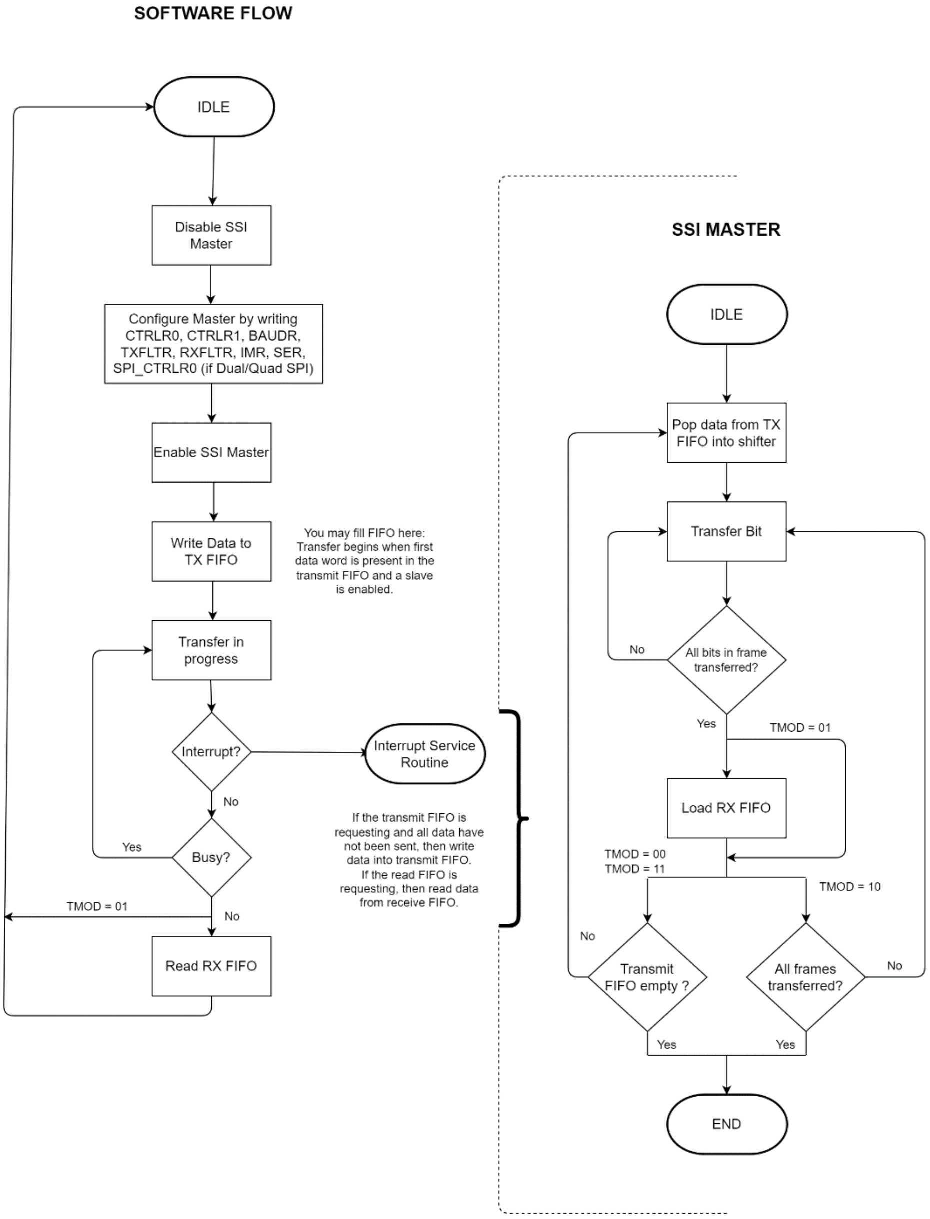

SOFTWARE FLOW
IDLE
Disable SSI Master
Configure Master by writing CTRLR0, CTRLR1, BAUDR, TXFLTR, RXFLTR, IMR, SER, SPI_CTRLR0 (if Dual/Quad SPI)
Enable SSI Master
Write Data to TX FIFO
You may fill FIFO here: Transfer begins when first data word is present in the transmit FIFO and a slave is enabled.
Transfer in progress
Interrupt?
Interrupt Service Routine
No
Yes
Busy?
If the transmit FIFO is requesting and all data have not been sent, then write data into transmit FIFO. If the read FIFO is requesting, then read data from receive FIFO.
TMOD = 01
No
Read RX FIFO
SSI MASTER
IDLE
Pop data from TX FIFO into shifter
Transfer Bit
No
All bits in frame transferred?
Yes
TMOD = 01
Load RX FIFO
TMOD = 00
TMOD = 11
TMOD = 10
No
Transmit FIFO empty ?
All frames transferred?
No
Yes
Yes
END

Figure 16.35. SSI Primary SPI/SSP Transfer Flow diagram

16.14.5.2 Primary Microwire Serial Transfers

A typical software flow for completing a Microwire serial transfer from the SSI Primary is outlined as follows:

1. If the SSI Primary is enabled, disable it by writing 0 to SSIENR.

2. Set up the SSI control registers for the transfer. These registers can be set in any order. Write CTRLR0 to set transfer parameters.

- ? If the transfer is sequential and the SSI Primary receives data, write CTRLR1 with the number of frames in the transfer minus 1.
- ? Write BAUDR to set the baud rate for the transfer.
- ? Write TXFTLR and RXFTLR to set FIFO threshold levels.
- ? Write the IMR register to set up interrupt masks.

Write the SER register to enable the target Secondary for selection. If a Secondary is enabled here, the transfer begins as soon as one valid data entry is present in the transmit FIFO. If no Secondaries are enabled prior to writing to the DR register, the transfer does not begin until a Secondary is enabled.

3. Enable the SSI Primary by writing 1 to the SSIENR register.

4. If the SSI Primary transmits data, write the control and data words into the transmit FIFO (write DR). If the SSI Primary receives data, write the control word(s) into the transmit FIFO. If no Secondaries were enabled in the SER register at this point, enable now to begin the transfer.

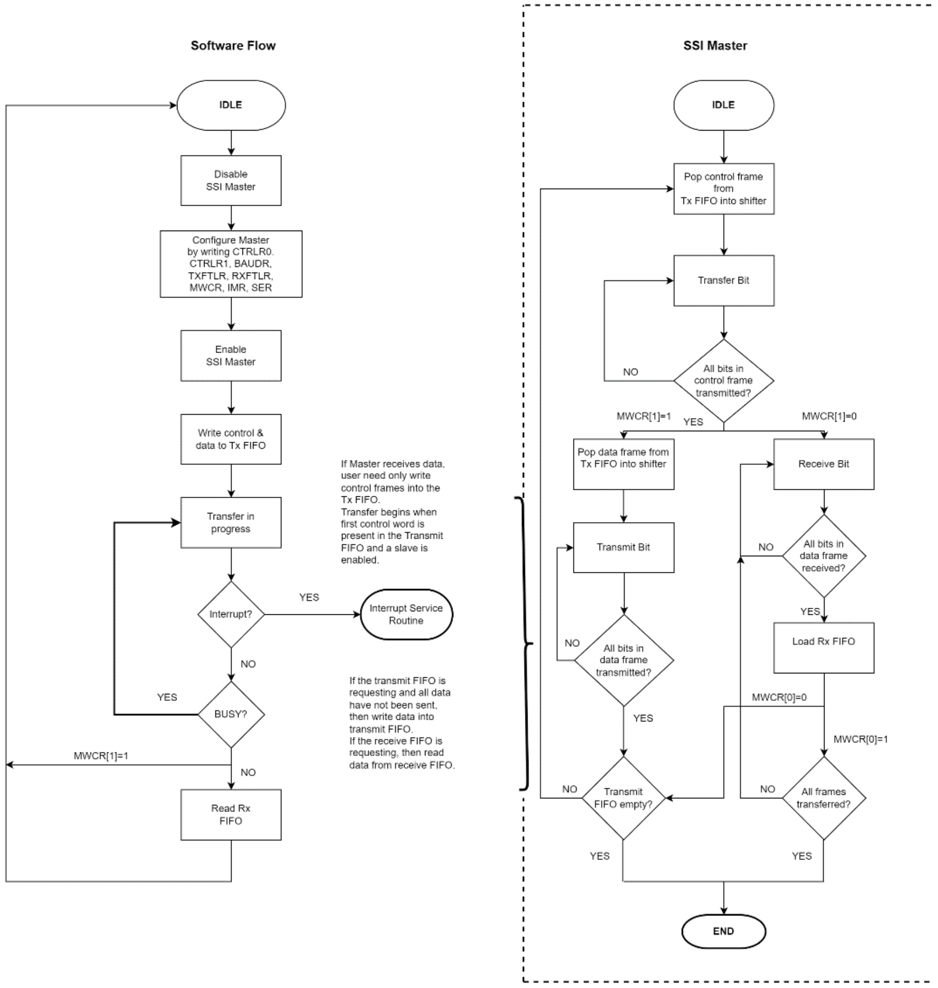
5. Poll the BUSY status to wait for completion of the transfer. The BUSY status cannot be polled immediately. If a transmit FIFO empty interrupt request is made, write the transmit FIFO (write DR). If a receive FIFO full interrupt request is made, read the receive FIFO (read DR).

6. The transfer is stopped by the shift control logic when the transmit FIFO is empty. If the transfer mode is sequential and the SSI Primary receives data, the transfer is stopped by the shift control logic when the specified number of data frames is received. When the transfer is done, the BUSY status is reset to 0.

7. If the SSI Primary receives data, read the receive FIFO until it is empty.

8. Disable the SSI Primary by writing 0 to SSIENR.

The figure below shows a typical software flow for starting Microwire serial transfer.


Software Flow
IDLE
Disable SSI Master
Configure Master by writing CTRLR0. CTRLR1, BAUDR, TXFTLR, RXFTLR, MWCR, IMR, SER
Enable SSI Master
Write control & data to Tx FIFO
Transfer in progress
Interrupt?
YES
Interrupt Service Routine
NO
YES
BUSY?
MWCR[1]=1
NO
Read Rx FIFO
SSI Master
IDLE
Pop control frame from Tx FIFO into shifter
Transfer Bit
NO
All bits in control frame transmitted?
MWCR[1]=1
YES
MWCR[1]=0
Pop data frame from Tx FIFO into shifter
Receive Bit
Transmit Bit
NO
All bits in data frame received?
YES
Load Rx FIFO
NO
All bits in data frame transmitted?
YES
MWCR[0]=0
MWCR[0]=1
NO
Transmit FIFO empty?
NO
All frames transferred?
YES
YES
END

If Master receives data, user need only write control frames into the Tx FIFO.
Transfer begins when first control word is present in the Transmit FIFO and a slave is enabled.

If the transmit FIFO is requesting and all data have not been sent, then write data into transmit FIFO.
If the receive FIFO is requesting, then read data from receive FIFO.

Figure 16.36. Microwire serial Transfer Flow Diagram Using SSI Primary

16.14.6 Register Summary

Base Address: 0x4402_0000

Table 16.361. Register Summary Table

Register Name	Offset	Description
Section 16.14.7.1 CTRLR0	0x00	Control Register 0
Section 16.14.7.2 CTRLR1	0x04	Control Register 1
Section 16.14.7.3 SSIENR	0x08	SSI Enable Register
Section 16.14.7.4 MWCR	0x0c	Microwire Control Register
Section 16.14.7.5 SER	0x10	Secondary Enable Register
Section 16.14.7.6 BAUDR	0x14	Baud Rate Select
Section 16.14.7.7 TXFLTR	0x18	Transmit FIFO Threshold Level Register
Section 16.14.7.8 RXFLTR	0x1c	Receive FIFO Threshold Level Register
Section 16.14.7.9 TXFLR	0x20	Transmit FIFO Level Register
Section 16.14.7.10 RXFLR	0x24	Receive FIFO Level Register
Section 16.14.7.11 SR	0x28	Status Register
Section 16.14.7.12 IMR	0x2c	Interrupt Mask Register
Section 16.14.7.13 ISR	0x30	Interrupt Status Register
Section 16.14.7.14 RISR	0x34	Raw Interrupt Status Register
Section 16.14.7.15 TXOICR	0x38	Transmit FIFO Overflow Interrupt Clear Register
Section 16.14.7.16 RXOICR	0x3c	Receive FIFO Overflow Interrupt Clear Register
Section 16.14.7.17 RXUICR	0x40	Receive FIFO Underflow Interrupt Clear Register
Section 16.14.7.18 MSTICR	0x44	Multi-Master Interrupt Clear Register
Section 16.14.7.19 ICR	0x48	Interrupt Clear Register
Section 16.14.7.20 DMACR	0x4c	DMA Control Register
Section 16.14.7.21 DMATDLR	0x50	DMA Transmit Data Level Register
Section 16.14.7.22 DMARDLR	0x54	DMA Receive Data Level Register
Section 16.14.7.23 IDR	0x58	Identification Register
Section 16.14.7.24 SSI_COMP_VERSION	0x5c	SSI Component Version
Section 16.14.7.25 DR	0x60-EC	Data Registers
Section 16.14.7.26 RXD_SAMPLE_DELAY	0xF0	RXD Sample Delay Register
Section 16.14.7.27 SPI_CTRLR0	0xF4	SPI Control Register
Reserved	0xF8	Reserved
Reserved	0xFC	Reserved

16.14.7 Register Description

Legend:
R = Read-only, W = Write-only, R/W = Read/Write, N/A = Reserved

16.14.7.1 CTRLR0

Table 16.362. Control Register 0 Description

Bit	Access	Function	POR Value	Description
31:23	N/A	Reserved	0	Reserved
22:21	R/W	SPI_FRF	0	SPI Frame Format: Selects data frame format for transmitting or receiving data. • 00 – Standard SPI Format • 01 – Dual SPI Format* • 10 – Quad SPI Format* • 11 – Reserved
20:16	R/W	DFS_32	0x7	Data Frame Size. Selects the data frame length. Range : 0011 -> 4 bit to 1111 -> 16 bit
15:12	R/W	CFS	0x0	Control Frame Size. Selects the length of the control word for the Microwire frame format. Ranges : 0000 -> 1 bit word control to 1111 -> 16 bit word control
11	R/W	SRL	0	Shift Register Loop used for testing purposes only. When internally active, connects the transmit shift register output to the receive shift register input. 0 – Normal Mode Operation 1 – Test Mode Operation
10	R/W	Reserved	0	Reserved
9:8	R/W	TMOD	0	Transfer Mode. Selects the mode of transfer for serial communication. 00 – Transmit & Receive 01 – Transmit Only 10 – Receive Only
7	R/W	SCPOL	0	Serial Clock Polarity. Valid when the frame format (FRF) is set to Motorola SPI. 0 – Inactive state of serial clock is low 1 – Inactive state of serial clock is high
6	R/W	SCPH	0	Serial Clock Phase. Valid when the frame format (FRF) is set to Motorola SPI. When SCPH = 0, data are captured on the first edge of the serial clock. When SCPH = 1, the serial clock starts toggling one cycle after the Slave select line is activated, and data are captured on the second edge of the serial clock. 0: Serial clock toggles in middle of first data bit 1: Serial clock toggles at start of first data bit
5:4	R/W	FRF	0	Frame Format. Selects which serial protocol transfers the data. 00 – Motorola SPI 01 – Texas Instruments SSP 10 – National Semiconductors Microwire 11 – Reserved
3:0	R/W	DFS	0x7	Data Frame Size. Selects the data frame length. Range : 0011 -> 4 bit to 1111 -> 16 bit

16.14.7.2 CTRLR1

Table 16.363. Control Register1 Description

Bit	Access	Function	POR Value	Description
31:16	N/A	Reserved		Reserved
15:0	R/W	NDF	0x0	Number of Data Frames. When TMOD = 10 or TMOD = 11, this register field sets the number of data frames to be continuously received by the SSI Master. The SSI Master continues to receive serial data until the number of data frames received is equal to this register value plus 1, which enables you to receive up to 64 KB of data in a continuous transfer.

16.14.7.3 SSIENR

Table 16.364. SSI Enable Register Description

Bit	Access	Function	POR Value	Description
31:1	R	Reserved	-	
0	R/W	SSI_EN	0x0	SSI Enable. Enables and disables all SSI Master operations. When disabled, all serial transfers are halted immediately. Transmit and receive FIFO buffers are cleared when the device is disabled.

16.14.7.4 MWCR

Table 16.365. Microwire Control Register Description

Bit	Access	Function	POR Value	Description
31:3	N/A	Reserved	0	Reserved
2	R/W	MHS	0	Microwire Handshaking. Used to enable and disable the “busy/ready” handshaking interface for the Microwire protocol. When enabled, the ssi checks for a ready status from the target Slave, after the transfer of the last data/control bit, before clearing the BUSY status in the SR register. 0: handshaking interface is disabled 1: handshaking interface is enabled
1	R/W	MDD	0	Microwire Control.When this bit is set to 0, the data word is received by the SSI MacroCell from the external serial device. When this bit is set to 1, the data word is transmitted from the SSI MacroCell to the external serial device.
0	R/W	MWMOD	0	Microwire Transfer Mode. 0 – non-sequential transfer 1 – sequential transfer When sequential mode is used, only one control word is needed to transmit or receive a block of data words. When non-sequential mode is used, there must be a control word for each data word that is transmitted or received.

16.14.7.5 SER

Table 16.366. Secondary Enable Register Description

Bit	Access	Function	POR Value	Description
31:4	N/A	Reserved	0	Reserved
3:0	R/W	SER	0	Secondary Select Enable Flag. Each bit in this register corresponds to a Secondary select line (ss_x_n) from the SSI Primary. When a bit in this register is set (1), the corresponding Secondary select line from the Primary is activated when a serial transfer begins. It should be noted that setting or clearing bits in this register have no effect on the corresponding Secondary select outputs until a transfer is started. Before beginning a transfer, the bit in this register that corresponds to the Secondary device with which the Primary wants to communicate should be enabled. When not operating in broadcast mode, only one bit in this field should be set. 1: Selected 0: Not Selected

16.14.7.6 BAUDR

Table 16.367. Baus Rate Select Register Description

Bit	Access	Function	POR Value	Description
31:16	N/A	Reserved	0	Reserved
15:0	R/W	SCKDV	0	SSI Clock Divider. The LSB for this field is always set to 0 and is unaffected by a write operation, which ensures an even value is held in this register. If the value is 0, the serial output clock (sclk_out) is disabled. The frequency of the sclk_out is derived from the following equation: Fsclk_out = Fssi_clk/SCKDV where SCKDV is any even value between 2 and 65534

16.14.7.7 TXFLTR

Table 16.368. Transmit FIFO Threshold Level Register Description

Bit	Access	Function	POR Value	Description
31:4	N/A	Reserved	0	Reserved
3:0	R/W	TFT	0	Transmit FIFO Threshold. Controls the level of entries (or below) at which the transmit FIFO controller triggers an interrupt. If this field is set to a value greater than or equal to the depth of the FIFO, this field is not written and retains its current value. When the number of transmit FIFO entries is less than or equal to this value, the transmit FIFO empty interrupt is triggered.

16.14.7.8 RXFLTR

Table 16.369. Receive FIFO Threshold Level Register

Bit	Access	Function	POR Value	Description
31:4	N/A	Reserved	0	Reserved
3:0	R/W	RFT	0	Receive FIFO Threshold. Controls the level of entries (or below) at which the receive FIFO controller triggers an interrupt. If this field is set to a value greater than or equal to the depth of the FIFO, this field is not written and retains its current value. When the number of receive FIFO entries is less than or equal to this value + 1, the receive FIFO full interrupt is triggered.

16.14.7.9 TXFLR

Table 16.370. Transmit FIFO Level Register Description

Bit	Access	Function	POR Value	Description
31:5	N/A	Reserved	0	Reserved
4:0	R	TXTFL	0	Transmit FIFO Level. Contains the number of valid data entries in the transmit FIFO.

16.14.7.10 RXFLR

Table 16.371. Receive FIFO Level Register Description

Bit	Access	Function	POR Value	Description
31:5	N/A	Reserved	0	Reserved
4:0	R	RXTFL	0	Receive FIFO Level. Contains the number of valid data entries in the receive FIFO.

16.14.7.11 SR

Table 16.372. Status Register Description

Bit	Access	Function	POR Value	Description
31:7	N/A	Reserved	0	Reserved
6	R	DCOL	0	Data Collision Error. This bit is set if the ss_in_n input is asserted by another Master, while the ssi Master is in the middle of the transfer. This informs the processor that the last data transfer was halted before completion. This bit is cleared when read. 0 – No error 1 – Transmit data collision error

Bit	Access	Function	POR Value	Description
5	N/A	Reserved	0	Reserved
4	R	RFF	0	Receive FIFO Full. 0 – Receive FIFO is not full 1 – Receive FIFO is full
3	R	RFNE	0	Receive FIFO Not Empty. 0 – Receive FIFO is empty 1 – Receive FIFO is not empty This bit can be polled by software to completely empty the receive FIFO.
2	R	TFE	1	Transmit FIFO Empty. 0 – Transmit FIFO is not empty 1 – Transmit FIFO is empty This bit field does not request an interrupt.
1	R	TFNF	1	Transmit FIFO Not Full. 0 – Transmit FIFO is full 1 – Transmit FIFO is not full
0	R	BUSY	0	SSI Busy Flag. 0 – SSI is idle or disabled 1 – SSI is actively transferring data

16.14.7.12 IMR

Table 16.373. Interrupt Mask Register Description

Bit	Access	Function	POR Value	Description
31:6	N/A	Reserved	0	Reserved
5	R/W	MSTIM	1	Multi-Master Contention Interrupt Mask. 0 – ssi_mst_intr interrupt is masked 1 – ssi_mst_intr interrupt is not masked
4	R/W	RXFIM	1	Receive FIFO Full Interrupt Mask 0 – ssi_rxf_intr interrupt is masked 1 – ssi_rxf_intr interrupt is not masked
3	R/W	RXOIM	1	Receive FIFO Overflow Interrupt Mask 0 – ssi_rxo_intr interrupt is masked 1 – ssi_rxo_intr interrupt is not masked
2	R/W	RXUIM	1	Receive FIFO Underflow Interrupt Mask 0 – ssi_rxu_intr interrupt is masked 1 – ssi_rxu_intr interrupt is not masked
1	R/W	TXOIM	1	Transmit FIFO Overflow Interrupt Mask 0 – ssi_txo_intr interrupt is masked 1 – ssi_txo_intr interrupt is not masked
0	R/W	TXEIM	1	Transmit FIFO Empty Interrupt Mask 0 – ssi_txe_intr interrupt is masked 1 – ssi_txe_intr interrupt is not masked

16.14.7.13 ISR

Table 16.374. Interrupt Status Register Description

Bit	Access	Function	POR Value	Description
31:6	N/A	Reserved	0	Reserved
5	R	MSTIS	0	Multi-Master Contention Interrupt Status. 0 – ssi_mst_intr interrupt not active after masking 1 – ssi_mst_intr interrupt is active after masking
4	R	RXFIS	0	Receive FIFO Full Interrupt Status 0 – ssi_rxf_intr interrupt is not active after masking 1 – ssi_rxf_intr interrupt is full after masking
3	R	RXOIS	0	Receive FIFO Overflow Interrupt Status 0 – ssi_rxo_intr interrupt is not active after masking 1 – ssi_rxo_intr interrupt is active after masking
2	R	RXUIS	0	Receive FIFO Underflow Interrupt Status 0 – ssi_rxu_intr interrupt is not active after masking 1 – ssi_rxu_intr interrupt is active after masking
1	R	TXOIS	0	Transmit FIFO Overflow Interrupt Status 0 – ssi_txo_intr interrupt is not active after masking 1 – ssi_txo_intr interrupt is active after masking
0	R	TXEIS	0	Transmit FIFO Empty Interrupt Status 0 – ssi_txe_intr interrupt is not active after masking 1 – ssi_txe_intr interrupt is active after masking

16.14.7.14 RISR

Table 16.375. RAW Interrupt Status Register Description

Bit	Access	Function	POR Value	Description
31:6	N/A	Reserved	0	Reserved
5	R	MSTIR	0	Multi-Master Contention Raw Interrupt Status. 0 – ssi_mst_intr interrupt is not active prior to asking 1 – ssi_mst_intr interrupt is active prior masking
4	R	RXFIR	0	Receive FIFO Full Raw Interrupt Status 0 – ssi_rxf_intr interrupt is not active prior to masking 1 – ssi_rxf_intr interrupt is active prior to masking
3	R	RXOIR	0	Receive FIFO Overflow Raw Interrupt Status 0 – ssi_rxo_intr interrupt is not active prior to masking 1 – ssi_rxo_intr interrupt is active prior masking
2	R	RXUIR	0	Receive FIFO Underflow Raw Interrupt Status 0 – ssi_rxu_intr interrupt is not active prior to masking 1 – ssi_rxu_intr interrupt is active prior to masking
1	R	TXOIR	0	Transmit FIFO Overflow Raw Interrupt Status 0 – ssi_txo_intr interrupt is not active prior to masking 1 – ssi_txo_intr interrupt is active prior masking

Bit	Access	Function	POR Value	Description
0	R	TXEIR	0	Transmit FIFO Empty Raw Interrupt Status 0 – ssi_txe_intr interrupt is not active prior to masking 1 – ssi_txe_intr interrupt is active prior masking

16.14.7.15 TXOICR

Table 16.376. Transmit FIFO Overflow Interrupt Clear Register Description

Bit	Access	Function	POR Value	Description
31:1	N/A	Reserved	0	Reserved
0	R	TXOICR	0	Clear Transmit FIFO Overflow Interrupt. This register reflects the status of the interrupt. A read from this register clears the ssi_txo_intr interrupt; writing has no effect.

16.14.7.16 RXOICR

Table 16.377. Receive FIFO Overflow Interrupt Clear Register Description

Bit	Access	Function	POR Value	Description
31:1	N/A	Reserved	0	Reserved
0	R	RXOICR	0	Clear Receive FIFO Overflow Interrupt. This register reflects the status of the interrupt. A read from this register clears the ssi_rxo_intr interrupt; writing has no effect.

16.14.7.17 RXUICR

Table 16.378. Receive FIFO Underflow Interrupt Clear Register Description

Bit	Access	Function	POR Value	Description
31:1	N/A	Reserved	0	Reserved
0	R	RXUICR	0	Clear Receive FIFO Underflow Interrupt. This register reflects the status of the interrupt. A read from this register clears the ssi_rxu_intr interrupt; writing has no effect.

16.14.7.18 MSTICR

Table 16.379. Multi-Primary Interrupt Clear Register Description

Bit	Access	Function	POR Value	Description
31:1	N/A	Reserved	0	Reserved
0	R	MSTICR	0	Clear Multi-Primary Contention Interrupt. This register reflects the status of the interrupt. A read from this register clears the ssi_mst_intr interrupt; writing has no effect.

16.14.7.19 ICR

Table 16.380. Interrupt Clear Register Description

Bit	Access	Function	POR Value	Description
31:1	N/A	Reserved	0	Reserved
0	R	ICR	0	Clear Interrupts. This register is set if any of the interrupts below are active. A read clears the ssi_txo_intr, ssi_rxu_intr, ssi_rxo_intr, and the ssi_mst_intr interrupts. Writing to this register has no effect.

16.14.7.20 DMACR

Table 16.381. DMA Control Register Description

Bit	Access	Function	POR Value	Description
31:2	N/A	Reserved	0	Reserved
1	R/W	TDMAE	0	Transmit DMA Enable. This bit enables/disables the transmit FIFO DMA channel. 0 – Transmit DMA disabled 1 – Transmit DMA enabled
0	R/W	RDMAE	0	Receive DMA Enable. This bit enables/disables the receive FIFO DMA channel 0 – Receive DMA disabled 1 – Receive DMA enabled

16.14.7.21 DMATDLR

Table 16.382. DMA Transmit Data Level Register Description

Bit	Access	Function	POR Value	Description
31:4	N/A	Reserved	0	Reserved
3:0	R/W	DMATDL	0	Transmit Data Level. This bit field controls the level at which a DMA request is made by the transmit logic. It is equal to the watermark level; that is, the dma_tx_req signal is generated when the number of valid data entries in the transmit FIFO is equal to or below this field value, and TDMAE = 1.

16.14.7.22 DMARDLR

Table 16.383. DMA Receive Data Level Register Description

Bit	Access	Function	POR Value	Description
31:4	N/A	Reserved	0	Reserved
3:0	R/W	DMARDL	0	Receive Data Level. This bit field controls the level at which a DMA request is made by the receive logic. The watermark level = DMARDL+1; that is, dma_rx_req is generated when the number of valid data entries in the receive FIFO is equal to or above this field value + 1, and RDMAE=1.

16.14.7.23 IDR

Table 16.384. Identification Register Description

Bit	Access	Function	POR Value	Description
31:0	R	IDCODE	0xFFFF_FFFF	Identification Code. This register contains the peripherals identification code.

16.14.7.24 SSI_COMP_VERSION

Table 16.385. SSI Component Version Register Description

Bit	Access	Function	POR Value	Description
31:0	R	SSI_COMP_VERSION	0x3430302a	Contains the hex representation of the component version.

16.14.7.25 DR

Table 16.386. Data Registers Description

Bit	Access	Function	POR Value	Description
31:0	R/W	DR	0	Data Register. When writing to this register, the user must right-justify the data. Read data are automatically right-justified. Read – Receive FIFO buffer Write – Transmit FIFO buffer

16.14.7.26 RXD_SAMPLE_DELAY

Table 16.387. RXD Sample Delay Register Description

Bit	Access	Function	POR Value	Description
31:8	N/A	Reserved		Reserved
7:0	R/W	RSD	0	Receive Data (rxd) Sample Delay. This register is used to delay the sample of the rxd input signal. Each value represents a single ssi_clk delay on the sample of the rxd signal. NOTE: If this register is programmed with a value that exceeds the depth of the internal shift registers (63), a zero (0) delay will be applied to the rxd sample.

16.14.7.27 SPI_CTRLR0

Table 16.388. SPI Control Register Description

Bit	Access	Function	POR Value	Description
31:15	R	Reserved	0	Reserved

Bit	Access	Function	POR Value	Description
14:11	R/W	WAIT_CYCLES	0	This bit defines the wait cycles in dual*/quad* mode between control frames transmit and data reception. Specified as number of SPI clock cycles. 0000 – No Wait Cycles 0001 – 1 Wait Cycle up to 1111 – 15 Wait Cycles
10	R	Reserved	0	Reserved
9:8	R/W	INST_L	0x2	Dual*/Quad* mode instruction length in bits. • 00 - 0 bit (No instruction) • 01 - 4 bits • 10 - 8 bits • 11 - 16 bits
7:6	R	Reserved	0	Reserved
5:2	R/W	ADDR_L	0x0	This bit defines length of address to be transmitted. The transfer begins only after these many bits are programmed into the FIFO. Address width = ADDR_L * 4 bits
1:0	R/W	TRANS_TYPE	0	Address and instruction transfer format. This bit selects whether ssi will transmit instruction/address either in Standard SPI mode or the SPI mode when the mode is specified in the CTRLR0.SPI_FRF field. • 00 - Instruction and Address will be sent in Standard SPI Mode. • 01 - Instruction will be sent in Standard SPI Mode and Address will be sent in the mode specified by CTRLR0.SPI_FRF. • 10 - Both Instruction and Address will be sent in the mode specified by CTRLR0.SPI_FRF. • 11 - Reserved.

16.15 SSI Secondary

16.15.1 General Description

The SSI Secondary is a programmable synchronous serial (SSI) peripheral. The SSI Secondary can connect to any serial-primary peripheral using one of the following interfaces:

- Motorola Serial Peripheral Interface (SPI)
- Texas Instruments Serial Protocol (SSP)
- National Semiconductor Microwire

16.15.2 Features

- Acts as a Serial Secondary
- DMA handshake present
- Independent masking of interrupts - Primary collision, transmit FIFO overflow, transmit FIFO empty, receive FIFO full, receive FIFO underflow, and receive FIFO overflow interrupts can all be masked independently.
- Works with Motorola SPI, Texas Instruments SSP and National Semiconductors Microwire
- Data Item size (4 to 16 bits) – Item size of each data transfer under the control of the programmer.
- Supported FIFO depth is 16 (Independent TX and RX FIFOs are present)
- Combined interrupt line for all interrupts
- Generates active high interrupts
- APB clock (pclk) and SSI secondary serial clock are identical

16.15.3 Functional Description

The MCU accesses data, control, and status information on the SSI Secondary through the APB interface. This may also interface with a DMA Controller using an optional set of DMA signals. The SSI Secondary can connect to any serial-Primary peripheral using one of the following interfaces:

- Motorola Serial Peripheral Interface (SPI): A four-wire, full- duplex serial protocol from Motorola. There are four possible combinations for the serial clock phase and polarity. The clock phase (SCPH) determines whether the serial transfer begins with the falling edge of the Secondary select signal or the first edge of the serial clock. The Secondary select line is held high when the SSI Primary is idle or disabled.
- Texas Instruments Serial Protocol (SSP): A four-wire, full-duplex serial protocol. The Secondary select line is used for SPI and Microwire protocols doubles as the frame indicator for the SSP protocol.
- National Semiconductor Microwire : A half-duplex serial protocol, which uses a control word transmitted from the serial Primary to the target serial Secondary.

FRF (frame format) bit field in the Control Register 0 (CTRLR0) is used to select which protocol is used. Frequency ratio restrictions between the bit-rate clock (sclk_in) and the

SSI Secondary peripheral clock (ssi_clk) is as follows:

Fssi_clk <= 4 × (maximum Fsclk_in)

16.15.4 SSI Interrupts

The SSI Secondary supports combined and individual interrupt requests, each of which can be masked. The combined interrupt request is the ORed result of all other interrupts after masking. All SSI Secondary interrupts are level interrupts and have the same active polarity level. This polarity level can be configured as active-high or active-low. The SSI Secondary interrupts are described as follows:

- Transmit FIFO Empty Interrupt (ssi_txe_intr) – Set when the transmit FIFO is equal to or below its threshold value and requires service to prevent an under-run. The threshold value, set through a software-programmable register, determines the level of transmit FIFO entries at which an interrupt is generated. This interrupt is cleared by hardware when data are written into the transmit FIFO buffer, bringing it over the threshold level.
- Transmit FIFO Overflow Interrupt (ssi_txo_intr) – Set when an APB access attempts to write into the transmit FIFO after it has been completely filled. When set, data written from the APB is discarded. This interrupt remains set until reading of the transmit FIFO overflow interrupt clear register (TXOICR).
- Receive FIFO Full Interrupt (ssi_rxf_intr) – Set when the receive FIFO is equal to or above its threshold value plus 1 and requires service to prevent an overflow. The threshold value, set through a software-programmable register, determines the level of receive FIFO entries at which an interrupt is generated. This interrupt is cleared by hardware when data are read from the receive FIFO buffer, bringing it below the threshold level.
- Receive FIFO Overflow Interrupt (ssi_rxo_intr) – Set when the receive logic attempts to place data into the receive FIFO after it has been completely filled. When set, newly received data are discarded. This interrupt remains set until reading of the receive FIFO overflow interrupt clear register (RXOICR).
- Receive FIFO Underflow Interrupt (ssi_rxu_intr) – Set when an APB access attempts to read from the receive FIFO when it is empty. When set, zeros are read back from the receive FIFO. This interrupt remains set until reading of the receive FIFO underflow interrupt clear register (RXUICR).
- Combined Interrupt Request (ssi_intr) – OR'ed result of all the above interrupt requests after masking. To mask this interrupt signal, mask all other SSI interrupt requests.

16.15.5 Programming Sequence for Data Transfer

This mode enables serial communication with primary peripheral devices. All serial transfers are initiated and controlled by the serial bus primary. The figure below shows an example of the SSI is configured as a serial Secondary in a single-Primary bus system. All data transfers to and from the serial Secondary are regulated on the serial clock line (sclk_in), driven from the serial Primary device. Data are propagated from the serial Secondary on one edge of the serial clock line and sampled on the opposite edge. The Secondary remains in an idle state until selected by the bus Primary. When not actively transmitting data, the Secondary must hold its txd line in a high impedance state to avoid interference with serial transfers to other Secondary devices.

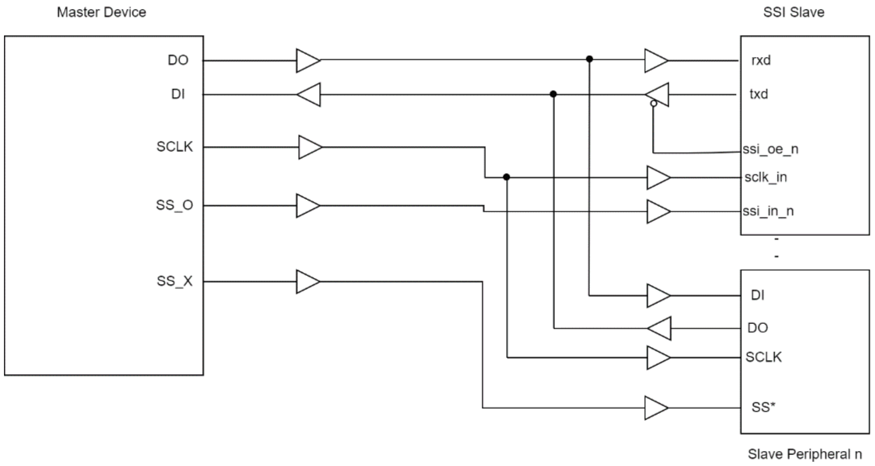


Figure 16.37. SSI Secondary in a Single-Primary Bus System

16.15.5.1 Secondary SPI and SSP Serial Transfers

A typical software flow for completing a continuous serial transfer from a serial Primary to the SSI

Secondary is described as follows:

1. If the SSI Secondary is enabled, disable it by writing 0 to SSIENR.
2. Set up the SSI Secondary control registers for the transfer. These registers can be set in any order.
   a. Write CTRLR0 (for SPI transfers SCPH and SCPOL must be set identical to the Primary device).
   b. Write TXFTLR and RXFTLR to set FIFO threshold levels.
   c. Write the IMR register to set up interrupt masks.
3. Enable the SSI by writing 1 to the SSIENR register.
4. If the transfer mode is transmit and receive (TMOD=2'b00) or transmit only (TMOD=2'b01), write data

   for transmission to the Primary into the transmit FIFO (Write DR).

   If the transfer mode is receive only (TMOD=2'b10), there is no need to write data into the transmit

   FIFO; the current value in the transmit shift register is re-transmitted.
5. The SSI Secondary is now ready for the serial transfer. The transfer begins when the SSI

   Secondary is selected by a serial-Primary device.
6. When the transfer is underway, the BUSY status can be polled to return the transfer status. If a

   transmit FIFO empty interrupt request is made, write the transmit FIFO (write DR). If a receive FIFO

   full interrupt request is made, read the receive FIFO (read DR).
7. The transfer ends when the serial Primary removes the select input to the SSI Secondary. When the

   transfer is completed, the BUSY status is reset to 0.
8. If the transfer mode is not transmit only (TMOD != 01), read the receive FIFO until empty.
9. Disable the SSI by writing 0 to SSIENR.

The figure below shows a typical software flow for a SSI Secondary SPI or SSP serial transfer.

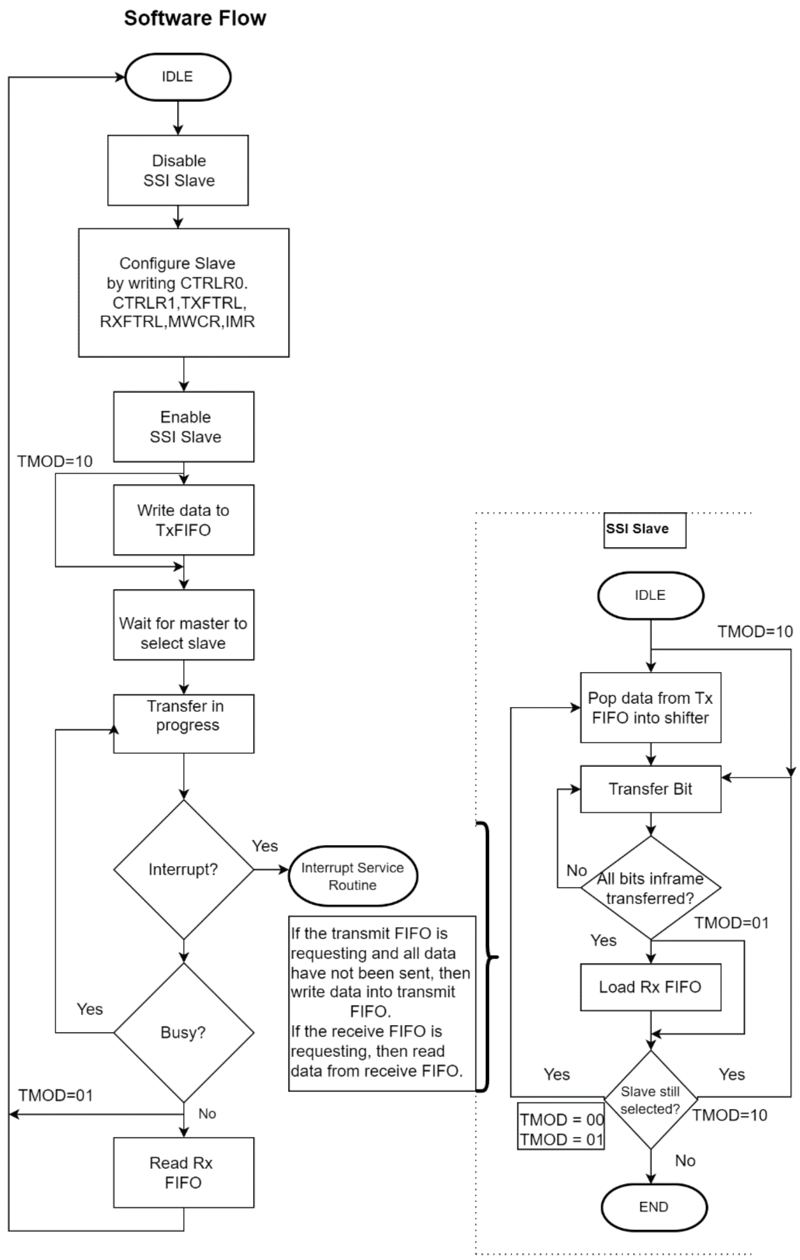

Software Flow
IDLE
Disable SSI Slave
Configure Slave by writing CTRLR0. CTRLR1,TXFTRL, RXFTRL,MWCR,IMR
Enable SSI Slave
TMOD=10
Write data to TxFIFO
Wait for master to select slave
Transfer in progress
Interrupt?
Yes
Interrupt Service Routine
If the transmit FIFO is requesting and all data have not been sent, then write data into transmit FIFO.
If the receive FIFO is requesting, then read data from receive FIFO.
Busy?
Yes
TMOD=01
No
Read Rx FIFO
SSI Slave
IDLE
TMOD=10
Pop data from Tx FIFO into shifter
Transfer Bit
No
All bits inframe transferred?
TMOD=01
Yes
Load Rx FIFO
Yes
Yes
Slave still selected?
TMOD = 00
TMOD = 01
TMOD=10
No
END

Figure 16.38. SSI Secondary SPI/SSP Transfer Flow

16.15.5.2 Secondary Microwire Serial Transfers

In SSI Secondary mode, the Microwire protocol operates in much the same way as the SPI protocol. There is no decode of the control frame by the SSI Secondary device.

16.15.6 Register Summary

Base Address: 0x4501_0000

Table 16.389. Register Summary Table

Register Name	Offset	Description
Section 16.15.7.1 CTRLR0	0x00	Control Register 0
Section 16.15.7.2 SSIENR	0x08	SSI Enable Register
Section 16.15.7.3 MWCR	0x0C	Microwire Control Register
Section 16.15.7.4 TXFTLR	0x18	Transmit FIFO Threshold Level Register
Section 16.15.7.5 RXFTLR	0x1C	Receive FIFO Threshold Level Register
Section 16.15.7.6 TXFLR	0x20	Transmit FIFO Level Register
Section 16.15.7.7 RXFLR	0x24	Receive FIFO Level Register
Section 16.15.7.8 SR	0x28	Status Register
Section 16.15.7.9 IMR	0x2C	Interrupt Mask Register
Section 16.15.7.10 ISR	0x30	Interrupt Status Register
Section 16.15.7.11 RISR	0x34	Raw Interrupt Status Register
Section 16.15.7.12 TXOICR	0x38	Transmit FIFO Overflow Interrupt Clear Register
Section 16.15.7.13 RXOICR	0x3C	Receive FIFO Overflow Interrupt Clear Register
Section 16.15.7.14 RXUICR	0x40	Receive FIFO Underflow Interrupt Clear Register
Section 16.15.7.15 MSTICR	0x44	Multi-Master (Master) Interrupt Clear Register
Section 16.15.7.16 ICR	0x48	Interrupt Clear Register
Section 16.15.7.17 DMACR	0x4C	DMA Control Register
Section 16.15.7.18 DMATDLR	0x50	DMA Transmit Data Level Register
Section 16.15.7.19 DMARDLR	0x54	DMA Receive Data Level Register
Section 16.15.7.20 IDR	0x58	Identification Register
Section 16.15.7.21 SSI_COMP_VERSION	0x5C	SSI Component Version
Section 16.15.7.22 DR	0x60-0xEC	Data Registers

16.15.7 Register Description

Legend: R = Read-only, W = Write-only, R/W = Read/Write, N/A = Reserved

16.15.7.1 CTRLR0

Table 16.390. Control Register 0 Description

Bit	Access	Function	POR Value	Description
31:16	N/A	Reserved	0	Reserved
15:12	R/W	CFS	0	Control Frame Size. Selects the length of the control word for the Microwire frame format. Ranges : 0000 -> 1 bit word control to 1111 -> 16 bit word control
11	R/W	SRL	0	Shift Register Loop used for testing purposes only. When internally active, connects the transmit shift register output to the receive shift register input. 0 – Normal Mode Operation 1 – Test Mode Operation Note : When SSI Secondary is configured in loop back mode, the ss_in_n and ssi_clk signals must be provided by an external source.
10	R/W	SLV_OE	0	Secondary Output Enable. This bit enables or disables the setting of the ssi_oe_n output from the SSI Secondary serial Secondary. 0 –- Secondary txd is enabled 1 –- Secondary txd is disabled
9:8	R/W	TMOD	0	Transfer Mode. Selects the mode of transfer for serial communication. 00 –- Transmit & Receive 01 –- Transmit Only 10 –- Receive Only
7	R/W	SCPOL	0	Serial Clock Polarity. Valid when the frame format (FRF) is set to Motorola SPI. 0 – Inactive state of serial clock is low 1 – Inactive state of serial clock is high
6	R/W	SCPH	0	Serial Clock Phase. Valid when the frame format (FRF) is set to Motorola SPI. When SCPH = 0, data are captured on the first edge of the serial clock. When SCPH = 1, the serial clock starts toggling one cycle after the Secondary select line is activated, and data are captured on the second edge of the serial clock. 0: Serial clock toggles in middle of first data bit 1: Serial clock toggles at start of first data bit
5:4	R/W	FRF	0	Frame Format. Selects which serial protocol transfers the data. 00 –- Motorola SPI 01 –- Texas Instruments SSP 10 –- National Semiconductors Microwire 11 –- Reserved
3:0	R/W	DFS	0x7	Data Frame Size. Selects the data frame length. Range : 0011 -> 4 bit to 1111 -> 16 bit

16.15.7.2 SSIENR

Table 16.391. SSI Enable Register Description

Bit	Access	Function	POR Value	Description
31:1	N/A	Reserved	0	Reserved

Bit	Access	Function	POR Value	Description
0	R/W	SSI_EN	0	SSI Enable. Enables and disables all SSI Secondary operations. When disabled, all serial transfers are halted immediately. Transmit and receive FIFO buffers are cleared when the device is disabled.

16.15.7.3 MWCR

Table 16.392. Microwire Control Register Description

Bit	Access	Function	POR Value	Description
2	N/A	Reserved	0	Reserved
1	R/W	MDD	0	Microwire Control. When this bit is set to 0, the data word is received by the SSI MacroCell from the external serial device. When this bit is set to 1, the data word is transmitted from the SSI MacroCell to the external serial device.
0	R/W	MWMOD	0	Microwire Transfer Mode. 0 – non-sequential transfer 1 – sequential transfer When sequential mode is used, only one control word is needed to transmit or receive a block of data words. When non-sequential mode is used, there must be a control word for each data word that is transmitted or received.

16.15.7.4 TXFTLR

Table 16.393. Transmit FIFO Threshold Level Register Description

Bit	Access	Function	POR Value	Description
31:4	N/A	Reserved	0	Reserved
3:0	R/W	TFT	0	Transmit FIFO Threshold. Controls the level of entries (or below) at which the transmit FIFO controller triggers an interrupt. If this field is set to a value greater than or equal to the depth of the FIFO, this field is not written and retains its current value. When the number of transmit FIFO entries is less than or equal to this value, the transmit FIFO empty interrupt is triggered.

16.15.7.5 RXFTLR

Table 16.394. Receive FIFO Threshold Level Register

Bit	Access	Function	POR Value	Description
31:4	N/A	Reserved	0	Reserved
3:0	R/W	RFT	0	Receive FIFO Threshold. Controls the level of entries (or below) at which the receive FIFO controller triggers an interrupt. If this field is set to a value greater than or equal to the depth of the FIFO, this field is not written and retains its current value. When the number of receive FIFO entries is less than or equal to this value + 1, the receive FIFO full interrupt is triggered.

16.15.7.6 TXFLR

Table 16.395. Transmit FIFO Level Register Description

Bit	Access	Function	POR Value	Description
31:4	N/A	Reserved	0	Reserved
4:0	R	TXTFL	0	Transmit FIFO Level. Contains the number of valid data entries in the transmit FIFO.

16.15.7.7 RXFLR

Table 16.396. Receive FIFO Level Register Description

Bit	Access	Function	POR Value	Description
31:4	N/A	Reserved	0	Reserved
4:0	R	RXTFL	0	Receive FIFO Level. Contains the number of valid data entries in the receive FIFO.

16.15.7.8 SR

Table 16.397. Status Register Description

Bit	Access	Function	POR Value	Description
31:6	N/A	Reserved	0	Reserved
5	R	TXE	0	Transmission Error. This bit is cleared when read. 0 – No error 1 – Transmission error
4	R	RFF	0	Receive FIFO Full. 0 – Receive FIFO is not full 1 – Receive FIFO is full

Bit	Access	Function	POR Value	Description
3	R	RFNE	0	Receive FIFO Not Empty. 0 – Receive FIFO is empty 1 – Receive FIFO is not empty This bit can be polled by software to completely empty the receive FIFO.
2	R	TFE	1	Transmit FIFO Empty. 0 – Transmit FIFO is not empty 1 – Transmit FIFO is empty This bit field does not request an interrupt.
1	R	TFNF	1	Transmit FIFO Not Full. 0 – Transmit FIFO is full 1 – Transmit FIFO is not full
0	R	BUSY	0	SSI Busy Flag. 0 – SSI is idle or disabled 1 – SSI is actively transferring data

16.15.7.9 IMR

Table 16.398. Interrupt Mask Register Description

Bit	Access	Function	POR Value	Description
31:5	N/A	Reserved	0	Reserved
4	R/W	RXFIM	1	Receive FIFO Full Interrupt Mask 0 – ssi_rxf_intr interrupt is masked 1 – ssi_rxf_intr interrupt is not masked
3	R/W	RXOIM	1	Receive FIFO Overflow Interrupt Mask 0 – ssi_rxo_intr interrupt is masked 1 – ssi_rxo_intr interrupt is not masked
2	R/W	RXUIM	1	Receive FIFO Underflow Interrupt Mask 0 – ssi_rxu_intr interrupt is masked 1 – ssi_rxu_intr interrupt is not masked
1	R/W	TXOIM	1	Transmit FIFO Overflow Interrupt Mask 0 – ssi_txo_intr interrupt is masked 1 – ssi_txo_intr interrupt is not masked
0	R/W	TXEIM	1	Transmit FIFO Empty Interrupt Mask 0 – ssi_txe_intr interrupt is masked 1 – ssi_txe_intr interrupt is not masked

16.15.7.10 ISR

Table 16.399. Interrupt Status Register Description

Bit	Access	Function	POR Value	Description
31:5	N/A	Reserved	0	Reserved
4	R	RXFIS	0	Receive FIFO Full Interrupt Status 0 – ssi_rxf_intr interrupt is not active after masking 1 – ssi_rxf_intr interrupt is full after masking

Bit	Access	Function	POR Value	Description
3	R	RXOIS	0	Receive FIFO Overflow Interrupt Status 0 – ssi_rxo_intr interrupt is not active after masking 1 – ssi_rxo_intr interrupt is active after masking
2	R	RXUIS	0	Receive FIFO Underflow Interrupt Status 0 – ssi_rxu_intr interrupt is not active after masking 1 – ssi_rxu_intr interrupt is active after masking
1	R	TXOIS	0	Transmit FIFO Overflow Interrupt Status 0 – ssi_txo_intr interrupt is not active after masking 1 – ssi_txo_intr interrupt is active after masking
0	R	TXEIS	0	Transmit FIFO Empty Interrupt Status 0 – ssi_txe_intr interrupt is not active after masking 1 – ssi_txe_intr interrupt is active after masking

16.15.7.11 RISR

Table 16.400. RAW Interrupt Status Register Description

Bit	Access	Function	POR Value	Description
31:5	N/A	Reserved	0	Reserved
4	R	RXFIR	0	Receive FIFO Full Raw Interrupt Status 0 – ssi_rxf_intr interrupt is not active prior to masking 1 – ssi_rxf_intr interrupt is active prior to masking
3	R	RXOIR	0	Receive FIFO Overflow Raw Interrupt Status 0 – ssi_rxo_intr interrupt is not active prior to masking 1 – ssi_rxo_intr interrupt is active prior masking
2	R	RXUIR	0	Receive FIFO Underflow Raw Interrupt Status 0 – ssi_rxu_intr interrupt is not active prior to masking 1 – ssi_rxu_intr interrupt is active prior to masking
1	R	TXOIR	0	Transmit FIFO Overflow Raw Interrupt Status 0 – ssi_txo_intr interrupt is not active prior to masking 1 – ssi_txo_intr interrupt is active prior masking
0	R	TXEIR	0	Transmit FIFO Empty Raw Interrupt Status 0 – ssi_txe_intr interrupt is not active prior to masking 1 – ssi_txe_intr interrupt is active prior masking

16.15.7.12 TXOICR

Table 16.401. Transmit FIFO Overflow Interrupt Clear Register Description

Bit	Access	Function	POR Value	Description
31:1	N/A	Reserved	0	Reserved
0	R	TXOICR	0	Clear Transmit FIFO Overflow Interrupt. This register reflects the status of the interrupt. A read from this register clears the ssi_txo_intr interrupt; writing has no effect.

16.15.7.13 RXOICR

Table 16.402. Receive FIFO Overflow Interrupt Clear Register Description

Bit	Access	Function	POR Value	Description
31:1	N/A	Reserved	0	Reserved
0	R	RXOICR	0	Clear Receive FIFO Overflow Interrupt. This register reflects the status of the interrupt. A read from this register clears the ssi_rxo_intr interrupt; writing has no effect.

16.15.7.14 RXUICR

Table 16.403. Receive FIFO Underflow Interrupt Clear Register Description

Bit	Access	Function	POR Value	Description
31:1	N/A	Reserved	0	Reserved
0	R	RXUICR	0	Clear Receive FIFO Underflow Interrupt. This register reflects the status of the interrupt. A read from this register clears the ssi_rxu_intr interrupt; writing has no effect.

16.15.7.15 MSTICR

Table 16.404. Multi-Master Interrupt Clear Register Description

Bit	Access	Function	POR Value	Description
31:1	N/A	Reserved	0	Reserved
0	R	MSTICR	0	Clear Multi-Master Contention Interrupt. This register reflects the status of the interrupt. A read from this register clears the ssi_mst_intr interrupt; writing has no effect.

16.15.7.16 ICR

Table 16.405. Interrupt Clear Register Description

Bit	Access	Function	POR Value	Description
31:1	N/A	Reserved	0	Reserved
0	R	ICR	0	Clear Interrupts. This register is set if any of the interrupts below are active. A read clears the ssi_txo_intr, ssi_rxu_intr, ssi_rxo_intr, and the ssi_mst_intr interrupts. Writing to this register has no effect.

16.15.7.17 DMACR

Table 16.406. DMA Control Register Description

Bit	Access	Function	POR Value	Description
31:2	N/A	Reserved	0	Reserved
1	R/W	TDMAE	0	Transmit DMA Enable. This bit enables/disables the transmit FIFO DMA channel. 0 – Transmit DMA disabled 1 – Transmit DMA enabled
0	R/W	RDMAE	0	Receive DMA Enable. This bit enables/disables the receive FIFO DMA channel 0 – Receive DMA disabled 1 – Receive DMA enabled

16.15.7.18 DMATDLR

Table 16.407. DMA Transmit Data Level Register Description

Bit	Access	Function	POR Value	Description
31:4	N/A	Reserved	0	Reserved
3:0	R/W	DMATDL	0	Transmit Data Level. This bit field controls the level at which a DMA request is made by the transmit logic. It is equal to the watermark level; that is, the dma_tx_req signal is generated when the number of valid data entries in the transmit FIFO is equal to or below this field value, and TDMAE = 1.

16.15.7.19 DMARDLR

Table 16.408. DMA Receive Data Level Register Description

Bit	Access	Function	POR Value	Description
31:4	N/A	Reserved	0	Reserved
3:0	R/W	DMARDL	0	Receive Data Level. This bit field controls the level at which a DMA request is made by the receive logic. The watermark level = DMARDL+1; that is, dma_rx_req is generated when the number of valid data entries in the receive FIFO is equal to or above this field value + 1, and RDMAE=1.

16.15.7.20 IDR

Table 16.409. Identification Register Description

Bit	Access	Function	POR Value	Description
31:0	R	IDCODE	0xFFFF_FFFF	Identification Code. This register contains the peripherals identification code.

16.15.7.21 SSI_COMP_VERSION

Table 16.410. SSI Component Version Register Description

Bit	Access	Function	POR Value	Description
31:0	R	SSI_COMP_VERSION	0x3430302a	Contains the hex representation of the component version.

16.15.7.22 DR

Table 16.411.

Bit	Access	Function	POR Value	Description
31:0	R/W	DR	0	Data Register. When writing to this register, the user must right-justify the data. Read data are automatically right-justified. Read – Receive FIFO buffer Write – Transmit FIFO buffer

16.16 UART

16.16.1 General Description

There are three UART controllers - two in the MCU HP peripherals (USART0, UART1) and one in the MCU ULP subsystem (ULP_UART). One of the controllers in the MCU HP peripherals (USART0) supports UART.

The UART is used for serial communication with:

- Peripherals
- Modems (data carrier equipment, DCE)
- Data sets

16.16.2 Features

Each of these UART controllers support the following features:

- Multi-drop RS485 interface support
- 5, 6, 7 and 8-bit character encoding with Even, Odd and No Parity
- 1, 1.5 (only with 5 bit character encoding) and 2 stop bits
- Hardware Auto flow control (RTS/CTS)
- Programmable baud rate as calculated by the following: baud rate = (serial clock frequency)/(16× divisor). Maximum supported baud rate is 921,600 (around 0.92Mbps) with 118MHz UART input clock.
- Supported standard baud rates are 300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200, 230400, 460800, 921600.

The UART controllers also support additional features listed below, which help in achieving better performance and reduce the burden on the processor:

- Programmable fractional baud rate support
- Programmable baud rate supporting up to 5 Mbps
- Programmable FIFO thresholds with maximum FIFO depth of 16 and support for DMA
- Prioritized interrupt identification

The UART controller in the MCU ULP subsystem (ULP_UART) supports the following additional power-save features:

- After the DMA is programmed in PS2 state for UART transfers, the MCU can switch to PS1 state (processor is shutdown) while the UART controller continues with the data transfer
- In PS1 state (ULP Peripheral mode) the UART controller completes the data transfer and, triggered by the Peripheral Interrupt, shifts to the PS2 active state

Note:

UART RTS and CTS both needs to be enabled. Standalone RTS/CTS is not supported.

In ULP_UART Multi-drop RS485 and Auto flow control (RTS/CTS) features are not supported

16.16.3 Functional Description

The figure below illustrates the functional block diagram. It contains APB interface, Register block, Modem sync block, Baud rate generator, Serial transmitter and serial receiver. The UART contains registers that control:

- Character length
- Baud rate
- Parity generation/checking
- Interrupt generation

Although there is only one interrupt output signal (intr) from the UART, there are several prioritized interrupt types that can be responsible for its assertion. Each of the interrupt types can be separately enabled or disabled by the control registers.

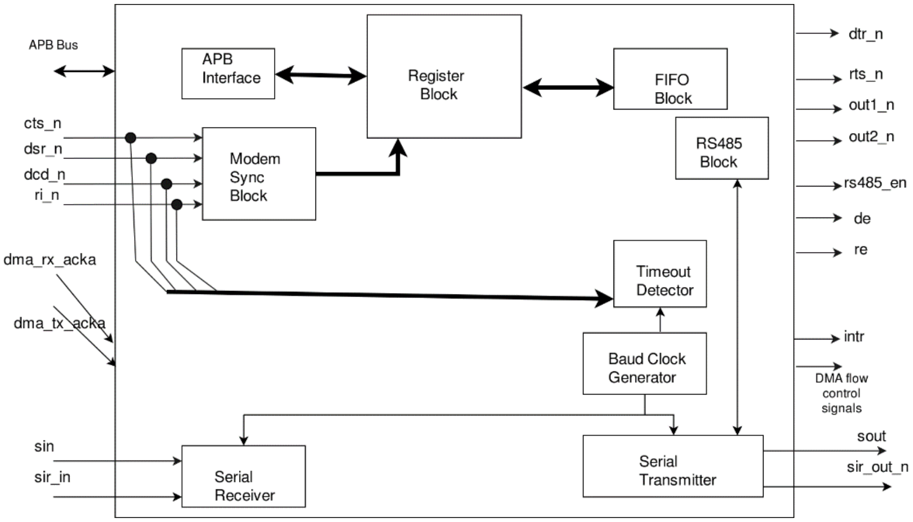


Figure 16.39. UART Functional Block Diagram

Because the serial communication between the UART and a selected device is asynchronous, additional bits (start and stop) are added to the serial data to indicate the beginning and end. Utilizing these bits allows two devices to be synchronized. This structure of serial data accompanied by start and stop bits is referred to as a character, as shown in the figure below. An additional parity bit can be added to the serial character. This bit appears after the last data bit and before the stop bits, which can be 1, 1.5, or 2 bits, in the character structure in order to provide the UART with the ability to perform simple error checking on the received data. The UART Line Control Register (LCR) is used to control the serial character characteristics. To ensure stability on the line, the receiver samples the serial input data at approximately the midpoint of the Bit Time once the start bit has been detected. Because the exact number of baud clocks is known for which each bit was transmitted, calculating the midpoint for sampling is not difficult; that is, every sixteen baud clocks after the midpoint sample of the start bit.

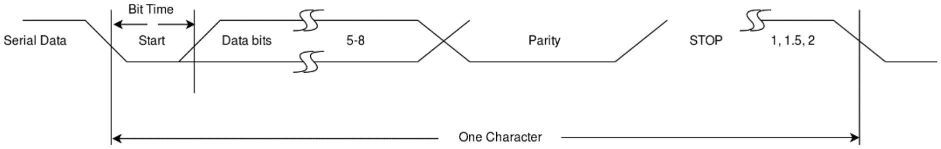


Figure 16.40. UART Serial Data Format

16.16.4 UART Transmission

The UART can be configured to have 9-bit data transfer in both transmit and receive mode. The 9th bit in the character appears after the 8th bit and before the parity bit in the character.

Configuration of the UART for 9-bit data transfer does the following:

- LCR_EXT[0] bit is used to enable or disable the 9-bit data transfer.
- LCR_EXT[1] bit is used to choose between hardware and software based address match in the case of receive.
- LCR_EXT[2] bit is used to enable to send the address in the case of transmit.
- LCR_EXT[3] bit is used to choose between hardware and software based address transmission.
- TAR and RAR registers are used to transmit address and to match the received address, respectively.
- THR, RBR, STHR and SRBR registers are of 9-bit which is used to do the data transfers in 9-bit mode.
- LSR[8] bit is used to indicate the address received interrupt.

16.16.4.1 Transmit Modes

UART supports two types of transmit modes:

- Transmit Mode 0 (when (LCR_EXT[3]) is set to 0)
- Transmit Mode 1 (when (LCR_EXT[3]) is set to 1)

Transmit Mode 0

In transmit mode 0, the address is programmed in the Transmit Address Register (TAR) register and data is written into the Transmit Holding Register (THR) or the Shadow Transmit Holding Register (STHR). The 9th bit of the THR and STHR register is not applicable in this mode. The figure below illustrates the transmission of address and data based on SEND_ADDR (LCR_EXT[2]), Halt Tx, and TxFIFO/THR empty conditions.

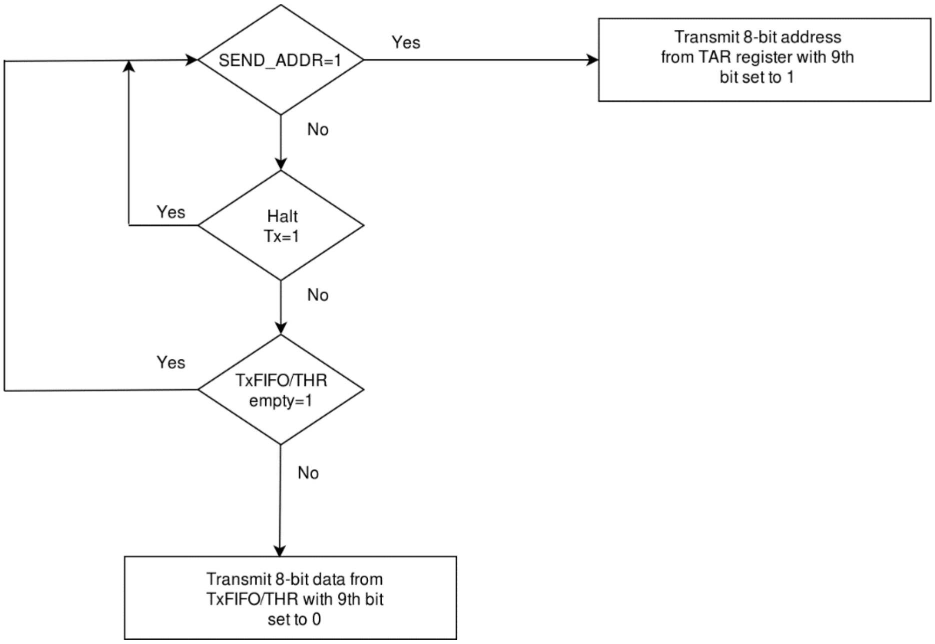

SEND_ADDR=1
Yes
Transmit 8-bit address from TAR register with 9th bit set to 1
No
Yes
Halt Tx=1
No
Yes
TxFIFO/THR empty=1
No
Transmit 8-bit data from TxFIFO/THR with 9th bit set to 0

Figure 16.41. UART Auto Address Transmit Flow Chart

The address of the target secondary to which the data is to be transmitted is programmed in the TAR register. Must enable the SEND_ADDR (LCR_EXT[2]) bit to transmit the target secondary address present in the TAR register on the serial UART line with 9th data bit set to 1 to indicate that the address is being sent to the secondary. The UART clears the SEND_ADDR bit after the address character starts transmitting on the UART line. The data required to transmit to the target secondary is programmed through Transmit Holding Register (THR). The data is transmitted on the UART line with 9th data bit set to 0 to indicate data is being sent to the secondary. If the application is required to fill the data bytes in the TxFIFO before sending the address on the UART line (before setting LCR_EXT[2]=1), then it is recommended to set the “Halt Tx” to 1 such that UART does not start sending out the data in the TxFIFO as data byte. Once the TxFIFO is filled, then program SEND_ADDR (LCR_EXT[2]) to 1 and then set "Halt Tx" to 0.

Transmit Mode 1

In transmit mode 1, THR and STHR registers are of 9-bit wide and both address and data are programmed through the THR and STHR registers. The UART does not differentiate between address and data, and both are taken from the TxFIFO. The SEND_ADDR (LCR_EXT[2]) bit and Transmit address register (TAR) are not applicable in this mode. The software must pack the 9th bit with 1/0 depending on whether address/data has to be sent.

16.16.5 UART Reception

16.16.5.1 Receive Modes

The UART supports two receive modes:

- Hardware Address Match Receive Mode (when ADDR_MATCH (LCR_EXT[1]) is set to 1)
- Software Address Match Receive Mode (when ADDR_MATCH (LCR_EXT[1]) is set to 0)

Hardware Address Match Receive Mode

In the hardware address match receive mode, the UART matches the received character with the address programmed in the Receive Address register (RAR), if the 9th bit of the received character is set to 1. If the received address is matched with the programmed address in RAR register, then subsequent data bytes (with 9th bit set to 0) are pushed into the RxFIFO. If the address matching fails, then UART controller discards further data characters until a matching address is received. The figure below illustrates the flow chart for the reception of data bytes based on the address matching feature.

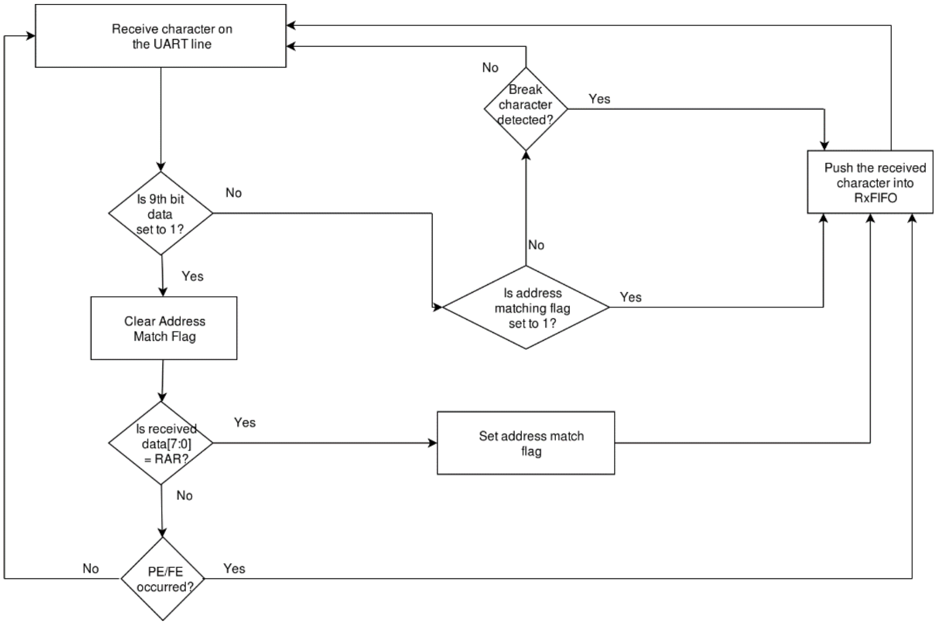

Receive character on the UART line
No
Break character detected?
Yes
Push the received character into RxFIFO
Is 9th bit data set to 1?
No
No
Yes
Is address matching flag set to 1?
Yes
Clear Address Match Flag
Is received data[7:0] = RAR?
Yes
Set address match flag
No
No
PE/FE occurred?
Yes

Figure 16.42. UART Hardware Address Match Receive Mode

UART receives the character irrespective of whether the 9th bit data is set to 1. If 9th bit of the received character is set to 1, then it clears internal address match flag and then compares the received 8-bit character information with the address programmed in the RAR register. If the received address character matches with the address programmed in the RAR register, then the address match flag is set to 1 and the received character is pushed to the RxFIFO in FIFO-mode or to RBR register in non-FIFO mode and the ADDR_RCVD bit in LSR register is set to indicate that the address has been received. In case of parity or if a framing error is found in the received address character and if the address is not matched with the RAR register, then the received address character is still pushed to RxFIFO or RBR register with ADDR_RCVD and PE/FE error bit set to 1. The subsequent data bytes (9th bit of received character is set to 0) are pushed to the Rx_FIFO in FIFO mode or to the RBR register in non-FIFO mode until the new address character is received. If any break character is received, UART treats it as a special character and pushes to the RxFIFO or RBR register based on the FIFO_MODE irrespective of address match flag.

Software Address Match Receive Mode

In this mode of operation, the UART does not perform the address matching for the received address character (9th bit data set to 1) with the RAR register. The UART always receives the 9-bit data and pushes in to RxFIFO in FIFO mode or to the RBR register in non-FIFO mode. The user must compare the address whenever address byte is received and indicated through ADDR_RCVD bit in the Line Status register. The user can flush/reset the RxFIFO in case of address not matched through 'RCVR FIFO Reset' bit in FIFO control register (FCR).

16.16.6 Interrupts

Assertion of the UART interrupt output signal (intr) a positive-level interrupt occurs whenever one of the several prioritized interrupt types are enabled and active. When an interrupt occurs, the primary accesses the IIR register. The following interrupt types can be enabled with the IER register:

- Receiver Error
- Receiver Data Available
- Character Timeout (in FIFO mode only)
- Transmitter Holding Register Empty at/below threshold (in Programmable THRE interrupt mode)
- Modem Status
- Busy Detect Indication

16.16.7 Auto Flow Control

The UART can be configured to have a 16750-compatible Auto RTS and Auto CTS serial data flow control mode available; if FIFOs are not implemented, this mode cannot be selected. When Auto Flow Control is not selected, none of the corresponding logic is implemented and the mode cannot be enabled, reducing overall gate counts. When Auto Flow Control mode is selected, it can be enabled with the Modem Control Register (MCR[5]). Auto RTS and Auto CTS are described as follows:

16.16.7.1 Auto RTS

It becomes active when the following occurs:

- RTS (MCR[1] bit and MCR[5]bit are both set)
- FIFOs are enabled (FCR[0]) bit is set)
- SIR mode is disabled (MCR[6] bit is not set)

When Auto RTS is enabled, the rts_n output is forced inactive (high) when the receiver FIFO level reaches the threshold set by FCR[7:6], but only if the RTS flow-control trigger is disabled. Otherwise, the rts_n output is forced inactive (high) when the FIFO is almost full, where “almost full” refers to two available slots in the FIFO. When rts_n is connected to the cts_n input of another UART device, the other UART stops sending serial data until the receiver FIFO has available space; that is, until it is completely empty. Once the receiver FIFO becomes completely empty by reading the Receiver Buffer Register (RBR), rts_n again becomes active (low), signalling the other UART to continue sending data.

16.16.7.2 Auto CTS

It becomes active when the following occurs:

- AFCE (MCR[5] bit = 1)
- FIFOs are enabled through FIFO Control Register FCR[0] bit
- SIR mode is disabled (MCR[6] bit = 0)

When Auto CTS is enabled (active), the UART transmitter is disabled whenever the cts_n input becomes inactive (high); this prevents overflowing the FIFO of the receiving UART. If the cts_n input is not inactivated before the middle of the last stop bit, another character is transmitted before the transmitter is disabled. While the transmitter is disabled, the transmitter FIFO can still be written to, and even overflowed. Therefore, when using this mode, the following happens:

- UART status register can be read to check if transmit FIFO is full (USR[1] set to 0)
- Current FIFO level can be read using TFL register
- Programmable THRE Interrupt mode must be enabled to access “FIFO full” status using Line Status Register (LSR)

When using the “FIFO full” status, software can poll this before each write to the Transmitter FIFO. When the cts_n input becomes active (low) again, transmission resumes.

Note: UART RTS and CTS both needs to be enabled. Standalone RTS/CTS is not supported

16.16.8 Register Summary

UART Base Address: 0x4502_0000

ULP_UART Base Address: 0x2404_1800

Table 16.412. Register Summary Table

Register Name	Offset	Description
Section 16.16.9.1 UART RECEIVE BUFFER REGISTER	0x00	Receive Buffer Register
Section 16.16.9.2 UART TRANSMIT HOLDING REGISTER	0x00	Transmit Holding Register
Section 16.16.9.3 UART DIVISOR LATCH LOW REGISTER	0x00	Divisor Latch (Low) Register
Section 16.16.9.4 UART DIVISOR LATCH HIGH REGISTER	0x04	Divisor Latch (High) Register
Section 16.16.9.5 UART INTERRUPT ENABLE REGISTER	0x04	Interrupt Enable Register
Section 16.16.9.6 UART INTERRUPT IDENTIFICATION REGISTER	0x08	Interrupt Identification Register
Section 16.16.9.7 UART FIFO CONTROL REGISTER	0x08	FIFO Control Register
Section 16.16.9.8 UART LINE CONTROL REGISTER	0x0C	Line Control Register
Section 16.16.9.9 UART MODEM CONTROL REGISTER	0x10	Modem Control Register
Section 16.16.9.10 UART LINE STATUS REGISTER	0x14	Line Status Register
Section 16.16.9.11 UART MODEM STATUS REGISTER	0x18	Modem Status Register
Section 16.16.9.12 UART SCRATCHPAD REGISTER	0x1C	Scratchpad Register
Section 16.16.9.13 UART LOW POWER DIVISOR LATCH (LOW) REGISTER	0x20	Low Power Divisor Latch (Low) Register
Section 16.16.9.14 UART LOW POWER DIVISOR LATCH (HIGH) REGISTER	0x24	Low Power Divisor Latch (High) Register
Reserved	0x28-0x2C	–
Section 16.16.9.15 UART SHADOW RECEIVE BUFFER REGISTER	0x30-0x6C	Shadow Receive Buffer Register
Section 16.16.9.16 UART SHADOW TRANSMIT HOLDING REGISTER	0x30-0x6C	Shadow Transmit Holding Register
Section 16.16.9.17 UART FIFO ACCESS REGISTER	0x70	FIFO Access Register
Section 16.16.9.18 UART TRANSMIT FIFO READ REGISTER	0x74	Transmit FIFO Read Register

Register Name	Offset	Description
Section 16.16.9.19 UART RECEIVE FIFO WRITE REGISTER	0x78	Receive FIFO Write Register
Section 16.16.9.20 UART STATUS REGISTER	0x7C	UART Status Register
Section 16.16.9.21 UART TRANSMIT FIFO LEVEL REGISTER	0x80	Transmit FIFO Level Register
Section 16.16.9.22 UART RECEIVE FIFO LEVEL REGISTER	0x84	Receive FIFO Level Register
Section 16.16.9.23 UART SOFTWARE RESET REGISTER	0x88	Software Reset Register
Section 16.16.9.24 UART SHADOW REQUEST TO SEND REGISTER	0x8C	Shadow Request to Send Register
Section 16.16.9.25 UART SHADOW BREAK CONTROL REGISTER	0x90	Shadow Break Control Register
Section 16.16.9.26 UART SHADOW DMA MODE REGISTER	0x94	Shadow DMA Mode Register
Section 16.16.9.27 UART SHADOW FIFO ENABLE REGISTER	0x98	Shadow FIFO Enable Register
Section 16.16.9.28 UART SHADOW RCVR TRIGGER REGISTER	0x9C	Shadow RCVR Trigger Register
Section 16.16.9.29 UART SHADOW TX EMPTY TRIGGER REGISTER	0xA0	Shadow TX Empty Trigger Register
Section 16.16.9.30 UART HALT TX REGISTER	0xA4	Halt TX Register
Section 16.16.9.31 UART DMA SOFTWARE ACKNOWLEDGE REGISTER	0xA8	DMA Software Acknowledge Register
Section 16.16.9.32 UART TRANSCEIVER CONTROL REGISTER	0xAC	Transceiver Control Register
Section 16.16.9.33 UART DRIVER OUTPUT ENABLE REGISTER	0xB0	Driver Output Enable Register
Section 16.16.9.34 UART RECEIVER OUTPUT ENABLE REGISTER	0xB4	Receiver Output Enable Register
Section 16.16.9.35 UART DRIVER OUTPUT ENABLE TIMING REGISTER	0xB8	Driver Output Enable Timing Register
Section 16.16.9.36 UART TURNAROUND TIMING REGISTER	0xBC	Turn Around Timing Register
Section 16.16.9.37 UART DIVISOR LATCH FRACTIONAL VALUE REGISTER	0xC0	Divisor Latch Fractional Value Register
Section 16.16.9.38 UART RECEIVE ADDRESS REGISTER	0xC4	Receive Address Register
Section 16.16.9.39 UART TRANSMIT ADDRESS REGISTER	0xC8	Transmit Address Register

Register Name	Offset	Description
Section 16.16.9.40 UART LINE EXTENDED CONTROL REGISTER	0xCC	Line Extended Control Register
Reserved	0xD0-0xF0	–
Section 16.16.9.41 UART COMPONENT PARAMETER REGISTER	0xF4	Component Parameter Register
Section 16.16.9.42 UART COMPONENT VERSION REGISTER	0xF8	UART Component Version Register
Section 16.16.9.43 UART COMPONENT TYPE REGISTER	0xFC	Component Type Register

16.16.9 Register Description

Legend:
R = Read-only, W = Write-only, R/W = Read/Write

16.16.9.1 UART RECEIVE BUFFER REGISTER

Table 16.413. Receive Buffer Register Description

Bit	Access	Function	POR Value	Description
31:9	R	Reserved	0	Reserved
8	R	Receive Buffer register (MSB 9th bit)	0	Data byte received on the serial input port (sin) in UART mode for the MSB 9th bit.
7:0	R	Receive Buffer Register (LSB 8 bits)	0	Data byte received on the serial input port (sin) in UART mode, or the serial infrared input (sir_in) in infrared mode. The data in this register is valid only if the Data Ready (DR) bit in the Line Status Register (LSR) is set. If in non-FIFO mode (FIFO_MODE = NONE) or FIFOs are disabled (FCR[0] set to 0), the data in the RBR must be read before the next data arrives, otherwise it is overwritten, resulting in an over-run error. If in FIFO mode (FIFO_MODE != NONE) and FIFOs are enabled (FCR[0] set to 1), this register accesses the head of the receive FIFO. If the receive FIFO is full and this register is not read before the next data character arrives, then the data already in the FIFO is preserved, but any incoming data are lost and an over-run error occurs.

16.16.9.2 UART TRANSMIT HOLDING REGISTER

Table 16.414. Transmit Holding Register Description

Bit	Access	Function	POR Value	Description
31:9	R	Reserved	0	Reserved

Bit	Access	Function	POR Val-ue	Description
8	W	THR (MSB 9th bit)	0	Data to be transmitted on the serial output port (sout) in UART mode for the MSB 9th bit.
7:0	W	THR (LSB 8 bits)	0	Data to be transmitted on the serial output port (sout) in UART mode or the serial infrared output (sir_out_n) in infrared mode. Data should only be written to the THR when the THR Empty (THRE) bit (LSR[5]) is set. If in non-FIFO mode or FIFOs are disabled (FCR[0] = 0) and THRE is set, writing a single character to the THR clears the THRE. Any additional writes to the THR before the THRE is set again causes the THR data to be overwritten. If in FIFO mode and FIFOs are enabled (FCR[0] = 1) and THRE is set, x number of characters of data may be written to the THR before the FIFO is full. The number x (default=16) is determined by the value of FIFO Depth that is set during configuration. Any attempt to write data when the FIFO is full results in the write data being lost.

16.16.9.3 UART DIVISOR LATCH LOW REGISTER

Table 16.415. Divisor Latch Low Register Description

Bit	Access	Function	POR Value	Description
31:8	R	Reserved	0	Reserved
7:0	R/W	Divisor Latch (Low)	0	Lower 8 bits of a 16-bit, read/write, Divisor Latch register that contains the baud rate divisor for the UART. The output baud rate is equal to the serial clock (pclk if one clock design, sclk if two clock design frequency divided by sixteen times the value of the baud rate divisor, as follows: baud rate = (serial clock freq) / (16 * divisor). Note that with the Divisor Latch Registers (DLL and DLH) set to 0, the baud clock is disabled and no serial communications occur. Also, once the DLL is set, at least 8 clock cycles of the slowest uart clock should be allowed to pass before transmitting or receiving data.

16.16.9.4 UART DIVISOR LATCH HIGH REGISTER

Table 16.416. Divisor Latch High Register Description

Bit	Access	Function	POR Value	Description
31:8	R	Reserved	0	Reserved

Bit	Access	Function	POR Value	Description
7:0	R/W	Divisor Latch (High)	0	Upper 8-bits of a 16-bit, read/write, Divisor Latch register that contains the baud rate divisor for the UART. The output baud rate is equal to the serial clock (pclk if one clock design, sclk if two clock design frequency divided by sixteen times the value of the baud rate divisor, as follows: baud rate = (serial clock freq) / (16 * divisor). Note that with the Divisor Latch Registers (DLL and DLH) set to 0, the baud clock is disabled and no serial communications occur. Also, once the DLH is set, at least 8 clock cycles of the slowest uart clock should be allowed to pass before transmitting or receiving data.

16.16.9.5 UART INTERRUPT ENABLE REGISTER

Table 16.417. Interrupt Enable Register Description

Bit	Access	Function	POR Value	Description
31:8	R	Reserved	0	Reserved
7	R/W	PTIME	0	Programmable THRE Interrupt Mode Enable. This is used to enable/disable the generation of THRE Interrupt. • 0 – disabled • 1 – enabled
6:4	R	Reserved	0	Reserved
3	R/W	EDSSI	0	Enable Modem Status Interrupt. This is used to enable/disable the generation of Modem Status Interrupt. This is the fourth highest priority interrupt. • 0 – disabled • 1 – enabled
2	R/W	ELSI	0	Enable Receiver Line Status Interrupt. This is used to enable/disable the generation of Receiver Line Status Interrupt. This is the highest priority interrupt. • 0 – disabled • 1 – enabled
1	R/W	ETBEI	0	Enable Transmit Holding Register Empty Interrupt. This is used to enable/disable the generation of Transmitter Holding Register Empty Interrupt. This is the third highest priority interrupt. • 0 – disabled • 1 – enabled
0	R/W	ERBFI	0	Enable Received Data Available Interrupt. These are the second highest priority interrupts. This is used to enable/disable the generation of Received Data Available Interrupt and the Character Timeout Interrupt (if in FIFO mode and FIFOs enabled). • 0 – disabled • 1 – enabled

16.16.9.6 UART INTERRUPT IDENTIFICATION REGISTER

Table 16.418. Interrupt Identity Register Description

Bit	Access	Function	POR Value	Description
31:8	R	Reserved	0	Reserved
7:6	R	FIFOs Enabled (or FIFOSE)	0	This is used to indicate whether the FIFOs are enabled or disabled. • 00 – disabled • 11 – enabled
5:4	R	Reserved	0	Reserved
3:0	R	Interrupt ID (IID)	1	This indicates the highest priority pending interrupt which can be one of the following types: • 0000 – modem status • 0001 – no interrupt pending • 0010 – THR empty • 0100 – received data available • 0110 – receiver line status • 0111 – busy detect • 1100 – character timeout Bit 3 indicates an interrupt can only occur when the FIFOs are enabled and used to distinguish a Character Timeout condition interrupt

16.16.9.7 UART FIFO CONTROL REGISTER

Table 16.419. FIFO Control Register Description

Bit	Access	Function	POR Value	Description
31:8	R	Reserved	0	Reserved
7:6	W	RCVR Trigger (or RT)	0	RCVR Trigger. This is used to select the trigger level in the receiver FIFO at which the Received Data Available Interrupt is generated. It also determines when the dma_rx_req_n signal is asserted in certain modes of operation. • 00 – 1 character in the FIFO • 01 – FIFO ¼ full • 10 – FIFO ½ full • 11 – FIFO 2 less than full
5:4	W	TX Empty Trigger (or TET)	0	TX Empty Trigger. This is used to select the empty threshold level at which the THRE Interrupts are generated when the mode is active. It also determines when the dma_tx_req_n signal is asserted when in certain modes of operation. The following trigger levels are supported: • 00 – FIFO empty • 01 – 2 characters in the FIFO • 10 – FIFO ¼ full • 11 – FIFO ½ full
3	W	DMA Mode (or DMAM)	0	DMA Mode. This determines the DMA signalling mode used for the dma_tx_req_n and dma_rx_req_n output signals. • 0 – mode 0 • 1 – mode 1

Bit	Access	Function	POR Value	Description
2	W	XMIT FIFO Reset (or XFIFOR)	0	XMIT FIFO Reset. This resets the control portion of the transmit FIFO and treats the FIFO as empty. Note that this bit is 'self-clearing'. It is not necessary to clear this bit.
1	W	RCVR FIFO Reset (or RFIFOR)	0	RCVR FIFO Reset. This resets the control portion of the receive FIFO and treats the FIFO as empty. Note that this bit is 'self-clearing'. It is not necessary to clear this bit.
0	W	FIFO Enable (or FIFOE)	0	FIFO Enable. This enables/disables the transmit (XMIT) and receive (RCVR) FIFOs. Whenever the value of this bit is changed both the XMIT and RCVR controller portion of FIFOs is reset.

16.16.9.8 UART LINE CONTROL REGISTER

Table 16.420. Line Control Register Description

Bit	Access	Function	POR Value	Description
31:8	R	Reserved	0	Reserved
7	R/W	DLAB	0	Divisor Latch Access Bit. This bit is used to enable reading and writing of the Divisor Latch register (DLL and DLH/LPDLL and LPDLH) to set the baud rate of the UART. This bit must be cleared after initial baud rate setup in order to access other registers
6	R/W	Break (or BC)	0	Break Control Bit.This is used to cause a break condition to be transmitted to the receiving device. If set to 1, the serial output is forced to the spacing (logic 0) state. When not in Loopback Mode, as determined by MCR[4], the sout line is forced low until the Break bit is cleared. If active (MCR[6] set to 1) the sir_out_n line is continuously pulsed. When in Loopback Mode, the break condition is internally looped back to the receiver and the sir_out_n line is forced low.
5	R/W	Stick Parity	0	Stick Parity. This bit is used to force parity value. When PEN, EPS and Stick Parity are set to 1, the parity bit is transmitted and checked as logic 0. If PEN and Stick Parity are set to 1 and EPS is a logic 0, then parity bit is transmitted and checked as a logic 1. If this bit is set to 0, Stick Parity is disabled.
4	R/W	EPS	0	Even Parity Select. This is used to select between even and odd parity, when parity is enabled (PEN set to 1). If set to 1, an even number of logic 1s is transmitted or checked. If set to 0, an odd number of logic 1s is transmitted or checked.
3	R/W	PEN	0	Parity Enable. This bit is used to enable and disable parity generation and detection in transmitted and received serial character respectively. • 0 – parity disabled • 1 – parity enabled

Bit	Access	Function	POR Value	Description
2	R/W	STOP	0	Number of stop bits. This is used to select the number of stop bits per character that the peripheral transmits and receives. If set to 0, one stop bit is transmitted in the serial data. If set to 1 and the data bits are set to 5 (LCR[1:0] set to 0) one and a half stop bits is transmitted. Otherwise, two stop bits are transmitted. Note that regardless of the number of stop bits selected, the receiver checks only the first stop bit. • 0 – 1 stop bit • 1 – 1.5 stop bits when DLS (LCR[1:0]) is 0, else 2 stop bit NOTE: The STOP bit duration implemented by uart may appear longer due to idle time inserted between characters for some configurations and baud clock divisor values in the transmit direction.
1:0	R/W	DLS (or CLS, as used in legacy)	0	Data Length Select. When DLS_E in LCR_EXT is set to 0, this register is used to select the number of data bits per character that the peripheral transmits and receives. The number of bits that may be selected are as follows: • 00 – 5 bits • 01 – 6 bits • 10 – 7 bits • 11 – 8 bits

16.16.9.9 UART MODEM CONTROL REGISTER

Table 16.421. ModemControl Register Description

Bit	Access	Function	POR Value	Description
31:6	R	Reserved	0	Reserved
5	R/W	AFCE	0	Auto Flow Control Enable. When FIFOs are enabled and the Auto Flow Control Enable (AFCE) bit is set, Auto Flow Control features are enabled • 0 – Auto Flow Control Mode disabled • 1 – Auto Flow Control Mode enabled
4	R/W	LoopBack (or LB)	0	LoopBack Bit. This is used to put the UART into a diagnostic mode for test purposes. If operating in UART mode, data on the sout line is held high, while serial data output is looped back to the sin line, internally. In this mode all the interrupts are fully functional. Also, in loopback mode, the modem control inputs (dsr_n, cts_n, ri_n, dcd_n) are disconnected and the modem control outputs (dtr_n, rts_n, out1_n, out2_n) are looped back to the inputs, internally. If operating in infrared mode (MCR[6] set to 1), data on the sir_out_n line is held low, while serial data output is inverted and looped back to the sir_in line.

Bit	Access	Function	POR Value	Description
3	R/W	OUT2	0	OUT2. This is used to directly control the user-designated Output2 (out2_n) output. The value written to this location is inverted and driven out on out2_n, that is: • 0 – out2_n de-asserted (logic 1) • 1 – out2_n asserted (logic 0) Note that in Loopback mode (MCR[4] set to 1), the out2_n output is held inactive high while the value of this location is internally looped back to an input.
2	R/W	OUT1	0	OUT1. This is used to directly control the user-designated Output1 (out1_n) output. The value written to this location is inverted and driven out on out1_n, that is: • 0 – out1_n de-asserted (logic 1) • 1 – out1_n asserted (logic 0) Note that in Loopback mode (MCR[4] set to 1), the out1_n output is held inactive high while the value of this location is internally looped back to an input.
1	R/W	RTS	0	Request to Send. This is used to directly control the Request to Send (rts_n) output. The Request To Send (rts_n) output is used to inform the modem or data set that the UART is ready to exchange data. When Auto RTS Flow Control is not enabled (MCR[5] set to 0), the rts_n signal is set low by programming MCR[1] (RTS) to a high. In Auto Flow Control, (MCR[5] set to 1) and FIFOs enable (FCR[0] set to 1), the rts_n output is controlled in the same way but is also gated with the receiver FIFO threshold trigger (rts_n is inactive high when above the threshold) only when the RTS Flow Trigger is disabled; otherwise it is gated by the receiver FIFO almost-full trigger, where “almost full” refers to two available slots in the FIFO (rts_n is inactive high when above the threshold). The rts_n signal is de-asserted when MCR[1] is set low. Note that in Loopback mode (MCR[4] set to 1), the rts_n output is held inactive high while the value of this location is internally looped back to an input. RTS should be enabled along with CTS. Otherwise its invalid
0	R/W	DTR	0	Data Terminal Ready. This is used to directly control the Data Terminal Ready (dtr_n) output. The value written to this location is inverted and driven out on dtr_n, that is: • 0 – dtr_n de-asserted (logic 1) • 1 – dtr_n asserted (logic 0) The Data Terminal Ready output is used to inform the modem or data set that the UART is ready to establish communications. Note that in Loopback mode (MCR[4] set to 1), the dtr_n output is held inactive high while the value of this location is internally looped back to an input.

16.16.9.10 UART LINE STATUS REGISTER

Table 16.422. Line Status Register Description

Bit	Access	Function	POR Value	Description
31:9	R	Reserved	0	Reserved

Bit	Access	Function	POR Value	Description
8	R/W	ADDR_RCVD	0	Address Received bit If 9-bit data mode (LCR_EXT[0]=1) is enabled, this bit is used to indicate that the 9th bit of the receive data is set to 1. This bit can also be used to indicate whether the incoming character is an address or data. • 1 - Indicates that the character is an address. • 0 - Indicates that the character is data. In the FIFO mode, since the 9th bit is associated with the received character, it is revealed when the character with the 9th bit set to 1 is at the top of the FIFO list. Reading the LSR clears the 9th bit. NOTE: You must ensure that an interrupt gets cleared (reading LSR register) before the next address byte arrives. If there is a delay in clearing the interrupt, then software will not be able to distinguish between multiple address related interrupt.
7	R	RFE	0	Receiver FIFO Error bit. This bit is only relevant when FIFO_MODE != NONE AND FIFOs are enabled (FCR[0] set to 1). This is used to indicate if there is at least one parity error, framing error, or break indication in the FIFO. • 0 – no error in RX FIFO • 1 – error in RX FIFO This bit is cleared when the LSR is read and the character with the error is at the top of the receiver FIFO and there are no subsequent errors in the FIFO.
6	R	TEMT	1	Transmitter Empty bit. If in FIFO mode (FIFO_MODE != NONE) and FIFOs enabled (FCR[0] set to 1), this bit is set whenever the Transmitter Shift Register and the FIFO are both empty. If in non-FIFO mode or FIFOs are disabled, this bit is set whenever the Transmitter Holding Register and the Transmitter Shift Register are both empty.
5	R	THRE	1	Transmit Holding Register Empty bit. If THRE mode is disabled (IER[7] set to 0) and regardless of FIFO's being implemented/enabled or not, this bit indicates that the THR or TX FIFO is empty. This bit is set whenever data is transferred from the THR or TX FIFO to the transmitter shift register and no new data has been written to the THR or TX FIFO. This also causes a THRE Interrupt to occur, if the THRE Interrupt is enabled. If both modes are active (IER[7] set to 1 and FCR[0] set to 1 respectively), the functionality is switched to indicate the transmitter FIFO is full and no longer controls THRE interrupts, which are then controlled by the FCR[5:4] threshold setting.

Bit	Access	Function	POR Value	Description
4	R	BI	0	Break Interrupt bit. This is used to indicate the detection of a break sequence on the serial input data. If in UART mode, it is set whenever the serial input, sin, is held in a logic '0' state for longer than the sum of start time + data bits + parity + stop bits. If in infrared mode, it is set whenever the serial input, sir_in, is continuously pulsed to logic '0' for longer than the sum of start time + data bits + parity + stop bits. A break condition on serial input causes one and only one character, consisting of all 0s, to be received by the UART. In FIFO mode, the character associated with the break condition is carried through the FIFO and is revealed when the character is at the top of the FIFO. Reading the LSR clears the BI bit. In non-FIFO mode, the BI indication occurs immediately and persists until the LSR is read. NOTE: If a FIFO is full when a break condition is received, a FIFO overrun occurs. The break condition and all the information associated with it—parity and framing errors—is discarded; any information that a break character was received is lost.
3	R	FE	0	A framing error occurs when the receiver does not detect a valid STOP bit in the received data. In the FIFO mode, since the framing error is associated with a character received, it is revealed when the character with the framing error is at the top of the FIFO. When a framing error occurs, the uart tries to re-synchronize. It does this by assuming that the error was due to the start bit of the next character and then continues receiving the other bit; that is, data, and/or parity and stop. It should be noted that the Framing Error (FE) bit (LSR[3]) is set if a break interrupt has occurred, as indicated by Break Interrupt (BI) bit (LSR[4]). This happens because the break character implicitly generates a framing error by holding the sin input to logic 0 for longer than the duration of a character. • 0 – no framing error • 1 – framing error Reading the LSR clears the FE bit.

Bit	Access	Function	POR Value	Description
2	R	PE	0	Parity Error bit. This is used to indicate the occurrence of a parity error in the receiver if the Parity Enable (PEN) bit (LCR[3]) is set. In the FIFO mode, since the parity error is associated with a character received, it is revealed when the character with the parity error arrives at the top of the FIFO. It should be noted that the Parity Error (PE) bit (LSR[2]) can be set if a break interrupt has occurred, as indicated by Break Interrupt (BI) bit (LSR[4]). In this situation, the Parity Error bit is set if parity generation and detection is enabled (LCR[3]=1) and the parity is set to odd (LCR[4]=0). • 0 – no parity error • 1 – parity error Reading the LSR clears the PE bit.
1	R	OE	0	Overrun error bit. This is used to indicate the occurrence of an overrun error. This occurs if a new data character was received before the previous data was read. In the non-FIFO mode, the OE bit is set when a new character arrives in the receiver before the previous character was read from the RBR. When this happens, the data in the RBR is overwritten. In the FIFO mode, an overrun error occurs when the FIFO is full and a new character arrives at the receiver. The data in the FIFO is retained and the data in the receive shift register is lost. • 0 – no overrun error • 1 – overrun error Reading the LSR clears the OE bit.
0	R	DR	0	Data Ready bit. This is used to indicate that the receiver contains at least one character in the RBR or the receiver FIFO. • 0 – no data ready • 1 – data ready This bit is cleared when the RBR is read in non-FIFO mode, or when the receiver FIFO is empty, in FIFO mode.

16.16.9.11 UART MODEM STATUS REGISTER

Table 16.423. Modem Status Register Description

Bit	Access	Function	POR Value	Description
31:8	R	Reserved	0	Reserved
7	R	DCD	0	Data Carrier Detect. This is used to indicate the current state of the modem control line dcd_n. This bit is the complement of dcd_n. When the Data Carrier Detect input (dcd_n) is asserted it is an indication that the carrier has been detected by the modem or data set. • 0 – dcd_n input is de-asserted (logic 1) • 1 – dcd_n input is asserted (logic 0) In Loopback Mode (MCR[4] set to 1), DCD is the same as MCR[3] (Out2).

Bit	Access	Function	POR Value	Description
6	R	RI	0	Ring Indicator. This is used to indicate the current state of the modem control line ri_n. This bit is the complement of ri_n. When the Ring Indicator input (ri_n) is asserted it is an indication that a telephone ringing signal has been received by the modem or data set. • 0 – ri_n input is de-asserted (logic 1) • 1 – ri_n input is asserted (logic 0) In Loopback Mode (MCR[4] set to 1), RI is the same as MCR[2] (Out1)
5	R	DSR	0	Data Set Ready. This is used to indicate the current state of the modem control line dsr_n. This bit is the complement of dsr_n. When the Data Set Ready input (dsr_n) is asserted it is an indication that the modem or data set is ready to establish communications with the uart. • 0 – dsr_n input is de-asserted (logic 1) • 1 – dsr_n input is asserted (logic 0) In Loopback Mode (MCR[4] set to 1), DSR is the same as MCR[0] (DTR).
4	R	CTS	0	This bit is the complement of cts_n. When the Clear to Send input (cts_n) is asserted it is an indication that the modem or data set is ready to exchange data with the uart. • 0 – cts_n input is de-asserted (logic 1) • 1 – cts_n input is asserted (logic 0) In Loopback Mode (MCR[4] = 1), CTS is the same as MCR[1] (RTS). CTS should be enabled along with RTS. Otherwise its invalid
3	R	DDCD	0	Delta Data Carrier Detect. This is used to indicate that the modem control line dcd_n has changed since the last time the MSR was read. • 0 – no change on dcd_n since last read of MSR • 1 – change on dcd_n since last read of MSR Reading the MSR clears the DDCD bit. In Loopback Mode (MCR[4] = 1), DDCD reflects changes on MCR[3] (Out2). Note, if the DDCD bit is not set and the dcd_n signal is asserted (low) and a reset occurs (software or otherwise), then the DDCD bit is set when the reset is removed if the dcd_n signal remains asserted.
2	R	TERI	0	Trailing Edge of Ring Indicator. This is used to indicate that a change on the input ri_n (from an active-low to an inactive-high state) has occurred since the last time the MSR was read. • 0 – no change on ri_n since last read of MSR • 1 – change on ri_n since last read of MSR Reading the MSR clears the TERI bit. In Loopback Mode (MCR[4] = 1), TERI reflects when MCR[2] (Out1) has changed state from a high to a low.

Bit	Access	Function	POR Value	Description
1	R	DDSR	0	Delta Data Set Ready. This is used to indicate that the modem control line dsr_n has changed since the last time the MSR was read. • 0 – no change on dsr_n since last read of MSR • 1 – change on dsr_n since last read of MSR Reading the MSR clears the DDSR bit. In Loopback Mode (MCR[4] = 1), DDSR reflects changes on MCR[0] (DTR). Note, if the DDSR bit is not set and the dsr_n signal is asserted (low) and a reset occurs (software or otherwise), then the DDSR bit is set when the reset is removed if the dsr_n signal remains asserted.
0	R	DCTS	0	Delta Clear to Send. This is used to indicate that the modem control line cts_n has changed since the last time the MSR was read. • 0 – no change on cts_n since last read of MSR • 1 – change on cts_n since last read of MSR Reading the MSR clears the DCTS bit. In Loopback Mode (MCR[4] = 1), DCTS reflects changes on MCR[1] (RTS). Note, if the DCTS bit is not set and the cts_n signal is asserted (low) and a reset occurs (software or otherwise), then the DCTS bit is set when the reset is removed if the cts_n signal remains asserted.

16.16.9.12 UART SCRATCHPAD REGISTER

Table 16.424. Scratchpad Register Description

Bit	Access	Function	POR Value	Description
31:8	R	Reserved	0	Reserved
7:0	R/W	Scratchpad Register	0	This register is for programmers to use as a temporary storage space. It has no defined purpose in the uart.

16.16.9.13 UART LOW POWER DIVISOR LATCH (LOW) REGISTER

Table 16.425. Low Power Divisor Latch Low Register Description

Bit	Access	Function	POR Value	Description
31:8	R	Reserved	0	Reserved

Bit	Access	Function	POR Value	Description
7:0	R/W	LPDLL	0	This register makes up the lower 8-bits of a 16-bit, read/write, Low Power Divisor Latch register that contains the baud rate divisor for the UART, which must give a baud rate of 115.2K. This is required for SIR Low Power (minimum pulse width) detection at the receiver. The output low-power baud rate is equal to the serial clock (sclk) frequency divided by sixteen times the value of the baud rate divisor, as follows: Low power baud rate = (serial clock frequency)/(16* divisor) Therefore, a divisor must be selected to give a baud rate of 115.2K. NOTE: When the Low Power Divisor Latch registers (LPDLL and LPDLH) are set to 0, the low-power baud clock is disabled and no low-power pulse detection (or any pulse detection) occurs at the receiver. Also, once the LPDLL is set, at least eight clock cycles of the slowest uart clock should be allowed to pass before transmitting or receiving data.

16.16.9.14 UART LOW POWER DIVISOR LATCH (HIGH) REGISTER

Table 16.426. Low Power Divisor Latch High Register Description

Bit	Access	Function	POR Value	Description
31:8	R	Reserved	0	Reserved
7:0	R/W	LPDLH	0	This register makes up the upper 8-bits of a 16-bit, read/write, Low Power Divisor Latch register that contains the baud rate divisor for the UART, which must give a baud rate of 115.2K. This is required for SIR Low Power (minimum pulse width) detection at the receiver. The output low-power baud rate is equal to the serial clock (sclk) frequency divided by sixteen times the value of the baud rate divisor, as follows: Low power baud rate = (serial clock frequency)/(16* divisor) Therefore, a divisor must be selected to give a baud rate of 115.2K. NOTE: When the Low Power Divisor Latch registers (LPDLL and LPDLH) are set to 0, the low-power baud clock is disabled and no low-power pulse detection (or any pulse detection) occurs at the receiver. Also, once the LPDLH is set, at least eight clock cycles of the slowest uart clock should be allowed to pass before transmitting or receiving data.

16.16.9.15 UART SHADOW RECEIVE BUFFER REGISTER

Table 16.427. Shadow Receive Buffer Register Description

Bit	Access	Function	POR Value	Description
31:9	R	Reserved	0	Reserved
8	R	Shadow Receive Buffer Register (MSB 9th bit)	0	This is a shadow register for the RBR[8] bit. It is applicable only when UART_9BIT_DATA_EN=1.

Bit	Access	Function	POR Value	Description
7:0	R	Shadow Receive Buffer Register (LSB 8 bits)	0	This is a shadow register for the RBR and has been allocated sixteen 32-bit locations so as to accommodate burst accesses from the Master. This register contains the data byte received on the serial input port (sin) in UART mode or the serial infrared input (sir_in) in infrared mode. The data in this register is valid only if the Data Ready (DR) bit in the Line status Register (LSR) is set. If in non-FIFO mode (FIFO_MODE = NONE) or FIFOs are disabled (FCR[0] set to 0), the data in the RBR must be read before the next data arrives, otherwise it is overwritten, resulting in an overrun error. If in FIFO mode (FIFO_MODE != NONE) and FIFOs are enabled (FCR[0] set to 1), this register accesses the head of the receive FIFO. If the receive FIFO is full and this register is not read before the next data character arrives, then the data already in the FIFO are preserved, but any incoming data is lost. An overrun error also occurs.

16.16.9.16 UART SHADOW TRANSMIT HOLDING REGISTER

Table 16.428. Shadow Transmit Holding Register Description

Bit	Access	Function	POR Value	Description
31:9	R	Reserved	0	Reserved
8	W	Shadow Transmit Holding Register (MSB 9th bit)	0	This is a shadow register for the THR[8] bit. It is applicable only when UART_9BIT_DATA_EN=1
7:0	W	Shadow Transmit Holding Register	0	This is a shadow register for the THR and has been allocated sixteen 32-bit locations so as to accommodate burst accesses from the Master. This register contains data to be transmitted on the serial output port (sout) in UART mode or the serial infrared output (sir_out_n) in infrared mode. Data should only be written to the THR when the THR Empty (THRE) bit (LSR[5]) is set. If in non-FIFO mode or FIFOs are disabled (FCR[0] set to 0) and THRE is set, writing a single character to the THR clears the THRE. Any additional writes to the THR before the THRE is set again causes the THR data to be overwritten. If in FIFO mode and FIFOs are enabled (FCR[0] set to 1) and THRE is set, x number of characters of data may be written to the THR before the FIFO is full. The number x (default=16) is determined by the value of FIFO Depth that is set during configuration. Any attempt to write data when the FIFO is full results in the write data being lost.

16.16.9.17 UART FIFO ACCESS REGISTER

Table 16.429. FIFO Access Register Description

Bit	Access	Function	POR Value	Description
31:1	R	Reserved	0	Reserved
0	R/W	FIFO Access	0	Writes have no effect when FIFO_ACCESS = No, always readable. This register is use to enable a FIFO access mode for testing, so that the receive FIFO can be written by the Master and the transmit FIFO can be read by the Master when FIFOs are implemented and enabled. When FIFOs are not implemented or not enabled it allows the RBR to be written by the Master and the THR to be read by the Master. • 0 – FIFO access mode disabled • 1 – FIFO access mode enabled Note, that when the FIFO access mode is enabled/disabled, the control portion of the receive FIFO and transmit FIFO is reset and the FIFOs are treated as empty.

16.16.9.18 UART TRANSMIT FIFO READ REGISTER

Table 16.430. Transmit FIFO Read Register Description

Bit	Access	Function	POR Value	Description
31:8	R	Reserved	0	Reserved
7:0	R	Transmit FIFO Read	0	These bits are only valid when FIFO access mode is enabled (FAR[0] is set to 1). When FIFOs are implemented and enabled, reading this register gives the data at the top of the transmit FIFO. Each consecutive read pops the transmit FIFO and gives the next data value that is currently at the top of the FIFO. When FIFOs are not implemented or not enabled, reading this register gives the data in the THR.

16.16.9.19 UART RECEIVE FIFO WRITE REGISTER

Table 16.431. Receive FIFO Write Register Description

Bit	Access	Function	POR Value	Description
31:10	R	Reserved	0	Reserved
9	W	RFFE	0	Receive FIFO Framing Error. These bits are only valid when FIFO access mode is enabled (FAR[0] is set to 1). When FIFOs are implemented and enabled, this bit is used to write framing error detection information to the receive FIFO. When FIFOs are not implemented or not enabled, this bit is used to write framing error detection information to the RBR.

Bit	Access	Function	POR Value	Description
8	W	RFPE	0	Receive FIFO Parity Error. These bits are only valid when FIFO access mode is enabled (FAR[0] is set to 1). When FIFOs are implemented and enabled, this bit is used to write parity error detection information to the receive FIFO. When FIFOs are not implemented or not enabled, this bit is used to write parity error detection information to the RBR.
7:0	W	RFWD	0	Receive FIFO Write Data. These bits are only valid when FIFO access mode is enabled (FAR[0] is set to 1). When FIFOs are implemented and enabled, the data that is written to the RFWD is pushed into the receive FIFO. Each consecutive write pushes the new data to the next write location in the receive FIFO. When FIFOs are not implemented or not enabled, the data that is written to the RFWD is pushed into the RBR.

16.16.9.20 UART STATUS REGISTER

Table 16.432. UART Status Register Description

Bit	Access	Function	POR Value	Description
31:5	R	Reserved	0	Reserved
4	R	RFF	0	Receive FIFO Full. This bit is only valid when FIFO_STAT = YES. This is used to indicate that the receive FIFO is completely full. • 0 – Receive FIFO not full • 1 – Receive FIFO Full This bit is cleared when the RX FIFO is no longer full.
3	R	RFNE	0	Receive FIFO Not Empty. This bit is only valid when FIFO_STAT = YES. This is used to indicate that the receive FIFO contains one or more entries. • 0 – Receive FIFO is empty • 1 – Receive FIFO is not empty This bit is cleared when the RX FIFO is empty.
2	R	TFE	0	Transmit FIFO Empty. This bit is only valid when FIFO_STAT = YES. This is used to indicate that the transmit FIFO is completely empty. • 0 – Transmit FIFO is not empty • 1 – Transmit FIFO is empty This bit is cleared when the TX FIFO is no longer empty.
1	R	TFNF	0	Transmit FIFO Not Full. This bit is only valid when FIFO_STAT = YES. This is used to indicate that the transmit FIFO in not full. • 0 – Transmit FIFO is full • 1 – Transmit FIFO is not full This bit is cleared when the TX FIFO is full.

Bit	Access	Function	POR Value	Description
0	R	BUSY	0	UART Busy. This bit indicates that a serial transfer is in progress; when cleared, indicates that the uart is idle or inactive. • 0 – uart is idle or inactive • 1 – uart is busy (actively transferring data) This bit will be set to 1 (busy) under any of the following conditions: 1. Transmission in progress on serial interface 2. Transmit data present in THR, when FIFO access mode is not being used (FAR = 0) and the baud divisor is non-zero ({DLH,DLL} does not equal 0) when the divisor latch access bit is 0 (LCR.DLAB = 0) 3. Reception in progress on the interface 4. Receive data present in RBR, when FIFO access mode is not being used (FAR = 0) NOTE: It is possible for the UART Busy bit to be cleared even though a new character may have been sent from another device. That is, if the uart has no data in THR and RBR and there is no transmission in progress and a start bit of a new character has just reached the uart. This is due to the fact that a valid start is not seen until the middle of the bit period and this duration is dependent on the baud divisor that has been programmed. If a second system clock has been implemented, the assertion of this bit is also delayed by several cycles of the slower clock.

16.16.9.21 UART TRANSMIT FIFO LEVEL REGISTER

Table 16.433. Transmit FIFO Level Register Description

Bit	Access	Function	POR Value	Description
31:30	R	Reserved	0	Reserved
29:0	R	Transmit FIFO Level	0	This indicates the number of data entries in the transmit FIFO

16.16.9.22 UART RECEIVE FIFO LEVEL REGISTER

Table 16.434. Receive FIFO Level Register Description

Bit	Access	Function	POR Value	Description
31:30	R	Reserved	0	Reserved
29:0	R	Receive FIFO Level	0	This indicates the number of data entries in the receive FIFO.

16.16.9.23 UART SOFTWARE RESET REGISTER

Table 16.435. Software Reset Register Description

Bit	Access	Function	POR Value	Description
31:3	R	Reserved	0	Reserved
2	W	XFR	0	XMIT FIFO Reset. This is a shadow register for the XMIT FIFO Reset bit (FCR[2]). This can be used to remove the burden on software having to store previously written FCR values (which are pretty static) just to reset the transmit FIFO. This resets the control portion of the transmit FIFO and treats the FIFO as empty. Note that this bit is 'self-clearing'. It is not necessary to clear this bit.
1	W	RFR	0	RCVR FIFO Reset. This is a shadow register for the RCVR FIFO Reset bit (FCR[1]). This can be used to remove the burden on software having to store previously written FCR values (which are pretty static) just to reset the receive FIFO This resets the control portion of the receive FIFO and treats the FIFO as empty. Note that this bit is 'self-clearing'. It is not necessary to clear this bit.
0	W	UR	0	UART Reset. This asynchronously resets the uart and synchronously removes the reset assertion. For a two clock implementation both pclk and sclk domains are reset.

16.16.9.24 UART SHADOW REQUEST TO SEND REGISTER

Table 16.436. Shadow Request to Send Register Description

Bit	Access	Function	POR Value	Description
31:1	R	Reserved	0	Reserved
0	R/W	Shadow Request to Send	0	This is a shadow register for the RTS bit (MCR[1]), this can be used to remove the burden of having to performing a read-modify-write on the MCR. This is used to directly control the Request to Send (rts_n) output. The Request To Send (rts_n) output is used to inform the modem or data set that the uart is ready to exchange data. When Auto RTS Flow Control is not enabled (MCR[5] = 0), the rts_n signal is set low by programming MCR[1] (RTS) to a high. In Auto Flow Control, (MCR[5] = 1) and FIFOs enable (FCR[0] = 1), the rts_n output is controlled in the same way, but is also gated with the receiver FIFO threshold trigger (rts_n is inactive high when above the threshold) only when RTS Flow Trigger is disabled; otherwise it is gated by the receiver FIFO almost-full trigger, where “almost full” refers to two available slots in the FIFO (rts_n is inactive high when above the threshold). Note that in Loopback mode (MCR[4] = 1), the rts_n output is held inactive-high while the value of this location is internally looped back to an input.

16.16.9.25 UART SHADOW BREAK CONTROL REGISTER

Table 16.437. Shadow Break Control Register Description

Bit	Access	Function	POR Value	Description
31:1	R	Reserved	0	Reserved
0	R/W	Shadow Break Control Register	0	This is a shadow register for the Break bit (LCR[6]), this can be used to remove the burden of having to performing a read modify write on the LCR. This is used to cause a break condition to be transmitted to the receiving device. If set to 1, the serial output is forced to the spacing (logic 0) state. When not in Loopback Mode, as determined by MCR[4], the sout line is forced low until the Break bit is cleared. If (MCR[6] = 1), then sir_out_n line is continuously pulsed. When in Loopback Mode, the break condition is internally looped back to the receiver.

16.16.9.26 UART SHADOW DMA MODE REGISTER

Table 16.438. Shadow DMA Mode Register Description

Bit	Access	Function	POR Value	Description
31:1	R	Reserved	0	Reserved
0	R/W	Shadow DMA Mode	0	This is a shadow register for the DMA mode bit (FCR[3]). This can be used to remove the burden of having to store the previously written value to the FCR in memory and having to mask this value so that only the DMA Mode bit gets updated. This determines the DMA signalling mode used for the dma_tx_req_n and dma_rx_req_n output signals. • 0 – mode 0 • 1 – mode 1

16.16.9.27 UART SHADOW FIFO ENABLE REGISTER

Table 16.439. Shadow FIFO Enable Register Description

Bit	Access	Function	POR Value	Description
31:1	R	Reserved	0	Reserved
0	R/W	Shadow FIFO Enable	0	Shadow FIFO Enable. This is a shadow register for the FIFO enable bit (FCR[0]). This can be used to remove the burden of having to store the previously written value to the FCR in memory and having to mask this value so that only the FIFO enable bit gets updated. This enables/disables the transmit (XMIT) and receive (RCVR) FIFOs. If this bit is set to 0 (disabled) after being enabled then both the XMIT and RCVR controller portion of FIFOs are reset.

16.16.9.28 UART SHADOW RCVR TRIGGER REGISTER

Table 16.440. Shadow RCVR Trigger Register Description

Bit	Access	Function	POR Value	Description
31:2	R	Reserved	0	Reserved
1:0	R/W	Shadow RCVR Trigger	0	Shadow RCVR Trigger. This is a shadow register for the RCVR trigger bits (FCR[7:6]). This can be used to remove the burden of having to store the previously written value to the FCR in memory and having to mask this value so that only the RCVR trigger bit gets updated. This is used to select the trigger level in the receiver FIFO at which the Received Data Available Interrupt is generated. It also determines when the dma_rx_req_n signal is asserted when DMA Mode (FCR[3]) = 1. The following trigger levels are supported: • 00 – 1 character in the FIFO • 01 – FIFO ¼ full • 10 – FIFO ½ full • 11 – FIFO 2 less than full

16.16.9.29 UART SHADOW TX EMPTY TRIGGER REGISTER

Table 16.441. Shadow TX Empty Trigger Register Description

Bit	Access	Function	POR Value	Description
31:2	R	Reserved	0	Reserved
1:0	R/W	Shadow TX Empty Trigger	0	Shadow TX Empty Trigger. This is a shadow register for the TX empty trigger bits (FCR[5:4]). This can be used to remove the burden of having to store the previously written value to the FCR in memory and having to mask this value so that only the TX empty trigger bit gets updated. This is used to select the empty threshold level at which the THRE Interrupts are generated when the mode is active. The following trigger levels are supported: • 00 – FIFO empty • 01 – 2 characters in the FIFO • 10 – FIFO ¼ full • 11 – FIFO ½ full

16.16.9.30 UART HALT TX REGISTER

Table 16.442. HALT TX Register Description

Bit	Access	Function	POR Value	Description
31:1	R	Reserved	0	Reserved

Bit	Access	Function	POR Value	Description
0	R/W	HALT TX	0	This register is use to halt transmissions for testing, so that the transmit FIFO can be filled by the Master when FIFOs are implemented and enabled. • 0 – Halt TX disabled • 1 – Halt TX enabled Note, if FIFOs are implemented and not enabled, the setting of the halt TX register has no effect on operation.

16.16.9.31 UART DMA SOFTWARE ACKNOWLEDGE REGISTER

Table 16.443. DMA Software Acknowledgment Register Description

Bit	Access	Function	POR Value	Description
31:1	R	Reserved	0	Reserved
0	W	DMA Software Acknowledge	0	This register is use to perform a DMA software acknowledge if a transfer needs to be terminated due to an error condition. For example, if the DMA disables the channel, then the uart should clear its request. This causes the TX request, TX single, RX request and RX single signals to de-assert. Note that this bit is 'self-clearing'. It is not necessary to clear this bit.

16.16.9.32 UART TRANSCEIVER CONTROL REGISTER

Table 16.444. Transceiver Control Register Description

Bit	Access	Function	POR Value	Description
31:5	R	Reserved	0	Reserved

Bit	Access	Function	POR Value	Description
4:3	R/W	XFER_MODE	0	Transfer Mode • 0 : In this mode, transmit and receive can happen simultaneously. You can enable DE_EN and RE_EN at any point of time. Turn around timing as programmed in the TAT register is not applicable in this mode. • 1 : In this mode, DE and RE are mutually exclusive. The hardware considers the turnaround timings that are programmed in the TAT register while switching from RE to DE or from DE to RE. Ensure that either DE or RE is expected to be enabled while programming. For transmission, hardware waits if it is in the midst of receiving any transfer, before it starts transmitting. • 2 : In this mode, DE and RE are mutually exclusive. Once DE_EN or RE_EN is programmed, 're' is enabled by default and uart controller will be ready to receive. If the user programs the TX FIFO with data, then uart, after ensuring no receive is in progress, disables the 're' and enables the 'de' signal. Once the TX FIFO becomes empty, the 're' signal gets enabled and the 'de' signal will be disabled. In this mode of operation, the hardware considers the turnaround timings that are programmed in the TAT register while switching from RE to DE or from DE to RE. In this mode, 'de' and 're' signals are strictly complementary to each other.
2	R/W	DE_POL	1	Driver Enable Polarity • 1: DE signal is active high • 0: DE signal is active low Reset Value: UART_DE_POL
1	R/W	RE_POL	1	Receiver Enable Polarity • 1: RE signal is active high • 0: RE signal is active low Reset Value: UART_RE_POL
0	R	Reserved	0	Reserved

16.16.9.33 UART DRIVER OUTPUT ENABLE REGISTER

Table 16.445. Driver Output Enable Register Description

Bit	Access	Function	POR Value	Description
31:1	R	Reserved	0	Reserved
0	R/W	DE Enable	0	DE Enable control The 'DE Enable' register bit is used to control assertion and de-assertion of 'de' signal. • 0: De-assert 'de' signal • 1: Assert 'de' signal

16.16.9.34 UART RECEIVER OUTPUT ENABLE REGISTER

Table 16.446. Receiver Output Enable Register Description

Bit	Access	Function	POR Value	Description
31:1	R	Reserved	0	Reserved
0	R/W	RE Enable	0	RE Enable control The 'RE Enable' register bit is used to control assertion and de-assertion of 're' signal. • 0: De-assert 're' signal • 1: Assert 're' signal

16.16.9.35 UART DRIVER OUTPUT ENABLE TIMING REGISTER

Table 16.447. Driver Output Enable Timing Register Description

Bit	Access	Function	POR Value	Description
31:24	R	Reserved	0	Reserved
23:16	R/W	DE de-assertion time	0	Driver enable de-assertion time. This field controls the amount of time (in terms of number of serial clock periods) between the end of stop bit on the serial output (sout) to the falling edge of Driver output enable signal
15:8	R	Reserved	0	Reserved
7:0	R/W	DE assertion time	0	Driver enable assertion time. This field controls the amount of time (in terms of number of serial clock periods) between the assertion of rising edge of Driver output enable signal to serial transmit enable. Any data in transmit buffer, will start on serial output (sout) after the transmit enable.

16.16.9.36 UART TURNAROUND TIMING REGISTER

Table 16.448. TurnAround Timing Register Description

Bit	Access	Function	POR Value	Description
31:16	R/W	RE to DE	0	Receiver Enable to Driver Enable Turn Around time. Turnaround time (in terms of serial clock) for RE de-assertion to DE assertion. Note: • If the DE assertion time in the DET register is 0, then the actual value is the programmed value + 3. • If the DE assertion time in the DET register is 1, then the actual value is the programmed value + 2. • If the DE assertion time in the DET register is greater than 1, then the actual value is the programmed value + 1.
15:0	R/W	DE to RE	0	Driver Enable to Receiver Enable Turn Around time. Turnaround time (in terms of serial clock) for DE de-assertion to RE assertion. Note: The actual time is the programmed value + 1.

16.16.9.37 UART DIVISOR LATCH FRACTIONAL VALUE REGISTER

Table 16.449. Divisor Latch Fraction Register Description

Bit	Access	Function	POR Value	Description
31:4	R	Reserved	0	Reserved
3:0	R/W	DLF	0	Fractional part of divisor. The fractional value is added to integer value set by DLH, DLL. Fractional value is determined by (Divisor Fraction value)/(2^4) The DLF Values to be programmed are DLF_Value Fraction Fractional Value 0000 0/16 0.0000 0001 1/16 0.0625 0010 2/16 0.125 0011 3/16 0.1875 0100 4/16 0.25 0101 5/16 0.3125 0110 6/16 0.375 0111 7/16 0.4375 1000 8/16 0.5 1001 9/16 0.5625 1010 10/16 0.625 1011 11/16 0.6875 1100 12/16 0.75 1101 13/16 0.8125 1110 14/16 0.875 1111 15/16 0.9375

16.16.9.38 UART RECEIVE ADDRESS REGISTER

Table 16.450. Receive Address Register Description

Bit	Access	Function	POR Value	Description
31:8	R	Reserved	0	Reserved

Bit	Access	Function	POR Value	Description
7:0	R/W	RAR	0	This is an address matching register during receive mode. If the 9-th bit is set in the incoming character then the remaining 8-bits will be checked against this register value. If the match happens then subsequent characters with 9-th bit set to 0 will be treated as data byte until the next address byte is received. Note: • This register is applicable only when 'ADDR_MATCH' (LCR_EXT[1]) and 'DLS_E' (LCR_EXT[0]) bits are set to 1. • RAR should be programmed only when UART is not busy. • RAR can be programmed at any point of the time. However, user must not change this register value when any receive is in progress.

16.16.9.39 UART TRANSMIT ADDRESS REGISTER

Table 16.451. Transmit Address Register Description

Bit	Access	Function	POR Value	Description
31:8	R	Reserved	0	Reserved
7:0	R/W	TAR	0	This is an address matching register during transmit mode. If DLS_E (LCR_EXT[0]) bit is enabled, then uart sends the 9-bit character with 9-th bit set to 1 and remaining 8-bit address will be sent from this register provided 'SEND_ADDR' (LCR_EXT[2]) bit is set to 1. NOTE: • This register is used only to send the address. The normal data should be sent by programming THR register. • Once the address is started to send on the uart serial lane, then 'SEND_ADDR' bit will be auto-cleared by the hardware.

16.16.9.40 UART LINE EXTENDED CONTROL REGISTER

Table 16.452. Line Extended Control Register Description

Bit	Access	Function	POR Value	Description
31:4	R	Reserved	0	Reserved

Bit	Access	Function	POR Value	Description
3	R/W	TRANSMIT_MODE	0	Transmit mode control bit. This bit is used to control the type of transmit mode during 9-bit data transfers. • 1 : In this mode of operation, Transmit Holding Register (THR) and Shadow Transmit Holding Register (STHR) are 9-bit wide. The user must ensure that the THR/STHR register is written correctly for address/data. Address: 9th bit is set to 1, Data: 9th bit is set to 0. NOTE: Transmit address register (TAR) is not applicable in this mode of operation. • 0 : In this mode of operation, Transmit Holding Register (THR) and Shadow Transmit Holding register (STHR) are 8-bit wide. The user needs to program the address into Transmit Address Register (TAR) and data into the THR/STHR register. SEND_ADDR bit is used as a control knob to indicate the uart on when to send the address.
2	R/W	SEND_ADDR	0	Send address control bit. This bit is used as a control knob for the user to determine when to send the address during transmit mode. • 1 - 9-bit character will be transmitted with 9-th bit set to 1 and the remaining 8-bits will match to what is being programmed in "Transmit Address Register". • 0 - 9-bit character will be transmitted with 9-th bit set to 0 and the remaining 8-bits will be taken from the TxFIFO which is programmed through 8-bit wide THR/STHR register. NOTE: 1. This bit is auto-cleared by the hardware, after sending out the address character. User is not expected to program this bit to 0. 2. This field is applicable only when DLS_E bit is set to 1 and TRANSMIT_MODE is set to 0.
1	R/W	ADDR_MATCH	0	Address Match Mode. This bit is used to enable the address match feature during receive. • 1 - Address match mode; uart will wait until the incoming character with 9-th bit set to 1. And, further checks to see if the address matches with what is programmed in "Receive Address Match Register". If match is found, then sub-sequent characters will be treated as valid data and uart starts receiving data. • 0 - Normal mode; uart will start to receive the data and 9-bit character will be formed and written into the receive RxFIFO. User is responsible to read the data and differentiate b/n address and data. NOTE: This field is applicable only when DLS_E is set to 1.
0	R/W	DLS_E	0	Extension for DLS. This bit is used to enable 9-bit data for transmit and receive transfers. • 1 = 9 bits per character • 0 = Number of data bits selected by DLS

16.16.9.41 UART COMPONENT PARAMETER REGISTER

Table 16.453. Component Parameter Register Description

Bit	Access	Function	POR Value	Description
31:24	R	Reserved	0	Reserved
23:16	R	FIFO_MODE	0x01	0x00 = 0 0x01 = 16 0x02 = 32 to 0x80 = 2048 0x81- 0xff = reserved
15:14	R	Reserved	0	Reserved
13	R	DMA_EXTRA	0x1	0 – FALSE 1 – TRUE
12	R	UART_ADD_ENCODED_PARAMS	0x0	0 – FALSE 1 – TRUE
11	R	SHADOW	0x0	0 – FALSE 1 – TRUE
10	R	FIFO_STAT	0x1	0 – FALSE 1 – TRUE
9	R	FIFO_ACCESS	0x0	0 – FALSE 1 – TRUE
8	R	ADDITIONAL_FEAT	0x1	0 – FALSE 1 – TRUE
7	R	SIR_LP_MODE	0x1	0 – FALSE 1 – TRUE
6	R	SIR_MODE	0x1	0 – FALSE 1 – TRUE
5	R	THRE_MODE	0x1	0 – FALSE 1 – TRUE
4	R	AFCE_MODE	0x1	0 – FALSE 1 – TRUE
3:2	R	Reserved	0	Reserved
1:0	R	APB_DATA_WIDTH	0x2	00 – 8 bits 01 – 16 bits 10 – 32 bits 11 – reserved

16.16.9.42 UART COMPONENT VERSION REGISTER

Table 16.454. UART Component Version Register Description

Bit	Access	Function	POR Value	Description
31:0	R	UART Component Version	32'h3430302a	This register contains UART Component Version.

16.16.9.43 UART COMPONENT TYPE REGISTER

Table 16.455. Component Type Register Description

Bit	Access	Function	POR Value	Description
31:0	R	Peripheral ID	0x44570110	This register contains the peripherals identification code

16.17 USART

16.17.1 General Description

USART is used in communication through wired medium in both Synchronous and Asynchronous fashion. In Synchronous mode both the full duplex and half duplex (single wire) modes are supported

16.17.2 Features

The following features are supported by the USART controller in the MCU HP peripherals (USART0):

- Support for both Synchronous and Asynchronous modes.
- Supports Full duplex and half duplex (single wire) mode of communication.
- 5-8 bit wide character support.
- Support all the baud rates available. Maximum supported baud rate is 7,372,800 (around 7.3Mbps) with 118MHz UART input clock. It supports upto 20Mbps in USART mode.
- Programmable FIFO thresholds with maximum FIFO depth of 16 and support for DMA
- Supports generation of interrupt for different events.

16.17.3 Functional Description

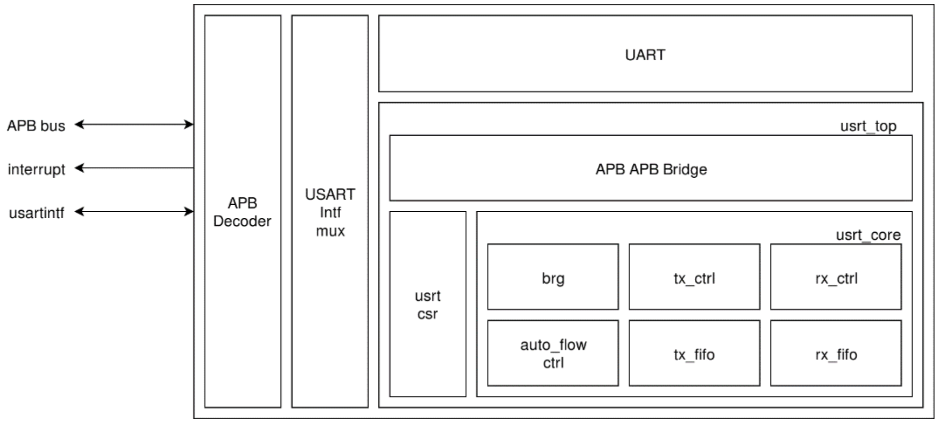


Figure 16.43. USART block diagram

USART supports both UART and USART functionality. UART is already explained in other sections. USART is going to explained in this section. USART module contains, Control and Status Registers(CSR), baud rate generator, auto flow control, tx and rx FIFOs, tx controller and rx controller.

This interface MUX muxes the interface signals from USART and UART. In sync_mode USART signal are given out on interface and interface signals are given to USART module and inputs to UART will be blocked. In async_mode UART signal are given out on interface and interface signals are given to UART module and inputs to USRT will be blocked.

USART is going to work in two modes.They are continuous clock mode and discontinuous clock mode.

Continuous Clock Mode:

Data from the external USART comes on rxd pin. Clock from the external USART comes on clk pin. These two are synchronized on using undivided clock (usart_clk). clk output of the synchronizer is registered once and passed through a multiplexer. In case of same edge sample and drive mode, this multiplexer is disabled to make sure the synchronized rxd (rxd_s) holds the valid data. The clk_s is passed through edge det logic where rising and falling edges are generated. These outputs are passed through a multiplexer to select the sampling edge. This edge is given to start/stop det logic where the rxd_s is also connected. For every pulse on the sampling signal rxd_s is seen and if it is matched with start_bit (usually 0), data_latch_en is path is activated and the sampling pulses from that point onwards are given to rx_shift_reg and rx_bit_cnt counter where rxd_s is sampled and bit cnt is incremented. When the number of bits is matched to the programmed value a fifo_wr signal is generated to capture the rx_shift_reg value into rx_fifo. Stop bit is detected by the start/stop det logic and given to bit_char_cnt. If stop bit is seen even before all the bits reception a frame error is generated and the rx_shift_reg is flushed and FIFO will not be filled. Second stop bit detection is ignored if two stop bits are adjacent (in case of 1.5/2 stop bits case, 2 stop bits might be seen some times). In Master mode case internally divided clock is used as input to edge det logic where as in Slave mode clk_s is used. Start/stop bit detection can be bypassed as to receive continuous data characters to increase the throughput.

Discontinuous Mode:

In this mode of operation, start/stop bits are not required and every data bit sampled on the sampling edge of clk is a valid data bit. In this mode, start/stop det logic can be fully bypassed. Fully divided clock is the actual baud clock where as fractional divided clock is the clock which is corrected clock to limit the error % with the actual baud rate.

16.17.4 Procedures to Use USRT

16.17.4.1 Full Duplex Mode

- Set sync_mode.
- Write data to TX FIFO.
- Write count to RX_CNT.
- Set auto_flow_ctrl if controlled transfers on interface.

16.17.4.2 Half Duplex Mode

RX mode

- Reset full_duplex_mode.
- Set rx_mode.
- Reset tx_mode.
- Write RX_CNT with number of bytes to be received.
- Read data from RX_FIFO by polling rx FIFO empty/aempty status.

TX mode

- Reset full_duplex_mode.
- Set tx_mode.
- Reset rx_mode.

Write data to TX_FIFO by polling the FIFO full/afull status.

Note: Serial clock selection and division factor register are mentioned here: Table 6.31 MCUHP_CLK_CONFIG_REG2 Description on page 85

16.17.5 Register Summary

UART Base Address: 0x4400_0000

USRT Base Address: 0x4400_0100

Table 16.456. Register Summary Table

Register Name	Offset	Description
Section 16.17.6.1 USART RECEIVE BUFFER REGISTER	0x0	Receive Buffer Register
Section 16.17.6.2 USART TRANSMIT HOLDING REGISTER	0x0	Transmit Holding Register
Section 16.17.6.3 USART DIVISOR LATCH LOW REGISTER	0x0	Divisor Latch (Low) Register
Section 16.17.6.4 USART DIVISOR LATCH HIGH REGISTER	0x4	Divisor Latch (High) Register
Section 16.17.6.5 USART INTERRUPT ENABLE REGISTER	0x4	Interrupt Enable Register
Section 16.17.6.6 USART INTERRUPT IDENTIFICATION REGISTER	0x8	Interrupt Identification Register
Section 16.17.6.7 USART FIFO CONTROL REGISTER	0x8	FIFO Control Register
Section 16.17.6.8 USART LINE CONTROL REGISTER	0xC	Line Control Register
Section 16.17.6.9 USART MODEM CONTROL REGISTER	0x10	Modem Control Register
Section 16.17.6.10 USART LINE STATUS REGISTER	0x14	Line Status Register
Section 16.17.6.11 USART MODEM STATUS REGISTER	0x18	Modem Status Register
Section 16.17.6.12 USART SCRATCHPAD REGISTER	0x1C	Scratchpad Register
Section 16.17.6.13 USART FDR REGISTER	0x28	Reserved
Section 16.17.6.14 USART HDEN REGISTER	0x40	Hardware Enable register
Section 16.17.6.15 USART SMCR REGISTER	0x58	Control register
Section 16.17.6.16 USART FIFO ACCESS REGISTER	0x70	FIFO Access Register
Section 16.17.6.17 USART TRANSMIT FIFO READ REGISTER	0x74	Transmit FIFO Read Register
Section 16.17.6.18 USART RECEIVE FIFO WRITE REGISTER	0x78	Receive FIFO Write Register
Section 16.17.6.19 USART STATUS REGISTER	0x7C	UART Status Register
Section 16.17.6.20 USART TRANSMIT FIFO LEVEL REGISTER	0x80	Transmit FIFO Level Register

Register Name	Offset	Description
Section 16.17.6.21 USART RECEIVE FIFO LEVEL REGISTER	0x84	Receive FIFO Level Register
Section 16.17.6.22 USART SOFTWARE RESET REGISTER	0x88	Software Reset Register
Section 16.17.6.23 USART SHADOW REQUEST TO SEND REGISTER	0x8C	Shadow Request to Send Register
Section 16.17.6.24 USART SHADOW BREAK CONTROL REGISTER	0x90	Shadow Break Control Register
Section 16.17.6.25 USART SHADOW DMA MODE REGISTER	0x94	Shadow DMA Mode Register
Section 16.17.6.26 USART SHADOW FIFO ENABLE REGISTER	0x98	Shadow FIFO Enable Register
Section 16.17.6.27 USART SHADOW RCVR TRIGGER REGISTER	0x9C	Shadow RCVR Trigger Register
Section 16.17.6.28 USART SHADOW TX EMPTY TRIGGER REGISTER	0xA0	Shadow TX Empty Trigger Register
Section 16.17.6.29 USART HALT TX REGISTER	0xA4	Halt TX Register
Section 16.17.6.30 USART DMA SOFTWARE ACKNOWLEDGE REGISTER	0xA8	DMA Software Acknowledge Register
Section 16.17.6.31 USART COMPONENT PARAMETER REGISTER	0xF4	Component Parameter Register
Section 16.17.6.32 UART COMPONENT VERSION REGISTER	0xF8	UART Component Version Register
Section 16.17.6.33 USART COMPONENT TYPE REGISTER	0xFC	Component Type Register

16.17.6 Register Description

Legend:
R = Read-only, W = Write-only, R/W = Read/Write

16.17.6.1 USART RECEIVE BUFFER REGISTER

Table 16.457. Receive Buffer Register Description

Bit	Access	Function	POR Value	Description
31:8	R	Reserved	0	Reserved

Bit	Access	Function	POR Value	Description
7:0	R	Receive Buffer Register	0x0	Data byte received on the serial input port (sin) in UART mode, or the serial infrared input (sir_in) in infrared mode. The data in this register is valid only if the Data Ready (DR) bit in the Line Status Register (LSR) is set. If in non-FIFO mode (FIFO_MODE = NONE) or FIFOs are disabled (FCR[0] set to 0), the data in the RBR must be read before the next data arrives, otherwise it is overwritten, resulting in an over-run error. If in FIFO mode (FIFO_MODE != NONE) and FIFOs are enabled (FCR[0] set to 1), this register accesses the head of the receive FIFO. If the receive FIFO is full and this register is not read before the next data character arrives, then the data already in the FIFO is preserved, but any incoming data are lost and an over-run error occurs.

16.17.6.2 USART TRANSMIT HOLDING REGISTER

Table 16.458. Transmit Holding Register Description

Bit	Access	Function	POR Value	Description
31:8	R	Reserved	0	Reserved
7:0	W	Transmit Holding Register	0x0	Data to be transmitted on the serial output port (sout) in UART mode or the serial infrared output (sir_out_n) in infrared mode. Data should only be written to the THR when the THR Empty (THRE) bit (LSR[5]) is set. If in non-FIFO mode or FIFOs are disabled (FCR[0] = 0) and THRE is set, writing a single character to the THR clears the THRE. Any additional writes to the THR before the THRE is set again causes the THR data to be overwritten. If in FIFO mode and FIFOs are enabled (FCR[0] = 1) and THRE is set, x number of characters of data may be written to the THR before the FIFO is full. The number x (default=16) is determined by the value of FIFO Depth that is set during configuration. Any attempt to write data when the FIFO is full results in the write data being lost.

16.17.6.3 USART DIVISOR LATCH LOW REGISTER

Table 16.459. Divisor Latch Low Register Description

Bit	Access	Function	POR Value	Description
31:8	R	Reserved	0	Reserved

Bit	Access	Function	POR Value	Description
7:0	R/W	Divisor Latch (Low)	0	Lower 8 bits of a 16-bit, read/write, Divisor Latch register that contains the baud rate divisor for the UART. The output baud rate is equal to the serial clock (pclk if one clock design, sclk if two clock design (CLOCK_MODE = Enabled) frequency divided by sixteen times the value of the baud rate divisor, as follows: baud rate = (serial clock freq) / (16 * divisor value). Note that with the Divisor Latch Registers (DLL and DLH) set to 0, the baud clock is disabled and no serial communications occur. Also, once the DLL is set, at least 8 clock cycles of the serial clock should be allowed to pass before transmitting or receiving data.

16.17.6.4 USART DIVISOR LATCH HIGH REGISTER

Table 16.460. Divisor Latch High Register Description

Bit	Access	Function	POR Value	Description
31:8	R	Reserved	0	Reserved
7:0	R/W	Divisor Latch (High)	0	Upper 8-bits of a 16-bit, read/write, Divisor Latch register that contains the baud rate divisor for the UART. The output baud rate is equal to the serial clock (pclk if one clock design, sclk if two clock design (CLOCK_MODE = Enabled) frequency divided by sixteen times the value of the baud rate divisor, as follows: baud rate = (serial clock freq) / (16 * divisor value). Note that with the Divisor Latch Registers (DLL and DLH) set to 0, the baud clock is disabled and no serial communications occur. Also, once the DLH is set, at least 8 clock cycles of the serial clock should be allowed to pass before transmitting or receiving data.

16.17.6.5 USART INTERRUPT ENABLE REGISTER

Table 16.461. Interrupt Enable Register Description

Bit	Access	Function	POR Value	Description
31:8	R	Reserved	0	Reserved
7	R/W	PTIME	0	Programmable THRE Interrupt Mode Enable that can be written to only when THRE_MODE_USER = Enabled, always readable. This is used to enable/disable the generation of THRE Interrupt. 0 – disabled 1 – enabled
6:4	R	Reserved	0	Reserved

Bit	Access	Function	POR Value	Description
3	R/W	EDSSI	0	Enable Modem Status Interrupt. This is used to enable/disable the generation of Modem Status Interrupt. This is the fourth highest priority interrupt. 0 – disabled 1 – enabled
2	R/W	ELSI	0	Enable Receiver Line Status Interrupt. This is used to enable/disable the generation of Receiver Line Status Interrupt. This is the highest priority interrupt. 0 – disabled 1 – enabled
1	R/W	ETBEI	0	Enable Transmit Holding Register Empty Interrupt. This is used to enable/disable the generation of Transmitter Holding Register Empty Interrupt. This is the third highest priority interrupt. 0 – disabled 1 – enabled
0	R/W	ERBFI	0	Enable Received Data Available Interrupt. This is used to enable/disable the generation of Received Data Available Interrupt and the Character Timeout Interrupt (if in FIFO mode and FIFOs enabled). These are the second highest priority interrupts. 0 – disabled 1 – enabled

16.17.6.6 USART INTERRUPT IDENTIFICATION REGISTER

Table 16.462. Interrupt Identity Register Description

Bit	Access	Function	POR Value	Description
31:8	R	Reserved	0	Reserved
7:6	R	FIFOs Enabled (or FIFOSE)	0	This is used to indicate whether the FIFOs are enabled or disabled. • 00 – disabled • 11 – enabled
5:4	R	Reserved	0	Reserved
3:0	R	Interrupt ID (or IID)	1	This indicates the highest priority pending interrupt which can be one of the following types: • 0000 – modem status • 0001 – no interrupt pending • 0010 – THR empty • 0100 – received data available • 0110 – receiver line status • 0111 – busy detect • 1100 – character timeout Bit 3 indicates an interrupt can only occur when the FIFOs are enabled and used to distinguish a Character Timeout condition interrupt.

16.17.6.7 USART FIFO CONTROL REGISTER

Table 16.463. FIFO Control Register Description

Bit	Access	Function	POR Value	Description
31:8	R	Reserved	0	Reserved
7:6	W	RCVR Trigger (or RT)	0	RCVR Trigger. This is used to select the trigger level in the receiver FIFO at which the Received Data Available Interrupt is generated. In auto flow control mode it is used to determine when the rts_n signal is de-asserted. It also determines when the dma_rx_req_n signal is asserted in certain modes of operation. The following trigger levels are supported: 00 – 1 character in the FIFO 01 – FIFO 1⁄4 full 10 – FIFO 1⁄2 full 11 – FIFO 2 less than full
5:4	W	TX Empty Trigger (or TET)	0	TX Empty Trigger. Writes have no effect when THRE_MODE_USER = Disabled. This is used to select the empty threshold level at which the THRE Interrupts are generated when the mode is active. It also determines when the dma_tx_req_n signal is asserted when in certain modes of operation. The following trigger levels are supported: 00 – FIFO empty 01 – 2 characters in the FIFO 10 – FIFO 1⁄4 full 11 – FIFO 1⁄2 full
3	W	DMA Mode (or DMAM)	0	DMA Mode. This determines the DMA signaling mode used for the dma_tx_req_n and dma_rx_req_n output signals when additional DMA handshaking signals are not selected (DMA_EXTRA = No). 0 – mode 0 1 – mode 1
2	W	XMIT FIFO Reset (or XFIFOR)	0	This resets the control portion of the transmit FIFO and treats the FIFO as empty. This also de-asserts the DMA TX request and single signals when additional DMA handshaking signals are selected (DMA_EXTRA = YES). Note that this bit is 'self-clearing'. It is not necessary to clear this bit.
1	W	RCVR FIFO Reset (or RFIFOR)	0	This resets the control portion of the receive FIFO and treats the FIFO as empty. This also de-asserts the DMA RX request and single signals when additional DMA handshaking signals are selected (DMA_EXTRA = YES). Note that this bit is 'self-clearing'. It is not necessary to clear this bit.

Bit	Access	Function	POR Value	Description
0	W	FIFO Enable (or FIFOE)	0	This enables/disables the transmit (XMIT) and receive (RCVR) FIFOs. Whenever the value of this bit is changed both the XMIT and RCVR controller portion of FIFOs is reset.

16.17.6.8 USART LINE CONTROL REGISTER

Table 16.464. Line Control Register Description

Bit	Access	Function	POR Value	Description
31:8	R	Reserved	0	Reserved
7	R/W	DLAB	0	Divisor Latch Access Bit. If UART_16550_COMPATIBLE = NO, then write-able only when UART is not busy (USR[0] is 0); otherwise always writable, always readable. This bit is used to enable reading and writing of the Divisor Latch register (DLL and DLH/LPDLL and LPDLH) to set the baud rate of the UART. This bit must be cleared after initial baud rate setup in order to access other registers.
6	R/W	Break (or BC)	0	Break Control Bit.This is used to cause a break condition to be transmitted to the receiving device. If set to 1, the serial output is forced to the spacing (logic 0) state. When not in Loopback Mode, as determined by MCR[4], the sout line is forced low until the Break bit is cleared. If SIR_MODE = Enabled and active (MCR[6] set to 1) the sir_out_n line is continuously pulsed. When in Loopback Mode, the break condition is internally looped back to the receiver and the sir_out_n line is forced low.
5	R/W	Stick Parity	0	If UART_16550_COMPATIBLE = NO, then write-able only when UART is not busy (USR[0] is 0); otherwise always writable, always readable. This bit is used to force parity value. When PEN, EPS, and Stick Parity are set to 1, the parity bit is transmitted and checked as logic 0. If PEN and Stick Parity are set to 1 and EPS is a logic 0, then parity bit is transmitted and checked as a logic 1. If this bit is set to 0, Stick Parity is disabled.
4	R/W	EPS	0	Even Parity Select. If UART_16550_COMPATIBLE = NO, then write-able only when UART is not busy (USR[0] is 0); otherwise always writable, always readable. This is used to select between even and odd parity, when parity is enabled (PEN set to 1). If set to 1, an even number of logic 1s is transmitted or checked. If set to 0, an odd number of logic 1s is transmitted or checked.
3	R/W	PEN	0	Parity Enable. If UART_16550_COMPATIBLE = NO, then write-able only when UART is not busy (USR[0] is 0); otherwise always writable, always readable. This bit is used to enable and disable parity generation and detection in transmitted and received serial character respectively. • 0 – parity disabled • 1 – parity enabled

Bit	Access	Function	POR Value	Description
2	R/W	STOP	0	Number of stop bits. If UART_16550_COMPATIBLE = NO, then writable only when UART is not busy (USR[0] is 0); otherwise always writable, always readable. This is used to select the number of stop bits per character that the peripheral transmits and receives. If set to 0, one stop bit is transmitted in the serial data. If set to 1 and the data bits are set to 5 (LCR[1:0] set to 0) one and a half stop bits is transmitted. Otherwise, two stop bits are transmitted. Note that regardless of the number of stop bits selected, the receiver checks only the first stop bit. • 0 – 1 stop bit • 1 – 1.5 stop bits when DLS (LCR[1:0]) is 0, else 2 stop bit NOTE: The STOP bit duration implemented by uart may appear longer due to idle time inserted between characters for some configurations and baud clock divisor values in the transmit direction.
1:0	R/W	DLS (or CLS, as used in legacy)	0	Data Length Select. If UART_16550_COMPATIBLE = NO, then writable only when UART is not busy (USR[0] is 0); otherwise always writable and readable. When DLS_E in LCR_EXT is set to 0, this register is used to select the number of data bits per character that the peripheral transmits and receives. The number of bits that may be selected are as follows: • 00 – 5 bits • 01 – 6 bits • 10 – 7 bits • 11 – 8 bits

16.17.6.9 USART MODEM CONTROL REGISTER

Table 16.465. ModemControl Register Description

Bit	Access	Function	POR Value	Description
31:6	R	Reserved	0	Reserved
5	R/W	AFCE	0	Auto Flow Control Enable. Write-able only when AFCE_MODE = Enabled, always readable. When FIFOs are enabled and the Auto Flow Control Enable (AFCE) bit is set, Auto Flow Control features are enabled • 0 – Auto Flow Control Mode disabled • 1 – Auto Flow Control Mode enabled

Bit	Access	Function	POR Value	Description
4	R/W	LoopBack (or LB)	0	LoopBack Bit. This is used to put the UART into a diagnostic mode for test purposes. If operating in UART mode (SIR_MODE != Enabled or not active, MCR[6] set to 0), data on the sout line is held high, while serial data output is looped back to the sin line, internally. In this mode all the interrupts are fully functional. Also, in loopback mode, the modem control inputs (dsr_n, cts_n, ri_n, dcd_n) are disconnected and the modem control outputs (dtr_n, rts_n, out1_n, out2_n) are looped back to the inputs, internally. If operating in infrared mode (SIR_MODE = Enabled AND active, MCR[6] set to 1), data on the sir_out_n line is held low, while serial data output is inverted and looped back to the sir_in line.
3	R/W	OUT2	0	OUT2. This is used to directly control the user-designated Output2 (out2_n) output. The value written to this location is inverted and driven out on out2_n, that is: • 0 – out2_n de-asserted (logic 1) • 1 – out2_n asserted (logic 0) Note that in Loopback mode (MCR[4] set to 1), the out2_n output is held inactive high while the value of this location is internally looped back to an input.
2	R/W	OUT1	0	OUT1. This is used to directly control the user-designated Output1 (out1_n) output. The value written to this location is inverted and driven out on out1_n, that is: • 0 – out1_n de-asserted (logic 1) • 1 – out1_n asserted (logic 0) Note that in Loopback mode (MCR[4] set to 1), the out1_n output is held inactive high while the value of this location is internally looped back to an input.
1	R/W	RTS	0	Request to Send. This is used to directly control the Request to Send (rts_n) output. The Request To Send (rts_n) output is used to inform the modem or data set that the UART is ready to exchange data. When Auto RTS Flow Control is not enabled (MCR[5] set to 0), the rts_n signal is set low by programming MCR[1] (RTS) to a high.In Auto Flow Control, AFCE_MODE = Enabled and active (MCR[5] set to 1) and FIFOs enable (FCR[0] set to 1), the rts_n output is controlled in the same way, but is also gated with the receiver FIFO threshold trigger (rts_n is inactive high when above the threshold) only when the RTS Flow Trigger is disabled; otherwise it is gated by the receiver FIFO almost-full trigger, where “almost full” refers to two available slots in the FIFO (rts_n is inactive high when above the threshold). The rts_n signal is de-asserted when MCR[1] is set low. Note that in Loopback mode (MCR[4] set to 1), the rts_n output is held inactive high while the value of this location is internally looped back to an input.

Bit	Access	Function	POR Val-ue	Description
0	R/W	DTR	0	Data Terminal Ready. This is used to directly control the Data Terminal Ready (dtr_n) output. The value written to this location is inverted and driven out on dtr_n, that is: • 0 – dtr_n de-asserted (logic 1) • 1 – dtr_n asserted (logic 0) The Data Terminal Ready output is used to inform the modem or data set that the UART is ready to establish communications. Note that in Loopback mode (MCR[4] set to 1), the dtr_n output is held inactive high while the value of this location is internally looped back to an input.

16.17.6.10 USART LINE STATUS REGISTER

Table 16.466. Line Status Register Description

Bit	Access	Function	POR Val-ue	Description
31:8	R	Reserved	0	Reserved
7	R	RFE	0	Receiver FIFO Error bit. This bit is only relevant when FIFO_MODE != NONE AND FIFOs are enabled (FCR[0] set to 1). This is used to indicate if there is at least one parity error, framing error, or break indication in the FIFO. 0 – no error in RX FIFO 1 – error in RX FIFO This bit is cleared when the LSR is read and the character with the error is at the top of the receiver FIFO and there are no subsequent errors in the FIFO.
6	R	TEMT	1	Transmitter Empty bit. If in FIFO mode (FIFO_MODE != NONE) and FIFOs enabled (FCR[0] set to 1), this bit is set whenever the Transmitter Shift Register and the FIFO are both empty. If in non-FIFO mode or FIFOs are disabled, this bit is set whenever the Transmitter Holding Register and the Transmitter Shift Register are both empty.
5	R	THRE	1	Transmit Holding Register Empty bit. If THRE_MODE_USER = Disabled or THRE mode is disabled (IER[7] set to 0) and regardless of FIFO's being implemented/ enabled or not, this bit indicates that the THR or TX FIFO is empty. This bit is set whenever data is transferred from the THR or TX FIFO to the transmitter shift register and no new data has been written to the THR or TX FIFO. This also causes a THRE Interrupt to occur, if the THRE Interrupt is enabled. If THRE_MODE_USER = Enabled AND FIFO_MODE != NONE and both modes are active (IER[7] set to 1 and FCR[0] set to 1 respectively), the functionality is switched to indicate the transmitter FIFO is full, and no longer controls THRE interrupts, which are then controlled by the FCR[5:4] threshold setting.

Bit	Access	Function	POR Value	Description
4	R	BI	0	Break Interrupt bit. This is used to indicate the detection of a break sequence on the serial input data. If in UART mode (SIR_MODE = Disabled), it is set whenever the serial input, sin, is held in a logic '0' state for longer than the sum of start time + data bits + parity + stop bits. If in infrared mode (SIR_MODE = Enabled), it is set whenever the serial input, sir_in, is continuously pulsed to logic '0' for longer than the sum of start time + data bits + parity + stop bits. A break condition on serial input causes one and only one character, consisting of all 0s, to be received by the UART. In FIFO mode, the character associated with the break condition is carried through the FIFO and is revealed when the character is at the top of the FIFO. Reading the LSR clears the BI bit. In non-FIFO mode, the BI indication occurs immediately and persists until the LSR is read. NOTE: If a FIFO is full when a break condition is received, a FIFO overrun occurs. The break condition and all the information associated with it—parity and framing errors—is discarded; any information that a break character was received is lost.
3	R	FE	0	Framing Error bit. This is used to indicate the occurrence of a framing error in the receiver. A framing error occurs when the receiver does not detect a valid STOP bit in the received data. In the FIFO mode, since the framing error is associated with a character received, it is revealed when the character with the framing error is at the top of the FIFO. When a framing error occurs, the uart tries to resynchronize. It does this by assuming that the error was due to the start bit of the next character and then continues receiving the other bit; that is, data, and/or parity and stop. It should be noted that the Framing Error (FE) bit (LSR[3]) is set if a break interrupt has occurred, as indicated by Break Interrupt (BI) bit (LSR[4]). This happens because the break character implicitly generates a framing error by holding the sin input to logic 0 for longer than the duration of a character. 0 – no framing error 1 – framing error Reading the LSR clears the FE bit.

Bit	Access	Function	POR Value	Description
2	R	PE	0	Parity Error bit. This is used to indicate the occurrence of a parity error in the receiver if the Parity Enable (PEN) bit (LCR[3]) is set. In the FIFO mode, since the parity error is associated with a character received, it is revealed when the character with the parity error arrives at the top of the FIFO. It should be noted that the Parity Error (PE) bit (LSR[2]) can be set if a break interrupt has occurred, as indicated by Break Interrupt (BI) bit (LSR[4]). In this situation, the Parity Error bit is set if parity generation and detection is enabled (LCR[3]=1) and the parity is set to odd (LCR[4]=0). 0 – no parity error 1 – parity error Reading the LSR clears the PE bit.
1	R	OE	0	Overrun error bit. This is used to indicate the occurrence of an overrun error. This occurs if a new data character was received before the previous data was read. In the non-FIFO mode, the OE bit is set when a new character arrives in the receiver before the previous character was read from the RBR. When this happens, the data in the RBR is overwritten. In the FIFO mode, an overrun error occurs when the FIFO is full and a new character arrives at the receiver. The data in the FIFO is retained and the data in the receive shift register is lost. 0 – no overrun error 1 – overrun error Reading the LSR clears the OE bit.
0	R	DR	0	Data Ready bit. This is used to indicate that the receiver contains at least one character in the RBR or the receiver FIFO. 0 – no data ready 1 – data ready This bit is cleared when the RBR is read in non-FIFO mode, or when the receiver FIFO is empty, in FIFO mode.

16.17.6.11 USART MODEM STATUS REGISTER

Table 16.467. Modem Status Register Description

Bit	Access	Function	POR Value	Description
31:8	R	Reserved	0	Reserved
7	R	DCD	0	Data Carrier Detect. This is used to indicate the current state of the modem control line dcd_n. This bit is the complement of dcd_n. When the Data Carrier Detect input (dcd_n) is asserted it is an indication that the carrier has been detected by the modem or data set. • 0 – dcd_n input is de-asserted (logic 1) • 1 – dcd_n input is asserted (logic 0) In Loopback Mode (MCR[4] set to 1), DCD is the same as MCR[3] (Out2).

Bit	Access	Function	POR Value	Description
6	R	RI	0	Ring Indicator. This is used to indicate the current state of the modem control line ri_n. This bit is the complement of ri_n. When the Ring Indicator input (ri_n) is asserted it is an indication that a telephone ringing signal has been received by the modem or data set. • 0 – ri_n input is de-asserted (logic 1) • 1 – ri_n input is asserted (logic 0) In Loopback Mode (MCR[4] set to 1), RI is the same as MCR[2] (Out1)
5	R	DSR	0	Data Set Ready. This is used to indicate the current state of the modem control line dsr_n. This bit is the complement of dsr_n. When the Data Set Ready input (dsr_n) is asserted it is an indication that the modem or data set is ready to establish communications with the uart. • 0 – dsr_n input is de-asserted (logic 1) • 1 – dsr_n input is asserted (logic 0) In Loopback Mode (MCR[4] set to 1), DSR is the same as MCR[0] (DTR).
4	R	CTS	0	This bit is the complement of cts_n. When the Clear to Send input (cts_n) is asserted it is an indication that the modem or data set is ready to exchange data with the uart. • 0 – cts_n input is de-asserted (logic 1) • 1 – cts_n input is asserted (logic 0) In Loopback Mode (MCR[4] = 1), CTS is the same as MCR[1] (RTS).
3	R	DDCD	0	Delta Data Carrier Detect. This is used to indicate that the modem control line dcd_n has changed since the last time the MSR was read. • 0 – no change on dcd_n since last read of MSR • 1 – change on dcd_n since last read of MSR Reading the MSR clears the DDCD bit. In Loopback Mode (MCR[4] = 1), DDCD reflects changes on MCR[3] (Out2). Note, if the DDCD bit is not set and the dcd_n signal is asserted (low) and a reset occurs (software or otherwise), then the DDCD bit is set when the reset is removed if the dcd_n signal remains asserted.
2	R	TERI	0	Trailing Edge of Ring Indicator. This is used to indicate that a change on the input ri_n (from an active-low to an inactive-high state) has occurred since the last time the MSR was read. • 0 – no change on ri_n since last read of MSR • 1 – change on ri_n since last read of MSR Reading the MSR clears the TERI bit. In Loopback Mode (MCR[4] = 1), TERI reflects when MCR[2] (Out1) has changed state from a high to a low.
1	R	DDSR	0	Delta Data Set Ready. This is used to indicate that the modem control line dsr_n has changed since the last time the MSR was read. • 0 – no change on dsr_n since last read of MSR • 1 – change on dsr_n since last read of MSR Reading the MSR clears the DDSR bit. In Loopback Mode (MCR[4] = 1), DDSR reflects changes on MCR[0] (DTR). Note, if the DDSR bit is not set and the dsr_n signal is asserted (low) and a reset occurs (software or otherwise), then the DDSR bit is set when the reset is removed if the dsr_n signal remains asserted.

Bit	Access	Function	POR Value	Description
0	R	DCTS	0	Delta Clear to Send. This is used to indicate that the modem control line cts_n has changed since the last time the MSR was read. • 0 – no change on cts_n since last read of MSR • 1 – change on cts_n since last read of MSR Reading the MSR clears the DCTS bit. In Loopback Mode (MCR[4] = 1), DCTS reflects changes on MCR[1] (RTS). Note, if the DCTS bit is not set and the cts_n signal is asserted (low) and a reset occurs (software or otherwise), then the DCTS bit is set when the reset is removed if the cts_n signal remains asserted.

16.17.6.12 USART SCRATCHPAD REGISTER

Table 16.468. Scratchpad Register Description

Bit	Access	Function	POR Value	Description
31:8	R	Reserved	0	Reserved
7:0	R/W	Scratchpad Register	0	This register is for programmers to use as a temporary storage space. It has no defined purpose in the UART.

16.17.6.13 USART FDR REGISTER

Table 16.469. FD Register Description

Bit	Access	Function	POR Value	Description
31:0	R/W	FDR register	0	This register is just written and read but not used anywhere in functionality. This is kept to make it compatible with USART standard

16.17.6.14 USART HDEN REGISTER

Table 16.470. HDEN Register Description

Bit	Access	Function	POR Value	Description
31:2	R	Reserved	0	Reserved
1	R/W	tx_mode/rx_mode	0	This signal is valid when full_duplex_mode is disabled 0 – tx_mode 1 – rx_mode
0	R/W	full_duplex_mode	0	0 – Full duplex mode enable 1 – Full duplex mode disable

16.17.6.15 USART SMCR REGISTER

Table 16.471. SMCR Register Description

Bit	Access	Function	POR Value	Description
31:6	R	Reserved	0	Reserved
5	R/W	start_stop_en	0	1 – Enable start stop 0 – Disable start stop
4	R/W	conti_clk_mode	0	1 – Continuous clock mode 0 – Non-continuous clock mode
3:2	R	Reserved	0	Reserved
1	R/W	mst_mode	0	1 – MST mode 0 – Non-MST mode
0	R/W	sync_mode	0	1 – Sync mode 0 – Non-Sync mode

16.17.6.16 USART FIFO ACCESS REGISTER

Table 16.472. FIFO Access Register Description

Bit	Access	Function	POR Value	Description
31:1	R	Reserved	0	Reserved
0	R/W	FIFO Access	0	Writes have no effect when FIFO_ACCESS = No, always readable. This register is use to enable a FIFO access mode for testing, so that the receive FIFO can be written by the Master and the transmit FIFO can be read by the Master when FIFOs are implemented and enabled. When FIFOs are not implemented or not enabled it allows the RBR to be written by the Master and the THR to be read by the Master. • 0 – FIFO access mode disabled • 1 – FIFO access mode enabled Note, that when the FIFO access mode is enabled/disabled, the control portion of the receive FIFO and transmit FIFO is reset and the FIFOs are treated as empty.

16.17.6.17 USART TRANSMIT FIFO READ REGISTER

Table 16.473. Transmit FIFO Read Register Description

Bit	Access	Function	POR Value	Description
31:8	R	Reserved	0	Reserved

Bit	Access	Function	POR Value	Description
7:0	R	Transmit FIFO Read	0	These bits are only valid when FIFO access mode is enabled (FAR[0] is set to 1). When FIFOs are implemented and enabled, reading this register gives the data at the top of the transmit FIFO. Each consecutive read pops the transmit FIFO and gives the next data value that is currently at the top of the FIFO. When FIFOs are not implemented or not enabled, reading this register gives the data in the THR.

16.17.6.18 USART RECEIVE FIFO WRITE REGISTER

Table 16.474. Receive FIFO Write Register Description

Bit	Access	Function	POR Value	Description
31:10	R	Reserved	0	Reserved
9	W	RFFE	0	Receive FIFO Framing Error. These bits are only valid when FIFO access mode is enabled (FAR[0] is set to 1). When FIFOs are implemented and enabled, this bit is used to write framing error detection information to the receive FIFO. When FIFOs are not implemented or not enabled, this bit is used to write framing error detection information to the RBR.
8	W	RFPE	0	Receive FIFO Parity Error. These bits are only valid when FIFO access mode is enabled (FAR[0] is set to 1). When FIFOs are implemented and enabled, this bit is used to write parity error detection information to the receive FIFO. When FIFOs are not implemented or not enabled, this bit is used to write parity error detection information to the RBR.
7:0	W	RFWD	0	Receive FIFO Write Data. These bits are only valid when FIFO access mode is enabled (FAR[0] is set to 1). When FIFOs are implemented and enabled, the data that is written to the RFWD is pushed into the receive FIFO. Each consecutive write pushes the new data to the next write location in the receive FIFO. When FIFOs are not implemented or not enabled, the data that is written to the RFWD is pushed into the RBR.

16.17.6.19 USART STATUS REGISTER

Table 16.475. UART Status Register Description

Bit	Access	Function	POR Value	Description
31:5	R	Reserved	0	Reserved

Bit	Access	Function	POR Value	Description
4	R	RFF	0	Receive FIFO Full. This bit is only valid when FIFO_STAT = YES. This is used to indicate that the receive FIFO is completely full. • 0 – Receive FIFO not full • 1 – Receive FIFO Full This bit is cleared when the RX FIFO is no longer full.
3	R	RFNE	0	Receive FIFO Not Empty. This bit is only valid when FIFO_STAT = YES. This is used to indicate that the receive FIFO contains one or more entries. • 0 – Receive FIFO is empty • 1 – Receive FIFO is not empty This bit is cleared when the RX FIFO is empty.
2	R	TFE	1	Transmit FIFO Empty. This bit is only valid when FIFO_STAT = YES. This is used to indicate that the transmit FIFO is completely empty. • 0 – Transmit FIFO is not empty • 1 – Transmit FIFO is empty This bit is cleared when the TX FIFO is no longer empty.
1	R	TFNF	1	Transmit FIFO Not Full. This bit is only valid when FIFO_STAT = YES. This is used to indicate that the transmit FIFO in not full. • 0 – Transmit FIFO is full • 1 – Transmit FIFO is not full This bit is cleared when the TX FIFO is full.
0	R	BUSY	0	UART Busy. This bit is valid only when UART_16550_COMPATIBLE = NO and indicates that a serial transfer is in progress; when cleared, indicates that the uart is idle or inactive. • 0 – uart is idle or inactive • 1 – uart is busy (actively transferring data) This bit will be set to 1 (busy) under any of the following conditions: 1. Transmission in progress on serial interface 2. Transmit data present in THR, when FIFO access mode is not being used (FAR = 0) and the baud divisor is non-zero ({DLH,DLL} does not equal 0) when the divisor latch access bit is 0 (LCR.DLAB = 0) 3. Reception in progress on the interface 4. Receive data present in RBR, when FIFO access mode is not being used (FAR = 0) NOTE: It is possible for the UART Busy bit to be cleared even though a new character may have been sent from another device. That is, if the uart has no data in THR and RBR and there is no transmission in progress and a start bit of a new character has just reached the uart. This is due to the fact that a valid start is not seen until the middle of the bit period and this duration is dependent on the baud divisor that has been programmed. If a second system clock has been implemented (CLOCK_MODE = Enabled), the assertion of this bit is also delayed by several cycles of the slower clock.

16.17.6.20 USART TRANSMIT FIFO LEVEL REGISTER

Table 16.476. Transmit FIFO Level Register Description

Bit	Access	Function	POR Value	Description
31:30	R	Reserved	0	Reserved
29:0	R	Transmit FIFO Level	0	This indicates the number of data entries in the transmit FIFO.

16.17.6.21 USART RECEIVE FIFO LEVEL REGISTER

Table 16.477. Receive FIFO Level Register Description

Bit	Access	Function	POR Value	Description
• 31:30	R	Reserved	0	Reserved
29:0	R	Receive FIFO Level	0	This indicates the number of data entries in the receive FIFO.

16.17.6.22 USART SOFTWARE RESET REGISTER

Table 16.478. Software Reset Register Description

Bit	Access	Function	POR Value	Description
31:3	R	Reserved	0	Reserved
2	W	XFR	0	XMIT FIFO Reset. This is a shadow register for the XMIT FIFO Reset bit (FCR[2]). This can be used to remove the burden on software having to store previously written FCR values (which are pretty static) just to reset the transmit FIFO. This resets the control portion of the transmit FIFO and treats the FIFO as empty. This also de-asserts the DMA TX request and single signals when additional DMA handshaking signals are selected (DMA_EXTRA = YES). Note that this bit is 'self-clearing'. It is not necessary to clear this bit.
1	W	RFR	0	RCVR FIFO Reset. This is a shadow register for the RCVR FIFO Reset bit (FCR[1]). This can be used to remove the burden on software having to store previously written FCR values (which are pretty static) just to reset the receive FIFO This resets the control portion of the receive FIFO and treats the FIFO as empty. This also de-asserts the DMA RX request and single signals when additional DMA handshaking signals are selected (DMA_EXTRA = YES). Note that this bit is 'self-clearing'. It is not necessary to clear this bit.
0	W	UR	0	UART Reset. This asynchronously resets the uart and synchronously removes the reset assertion. For a two clock implementation both pclk and sclk domains are reset.

16.17.6.23 USART SHADOW REQUEST TO SEND REGISTER

Table 16.479. Shadow Request to Send Register Description

Bit	Access	Function	POR Value	Description
31:1	R	Reserved	0	Reserved
0	R/W	Shadow Request to Send	0	This is a shadow register for the RTS bit (MCR[1]), this can be used to remove the burden of having to performing a read-modify-write on the MCR. This is used to directly control the Request to Send (rts_n) output. The Request To Send (rts_n) output is used to inform the modem or data set that the uart is ready to exchange data. When Auto RTS Flow Control is not enabled (MCR[5] = 0), the rts_n signal is set low by programming MCR[1] (RTS) to a high. In Auto Flow Control, AFCE_MODE = Enabled and active (MCR[5] = 1) and FIFOs enable (FCR[0] = 1), the rts_n output is controlled in the same way, but is also gated with the receiver FIFO threshold trigger (rts_n is inactive high when above the threshold) only when RTS Flow Trigger is disabled; otherwise it is gated by the receiver FIFO almost-full trigger, where “almost full” refers to two available slots in the FIFO (rts_n is inactive high when above the threshold). Note that in Loopback mode (MCR[4] = 1), the rts_n output is held inactive-high while the value of this location is internally looped back to an input.

16.17.6.24 USART SHADOW BREAK CONTROL REGISTER

Table 16.480. Shadow Break Control Register Description

Bit	Access	Function	POR Value	Description
31:1	R	Reserved	0	Reserved
0	R/W	Shadow Break Control Register	0	This is a shadow register for the Break bit (LCR[6]), this can be used to remove the burden of having to performing a read modify write on the LCR. This is used to cause a break condition to be transmitted to the receiving device. If set to 1, the serial output is forced to the spacing (logic 0) state. When not in Loopback Mode, as determined by MCR[4], the sout line is forced low until the Break bit is cleared. If SIR_MODE = Enabled and active (MCR[6] = 1) the sir_out_n line is continuously pulsed. When in Loopback Mode, the break condition is internally looped back to the receiver.

16.17.6.25 USART SHADOW DMA MODE REGISTER

Table 16.481. Shadow DMA Mode Register Description

Bit	Access	Function	POR Value	Description
31:1	R	Reserved	0	Reserved

Bit	Access	Function	POR Value	Description
0	R/W	Shadow DMA Mode	0	This is a shadow register for the DMA mode bit (FCR[3]). This can be used to remove the burden of having to store the previously written value to the FCR in memory and having to mask this value so that only the DMA Mode bit gets updated. This determines the DMA signalling mode used for the dma_tx_req_n and dma_rx_req_n output signals when additional DMA handshaking signals are not selected (DMA_EXTRA = NO). • 0 – mode 0 • 1 – mode 1

16.17.6.26 USART SHADOW FIFO ENABLE REGISTER

Table 16.482. Shadow FIFO Enable Register Description

Bit	Access	Function	POR Value	Description
31:1	R	Reserved	0	Reserved
0	R/W	Shadow FIFO Enable	0	Shadow FIFO Enable. This is a shadow register for the FIFO enable bit (FCR[0]). This can be used to remove the burden of having to store the previously written value to the FCR in memory and having to mask this value so that only the FIFO enable bit gets updated.This enables/disables the transmit (XMIT) and receive (RCVR) FIFOs. If this bit is set to 0 (disabled) after being enabled then both the XMIT and RCVR controller portion of FIFOs are reset.

16.17.6.27 USART SHADOW RCVR TRIGGER REGISTER

Table 16.483. Shadow RCVR Trigger Register Description

Bit	Access	Function	POR Value	Description
31:2	R	Reserved	0	Reserved
1:0	R/W	Shadow RCVR Trigger	0	Shadow RCVR Trigger. This is a shadow register for the RCVR trigger bits (FCR[7:6]). This can be used to remove the burden of having to store the previously written value to the FCR in memory and having to mask this value so that only the RCVR trigger bit gets updated. This is used to select the trigger level in the receiver FIFO at which the Received Data Available Interrupt is generated. It also determines when the dma_rx_req_n signal is asserted when DMA Mode (FCR[3]) = 1. The following trigger levels are supported: • 00 – 1 character in the FIFO • 01 – FIFO ¼ full • 10 – FIFO ½ full • 11 – FIFO 2 less than full

16.17.6.28 USART SHADOW TX EMPTY TRIGGER REGISTER

Table 16.484. Shadow TX Empty Trigger Register Description

Bit	Access	Function	POR Value	Description
31:2	R	Reserved	0	Reserved
1:0	R/W	Shadow TX Empty Trigger	0	Shadow TX Empty Trigger. This is a shadow register for the TX empty trigger bits (FCR[5:4]). This can be used to remove the burden of having to store the previously written value to the FCR in memory and having to mask this value so that only the TX empty trigger bit gets updated. This is used to select the empty threshold level at which the THRE Interrupts are generated when the mode is active. The following trigger levels are supported: • 00 – FIFO empty • 01 – 2 characters in the FIFO • 10 – FIFO ¼ full • 11 – FIFO ½ full

16.17.6.29 USART HALT TX REGISTER

Table 16.485. HALT TX Register Description

Bit	Access	Function	POR Value	Description
31:1	R	Reserved	0	Reserved
0	R/W	HALT TX	0	This register is use to halt transmissions for testing, so that the transmit FIFO can be filled by the Master when FIFOs are implemented and enabled. • 0 – Halt TX disabled • 1 – Halt TX enabled Note, if FIFOs are implemented and not enabled, the setting of the halt TX register has no effect on operation.

16.17.6.30 USART DMA SOFTWARE ACKNOWLEDGE REGISTER

Table 16.486. DMA Software Acknowledgment Register Description

Bit	Access	Function	POR Value	Description
31:1	R	Reserved	0	Reserved
0	W	DMA Software Acknowledge	0	This register is use to perform a DMA software acknowledge if a transfer needs to be terminated due to an error condition. For example, if the DMA disables the channel, then the uart should clear its request. This causes the TX request, TX single, RX request and RX single signals to de-assert. Note that this bit is 'self-clearing'. It is not necessary to clear this bit.

16.17.6.31 USART COMPONENT PARAMETER REGISTER

Table 16.487. Component Parameter Register Description

Bit	Access	Function	POR Value	Description
31:24	R	Reserved	0	Reserved
23:16	R	FIFO_MODE	0x01	0x00 = 0 0x01 = 16 0x02 = 32 to 0x80 = 2048 0x81- 0xff = reserved
15:14	R	Reserved	0	Reserved
13	R	DMA_EXTRA	0x1	0 – FALSE 1 – TRUE
12	R	UART_ADD_ENCODED_PARAMS	0x0	0 – FALSE 1 – TRUE
11	R	SHADOW	0x0	0 – FALSE 1 – TRUE
10	R	FIFO_STAT	0x1	0 – FALSE 1 – TRUE
9	R	FIFO_ACCESS	0x0	0 – FALSE 1 – TRUE
8	R	ADDITIONAL_FEAT	0x1	0 – FALSE 1 – TRUE
7	R	SIR_LP_MODE	0x1	0 – FALSE 1 – TRUE
6	R	SIR_MODE	0x1	0 – FALSE 1 – TRUE
5	R	THRE_MODE	0x1	0 – FALSE 1 – TRUE
4	R	AFCE_MODE	0x1	0 – FALSE 1 – TRUE
3:2	R	Reserved	0	Reserved
1:0	R	APB_DATA_WIDTH	0x10	00 – 8 bits 01 – 16 bits 10 – 32 bits 11 – reserved

16.17.6.32 UART COMPONENT VERSION REGISTER

Table 16.488. UART Component Version Register Description

Bit	Access	Function	POR Value	Description
31:0	R	UART Component Version	0x3430302a	UART Component version

16.17.6.33 USART COMPONENT TYPE REGISTER

Table 16.489. Component Type Register Description

Bit	Access	Function	POR Value	Description
31:0	R	Peripheral ID	0x44570110	This register contains the peripherals identification code

16.18 Sensor Data Collector (SDC)

16.18.1 General Description

Sensor Data Collector (SDC) is a low energy sensor sample collection mode where AUX-ADC is used for sample collection. It also has the option that utilizes on chip analog peripheral such as OPAMP to perform measurements. The result from measurement will be stored in a buffer of 16 samples to be used by MCU for further processing.

16.18.2 Features

- Low energy sensor block which uses ADC, utilizes on chip OPAMP for measurement
- Support for 13 Single-Ended mode Configuration
  - 12 External Inputs
  - 1 Internal Inputs
    - Opamp along with ADC
- Support for 7 Differential Configuration.
- Stores 16 samples of ADC data in internal Buffer for 1 Channel.
- Stores 8 samples of ADC data in internal Buffer for 2 Channel.
- Stores 4 samples of ADC data in internal Buffer for 4 Channel.
- Initiates Interrupt/wakeup once Internal Buffer threshold has reached.

16.18.3 Functional Description

16.18.3.1 Block Diagram

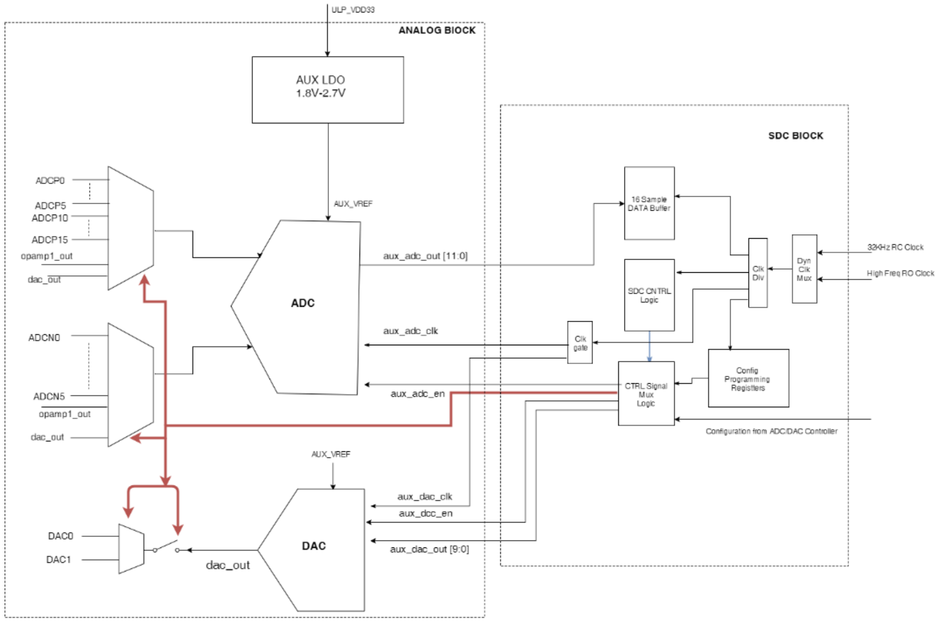


Figure 16.44. Block Diagram of Sensor Data Collector (SDC)

16.18.3.2 GPIO MUXING

Please refer to ULP GPIO and analog pin multiplexing sections in datasheet chapter for more information on analog pin multiplexing.

16.18.3.3 Clock Section

SDC clock source should be selected for initiating configuration register programming. There are 2 clock option for SDC, 32MHz RC clock and High Frequency RO clock.

The following procedure should be used for clock source selection.

- Configure **SDCSS_CLK_SEL** bit to select 32MHz RC clock in register **MCU_FSM_REF_CLK**
- Enable **SDCSS_CLK_EN** bit in register **MCU_FSM_REF_CLK**
- Enable **SDCSS_STATIC_CLK_EN** bit in register **MCU_FSM_REF_CLK**

Once the configuration register are programmed, program **SDCSS_STATIC_CLK_EN** bit as '0' in register **MCU_FSM_REF_CLK** to disable free running clock. Clock will be enabled once the configured trigger event occurs and it will remain, till sampling operation is completed.

16.18.3.4 Programming Sequence

1. Configure SDC clock source to initiate programming for SDC Configuration.
    a. Configure **SDCSS_CLK_SEL** bit to select 32MHz RC clock in register **MCU_FSM_REF_CLK**
    b. Enable **SDCSS_CLK_EN** bit in register **MCU_FSM_REF_CLK**
    c. Enable **SDCSS_STATIC_CLK_EN** bit in register **MCU_FSM_REF_CLK**
2. Configure SDCSS Parameters
    a. Program **SAMP_THRESH** bits in register **SDC_GEN_CONFIG_1** for number of sample required after which wakeup is initiated.
    b. Program **NUM_CH_SEL** bits in register **SDC_GEN_CONFIG_2** for selecting number of channel that need to be swept for sample collection.
    c. Program **CNT_TRIG_EVNT** bits in register **SDC_GEN_CONFIG_3**, which indicated the number of trigger event that need to be skipped.
    d. Select Trigger event on which AUX-ADC data should be sampled by programming **SAMP_TRIG_SEL** bits in register **SDC_GEN_CONFIG_3**.
3. AUX-ADC Should be calibrated before triggering SDC block. Refer to Analog to Digital converter section in Analog peripheral chapter for calibration procedure.
4. Configure Aux-ADC parameters
    a. Program SDC_AUXADC_INPUT_N_SEL_CH1 , SDC_AUXADC_INPUT_P_SEL_CH1 , SDC_AUXADC_DIFF_MODE_CH1 bits in register **SDC_AUXADC_CONFIG_1** for Channel-1 ADC selection. Refer to Analog to Digital converter section in Analog peripheral chapter for AUX-ADC Channel selection.
    b. Program SDC_AUXADC_INPUT_N_SEL_CH2 , SDC_AUXADC_INPUT_P_SEL_CH2 , SDC_AUXADC_DIFF_MODE_CH2 bits in register **SDC_AUXADC_CONFIG_2** for Channel-2 ADC selection. Refer to Analog to Digital converter section in Analog peripheral chapter for AUX-ADC Channel selection.
    c. Program SDC_AUXADC_INPUT_N_SEL_CH3 , SDC_AUXADC_INPUT_P_SEL_CH3 , SDC_AUXADC_DIFF_MODE_CH3 bits in register **SDC_AUXADC_CONFIG_3** for Channel-3 ADC selection. Refer to Analog to Digital converter section in Analog peripheral chapter for AUX-ADC Channel selection.
    d. Program SDC_AUXADC_INPUT_N_SEL_CH4 , SDC_AUXADC_INPUT_P_SEL_CH4 , SDC_AUXADC_DIFF_MODE_CH4 bits in register **SDC_AUXADC_CONFIG_4** for Channel-3 ADC selection. Refer to Analog to Digital converter section in Analog peripheral chapter for AUX-ADC Channel selection.
    e. Program SDC_AUXADC_EN bits in register **SDC_AUXADC_CONFIG_1** for enabling AUX-ADC through SDC block.
    f. Program SDC_ADC_CONFIG_EN bits in register **SDC_AUXADC_CONFIG_1** for enabling SDC AUX-ADC Configuration .
5. Configure OP-AMP parameters if Op-AMP output need to feed to ADC.
    a. Program SDC_OPAMP_IN_N_SEL bit in register **SDC_AUXOPAMP_CONFIG_1** for selecting N-Channel of OP-AMP1. N Channel is common for all 4 Channel. Refer to OPAMP section in Analog peripheral chapter for OPAMP Channel selection.
    b. Program SDC_OPAMP_IN_P_SEL_CH1, SDC_OPAMP_EN_CH1 bits in register **SDC_AUXOPAMP_CONFIG_1** for selecting P-Channel of OP-AMP1.
    c. Program SDC_OPAMP_IN_P_SEL_CH2, SDC_OPAMP_EN_CH2 bits in register **SDC_AUXOPAMP_CONFIG_2** for selecting P-Channel of OP-AMP1.
    d. Program SDC_OPAMP_IN_P_SEL_CH3, SDC_OPAMP_EN_CH3 bits in register **SDC_AUXOPAMP_CONFIG_3** for selecting P-Channel of OP-AMP1.
    e. Program SDC_OPAMP_IN_P_SEL_CH4, SDC_OPAMP_EN_CH4 bits in register **SDC_AUXOPAMP_CONFIG_4** for selecting P-Channel of OP-AMP1.
    f. Program SDC_OPAMP_CONFIG_EN bit in register **SDC_AUXOPAMP_CONFIG_1** for enabling SDC OPAMP Configuration .
6. Configure ADC Clock Division factor to generated clock to ADC by programming SDC_CLK_DIV bits in register **SDC_GEN_CONFIG_3**.
7. Enable Data Collection
    a. Program SDC_SAMP_EN bits in register **SDC_GEN_CONFIG_2**
8. Disable **SDCSS_STATIC_CLK_EN** bit in register address **MCU_FSM_REF_CLK**
9. Initiate Sleep sequence for entering PS1 with SDC a wakeup source.
10. Upon Wakeup read register SDC_DATA_REG0 to SDC_DATA_REG15. These register will contain sample collected from AUX-ADC and samples that are needed will be moved to ULP SRAM Banks.
11. Re-Initiating the SDC block
    a. Program RST_WRT_PTR bit in register **SDC_GEN_CONFIG_1** to reset the write pointer of FIFO.
    b. Program INTR_STATUS_CLEAR bit in register **SDC_GEN_CONFIG_0** to clear interrupt to NVIC.
12. For Collecting mode sample Re-Initiate Sleep Sequence.

16.18.4 Register Summary

Base Address: 0x2404_8100

Table 16.490. SDC Clock Register Summary

Register Name	Offset	Description
Section 16.18.5.1 MCU_FSM_REF_CLK	0x1C	

Base Address: 0x2404_2400

Table 16.491. SDC Register Summary

Register Name	Offset
Section 16.18.5.2 SDC_GEN_CONFIG_0	0x00
Section 16.18.5.3 SDC_GEN_CONFIG_1	0x04
Section 16.18.5.4 SDC_GEN_CONFIG_2	0x08
Section 16.18.5.5 SDC_GEN_CONFIG_3	0x0C
Section 16.18.5.6 SDC_AUXADC_CONFIG_1	0x18
Section 16.18.5.7 SDC_AUXLDO_CONFIG	0x20
Section 16.18.5.8 SDC_AUXOPAMP_CONFIG_1	0x24
Section 16.18.5.9 SDC_AUXADC_CONFIG_2	0x28
Section 16.18.5.10 SDC_AUXADC_CONFIG_3	0x2C
Section 16.18.5.11 SDC_AUXADC_CONFIG_4	0x30
Section 16.18.5.12 SDC_AUXOPAMP_CONFIG_2	0x34
Section 16.18.5.13 SDC_DATA_REG0	0x38
Section 16.18.5.14 SDC_DATA_REG1	0x3C
Section 16.18.5.15 SDC_DATA_REG2	0x40
Section 16.18.5.16 SDC_DATA_REG3	0x44
Section 16.18.5.17 SDC_DATA_REG4	0x48
Section 16.18.5.18 SDC_DATA_REG5	0x4C
Section 16.18.5.19 SDC_DATA_REG6	0x50
Section 16.18.5.20 SDC_DATA_REG7	0x54
Section 16.18.5.21 SDC_DATA_REG8	0x58
Section 16.18.5.22 SDC_DATA_REG9	0x5C
Section 16.18.5.23 SDC_DATA_REG10	0x60
Section 16.18.5.24 SDC_DATA_REG11	0x64
Section 16.18.5.25 SDC_DATA_REG12	0x68
Section 16.18.5.26 SDC_DATA_REG13	0x6C
Section 16.18.5.27 SDC_DATA_REG14	0x70
Section 16.18.5.28 SDC_DATA_REG15	0x74

16.18.5 Register Description

16.18.5.1 MCU_FSM_REF_CLK

Table 16.492. MCU_FSM_REF_CLK

Bit	Access	Function	Reset Value	Description
31	R/W	SDCSS_STATIC_CLK_EN	0	To enable static clk for sensor data collector subsystem
30	R/W	SDCSS_CLK_EN	0	To enable dynamic clock for sdcss
29:28	R/W	SDCSS_CLK_SEL	0	SDCSS Clock Selection to be used for Configuration 01 – 32MHz RC Clock 10 – High Frequency RO Clock
27:25				
24	R/W	ULPSS_REF_CLK_CLNR_ON	1	Clk cleaner On signal for ulpss ref clock
23	R/W	ULPSS_REF_CLK_CLNR_OFF	0	clk cleaner Off signal for ulpss ref clock
22:19				
18:16	R/W	ULPSS_REF_CLK_SEL	1	Dynamic Reference Clock Mux select of ULPSS 0 : Clock will be gated at dynamic mux output of ULPSS 1 : ulp_32mhz_rc_byp_clk 2 : ulp_32mhz_rc_clk 3 : rf_ref_clk 4 : mems_ref_clk 5 : ulp_20mhz_ringosc_clk 6 : ulp_doubler_clk 7 : ref_byp_clk to NWP
15				
14:12	R/W	TASS_REF_CLK_SEL	1	Dynamic Reference Clock Mux select of NWP controlled by M4. 0 : Clock will be gated at dynamic mux output of NWP 1 : ulp_32mhz_rc_byp_clk 2 : ulp_32mhz_rc_clk 3 : rf_ref_clk 4 : mems_ref_clk 5 : ulp_20mhz_ringosc_clk 6 : ref_byp_clk to NWP
11:9				
8	R/W	M4SS_REF_CLK_CLNR_ON	1	Enable clk cleaner for m4ss reference clock

Bit	Access	Function	Reset Value	Description
7	R/W	M4SS_REF_CLK_CLNR_OFF	0	Disable signal for m4ss reference clock
6:3				
2:0	R/W	M4SS_REF_CLK_SEL	1	Dynamic Reference Clock Mux select of M4SS 0 : Clock will be gated at dynamic mux output of M4SS 1 : ulp_32mhz_rc_byp_clk 2 : ulp_32mhz_rc_clk 3 : rf_ref_clk 4 : mems_ref_clk 5 : ulp_20mhz_ringosc_clk 6 : ulp_doubler_clk 7 : ref_byp_clk to NWP

16.18.5.2 SDC_GEN_CONFIG_0

Table 16.493. SDC_GEN_CONFIG_0

Bit	Access	Function	Reset Value	Description
31:1				
0	R/W	INTR_STATUS_CLEAR	0	Reading this register will returns SDC's interrupt status. Writing 1 to this register will clear interrupt.

16.18.5.3 SDC_GEN_CONFIG_1

Table 16.494. SDC_GEN_CONFIG_1

Bit	Access	Function	Reset Value	Description
31:9				
8:5	R/W	SAMP_THRESH	0	Number of data sampled to be collected from Aux-ADC and stored in Buffer before interrupt is raised/ wakeup is initialed. If SAMP_THRESH is 1, then 2 samples are stored in the SDC data register and wake up is initaited SAMP_THRESH value 15 is invalid. Valid values are 0 to 14
4:1	R	WRT_PTR	0	Write pointer Value
0	R/W	RST_WRT_PTR	0	Writing 1 to this register will resets the write pointer so that new samples can be filled in Buffer.

16.18.5.4 SDC_GEN_CONFIG_2

Table 16.495. SDC_GEN_CONFIG_2

Bit	Access	Function	Reset Value	Description
31:4				
3:1	R/W	NUM_CH_SEL	0	Number of Channels to be used 0 - Single channel is used 1 - Two channels are used 2 - Three channels are used 3 - Four channels are used Ex: If 0 , Only 1 sample is stored into the buffer for Channel-1's configuration on trigger event. Ex: If 1, 2 sampled will be stored into the buffer. 1st sample will be for Channel-1's configuration and 2nd sampled will be for Channel-2's Configuration
0	R/W	SDC_SAMP_EN	0	SDC Data Sampling mode 1 - Enable 0 - Disable

16.18.5.5 SDC_GEN_CONFIG_3

Table 16.496. SDC_GEN_CONFIG_3

Bit	Access	Function	Reset Value	Description
31:21				
20:11	R/W	SDC_CLK_DIV	0	SDCSS clock division factor 0: Passthrough N: Division by 2N Note: Value should not exceed 160 value equal for generating 20Khz clock.
10:1	R/W	CNT_TRIG_EVNT	0	The register will indicate in which trigger event AUX-ADC Data will sampled 0 - Sample Data on every Trigger Event 1 - Sample Data on every alternate Trigger Event 2 - Sample Data on every 3rd Trigger Event 3 - Sample Data on every 4th Trigger Event and so on.
0	R/W	SAMP_TRIG_SEL	0	The register is used to select the trigger event on which AUX-ADC Data is sampled 1 – 1ms Pulse will be used as Trigger event 0 – 1sec Pulse will be used as Trigger event Note : Calendar/RTC Block should be configured for generating 1 milli-sec and 1 sec pulse/

16.18.5.6 SDC_AUXADC_CONFIG_1

Table 16.497. SDC_AUXADC_CONFIG_1

Bit	Access	Function	Reset Value	Description
31:12				
11	R/W	SDC_ADC_CONFIG_EN	0	On Enabling this register, SDC ADC Configuration will be Applied.
10	R/W	SDC_AUXADC_EN	0	AUXADC Enable from SDC Block
9	R/W	SDC_AUXADC_DIFF_MODE_CH1	0	Enable Differential Mode in AUX ADC for Channel -1
8:5	R/W	SDC_AUXADC_INPUT_N_SEL_CH1	0	AUXADC's Negative Input Mux Select for Channel-1 Please refer to Input selection section in Analog to Digital conversion chapter for programming options.
4:0	R/W	SDC_AUXADC_INPUT_P_SEL_CH1	0	AUXADC's Positive Input Mux Select for Channel-1 Please refer to Input selection section in Analog to Digital conversion chapter for programming options.

16.18.5.7 SDC_AUXLDO_CONFIG

Table 16.498. SDC_AUXLDO_CONFIG

Bit	Access	Function	Reset Value	Description
31:8				
7	R/W	SDC_AUXLDO_CONFIG_EN	0	SDC Aux LDO Configuration Control Enable
6	R/W	SDC_AUXLDO_EN	0	Turn-On AUX LDO 1 - Turn-ON 0 - Turn-OFF
5	R/W	SDC_AUXLDO_BYP_EB	0	Configure AUXLDO in Buypass mode. When Enabled, Ouput supply of LDO will be same as Input supply.
4				
3:0	R/W	SDC_AUXLDO_VOLT_CTRL	0	SDC AUX LDO Voltage Control Selection 0 - 1.60v 1 - 1.68v 2 - 1.76v 3 - 1.84v 4 - 1.92v 5 - 2.00v 6 - 2.08v 7 - 2.16v 8 - 2.24v 9 - 2.32v 10 - 2.4v 11 - 2.48v 12 - 2.56v 13 - 2.64v 14 - 2.72v 15 - 2.90v

16.18.5.8 SDC_AUXOPAMP_CONFIG_1

Table 16.499. SDC_AUXOPAMP_CONFIG_1

Bit	Access	Function	Reset Value	Description
31	R/W	SDC_OPAMP_CONFIG_EN	0	On Enabling this register, SDC OPAMP Configuration will be Applied.
30	R/W	SDC_VREF_MUX_4_SEL	0	Selection register for choosing Voltage reference to external use on GPIO(GPIO_30) 1 – AUX_Vref 0 – 1.0v
29	R/W	SDC_VREF_MUX_3_SEL	0	Selection register for choosing Voltage reference to external use on GPIO(ULP_GPIO_4) 1 – AUX_Vref 0 – 1.0v
28	R/W	SDC_VREF_MUX_2_SEL	0	Selection register for choosing Voltage reference to external use on GPIO(ULP_GPIO_3) 1 – AUX_Vref 0 – 1.0v
27	R/W	SDC_VREF_MUX_1_SEL	0	Selection register for choosing Voltage reference to external use on GPIO(ULP_GPIO_1) 1 – AUX_Vref 0 – 1.0v
26				
25	R/W	SDC_VREF_MUX_4_EN	0	Enable for connecting Low Drive strength Voltage reference to GPIO for external use. 1 – Connect Voltage reference to GPIO(GPIO_30) 0 – No Connect to GPIO Please refer to Analog Functions mapping section in Pin Multiplexing for MCU and WiSeMCU chapter for more description.

Bit	Access	Function	Reset Value	Description
24	R/W	SDC_VREF_MUX_3_EN	0	Enable for connecting Low Drive strength Voltage reference to GPIO for external use. 1 – Connect Voltage reference to GPIO(ULP_GPIO_4) 0 – No Connect to GPIO Please refer to Analog Functions mapping section in Pin Multiplexing for MCU and WiSeMCU chapter for more description.
23	R/W	SDC_VREF_MUX_3_EN	0	Enable for connecting Low Drive strength Voltage reference to GPIO for external use. 1 – Connect Voltage reference to GPIO(ULP_GPIO_3) 0 – No Connect to GPIO Please refer to Analog Functions mapping section in Pin Multiplexing for MCU and WiSeMCU chapter for more description.
22	R/W	SDC_VREF_MUX_1_EN	0	Enable for connecting Low Drive strength Voltage reference to GPIO for external use. 1 – Connect Voltage reference to GPIO(ULP_GPIO_1) 0 – No Connect to GPIO Please refer to Analog Functions mapping section in Pin Multiplexing for MCU and WiSeMCU chapter for more description.
21	R	Reserved	0	Reserved bit
20	R/W	SDC_OPAMP_OUT_MUX_SEL	0	Configuration register for connecting OPAMP1 output to GPIO 0 – Connect OPAMP1 Output to ULP_GPIO_4 1 – Connect OPAMP1 Output to GPIO_30 Please refer to Analog Functions mapping section in Pin Multiplexing for MCU and WiSeMCU chapter for more description.

Bit	Access	Function	Reset Value	Description
19:16	R/W	SDC_OPAMP_IN_P_SEL_CH1	0	Configuration register for selecting P Input of OPAMP1 for Channel-1 Please refer to Input Selection in OPAMP chapter.
15:13	R/W	SDC_OPAMP_IN_N_SEL	0	Configuration register for selecting N Input of OPAMP1. This selection will be common for all Channels Please refer to Input Selection in OPAMP chapter.
12	R/W	SDC_OPAMP_OUT_MUX_EN	0	On Configuring this register, OPAMP1 Output will be connected to GPIO 1 – Output is connected to GPIO 0 – Output is Not connected.
11	R/W	SDC_OPAMP_RES_TO_OUT_VDD	0	Configuration register for Connecting R2 Resistor Ladder input 0 – Connect R2 to Opamp1 Output 1 – Connect R2 to VDD(AUX_Vref)
10:8	R/W	SDC_OPAMP_RES_MUX_SEL	0	Configuration register for Connecting R1 Resistor Ladder input Please refer to Input Selection in OPAMP chapter.
7	R/W	SDC_OPAMP_RES_BACK_EN	0	Configuration register for controlling Resistor Bank of OPAMP 0 – Disable 1 – Enable
6:4	R/W	SDC_OPAMP_R2_SEL	0	Configuration for Resistor Ladder R2 of OPAMP1 for controlling it gain. Please refer to Resistor banks
3:2	R/W	SDC_OPAMP_R1_SEL	0	Configuration for Resistor Ladder R1 of OPAMP1 for controlling it gain. Please refer to Resistor banks

Bit	Access	Function	Reset Value	Description
1	R/W	SDC_OPAMP_LP_MODE	0	Configuration of OPAMP1 Operation mode 0 – Normal mode of Operation 1 – Low power mode of Operation
0	R/W	SDC_OPAMP_EN_CH1	0	Enable signal for turning OPAMP to used for Channel-1 Operation 1 – Enable 0 – Disable

16.18.5.9 SDC_AUXADC_CONFIG_2

Table 16.500. SDC_AUXADC_CONFIG_2

Bit	Access	Function	Reset Value	Description
31:10				
9	R/W	SDC_AUXADC_DIFF_MODE_CH2	0	Enable Differential Mode in AUX ADC for Channel -2 1 - AUX ADC Operates in Differential mode 0 - AUX ADC Operates in Single Ended mode
8:5	R/W	SDC_AUXADC_INPUT_N_SEL_CH2	0	AUXADC's Negative Input Mux Select for Channel-2 Please refer to Input selection section in Analog to Digital conversion chapter for programming options.
4:0	R/W	SDC_AUXADC_INPUT_P_SEL_CH2	0	AUXADC's Positive Input Mux Select for Channel-2 Please refer to Input selection section in Analog to Digital conversion chapter for programming options.

16.18.5.10 SDC_AUXADC_CONFIG_3

Table 16.501. SDC_AUXADC_CONFIG_3

Bit	Access	Function	Reset Value	Description
31:10				
9	R/W	SDC_AUXADC_DIFF_MODE_CH3	0	Enable Differential Mode in AUX ADC for Channel 3 1 - AUX ADC Operates in Differential mode 0 - AUX ADC Operates in Single Ended mode
8:5	R/W	SDC_AUXADC_INPUT_N_SEL_CH3	0	AUXADC's Negative Input Mux Select for Channel-3 Please refer to Input selection section in Analog to Digital conversion chapter for programming options.
4:0	R/W	SDC_AUXADC_INPUT_P_SEL_CH3	0	AUXADC's Positive Input Mux Select for Channel-3 Please refer to Input selection section in Analog to Digital conversion chapter for programming options.

16.18.5.11 SDC_AUXADC_CONFIG_4

Table 16.502. SDC_AUXADC_CONFIG_4

Bit	Access	Function	Reset Value	Description
31:10				
9	R/W	SDC_AUXADC_DIFF_MODE_CH4	0	Enable Differential Mode in AUX ADC for Channel -4 1 - AUX ADC Operates in Differential mode 0 - AUX ADC Operates in Single Ended mode
8:5	R/W	SDC_AUXADC_INPUT_N_SEL_CH4	0	AUXADC's Negative Input Mux Select for Channel-4 Please refer to Input selection section in Analog to Digital conversion chapter for programming options.
4:0	R/W	SDC_AUXADC_INPUT_P_SEL_CH4	0	AUXADC's Positive Input Mux Select for Channel-4 Please refer to Input selection section in Analog to Digital conversion chapter for programming options.

16.18.5.12 SDC_AUXOPAMP_CONFIG_2

Table 16.503. SDC_AUXOPAMP_CONFIG_2

Bit	Access	Function	Reset Value	Description
31:15				
14:11	R/W	SDC_OPAMP_IN_P_SEL_CH4	0	Configuration register for selecting P Input of OPAMP1 for Channel-4 Please refer to Input Selection in OPAMP chapter.
10	R/W	SDC_OPAMP_EN_CH4	0	Enable signal for turning OPAMP to used for Channel-4 Operation 1 – Enable 0 – Disable
9:6	R/W	SDC_OPAMP_IN_P_SEL_CH3	0	Configuration register for selecting P Input of OPAMP1 for Channel-3 Please refer to Input Selection in OPAMP chapter.
5	R/W	SDC_OPAMP_EN_CH3	0	Enable signal for turning OPAMP to used for Channel-4 Operation 1 – Enable 0 – Disable
4:1	R/W	SDC_OPAMP_IN_P_SEL_CH2	0	Configuration register for selecting P Input of OPAMP1 for Channel-2 Please refer to Input Selection in OPAMP chapter.
0	R/W	SDC_OPAMP_EN_CH2	0	Enable signal for turning OPAMP to used for Channel-2 Operation 1 – Enable 0 – Disable

16.18.5.13 SDC_DATA_REG0

Table 16.504. SDC_DATA_REG0

Bit	Access	Function	Reset Value	Description
31:16				
13:12	R	SMP_ID_CH_0	0	Channel iD for sample 0
11:0	R	SDC_DATA_SAMPLE_0	0	Sample 0 collected from Sensor through Aux ADC.

16.18.5.14 SDC_DATA_REG1

Table 16.505. SDC_DATA_REG1

Bit	Access	Function	Reset Value	Description
31:16				
13:12	R	SMP_ID_CH_1	0	Channel iD for sample 1
11:0	R	SDC_DATA_SAMPLE_1	0	Sample 1 collected from Sensor through Aux ADC.

16.18.5.15 SDC_DATA_REG2

Table 16.506. SDC_DATA_REG2

Bit	Access	Function	Reset Value	Description
31:16				
13:12	R	SMP_ID_CH_2	0	Channel iD for sample 2
11:0	R	SDC_DATA_SAMPLE_2	0	Sample 2 collected from Sensor through Aux ADC.

16.18.5.16 SDC_DATA_REG3

Table 16.507. SDC_DATA_REG3

Bit	Access	Function	Reset Value	Description
31:16				
13:12	R	SMP_ID_CH_3	0	Channel iD for sample 3
11:0	R	SDC_DATA_SAMPLE_3	0	Sample 3 collected from Sensor through Aux ADC.

16.18.5.17 SDC_DATA_REG4

Table 16.508. SDC_DATA_REG4

Bit	Access	Function	Reset Value	Description
31:16				
13:12	R	SMP_ID_CH_4	0	Channel iD for sample 4
11:0	R	SDC_DATA_SAMPLE_4	0	Sample 4 collected from Sensor through Aux ADC

16.18.5.18 SDC_DATA_REG5

Table 16.509. SDC_DATA_REG5

Bit	Access	Function	Reset Value	Description
31:16				
13:12	R	SMP_ID_CH_5	0	Channel iD for sample 5
11:0	R	SDC_DATA_SAMPLE_5	0	Sample 5 collected from Sensor through Aux ADC

16.18.5.19 SDC_DATA_REG6

Table 16.510. SDC_DATA_REG6

Bit	Access	Function	Reset Value	Description
31:16				
13:12	R	SMP_ID_CH_6	0	Channel iD for sample 6
11:0	R	SDC_DATA_SAMPLE_6	0	Sample 6 collected from Sensor through Aux ADC

16.18.5.20 SDC_DATA_REG7

Table 16.511. SDC_DATA_REG7

Bit	Access	Function	Reset Value	Description
31:16				
13:12	R	SMP_ID_CH_7	0	Channel iD for sample 7
11:0	R	SDC_DATA_SAMPLE_7	0	Sample 7 collected from Sensor through Aux ADC.

16.18.5.21 SDC_DATA_REG8

Table 16.512. SDC_DATA_REG8

Bit	Access	Function	Reset Value	Description
31:16				
13:12	R	SMP_ID_CH_8	0	Channel iD for sample 8
11:0	R	SDC_DATA_SAMPLE_8	0	Sample 8 collected from Sensor through Aux ADC.

16.18.5.22 SDC_DATA_REG9

Table 16.513. SDC_DATA_REG9

Bit	Access	Function	Reset Value	Description
31:16				
13:12	R	SMP_ID_CH_9	0	Channel iD for sample 9
11:0	R	SDC_DATA_SAMPLE_9	0	Sample 9 collected from Sensor through Aux ADC.

16.18.5.23 SDC_DATA_REG10

Table 16.514. SDC_DATA_REG10

Bit	Access	Function	Reset Value	Description
31:16				
13:12	R	channel_id_10	0	Channel iD for sample 10
11:0	R	SDC_DATA_SAMPLE_10	0	Sample 10 collected from Sensor through Aux ADC.

16.18.5.24 SDC_DATA_REG11

Table 16.515. SDC_DATA_REG11

Bit	Access	Function	Reset Value	Description
31:16				
13:12	R	SMP_ID_CH_11	0	Channel iD for sample 11
11:0	R	SDC_DATA_SAMPLE_11	0	Sample 11 collected from Sensor through Aux ADC.

16.18.5.25 SDC_DATA_REG12

Table 16.516. SDC_DATA_REG12

Bit	Access	Function	Reset Value	Description
31:16				
13:12	R	SMP_ID_CH_12	0	Channel iD for sample 12
11:0	R	SDC_DATA_SAMPLE_12	0	Sample 12 collected from Sensor through Aux ADC.

16.18.5.26 SDC_DATA_REG13

Table 16.517. SDC_DATA_REG13

Bit	Access	Function	Reset Value	Description
31:16				
13:12	R	SMP_ID_CH_13	0	Channel iD for sample 13
11:0	R	SDC_DATA_SAMPLE_13	0	Sample 13 collected from Sensor through Aux ADC.

16.18.5.27 SDC_DATA_REG14

Table 16.518. SDC_DATA_REG14

Bit	Access	Function	Reset Value	Description
31:16				
13:12	R	SMP_ID_CH_14	0	Channel iD for sample 14
11:0	R	SDC_DATA_SAMPLE_14	0	Sample 14 collected from Sensor through Aux ADC.

16.18.5.28 SDC_DATA_REG15

Table 16.519. SDC_DATA_REG15

Bit	Access	Function	Reset Value	Description
31:16				
13:12	R	SMP_ID_CH_15	0	Channel ID for sample 15
11:0	R	SDC_DATA_SAMPLE_15	0	Sample 15 collected from Sensor through Aux ADC.

16.19 AUX ADC/DAC Controller

16.19.1 General Description

The AUXADC-AUXDAC Controller works on a ADC with a resolution of 12bits at 2.5 Msps.

16.19.2 AUX ADC Features

- Programmable clock divider to getting ADC_CLK with Option of controlling its Duty-cycle
- 12 bit ADC Output in 2's complement representation
- GPIOs in High Power mode for ADC Operation
  - Single-ended Mode
    - 18 External configuration selection
    - 5 Internal configuration selection
      - Internal Temperature sensor
      - 3 Opamps Outputs
      - DAC output for internal reference
  - Differential Mode
    - 9 external differential mode configuration selection
    - 4 Internal configuration selection.
      - 3 Opamps Outputs
      - DAC output for internal reference
- GPIOs in Low Power mode for ADC Operation
  - Single-ended Mode
    - 12 External configuration selection.
  - Differential Mode
    - 6 external differential mode configuration selection.
- 10 MHz to 32 KHz allowed ADC_CLK
- Configurable DMA to support 16 channels for storing AUXADC data. Date is ULP SRAM.
- Measurement range 0 to AUXADC_VREF (1.8 to 3.3 V)
- Increased FIFO depth to 16 to meet burst requirement
- Enhanced DMA support for ADC
- Support 2.5 MHz ADC-DAC without losing samples.

16.19.3 AUX DAC Features

- Programmable clock divider to getting DAC_CLK
- 10-bit resolution
- Single ended DAC
- Monotonic by design
- Max sampling frequency is 2.5MHz for DAC_CLK
- Functional Description
- Increased FIFO depth to 16

16.19.3.1 Block Diagram

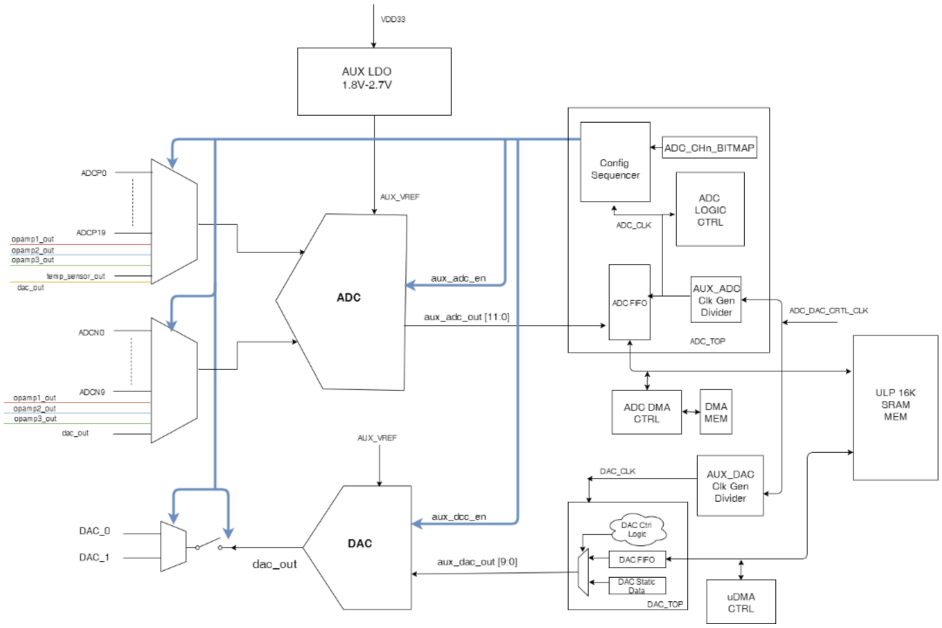


Figure 16.45. Block Diagram of AUX ADC/DAC Controller

16.19.3.2 GPIO MUXING

Please refer to ULP GPIO and analog pin multiplexing sections in datasheet chapter for more information on analog Pin Muxing.

16.19.3.3 Clock Selection

For configuring the AUX ADC-DAC controller, clock source need to be selected before initiating configuration sequence. Please refer to ULPSS Clock Architecture section for selection options

AUXADC CLK Selection

Clock to ADC is generated by programming register **AUXADC_CLK_DIV_FAC** and enabling the clock circuit to ADC by programming **EN_ADC_CLK** bit in register **AUXADC_CTRL_1**.

In register AUXADC_CLK_DIV_FAC, the are programming option for controlling On-Duration of the ADC clock and Total duration of ADC clock. With these option we can generate a duty-cycled clock for better time optimization. On-Duration of the ADC clock is controlled by **ADC_CLK_ON_DUR** bits and Total duration of clock is controlled by programming ADC_CLK_DIV_FAC bits.

Ex: If the AUX-ADC controller clock is selected as 32MHz and the required ADC clock is 4MHz with 50% duty cycle. Then program 4 to **ADC_CLK_ON_DUR** and 8 to **ADC_CLK_DIV_FAC**.

AUXDAC CLK Selection

Clock to DAC is generated by programming register **AUXDAC_CLK_DIV_FAC**.

16.19.3.4 AUXDAC Controller

There are two operating modes AUXDAC controller.

- Static Mode
- FIFO Mode

Static Mode

In static mode, DAC will give out constant voltage output for a programmed DAC.

Programming Sequence :

1. Program **AUXDAC_CLK_DIV_FAC** register to select clock frequency on which DAC should be operating.
2. Program **AUXDAC_DYN_MODE** bit in register **AUXDAC_CONFIG_1** to 0.
3. Program DAC parameter in AUXDAC_CONIG_1 register.
   a. AUXDAC_OUT_MUX_EN bit when set will enable DAC output to be seen on GPIO.
   b. AUXDAC_OUT_MUX_SEL bit will choose on which GPIO DAC Output data is feed.
   c. AUXDAC_DATA_S value will decide DAC output voltage in static mode.

      AUXDAC_EN_S bit will Enable DAC operation in Static mode.

FIFO Mode

DAC controller can be kept in FIFO mode to play continuously digital word on DAC. This mode can be used for playing Single tone waveform in DAC.

Programming Sequence:

1. Program **AUXDAC_CLK_DIV_FAC** register to select clock frequency on which DAC should be operating.
2. Program **AUXDAC_DYN_MODE** bit in register **AUXDAC_CONFIG_1** to 1.
3. Configure DAC FIFO Parameter in register **AUXDAC_CTRL_1**
   a. Program:

      DAC_FIFI_THRESHOLD, DAC_FIFO_FLUSH, ENDAC_FIFO_CONFIG bit in register **AUXDAC_CONFIG_1**.
4. Load DAC digital in ULP SRAM and Configure **µDAM controller** to send data from ULP memories to AUXDAC_DATA register. Please refer to µDAM Controller section for configuration details.
5. Program **AUXDAC_EN_F** bit in register **AUXDAC_CONFIG_1** to l Enable DAC operation in FIFO mode.

Note: In FIFO mode, DAC output is available only on AGPIO4

16.19.3.5 AUXADC Controller

There are two operating modes AUXADC controller.

- Static Mode
- DMA Mode

Static Mode

In static mode, ADC will sample data from a single configured channel.

Programming Sequence:

1. Program **AUXADC_CLK_DIV_FAC** register to select clock frequency on which ADC should be operating.
2. Enable ADC module and ADC static mode by setting ADC_ENABLE and ADC_STATIC_MODE bits in **AUXADC_CRTL_1**. Also, select the channel number from which the data has to be sampled.
3. Data can be read from **AUXADC_DATA** as output.
4. To compare the output of ADC with any threshold value and in order to get threshold interrupt, configure **ADC_DET_THR_CTRL_0** and **ADC_DET_THR_CTRL_1** according to the need. The interrupt can be seen on bit[0] of the **INTR_STATUS_REG** register.
5. Per channel interrupt can also be seen on bits[22:7] of the **INTR_STATUS_REG** register.

DMA Mode

Channel BITMAP

BITMAP Location	Signal Name
16 - 0	Reserved
21 - 17	aux_adc_ip_sel
25 - 22	aux_adc_in_sel
26	aux_adc_diffmode
27	Reserved
28	aux_dac_en
29	aux_dac_out_mux_en
30	aux_dac_out_mux_sel
40 - 31	aux_dac_data
41	opamp1_enable
42	opamp1_lp_mode
44 - 43	opamp1_R1_sel
47 - 45	opamp1_R2_sel
48	opamp1_en_res_bank
51 - 49	opamp1_res_mux_sel
52	opamp1_res_to_out_vdd
53	opamp1_out_mux_en
56 - 54	opamp1_inn_sel
60 - 57	opamp1_inp_sel
61	opamp1_out_mux_sel
62	opamp2_enable
63	opamp2_lp_mode
65 - 64	opamp2_R1_sel
68 - 66	opamp2_R2_sel
69	opamp2_en_res_bank
72 - 70	opamp2_res_mux_sel
74 - 73	opamp2_res_to_out_vdd
75	opamp2_out_mux_en
77 - 76	opamp2_inn_sel
80 - 78	opamp2_inp_sel
81	opamp3_enable
82	opamp3_lp_mode
84 - 83	opamp3_R1_sel
87 - 85	opamp3_R2_sel
88	opamp3_en_res_bank

BITMAP Location	Signal Name
91 - 89	opamp3_res_mux_sel
92	opamp3_res_to_out_vdd
93	opamp3_out_mux_en
95 - 94	opamp3_inn_sel
98 - 96	opamp3_inp_sel
99	ldo_bypass
100	ldo_enable

DMA Memory Configuration

There a dedicated ADC DMA to support 16 channels. The DMA is enabled by setting bit **Internal_DMA_Enable** in register **INTERNAL_DMA_CH_ENABLE**to 1. Internal ADC DMA has a small memory where destination address for storing Aux-ADC sample in ULP SRAM is written along with number of sample to be stored. There are 32 location(2 per channel) in the DMA memory for storing page1(ping) and page2(pong) memory location are stored per channel. Ex: Address 0,1 is used for storing destination memory address to storing Aux-ADC sampled for Channel-1 and Address 2,3 is used for storing destination memory address to storing Aux-ADC sampled for Channel-2 and so on.

The first internal RAM should be written by the user through APB with the information of “start_addr_1, buf_len_1,valid_1, start_addr_2, buf_len_2, valid_2” (mentioned in fig.5) for each channel out of total 16 channels. These “start_addr_1, buf_len_1,valid_1, start_addr_2, buf_len_2, valid_2” information corresponds to the addresses of external memory in which the data from the corresponding channels are stored through AHB Bus.

DMA mode supports dual buffer cyclic mode to avoid loss of data when buffer is full. In dual buffer cyclic mode, if buffer 1 is full for particular channel, incoming sampled data is written into buffer 2 such that, samples from buffer 1 are read back by controller during this time. That’s why there are two start addresses, two buffer lengths and two valid signals for each channel.

Note: Buffer-1 should be read while the controller is populating buffer-2 after memory switch; otherwise after buffer-2 is full, the controller again writes to buffer-1 and as the data of buffer-1 will be overwritten this time, user should make sure that data in buffer-1 was already read.

This information is written in to internal RAM through registers “adc_int_mem_1” and “adc_int_mem_2”. Two consecutive address of RAM corresponds to one channel information. For example RAM address “0” and “1” holds information regarding “Channel-1”, RAM addresses “2” and “3” holds information regarding “Channel-2” and so on. Bit locations adc_int_mem_2[14:10] are to be programmed with the RAM address depending on which channel information the user want to program. For example if user wants to program the information regarding channel-1, first make adc_int_mem_2[14:10] to 5’d0 and should program information through remaining bits of “adc_int_mem_2” and “adc_int_mem_1” (***as mentioned in section.7***) and then make “adc_int_mem_2[14:10] to 5’d1 and should program information through remaining bits of “adc_int_mem_2” and “adc_int_mem_1” (***as mentioned in section.7***).

Per channel memory address information from internal RAM can be read back through APB interface valid bit is used to indicate if buffer address is valid or not in case of dual buffer mode.

Out of two memory chunks allocated for each of the channel to enable “ping-pong” operation in DMA mode, memory_switch interrupt will be asserted every time a ping or pong occurs i.e. every time either of the chunk fills up. So, if first memory chunk is filled and if the interrupt is asserted, then the responsibility of clearing that interrupt lies with the processor before second memory chunk fills up so that interrupt corresponding to second memory chunk will be asserted.

Note: The ports to outer memory will be ULI interface and not AHB bus and the architecture will be as below

Aux-ADC Input Selection

Please refer to Input selection section in Analog to Digital Converter sub-chapter in Analog peripheral chapter for selecting input to AUXADC. If OPAMPs are selected to input to AUXADC, Please refer to Input selection section of OPAMP sub-chapter in Analog peripheral chapter.

16.19.4 Register Summary

Base Address: 0x24043800

Table 16.520. Base Address: 0x24043800

Register Name	Offset
Section 16.19.5.1 AUXDAC_CTRL_1	0x00
Section 16.19.5.2 AUXADC_CTRL_1	0x04
Section 16.19.5.3 AUXDAC_CLK_DIV_FAC	0x08
Section 16.19.5.4 AUXADC_CLK_DIV_FAC	0x0C
Section 16.19.5.5 AUXDAC_DATA	0x10
Section 16.19.5.6 AUXADC_DATA	0x14
Section 16.19.5.7 ADC_DET_THR_CTRL_0	0x18
Section 16.19.5.8 ADC_DET_THR_CTRL_1	0x1C
Section 16.19.5.9 INTR_CLEAR_REG	0x20
Section 16.19.5.10 INTR_MASK_REG	0x24
Section 16.19.5.11 INTR_STATUS_REG	0x28
Section 16.19.5.12 INTR_MASKED_STATUS_REG	0x2C
Section 16.19.5.13 FIFO_STATUS_REG	0x30
Section	0x34
Section 16.19.5.14 ADC_CH1_BIT_MAP_CONFIG_0	0x38
Section 16.19.5.15 ADC_CH1_BIT_MAP_CONFIG_1	0x3C
Section 16.19.5.16 ADC_CH1_BIT_MAP_CONFIG_2	0x40
Section 16.19.5.17 ADC_CH1_BIT_MAP_CONFIG_3	0x44
Section 16.19.5.18 ADC_CH2_BIT_MAP_CONFIG_0	0x48
Section 16.19.5.19 ADC_CH2_BIT_MAP_CONFIG_1	0x4C
Section 16.19.5.20 ADC_CH2_BIT_MAP_CONFIG_2	0x50
Section 16.19.5.21 ADC_CH2_BIT_MAP_CONFIG_3	0x54
Section 16.19.5.22 ADC_CH3_BIT_MAP_CONFIG_0	0x58
Section 16.19.5.23 ADC_CH3_BIT_MAP_CONFIG_1	0x5C
Section 16.19.5.24 ADC_CH3_BIT_MAP_CONFIG_2	0x60
Section 16.19.5.25 ADC_CH3_BIT_MAP_CONFIG_3	0x64
Section 16.19.5.26 ADC_CH4_BIT_MAP_CONFIG_0	0x68
Section 16.19.5.27 ADC_CH4_BIT_MAP_CONFIG_1	0x6C
Section 16.19.5.28 ADC_CH4_BIT_MAP_CONFIG_2	0x70
Section 16.19.5.29 ADC_CH4_BIT_MAP_CONFIG_3	0x74
Section 16.19.5.30 ADC_CH5_BIT_MAP_CONFIG_0	0x78
Section 16.19.5.31 ADC_CH5_BIT_MAP_CONFIG_1	0x7C
Section 16.19.5.32 ADC_CH5_BIT_MAP_CONFIG_2	0x80

Register Name	Offset
Section 16.19.5.33 ADC_CH5_BIT_MAP_CONFIG_3	0x84
Section 16.19.5.34 ADC_CH6_BIT_MAP_CONFIG_0	0x88
Section 16.19.5.35 ADC_CH6_BIT_MAP_CONFIG_1	0x8C
Section 16.19.5.36 ADC_CH6_BIT_MAP_CONFIG_2	0x90
Section 16.19.5.37 ADC_CH6_BIT_MAP_CONFIG_3	0x94
Section 16.19.5.38 ADC_CH7_BIT_MAP_CONFIG_0	0x98
Section 16.19.5.39 ADC_CH7_BIT_MAP_CONFIG_1	0x9C
Section 16.19.5.40 ADC_CH7_BIT_MAP_CONFIG_2	0xA0
Section 16.19.5.41 ADC_CH7_BIT_MAP_CONFIG_3	0xA4
Section 16.19.5.42 ADC_CH8_BIT_MAP_CONFIG_0	0xA8
Section 16.19.5.43 ADC_CH8_BIT_MAP_CONFIG_1	0xAC
Section 16.19.5.44 ADC_CH8_BIT_MAP_CONFIG_2	0xB0
Section 16.19.5.45 ADC_CH8_BIT_MAP_CONFIG_3	0xB4
Section 16.19.5.46 ADC_CH9_BIT_MAP_CONFIG_0	0xB8
Section 16.19.5.47 ADC_CH9_BIT_MAP_CONFIG_1	0xBC
Section 16.19.5.48 ADC_CH9_BIT_MAP_CONFIG_2	0xC0
Section 16.19.5.49 ADC_CH9_BIT_MAP_CONFIG_3	0xC4
Section 16.19.5.50 ADC_CH10_BIT_MAP_CONFIG_0	0xC8
Section 16.19.5.51 ADC_CH10_BIT_MAP_CONFIG_1	0xCC
Section 16.19.5.52 ADC_CH10_BIT_MAP_CONFIG_2	0xD0
Section 16.19.5.53 ADC_CH10_BIT_MAP_CONFIG_3	0xD4
Section 16.19.5.54 ADC_CH11_BIT_MAP_CONFIG_0	0xD8
Section 16.19.5.55 ADC_CH11_BIT_MAP_CONFIG_1	0xDC
Section 16.19.5.56 ADC_CH11_BIT_MAP_CONFIG_2	0xE0
Section 16.19.5.57 ADC_CH11_BIT_MAP_CONFIG_3	0xE4
Section 16.19.5.58 ADC_CH12_BIT_MAP_CONFIG_0	0xE8
Section 16.19.5.59 ADC_CH12_BIT_MAP_CONFIG_1	0xEC
Section 16.19.5.60 ADC_CH12_BIT_MAP_CONFIG_2	0xF0
Section 16.19.5.61 ADC_CH12_BIT_MAP_CONFIG_3	0xF4
Section 16.19.5.62 ADC_CH13_BIT_MAP_CONFIG_0	0xF8
Section 16.19.5.63 ADC_CH13_BIT_MAP_CONFIG_1	0xFC
Section 16.19.5.64 ADC_CH13_BIT_MAP_CONFIG_2	0x100
Section 16.19.5.65 ADC_CH13_BIT_MAP_CONFIG_3	0x104
Section 16.19.5.66 ADC_CH14_BIT_MAP_CONFIG_0	0x108
Section 16.19.5.67 ADC_CH14_BIT_MAP_CONFIG_1	0x10C
Section 16.19.5.68 ADC_CH14_BIT_MAP_CONFIG_2	0x110
Section 16.19.5.69 ADC_CH14_BIT_MAP_CONFIG_3	0x114

Register Name	Offset
Section 16.19.5.70 ADC_CH15_BIT_MAP_CONFIG_0	0x118
Section 16.19.5.71 ADC_CH15_BIT_MAP_CONFIG_1	0x11C
Section 16.19.5.72 ADC_CH15_BIT_MAP_CONFIG_2	0x120
Section 16.19.5.73 ADC_CH15_BIT_MAP_CONFIG_3	0x124
Section 16.19.5.74 ADC_CH16_BIT_MAP_CONFIG_0	0x128
Section 16.19.5.75 ADC_CH16_BIT_MAP_CONFIG_1	0x12C
Section 16.19.5.76 ADC_CH16_BIT_MAP_CONFIG_2	0x130
Section 16.19.5.77 ADC_CH16_BIT_MAP_CONFIG_3	0x134
Section 16.19.5.78 ADC_CH1_OFFSET	0x138
Section 16.19.5.79 ADC_CH2_OFFSET	0x13C
Section 16.19.5.80 ADC_CH3_OFFSET	0x140
Section 16.19.5.81 ADC_CH4_OFFSET	0x144
Section 16.19.5.82 ADC_CH5_OFFSET	0x148
Section 16.19.5.83 ADC_CH6_OFFSET	0x14C
Section 16.19.5.84 ADC_CH7_OFFSET	0x150
Section 16.19.5.85 ADC_CH8_OFFSET	0x154
Section 16.19.5.86 ADC_CH9_OFFSET	0x158
Section 16.19.5.87 ADC_CH10_OFFSET	0x15C
Section 16.19.5.88 ADC_CH11_OFFSET	0x160
Section 16.19.5.89 ADC_CH12_OFFSET	0x164
Section 16.19.5.90 ADC_CH13_OFFSET	0x168
Section 16.19.5.91 ADC_CH14_OFFSET	0x16C
Section 16.19.5.92 ADC_CH15_OFFSET	0x170
Section 16.19.5.93 ADC_CH16_OFFSET	0x174
Section 16.19.5.94 ADC_CH1_FREQ	0x178
Section 16.19.5.95 ADC_CH2_FREQ	0x17C
Section 16.19.5.96 ADC_CH3_FREQ	0x180
Section 16.19.5.97 ADC_CH4_FREQ	0x184
Section 16.19.5.98 ADC_CH5_FREQ	0x188
Section 16.19.5.99 ADC_CH6_FREQ	0x18C
Section 16.19.5.100 ADC_CH7_FREQ	0x190
Section 16.19.5.101 ADC_CH8_FREQ	0x194
Section 16.19.5.102 ADC_CH9_FREQ	0x198
Section 16.19.5.103 ADC_CH10_FREQ	0x19C
Section 16.19.5.104 ADC_CH11_FREQ	0x1A0
Section 16.19.5.105 ADC_CH12_FREQ	0x1A4
Section 16.19.5.106 ADC_CH13_FREQ	0x1A8

Register Name	Offset
Section 16.19.5.107 ADC_CH14_FREQ	0x1AC
Section 16.19.5.108 ADC_CH15_FREQ	0x1B0
Section 16.19.5.109 ADC_CH16_FREQ	0x1B4
Section 16.19.5.110 ADC_CH_PHASE_1	0x1B8
Section 16.19.5.111 ADC_CH_PHASE_2	0x1BC
Reserved	0x1C0
Section 16.19.5.112 ADC_SINGLE_CH_CTRL_1	0x1C4
Section 16.19.5.113 ADC_SINGLE_CH_CTRL_2	0x1C8
Section 16.19.5.114 ADC_SEQ_CTRL	0x1CC
Section 16.19.5.115 VAD_BBP_ID	0x1D0
Section 16.19.5.116 ADC_INT_MEM_1	0x1D4
Section 16.19.5.117 ADC_INT_MEM_2	0x1D8
Section 16.19.5.118 INTERNAL_DMA_CH_ENABLE	0x1DC
Section 16.19.5.119 TS_PTAT_ENABLE	0x1E0
Section 16.19.5.120 ADC_FIFO_THRESHOLD	0x1E4
Reserved	0x1E8-0x204
Section 16.19.5.121 BOD	0x200
Section 16.19.5.122 COMPARATOR	0x204
Section 16.19.5.123 AUXADC_CONFIG_2	0x208
Section 16.19.5.124 AUXDAC_CONFIG_1	0x20C
Section 16.19.5.125 OPAMP_1	0x214
Section 16.19.5.126 OPAMP_2	0x218
Section 16.19.5.127 OPAMP_3	0x21C
Section 16.19.5.128 AUX_LDO	0x210

16.19.5 Register Description

16.19.5.1 AUXDAC_CTRL_1

Table 16.521. AUXDAC_CTRL_1 Register Description

Bit	Access	Function	Reset Value	Description
31:17	R	Reserved	-	Reserved
16:13	R/W	DAC_FIFO_AFULL_THRESHOLD	0xF	These bits control the DAC FIFO almost full threshold
12:9	R/W	DAC_FIFO_AEMPTY_THRESHOLD	0x1	These bits control the DAC FIFO almost empty threshold
8:7	-	Reserved	-	Reserved

Bit	Access	Function	Reset Value	Description
6	R/W	DAC_ENABLE_F	0x0	This bit is used to enable AUX DAC controller. (Valid only when dac_enable is set) 1 – Enable DAC Controller 0 – Disable DAC Controller
5:3	-	Reserved	-	Reserved
2	R/W	DAC_FIFO_FLUSH	0x0	This bit is used to flush the DAC FIFO ‘1’ – Flush dac FIFO ‘0’ – Do not flush This bit is self-clearing
1	R/W	DAC_STATIC_MODE	0x0	This bit is used to select non-FIFO mode in DAC. ‘1’ – Static mode is enabled. Data written to the DAC_DATA_REG will not be written to the FIFO. It will be played on DAC directly. Only single sample can be held at a time. ‘0’ – FIFO mode enabled. Data written to the DAC_DATA_REG is written to the FIFO in this mode. In either of these modes, data will be driven to the DAC only when dac_enable is set.
0	R/W	ENDAC_FIFO_CONFIG	0x0	This bit activates the DAC path in Aux ADC-DAC controller. Data samples will be played on DAC only when this bit is set. 1 – Enable 0 - Disable

16.19.5.2 AUXADC_CTRL_1

Table 16.522. AUXADC_CTRL_1 Register Description

Bit	Access	Function	Reset Value	Description
31:28	R	Reserved	-	Reserved
27	R/W	ADC_NUM_PHASE	0x0	
26:14	R	Reserved	-	Reserved

Bit	Access	Function	Reset Value	Description
13:12	R/W	ADC_CH_SEL_LS	0x0	Aux ADC channel number from which the data has to be sampled 2'b00 – channel 0 2'b01 – channel 1 2'b10 – channel 2 2'b11 – channel 3 When channel number is greater than 3, upper bits should also be programmed ADC_CH_SEL_MS to bits in this register.
11	R	Reserved	-	Reserved
10	R/W	EN_ADC_CLK	0x0	Enable AUX ADC Divider output clock.
9	R/W	BYPASS_NOISE_AVG	0x0	ADC in Bypass noise avg mode
8:7	R/W	ADC_CH_SEL_MS	0x0	Upper 2-bits of adc chan select.When the channel number is greater than 3, these have to be used. The hardware uses {{8:7},{13:12}} as 4-bit channel select
6:3	-	Reserved	-	Reserved
2	R/W	ADC_FIFO_FLUSH	0x0	This bit is used to flush the ADC FIFO '1' – Flush adc FIFO '0' – Do not flush This bit is self clearing
1	R/W	ADC_STATIC_MODE	0x0	This bit is used to select non-FIFO mode in ADC. '1' – Static mode is enabled. ADC data input will be sampled and written to a register in this mode. It will not be written to the FIFO. '0' – FIFO mode enabled. ADC data input will be sampled and written to the ADC FIFO in this mode. In either of these modes, input data from ADC will be sampled only when adc_enable is set. Reading the ADC_DATA_REG provides the value of the sampled data in both the modes.
0	R/W	ADC_ENABLE	0x0	This bits activates the ADC path in Aux ADC-DAC controller. Data will be sampled from ADC only when this bit is set. 1 – Enable 0 - Disable

16.19.5.3 AUXDAC_CLK_DIV_FAC

Table 16.523. AUXDAC_CLK_DIV_FAC Register Description

Bit	Access	Function	Reset Value	Description
31:10	R	Reserved	-	Reserved
9:0	R/W	AUXDAC_CLK_DIV_FAC	0x0	These bits control the DAC clock division factor . clock_freq = input_clock_freq/(2*division factor) Note: '0' value will not generate clock.

16.19.5.4 AUXADC_CLK_DIV_FAC

Table 16.524. AUXADC_CLK_DIV_FAC Register Description

Bit	Access	Function	Reset Value	Description
31:25	R	Reserved	-	Reserved
24:16	R/W	ADC_CLK_ON_DUR	0x0	These bits control the On-Duration of the ADC clock.
15:10	R	Reserved	-	Reserved
9:0	R/W	ADC_CLK_DIV_FAC	0x0	These bits control the Total-Duration of the ADC clock. Note : 'EN_ADC_CLK' bit need to be enable to get clock divider output.

16.19.5.5 AUXDAC_DATA

Table 16.525. AUXDAC_DATA Register Description

Bit	Access	Function	Reset Value	Description
31:10	R	Reserved	-	Reserved
9:0	R/W	AUXDAC_DATA	0x0	DAC Data register for dynamic and static mode Reading this register returns the last data played on DAC

16.19.5.6 AUXADC_DATA

Table 16.526. AUXADC_DATA Register Description

Bit	Access	Function	Reset Value	Description
31:16	R	Reserved	-	Reserved
15:12	R	Dynamic Channel ID	0	Channel ID relevant to the ADC data in DMA dynamic mode
11:0	R	AUXADC_DATA	0	AUXADC Data Read through Register.

16.19.5.7 ADC_DET_THR_CTRL_0

Table 16.527. ADC_DET_THR_CTRL_0 Register Description

Bit	Access	Function	Reset Value	Description
31:16	R	Reserved	-	Reserved
15:12	R/W	ADC input detection threshold 1	0x0	Carries upper four bits of ADC detection threshold
11	R/W	range_comparison_enable	0x0	When set, Aux ADC-DAC controller raises an interrupt to processor when the Aux ADC output falls within the range specified in AUX_ADC_DET_THRESHOLD_0 & AUX_ADC_DET_THRESHOLD_1.
10	R/W	cmp_eq_en	0x0	When set, Aux ADC-DAC controller raises an interrupt to processor when the Aux ADC output is equal to the programmed Aux ADC detection threshold. (Bits [7:0] of this register) When range comparison is set, interrupt will be raised only if the comparison conditions specified AUX_ADC_DET_THRESHOLD_1 are also met.
9	R/W	cmp_grtr_than_en	0x0	When set, Aux ADC-DAC controller raises an interrupt to processor when the Aux ADC output is greater than the programmed Aux ADC detection threshold. (Bits [7:0] of this register) When range comparison is set, interrupt will be raised only if the comparison conditions specified AUX_ADC_DET_THRESHOLD_1 are also met.
8	R/W	cmp_less_than_en	0x0	When set, Aux ADC-DAC controller raises an interrupt to processor when the Aux ADC output falls below the programmed Aux ADC detection threshold.(Bits [7:0] of this register) When range comparison is set, interrupt will be raised only if the comparison conditions specified AUX_ADC_DET_THRESHOLD_1 are also met.
7:0	R/W	ADC_INPUT_DETECTION_THRESHOLD_0	0x0	The value against which the ADC output has to be compared is to be programmed in this register. Bits {15:12,7:0} represent the 12-bit comparison value mode. Interrupt will be raised to the processor as soon as any of the comparison conditions becomes valid. With the existing comparison conditions, the following can be achieved: > , < , = , >= , <= When range comparison is enabled, the following can be achieved: threshold1 < adc output < threshold2 threshold1 <= adc output < threshold2 threshold1 <= adc output <= threshold2 Note: This is valid in adc_static_mode only

16.19.5.8 ADC_DET_THR_CTRL_1

Table 16.528. ADC_DET_THR_CTRL_1 Register Description

Bit	Access	Function	Reset Value	Description
31:15	R	Reserved	-	Reserved
14:11	R/W	ADC_DETECTION_THRESHOLD_4_UPPER_BITS	0x0	Upper 4 bits of ADC detection threshold 2
10	R/W	cmp_eq	0x0	When set, Aux ADC-DAC controller raises an interrupt to NWP when the Aux ADC output is equal to the programmed Aux ADC detection threshold. This is valid only when range comparison is enabled.
9	R/W	cmp_grtr_than	0x0	When set, Aux ADC-DAC controller raises an interrupt to NWP when the Aux ADC output is greater than the programmed Aux ADC detection threshold. This is valid only when range comparison is enabled.
8	R/W	cmp_less_than	0x0	When set, Aux ADC-DAC controller raises an interrupt to NWP when the Aux ADC output falls below the programmed Aux ADC detection threshold. This is valid only when range comparison is enabled.
7:0	R/W	ADC input detection threshold 2	0x0	The value against which the ADC output has to be compared is to be programmed in this register. Bits {14:11,7:0} represent the 12-bit comparison value. Interrupt will be raised to the processor as soon as any of the comparison conditions becomes valid. With the existing comparison conditions, the following can be achieved: > , < , = , >= , <= Note: This is valid in adc_static_mode and when range comparison is enabled

16.19.5.9 INTR_CLEAR_REG

Table 16.529. INTR_CLEAR_REG Register Description

Bit	Access	Function	Reset Value	Description
31:24				
23:8	R/W	intr_clr_reg	0x0	If enabled, corresponding first_mem_switch_intr bits will be cleared.
7:1				
0	R/WC	Clear_intr	0x0	This bit is used to clear threshold detection interrupt ‘1’ – Clear the interrupt ‘0’ – No effect This bit is self-clearing. Upon read, this provides the status of the ADC detection threshold interrupt.

16.19.5.10 INTR_MASK_REG

Table 16.530. INTR_MASK_REG Register Description

Bit	Access	Function	Reset Value	Description
31:25				
24	R/W	dac_static_mode_data_intr_mask	0x1	When Cleared, dac_static_mode_data_intr will be unmasked
23	R/W	adc_static_mode_data_intr_mask	0x1	When Cleared, adc_static_mode_data_intr will be unmasked
22:7	R/W	DMA_Channel_intr_mask[15:0]	0xFFFF	When Cleared, first_mem_switch_intr will be unmasked
6	R/W	dac_fifo_underrun_intr_mask	0x1	When Cleared, dac FIFO underrun interrupt will be unmasked
5	R/W	adc_fifo_overflow_intr_mask	0x1	When Cleared, adc FIFO overflow interrupt will be unmasked
4	R/W	adc_fifo_afull_intr_mask	0x1	When Cleared, adc FIFO afull interrupt will be unmasked
3	R/W	adc_fifo_full_intr_mask	0x1	When Cleared, adc FIFO full interrupt will be unmasked
2	R/W	dac_fifo_aempty_intr_mask	0x1	When Cleared, dac FIFO aempty interrupt will be unmasked
1	R/W	dac_fifo_empty_intr_mask	0x1	When Cleared, dac_FIFO_empty interrupt will be unmasked.
0	R/W	threshold detection intr en	0x1	When Cleared, threshold detection interrupt will be unmasked

16.19.5.11 INTR_STATUS_REG

Table 16.531. INTR_STATUS_REG Register Description

Bit	Access	Function	Reset Value	Description
31:25				
24	R	dac_static_mode_data_intr	0x0	Set when a proper data packet is ready to read in static mode for DAC, it will get cleared automatically after reading the AUXDAC_DATA
23	R	adc_static_mode_data_intr	0x0	Set when a proper data packet is ready to read in static mode for ADC, it will get cleared automatically after reading the AUXADC_DATA
22:7	R	DMA_Channel_intr[15:0]	0x0	Interrupt indicating the first memory has been filled and the DMA write is being shifted to second memory chunk for ping-pong operation. Each bit is for each channel. For example, 15th bit is for 15th channel and so on.
6	R	Dac_fifo_underrun	0x0	Set when a read is done on DAC FIFO when the FIFO is empty. This happens when the FIFO is empty at the driving edge of the AUX DAC clock. This bit gets cleared as soon as the FIFO is written.

Bit	Access	Function	Reset Value	Description
5	R	Adc_fifo_overflow	0x0	Set when a write attempt is made to ADC FIFO when the FIFO is already full. This happens when the FIFO is full at the sampling edge of the AUX ADC clock. This bit gets cleared as soon as the FIFO is read.
4	R	adc_fifo_afull	0x0	Set when ADC FIFO occupancy >= ADC FIFO threshold. This bit gets cleared as soon as the FIFO level falls below the threshold.
3	R	adc_fifo_full	0x0	Set when ADC FIFO is full. This bit gets cleared when data is read from the FIFO.
2	R	Dac_fifo_aempty	0x1	Set when the FIFO occupancy <= DAC FIFO threshold. This bit gets cleared as soon as soon as DAC FIFO occupancy level crosses the programmed threshold.
1	R	dac_fifo_empty	0x1	Set when DAC FIFO is empty. This bit gets cleared when the DAC FIFO at least a single sample is available in DAC FIFO.
0	R	Adc_threshold_detection_intr	0x0	This bit is set when ADC threshold matches with the programmed conditions. This will be be cleared as soon as this interrupt is acknowledged by processor.

16.19.5.12 INTR_MASKED_STATUS_REG

Table 16.532. INTR_MASKED_STATUS_REG Register Description

Bit	Access	Function	Reset Value	Description
31:25				
24	R	dac_static_mode_data_intr_masked	0x0	Masked Interrupt. Set when a proper data packet is ready to read in static mode for DAC
23	R	adc_static_mode_data_intr_masked	0x0	Masked Interrupt. Set when a proper data packet is ready to read in static mode for ADC
22:7	R	DMA_Channel_intr_masked[15:0]	0x0	Masked Interrupt status indicating the first memory has been filled and the DMA write is being shifted to second memory chunk for ping-pong operation. Each bit is for each channel. For example, 15th bit is for 15th channel and so on.
6	R	Dac_fifo_underrun_masked	0x0	Masked Interrupt. Set when a read is done on DAC FIFO when the FIFO is empty. This happens when the FIFO is empty at the driving edge of the AUX DAC clock. This bit gets cleared as soon as the FIFO is written.
5	R	Adc_fifo_overflow_masked	0x0	Masked Interrupt. Set when a write attempt is made to ADC FIFO when the FIFO is already full. This happens when the FIFO is full at the sampling edge of the AUX ADC clock. This bit gets cleared as soon as the FIFO is read.

Bit	Access	Function	Reset Value	Description
4	R	adc_fifo_afull_masked	0x0	Masked Interrupt. Set when ADC FIFO occupancy >= ADC FIFO threshold. This bit gets cleared as soon as the FIFO level falls below the threshold.
3	R	adc_fifo_full_masked	0x0	Masked Interrupt. Set when ADC FIFO is full. This bit gets cleared when data is read from the FIFO.
2	R	Dac_fifo_aempty_masked	0x1	Masked Interrupt. Set when the FIFO occupancy <= DAC FIFO threshold. This bit gets cleared as soon as soon as DAC FIFO occupancy level crosses the programmed threshold.
1	R	dac_fifo_empty_masked	0x1	Masked Interrupt. Set when DAC FIFO is empty. This bit gets cleared when the DAC FIFO at least a single sample is available in DAC FIFO.
0	R	Adc_threshold_detection_intr_masked	0x0	Masked Interrupt. This bit is set when ADC threshold matches with the programmed conditions. This will be be cleared as soon as this interrupt is acknowledged by processor.

16.19.5.13 FIFO_STATUS_REG

Table 16.533. FIFO_STATUS_REG Register Description

Bit	Access	Function	Reset Value	Description
31:8				
7	R	adc_fifo_afull	0x0	Set when ADC FIFO occupancy >= ADC FIFO threshold. This bit gets cleared as soon as the FIFO level falls below the threshold.
6	R	adc_fifo_full	0x0	Set when ADC FIFO is full. This bit gets cleared when data is read from the FIFO.
5	R	Dac_fifo_aempty	0x1	Set when the FIFO occupancy <= DAC FIFO threshold. This bit gets cleared as soon as soon as DAC FIFO occupancy level crosses the programmed threshold.
4	R	dac_fifo_empty	0x1	Set when FIFO is empty. This bit gets cleared when the DAC FIFO is not empty.
3	R	Adc_fifo_aempty	0x1	Set when the FIFO occupancy <= ADC FIFO threshold. This bit gets cleared as soon as soon as ADC FIFO occupancy level crosses the programmed threshold.
2	R	Adc_fifo_empty	0x1	Set when FIFO is empty. This bit gets cleared when the ADC FIFO is not empty. In word mode, FIFO will be shown an non-empty only when occupancy >= 2.
1	R	Dac_fifo_afull	0x0	Set when DAC FIFO occupancy >= FIFO threshold. This bit gets cleared as soon as the FIFO level falls below the threshold.
0	R	Dac_fifo_full	0x0	Set when DAC FIFO is full. In word mode, FIFO will be shown as full unless there is space for 16-bits.This bit gets cleared when data is read from the FIFO and launched to the AUX DAC.

16.19.5.14 ADC_CH1_BIT_MAP_CONFIG_0

Table 16.534. ADC_CH1_BIT_MAP_CONFIG_0 Register Description

Bit	Access	Function	Default Value	Description
31:0	R/W	Channel_1_BitMap [31:0]	-	Refer Bitmap Description

16.19.5.15 ADC_CH1_BIT_MAP_CONFIG_1

Table 16.535. ADC_CH1_BIT_MAP_CONFIG_1 Register Description

Bit	Access	Function	Default Value	Description
31:0	R/W	Channel_1_BitMap [63:32]	-	Refer Bitmap Description

16.19.5.16 ADC_CH1_BIT_MAP_CONFIG_2

Table 16.536. ADC_CH1_BIT_MAP_CONFIG_2 Register Description

Bit	Access	Function	Default Value	Description
31:0	R/W	Channel_1_BitMap [95:64]	-	Refer Bitmap Description

16.19.5.17 ADC_CH1_BIT_MAP_CONFIG_3

Table 16.537. ADC_CH1_BIT_MAP_CONFIG_3 Register Description

Bit	Access	Function	Default Value	Description
31:3				
2:0	R/W	Channel_1_BitMap [98:96]	-	Refer Bitmap Description

16.19.5.18 ADC_CH2_BIT_MAP_CONFIG_0

Table 16.538. ADC_CH2_BIT_MAP_CONFIG_0 Register Description

Bit	Access	Function	Default Value	Description
31:0	R/W	Channel_2_BitMap [31:0]	-	Refer Bitmap Description

16.19.5.19 ADC_CH2_BIT_MAP_CONFIG_1

Table 16.539. ADC_CH2_BIT_MAP_CONFIG_1 Register Description

Bit	Access	Function	Default Value	Description
31:0	R/W	Channel_2_BitMap [63:32]	-	Refer Bitmap Description

16.19.5.20 ADC_CH2_BIT_MAP_CONFIG_2

Table 16.540. ADC_CH2_BIT_MAP_CONFIG_2 Register Description

Bit	Access	Function	Default Value	Description
31:0	R/W	Channel_2_BitMap [95:64]	-	Refer Bitmap Description

16.19.5.21 ADC_CH2_BIT_MAP_CONFIG_3

Table 16.541. ADC_CH2_BIT_MAP_CONFIG_3 Register Description

Bit	Access	Function	Default Value	Description
31:3				
2:0	R/W	Channel_2_BitMap [98:96]	-	Refer Bitmap Description

16.19.5.22 ADC_CH3_BIT_MAP_CONFIG_0

Table 16.542. ADC_CH3_BIT_MAP_CONFIG_0 Register Description

Bit	Access	Function	Default Value	Description
31:0	R/W	Channel_3_BitMap [31:0]	-	Refer Bitmap Description

16.19.5.23 ADC_CH3_BIT_MAP_CONFIG_1

Table 16.543. ADC_CH3_BIT_MAP_CONFIG_1 Register Description

Bit	Access	Function	Default Value	Description
31:0	R/W	Channel_3_BitMap [63:32]	-	Refer Bitmap Description

16.19.5.24 ADC_CH3_BIT_MAP_CONFIG_2

Table 16.544. ADC_CH3_BIT_MAP_CONFIG_2 Register Description

Bit	Access	Function	Default Value	Description
31:0	R/W	Channel_3_BitMap [95:64]	-	Refer Bitmap Description

16.19.5.25 ADC_CH3_BIT_MAP_CONFIG_3

Table 16.545. ADC_CH3_BIT_MAP_CONFIG_3 Register Description

Bit	Access	Function	Default Value	Description
31:5				
4:0	R/W	Channel_3_BitMap [100:96]	-	Refer Bitmap Description

16.19.5.26 ADC_CH4_BIT_MAP_CONFIG_0

Table 16.546. ADC_CH4_BIT_MAP_CONFIG_0 Register Description

Bit	Access	Function	Default Value	Description
31:0	R/W	Channel_4_BitMap [31:0]	-	Refer Bitmap Description

16.19.5.27 ADC_CH4_BIT_MAP_CONFIG_1

Table 16.547. ADC_CH4_BIT_MAP_CONFIG_1 Register Description

Bit	Access	Function	Default Value	Description
31:0	R/W	Channel_4_BitMap [63:32]	-	Refer Bitmap Description

16.19.5.28 ADC_CH4_BIT_MAP_CONFIG_2

Table 16.548. ADC_CH4_BIT_MAP_CONFIG_2 Register Description

Bit	Access	Function	Default Value	Description
31:0	R/W	Channel_4_BitMap [95:64]	-	Refer Bitmap Description

16.19.5.29 ADC_CH4_BIT_MAP_CONFIG_3

Table 16.549. ADC_CH4_BIT_MAP_CONFIG_3 Register Description

Bit	Access	Function	Default Value	Description
31:3				
2:0	R/W	Channel_4_BitMap [98:96]	-	Refer Bitmap Description

16.19.5.30 ADC_CH5_BIT_MAP_CONFIG_0

Table 16.550. ADC_CH5_BIT_MAP_CONFIG_0 Register Description

Bit	Access	Function	Default Value	Description
31:0	R/W	Channel_5_BitMap [31:0]	-	Refer Bitmap Description

16.19.5.31 ADC_CH5_BIT_MAP_CONFIG_1

Table 16.551. ADC_CH5_BIT_MAP_CONFIG_1 Register Description

Bit	Access	Function	Default Value	Description
31:0	R/W	Channel_5_BitMap [63:32]	-	Refer Bitmap Description

16.19.5.32 ADC_CH5_BIT_MAP_CONFIG_2

Table 16.552. ADC_CH5_BIT_MAP_CONFIG_2 Register Description

Bit	Access	Function	Default Value	Description
31:0	R/W	Channel_5_BitMap [95:64]	-	Refer Bitmap Description

16.19.5.33 ADC_CH5_BIT_MAP_CONFIG_3

Table 16.553. ADC_CH5_BIT_MAP_CONFIG_3 Register Description

Bit	Access	Function	Default Value	Description
31:3				
2:0	R/W	Channel_5_BitMap [98:96]	-	Refer Bitmap Description

16.19.5.34 ADC_CH6_BIT_MAP_CONFIG_0

Table 16.554. ADC_CH6_BIT_MAP_CONFIG_0 Register Description

Bit	Access	Function	Default Value	Description
31:0	R/W	Channel_6_BitMap [31:0]	-	Refer Bitmap Description

16.19.5.35 ADC_CH6_BIT_MAP_CONFIG_1

Table 16.555. ADC_CH6_BIT_MAP_CONFIG_1 Register Description

Bit	Access	Function	Default Value	Description
31:0	R/W	Channel_6_BitMap [63:32]	-	Refer Bitmap Description

16.19.5.36 ADC_CH6_BIT_MAP_CONFIG_2

Table 16.556. ADC_CH6_BIT_MAP_CONFIG_2 Register Description

Bit	Access	Function	Default Value	Description
31:0	R/W	Channel_6_BitMap [95:64]	-	Refer Bitmap Description

16.19.5.37 ADC_CH6_BIT_MAP_CONFIG_3

Table 16.557. ADC_CH6_BIT_MAP_CONFIG_3 Register Description

Bit	Access	Function	Default Value	Description
31:3				
2:0	R/W	Channel_6_BitMap [98:96]	-	Refer Bitmap Description

16.19.5.38 ADC_CH7_BIT_MAP_CONFIG_0

Table 16.558. ADC_CH7_BIT_MAP_CONFIG_0 Register Description

Bit	Access	Function	Default Value	Description
31:0	R/W	Channel_7_BitMap [31:0]	-	Refer Bitmap Description

16.19.5.39 ADC_CH7_BIT_MAP_CONFIG_1

Table 16.559. ADC_CH7_BIT_MAP_CONFIG_1 Register Description

Bit	Access	Function	Default Value	Description
31:0	R/W	Channel_7_BitMap [63:32]	-	Refer Bitmap Description

16.19.5.40 ADC_CH7_BIT_MAP_CONFIG_2

Table 16.560. ADC_CH7_BIT_MAP_CONFIG_2 Register Description

Bit	Access	Function	Default Value	Description
31:0	R/W	Channel_7_BitMap [95:64]	-	Refer Bitmap Description

16.19.5.41 ADC_CH7_BIT_MAP_CONFIG_3

Table 16.561. ADC_CH7_BIT_MAP_CONFIG_3 Register Description

Bit	Access	Function	Default Value	Description
31:3				
2:0	R/W	Channel_7_BitMap [98:96]	-	Refer Bitmap Description

16.19.5.42 ADC_CH8_BIT_MAP_CONFIG_0

Table 16.562. ADC_CH8_BIT_MAP_CONFIG_0 Register Description

Bit	Access	Function	Default Value	Description
31:0	R/W	Channel_2_BitMap [31:0]	-	Refer Bitmap Description

16.19.5.43 ADC_CH8_BIT_MAP_CONFIG_1

Table 16.563. ADC_CH8_BIT_MAP_CONFIG_1 Register Description

Bit	Access	Function	Default Value	Description
31:0	R/W	Channel_2_BitMap [63:32]	-	Refer Bitmap Description

16.19.5.44 ADC_CH8_BIT_MAP_CONFIG_2

Table 16.564. ADC_CH8_BIT_MAP_CONFIG_2 Register Description

Bit	Access	Function	Default Value	Description
31:0	R/W	Channel_2_BitMap [95:64]	-	Refer Bitmap Description

16.19.5.45 ADC_CH8_BIT_MAP_CONFIG_3

Table 16.565. ADC_CH8_BIT_MAP_CONFIG_3 Register Description

Bit	Access	Function	Default Value	Description
31:3				
2:0	R/W	Channel_2_BitMap [98:96]	-	Refer Bitmap Description

16.19.5.46 ADC_CH9_BIT_MAP_CONFIG_0

Table 16.566. ADC_CH9_BIT_MAP_CONFIG_0 Register Description

Bit	Access	Function	Default Value	Description
31:0	R/W	Channel_9_BitMap [31:0]	-	Refer Bitmap Description

16.19.5.47 ADC_CH9_BIT_MAP_CONFIG_1

Table 16.567. ADC_CH9_BIT_MAP_CONFIG_1 Register Description

Bit	Access	Function	Default Value	Description
31:0	R/W	Channel_9_BitMap [63:32]	-	Refer Bitmap Description

16.19.5.48 ADC_CH9_BIT_MAP_CONFIG_2

Table 16.568. ADC_CH9_BIT_MAP_CONFIG_2 Register Description

Bit	Access	Function	Default Value	Description
31:0	R/W	Channel_9_BitMap [95:64]	-	Refer Bitmap Description

16.19.5.49 ADC_CH9_BIT_MAP_CONFIG_3

Table 16.569. ADC_CH9_BIT_MAP_CONFIG_3 Register Description

Bit	Access	Function	Default Value	Description
31:3				
2:0	R/W	Channel_9_BitMap [98:96]	-	Refer Bitmap Description

16.19.5.50 ADC_CH10_BIT_MAP_CONFIG_0

Table 16.570. ADC_CH10_BIT_MAP_CONFIG_0 Register Description

Bit	Access	Function	Default Value	Description
31:0	R/W	Channel_10_BitMap [31:0]	-	Refer Bitmap Description

16.19.5.51 ADC_CH10_BIT_MAP_CONFIG_1

Table 16.571. ADC_CH10_BIT_MAP_CONFIG_1 Register Description

Bit	Access	Function	Default Value	Description
31:0	R/W	Channel_10_BitMap [63:32]	-	Refer Bitmap Description

16.19.5.52 ADC_CH10_BIT_MAP_CONFIG_2

Table 16.572. ADC_CH10_BIT_MAP_CONFIG_2 Register Description

Bit	Access	Function	Default Value	Description
31:0	R/W	Channel_10_BitMap [95:64]	-	Refer Bitmap Description

16.19.5.53 ADC_CH10_BIT_MAP_CONFIG_3

Table 16.573. ADC_CH10_BIT_MAP_CONFIG_3 Register Description

Bit	Access	Function	Default Value	Description
31:3				
2:0	R/W	Channel_10_BitMap [98:96]	-	Refer Bitmap Description

16.19.5.54 ADC_CH11_BIT_MAP_CONFIG_0

Table 16.574. ADC_CH11_BIT_MAP_CONFIG_0 Register Description

Bit	Access	Function	Default Value	Description
31:0	R/W	Channel_11_BitMap [31:0]	-	Refer Bitmap Description

16.19.5.55 ADC_CH11_BIT_MAP_CONFIG_1

Table 16.575. ADC_CH11_BIT_MAP_CONFIG_1 Register Description

Bit	Access	Function	Default Value	Description
31:0	R/W	Channel_11_BitMap [63:32]	-	Refer Bitmap Description

16.19.5.56 ADC_CH11_BIT_MAP_CONFIG_2

Table 16.576. ADC_CH11_BIT_MAP_CONFIG_2 Register Description

Bit	Access	Function	Default Value	Description
31:0	R/W	Channel_11_BitMap [95:64]	-	Refer Bitmap Description

16.19.5.57 ADC_CH11_BIT_MAP_CONFIG_3

Table 16.577. ADC_CH11_BIT_MAP_CONFIG_3 Register Description

Bit	Access	Function	Default Value	Description
31:3				
2:0	R/W	Channel_11_BitMap [98:96]	-	Refer Bitmap Description

16.19.5.58 ADC_CH12_BIT_MAP_CONFIG_0

Table 16.578. ADC_CH12_BIT_MAP_CONFIG_0 Register Description

Bit	Access	Function	Default Value	Description
31:0	R/W	Channel_12_BitMap [31:0]	-	Refer Bitmap Description

16.19.5.59 ADC_CH12_BIT_MAP_CONFIG_1

Table 16.579. ADC_CH12_BIT_MAP_CONFIG_1 Register Description

Bit	Access	Function	Default Value	Description
31:0	R/W	Channel_12_BitMap [63:32]	-	Refer Bitmap Description

16.19.5.60 ADC_CH12_BIT_MAP_CONFIG_2

Table 16.580. ADC_CH12_BIT_MAP_CONFIG_2 Register Description

Bit	Access	Function	Default Value	Description
31:0	R/W	Channel_12_BitMap [95:64]	-	Refer Bitmap Description

16.19.5.61 ADC_CH12_BIT_MAP_CONFIG_3

Table 16.581. ADC_CH12_BIT_MAP_CONFIG_3 Register Description

Bit	Access	Function	Default Value	Description
31:3				
2:0	R/W	Channel_12_BitMap [98:96]	-	Refer Bitmap Description

16.19.5.62 ADC_CH13_BIT_MAP_CONFIG_0

Table 16.582. ADC_CH13_BIT_MAP_CONFIG_0 Register Description

Bit	Access	Function	Default Value	Description
31:0	R/W	Channel_13_BitMap [31:0]	-	Refer Bitmap Description

16.19.5.63 ADC_CH13_BIT_MAP_CONFIG_1

Table 16.583. ADC_CH13_BIT_MAP_CONFIG_1 Register Description

Bit	Access	Function	Default Value	Description
31:0	R/W	Channel_13_BitMap [63:32]	-	Refer Bitmap Description

16.19.5.64 ADC_CH13_BIT_MAP_CONFIG_2

Table 16.584. ADC_CH13_BIT_MAP_CONFIG_2 Register Description

Bit	Access	Function	Default Value	Description
31:0	R/W	Channel_13_BitMap [95:64]	-	Refer Bitmap Description

16.19.5.65 ADC_CH13_BIT_MAP_CONFIG_3

Table 16.585. ADC_CH13_BIT_MAP_CONFIG_3 Register Description

Bit	Access	Function	Default Value	Description
31:3				
2:0	R/W	Channel_13_BitMap [98:96]	-	Refer Bitmap Description

16.19.5.66 ADC_CH14_BIT_MAP_CONFIG_0

Table 16.586. ADC_CH14_BIT_MAP_CONFIG_0 Register Description

Bit	Access	Function	Default Value	Description
31:0	R/W	Channel_14_BitMap [31:0]	-	Refer Bitmap Description

16.19.5.67 ADC_CH14_BIT_MAP_CONFIG_1

Table 16.587. ADC_CH14_BIT_MAP_CONFIG_1 Register Description

Bit	Access	Function	Default Value	Description
31:0	R/W	Channel_14_BitMap [63:32]	-	Refer Bitmap Description

16.19.5.68 ADC_CH14_BIT_MAP_CONFIG_2

Table 16.588. ADC_CH14_BIT_MAP_CONFIG_2 Register Description

Bit	Access	Function	Default Value	Description
31:0	R/W	Channel_14_BitMap [95:64]	-	Refer Bitmap Description

16.19.5.69 ADC_CH14_BIT_MAP_CONFIG_3

Table 16.589. ADC_CH14_BIT_MAP_CONFIG_3 Register Description

Bit	Access	Function	Default Value	Description
31:3				
2:0	R/W	Channel_14_BitMap [98:96]	-	Refer Bitmap Description

16.19.5.70 ADC_CH15_BIT_MAP_CONFIG_0

Table 16.590. ADC_CH15_BIT_MAP_CONFIG_0 Register Description

Bit	Access	Function	Default Value	Description
31:0	R/W	Channel_15_BitMap [31:0]	-	Refer Bitmap Description

16.19.5.71 ADC_CH15_BIT_MAP_CONFIG_1

Table 16.591. ADC_CH15_BIT_MAP_CONFIG_1 Register DescriptionRegister Description

Bit	Access	Function	Default Value	Description
31:0	R/W	Channel_15_BitMap [63:32]	-	Refer Bitmap Description

16.19.5.72 ADC_CH15_BIT_MAP_CONFIG_2

Table 16.592. ADC_CH15_BIT_MAP_CONFIG_2 Register DescriptionRegister Description

Bit	Access	Function	Default Value	Description
31:0	R/W	Channel_15_BitMap [95:64]	-	Refer Bitmap Description

16.19.5.73 ADC_CH15_BIT_MAP_CONFIG_3

Table 16.593. ADC_CH15_BIT_MAP_CONFIG_3 Register Description

Bit	Access	Function	Default Value	Description
31:3				
2:0	R/W	Channel_15_BitMap [98:96]	-	Refer Bitmap Description

16.19.5.74 ADC_CH16_BIT_MAP_CONFIG_0

Table 16.594. ADC_CH16_BIT_MAP_CONFIG_0 Register Description

Bit	Access	Function	Default Value	Description
31:0	R/W	Channel_16_BitMap [31:0]	-	Refer Bitmap Description

16.19.5.75 ADC_CH16_BIT_MAP_CONFIG_1

Table 16.595. ADC_CH16_BIT_MAP_CONFIG_1 Register Description

Bit	Access	Function	Default Value	Description
31:0	R/W	Channel_16_BitMap [63:32]	-	Refer Bitmap Description

16.19.5.76 ADC_CH16_BIT_MAP_CONFIG_2

Table 16.596. ADC_CH16_BIT_MAP_CONFIG_2 Register Description

Bit	Access	Function	Default Value	Description
31:0	R/W	Channel_16_BitMap [95:64]	-	Refer Bitmap Description

16.19.5.77 ADC_CH16_BIT_MAP_CONFIG_3

Table 16.597. ADC_CH16_BIT_MAP_CONFIG_3 Register Description

Bit	Access	Function	Default Value	Description
31:3				
2:0	R/W	Channel_16_BitMap [98:96]	-	Refer Bitmap Description

16.19.5.78 ADC_CH1_OFFSET

Table 16.598. ADC_CH1_OFFSET Register Description

Bit	Access	Function	Default Value	Description
31:16				
15:0	R/W	ch1_offset	0	This Register specifies initial offset value with respect to AUX_ADC clock after which Channel-1 should be sampled.

16.19.5.79 ADC_CH2_OFFSET

Table 16.599. ADC_CH2_OFFSET Register Description

Bit	Access	Function	Default Value	Description
31:16				

Bit	Access	Function	Default Value	Description
15:0	R/W	ch2_offset	0	This Register specifies initial offset value with respect to AUX_ADC clock after which Channel-2 should be sampled.

16.19.5.80 ADC_CH3_OFFSET

Table 16.600. ADC_CH3_OFFSET Register Description

Bit	Access	Function	Default Value	Description
31:16				
15:0	R/W	ch3_offset	0	This Register specifies initial offset value with respect to AUX_ADC clock after which Channel-3 should be sampled.

16.19.5.81 ADC_CH4_OFFSET

Table 16.601. ADC_CH4_OFFSET Register Description

Bit	Access	Function	Default Value	Description
31:16				
15:0	R/W	ch4_offset	0	This Register specifies initial offset value with respect to AUX_ADC clock after which Channel-4 should be sampled.

16.19.5.82 ADC_CH5_OFFSET

Table 16.602. ADC_CH5_OFFSET Register Description

Bit	Access	Function	Default Value	Description
31:16				
15:0	R/W	ch5_offset	0	This Register specifies initial offset value with respect to AUX_ADC clock after which Channel-5 should be sampled.

16.19.5.83 ADC_CH6_OFFSET

Table 16.603. ADC_CH6_OFFSET Register Description

Bit	Access	Function	Default Value	Description
31:16				
15:0	R/W	ch6_offset	0	This Register specifies initial offset value with respect to AUX_ADC clock after which Channel-6 should be sampled.

16.19.5.84 ADC_CH7_OFFSET

Table 16.604. ADC_CH7_OFFSET Register Description

Bit	Access	Function	Default Value	Description
31:16				
15:0	R/W	ch7_offset	0	This Register specifies initial offset value with respect to AUX_ADC clock after which Channel-7 should be sampled.

16.19.5.85 ADC_CH8_OFFSET

Table 16.605. ADC_CH8_OFFSET Register Description

Bit	Access	Function	Default Value	Description
31:16				
15:0	R/W	ch8_offset	0	This Register specifies initial offset value with respect to AUX_ADC clock after which Channel-8 should be sampled.

16.19.5.86 ADC_CH9_OFFSET

Table 16.606. ADC_CH9_OFFSET Register Description

Bit	Access	Function	Default Value	Description
31:16				
15:0	R/W	ch9_offset	0	This Register specifies initial offset value with respect to AUX_ADC clock after which Channel-9 should be sampled.

16.19.5.87 ADC_CH10_OFFSET

Table 16.607. ADC_CH10_OFFSET Register Description

Bit	Access	Function	Default Value	Description
31:16				
15:0	R/W	ch10_offset	0	This Register specifies initial offset value with respect to AUX_ADC clock after which Channel-10 should be sampled.

16.19.5.88 ADC_CH11_OFFSET

Table 16.608. ADC_CH11_OFFSET Register Description

Bit	Access	Function	Default Value	Description
31:16				
15:0	R/W	ch11_offset	0	This Register specifies initial offset value with respect to AUX_ADC clock after which Channel-11 should be sampled.

16.19.5.89 ADC_CH12_OFFSET

Table 16.609. ADC_CH12_OFFSET Register Description

Bit	Access	Function	Default Value	Description
31:16				
15:0	R/W	ch12_offset	0	This Register specifies initial offset value with respect to AUX_ADC clock after which Channel-12 should be sampled.

16.19.5.90 ADC_CH13_OFFSET

Table 16.610. ADC_CH13_OFFSET Register Description

Bit	Access	Function	Default Value	Description
31:16				
15:0	R/W	ch13_offset	0	This Register specifies initial offset value with respect to AUX_ADC clock after which Channel-13 should be sampled.

16.19.5.91 ADC_CH14_OFFSET

Table 16.611. ADC_CH14_OFFSET Register Description

Bit	Access	Function	Default Value	Description
31:16				
15:0	R/W	ch14_offset	0	This Register specifies initial offset value with respect to AUX_ADC clock after which Channel-14 should be sampled.

16.19.5.92 ADC_CH15_OFFSET

Table 16.612. ADC_CH15_OFFSET Register Description

Bit	Access	Function	Default Value	Description
31:16				
15:0	R/W	ch15_offset	0	This Register specifies initial offset value with respect to AUX_ADC clock after which Channel-15 should be sampled.

16.19.5.93 ADC_CH16_OFFSET

Table 16.613. ADC_CH16_OFFSET Register Description

Bit	Access	Function	Default Value	Description
31:16				
15:0	R/W	ch16_offset	0	This Register specifies initial offset value with respect to AUX_ADC clock after which Channel-16 should be sampled.

16.19.5.94 ADC_CH1_FREQ

Table 16.614. ADC_CH1_FREQ Register Description

Bit	Access	Function	Default Value	Description
31:16				
15:0	R/W	ch1_freq_value	0	This register specifies Sampling frequency rate at which AUX ADC Date is sampled for Channel-1. The minimum allowed value is 3 to satisfy the Nyquist criteria of sampling rates. Freq_value 1 and 2 are not allowed.

16.19.5.95 ADC_CH2_FREQ

Table 16.615. ADC_CH2_FREQ Register Description

Bit	Access	Function	Default Value	Description
31:16				
15:0	R/W	ch2_freq_value	0	This register specifies Sampling frequency rate at which AUX ADC Date is sampled for Channel-2. The minimum allowed value is 3 to satisfy the Nyquist criteria of sampling rates. Freq_value 1 and 2 are not allowed.

16.19.5.96 ADC_CH3_FREQ

Table 16.616. ADC_CH3_FREQ Register Description

Bit	Access	Function	Default Value	Description
31:16				
15:0	R/W	ch3_freq_value	0	This register specifies Sampling frequency rate at which AUX ADC Date is sampled for Channel-3. The minimum allowed value is 3 to satisfy the Nyquist criteria of sampling rates. Freq_value 1 and 2 are not allowed.

16.19.5.97 ADC_CH4_FREQ

Table 16.617. ADC_CH4_FREQ Register Description

Bit	Access	Function	Default Value	Description
31:16				
15:0	R/W	ch4_freq_value	0	This register specifies Sampling frequency rate at which AUX ADC Date is sampled for Channel-4. The minimum allowed value is 3 to satisfy the Nyquist criteria of sampling rates. Freq_value 1 and 2 are not allowed.

16.19.5.98 ADC_CH5_FREQ

Table 16.618. ADC_CH5_FREQ Register Description

Bit	Access	Function	Default Value	Description
31:16				
15:0	R/W	ch5_freq_value	0	This register specifies Sampling frequency rate at which AUX ADC Date is sampled for Channel-5. The minimum allowed value is 3 to satisfy the Nyquist criteria of sampling rates. Freq_value 1 and 2 are not allowed.

16.19.5.99 ADC_CH6_FREQ

Table 16.619. ADC_CH6_FREQ Register Description

Bit	Access	Function	Default Value	Description
31:16				
15:0	R/W	ch6_freq_value	0	This register specifies Sampling frequency rate at which AUX ADC Date is sampled for Channel-6. The minimum allowed value is 3 to satisfy the Nyquist criteria of sampling rates. Freq_value 1 and 2 are not allowed.

16.19.5.100 ADC_CH7_FREQ

Table 16.620. ADC_CH7_FREQ Register Description

Bit	Access	Function	Default Value	Description
31:16				
15:0	R/W	ch7_freq_value	0	This register specifies Sampling frequency rate at which AUX ADC Date is sampled for Channel-7. The minimum allowed value is 3 to satisfy the Nyquist criteria of sampling rates. Freq_value 1 and 2 are not allowed.

16.19.5.101 ADC_CH8_FREQ

Table 16.621. ADC_CH8_FREQ Register Description

Bit	Access	Function	Default Value	Description
31:16				
15:0	R/W	ch8_freq_value	0	This register specifies Sampling frequency rate at which AUX ADC Date is sampled for Channel-8. The minimum allowed value is 3 to satisfy the Nyquist criteria of sampling rates. Freq_value 1 and 2 are not allowed.

16.19.5.102 ADC_CH9_FREQ

Table 16.622. ADC_CH9_FREQ Register Description

Bit	Access	Function	Default Value	Description
31:16				
15:0	R/W	ch9_freq_value	0	This register specifies Sampling frequency rate at which AUX ADC Date is sampled for Channel-9. The minimum allowed value is 3 to satisfy the Nyquist criteria of sampling rates. Freq_value 1 and 2 are not allowed.

16.19.5.103 ADC_CH10_FREQ

Table 16.623. ADC_CH10_FREQ Register Description

Bit	Access	Function	Default Value	Description
31:16				
15:0	R/W	ch10_freq_value	0	This register specifies Sampling frequency rate at which AUX ADC Date is sampled for Channel-10. The minimum allowed value is 3 to satisfy the Nyquist criteria of sampling rates. Freq_value 1 and 2 are not allowed.

16.19.5.104 ADC_CH11_FREQ

Table 16.624. ADC_CH11_FREQ Register Description

Bit	Access	Function	Default Value	Description
31:16				
15:0	R/W	ch11_freq_value	0	This register specifies Sampling frequency rate at which AUX ADC Date is sampled for Channel-11. The minimum allowed value is 3 to satisfy the Nyquist criteria of sampling rates. Freq_value 1 and 2 are not allowed.

16.19.5.105 ADC_CH12_FREQ

Table 16.625. ADC_CH12_FREQ Register Description

Bit	Access	Function	Default Value	Description
31:16				

Bit	Access	Function	Default Value	Description
15:0	R/W	ch12_freq_value	0	This register specifies Sampling frequency rate at which AUX ADC Date is sampled for Channel-12. The minimum allowed value is 3 to satisfy the Nyquist criteria of sampling rates. Freq_value 1 and 2 are not allowed.

16.19.5.106 ADC_CH13_FREQ

Table 16.626. ADC_CH13_FREQ Register Description

Bit	Access	Function	Default Value	Description
31:16				
15:0	R/W	ch13_freq_value	0	This register specifies Sampling frequency rate at which AUX ADC Date is sampled for Channel-13. The minimum allowed value is 3 to satisfy the Nyquist criteria of sampling rates. Freq_value 1 and 2 are not allowed.

16.19.5.107 ADC_CH14_FREQ

Table 16.627. ADC_CH14_FREQ Register Description

Bit	Access	Function	Default Value	Description
31:16				
15:0	R/W	ch14_freq_value	0	This register specifies Sampling frequency rate at which AUX ADC Date is sampled for Channel-14. The minimum allowed value is 3 to satisfy the Nyquist criteria of sampling rates. Freq_value 1 and 2 are not allowed.

16.19.5.108 ADC_CH15_FREQ

Table 16.628. ADC_CH15_FREQ Register Description

Bit	Access	Function	Default Value	Description
31:16				
15:0	R/W	ch15_freq_value	0	This register specifies Sampling frequency rate at which AUX ADC Date is sampled for Channel-15. The minimum allowed value is 3 to satisfy the Nyquist criteria of sampling rates. Freq_value 1 and 2 are not allowed.

16.19.5.109 ADC_CH16_FREQ

Table 16.629. ADC_CH16_FREQ Register Description

Bit	Access	Function	Default Value	Description
31:16				
15:0	R/W	ch16_freq_value	0	This register specifies Sampling frequency rate at which AUX ADC Date is sampled for Channel-16. The minimum allowed value is 3 to satisfy the Nyquist criteria of sampling rates. Freq_value 1 and 2 are not allowed.

16.19.5.110 ADC_CH_PHASE_1

Table 16.630. ADC_CH_PHASE_1 Register Description

Bit	Access	Function	Default Value	Description
[3:0]	R/W	ch1_phase	0	Phase corresponding to channel-1. Phase in this particular context means the number of the clock tick. For example, “adc_num_phase” is 4, then clock tick-1 to 4 belongs to phase-1 to 4, then clock tick -5 to 8 will again belong to phase – 1 to 4 and so on. So, these phases are assigned to each of the channels. So for each channel is sampled on its respective phase (clock tick) depending on the offset and frequency (from registers, ADC_CHn_OFFSET, ADC_CHn_FREQ_VALUE).
[7:4]	R/W	ch2_phase	0	Phase corresponding to channel-2
[11:8]	R/W	ch3_phase	0	Phase corresponding to channel-3
[15:12]	R/W	ch4_phase	0	Phase corresponding to channel-4
[19:16]	R/W	ch5_phase	0	Phase corresponding to channel-5
[23:20]	R/W	ch6_phase	0	Phase corresponding to channel-6
[27:24]	R/W	ch7_phase	0	Phase corresponding to channel-7
[31:28]	R/W	ch8_phase	0	Phase corresponding to channel-8

16.19.5.111 ADC_CH_PHASE_2

Table 16.631. ADC_CH_PHASE_1 Register Description

Bit	Access	Function	Default Value	Description
[3:0]	R/W	ch9_phase	0	Phase corresponding to channel-9
[7:4]	R/W	ch10_phase	0	Phase corresponding to channel-10
[11:8]	R/W	ch11_phase	0	Phase corresponding to channel-11
[15:12]	R/W	ch12_phase	0	Phase corresponding to channel-12
[19:16]	R/W	ch13_phase	0	Phase corresponding to channel-13
[23:20]	R/W	ch14_phase	0	Phase corresponding to channel-14
[27:24]	R/W	ch15_phase	0	Phase corresponding to channel-15
[31:28]	R/W	ch16_phase	0	Phase corresponding to channel-16

16.19.5.112 ADC_SINGLE_CH_CTRL_1

Table 16.632. ADC_SINGLE_CH_CTRL_1 Register Description

Bit	Access	Function	Default Value	Description
[31:0]	R/W	adc_ch_index_single_chan[31:0]	0	[31:0] out of total 48 bits of bit map, for single-channel mode of a particular channel. This register field contains the Bit-map information for the single-channel mode. Channel is decided by the auxadcsel bits of the ADC_CTRL_REG register.

16.19.5.113 ADC_SINGLE_CH_CTRL_2

Table 16.633. ADC_SINGLE_CH_CTRL_2 Register Description

Bit	Access	Function	Default Value	Description
[31:15]	-	Reserved	-	
[15:0]	R/W	adc_ch_index_single_chan[47:32]	0	[47:32] out of total 48 bits of bit map, for single-channel mode of a particular channel. This register field contains the Bit-map information for the single-channel mode. Channel is decided by the auxadcsel bits of the ADC_CTRL_REG register.

16.19.5.114 ADC_SEQ_CTRL

Table 16.634. ADC_SEQ_CTRL Register Description

Bit	Access	Function	Default Value	Description
31:16	R/W	adc_seq_ctrl_dma_mode	0x0	To enable/disable per channel DMA mode (One-hot coding)Bit 31 :Enable for Channel 16Bit 30 :Enable for Channel 15Bit 29 :Enable for Channel 14Bit 28 :Enable for Channel 13Bit 27 :Enable for Channel 12Bit 26 :Enable for Channel 11Bit 25 :Enable for Channel 10Bit 24 :Enable for Channel 9Bit 23 :Enable for Channel 8Bit 22 :Enable for Channel 7Bit 21 :Enable for Channel 6Bit 20 :Enable for Channel 5Bit 19 :Enable for Channel 4Bit 18 :Enable for Channel 3Bit 17 :Enable for Channel 2Bit 16 :Enable for Channel 1
15:0	R/W	adc_seq_ctrl_ping_pong	0x0	To enable/disable per channel ping-pong operation (One-hot coding)Bit 15 :Enable for Channel 16Bit 14 :Enable for Channel 15Bit 13 :Enable for Channel 14Bit 12 :Enable for Channel 13Bit 11 :Enable for Channel 12Bit 10 :Enable for Channel 11Bit 9 :Enable for Channel 10Bit 8 :Enable for Channel 9Bit 7 :Enable for Channel 8Bit 6 :Enable for Channel 7Bit 5 :Enable for Channel 6Bit 4 :Enable for Channel 5Bit 3 :Enable for Channel 4Bit 2 :Enable for Channel 3Bit 1 :Enable for Channel 2Bit 0 :Enable for Channel 1

16.19.5.115 VAD_BBP_ID

Table 16.635. VAD_BBP_ID Register Description

Bit	Access	Function	Default Value	Description
31:16	R/W	discont_mode	0x0	Per channel discontinuous mode enable signal. When discontinuous mode is enabled, data is sampled only once from that channel and the enable bit is reset to 0.
15:6				
5:0	-	Reserved	-	

16.19.5.116 ADC_INT_MEM_1

Table 16.636. ADC_INT_MEM_1 Register Description

Bit	Access	Function	Default Value	Description
31:0	R/W	prog_wr_data[31:0]	0x0	These 32-bits specifies the start address of first/second buffer corresponding to the channel location ADC_INT_MEM_2[14:10]

16.19.5.117 ADC_INT_MEM_2

Table 16.637. ADC_INT_MEM_2 Register Description

Bit	Access	Function	Default Value	Description
31:16				
15	R/W	prog_wr_data[42]	0x0	Valid bit for first/second buffers corresponding to ADC_INT_MEM_2[14:10]
14:10	R/W	prog_wr_addr	0x0	These bits correspond to the address of the internal memory basing on the channel number, whose information we want to program. For example this will be “0” or “1’ for channel number 1 and so on.
9:0	R/W	prog_wr_data[41:32]	0x0	These 10-bits specify the buffer length of first/second buffer corresponding to the channel location ADC_INT_MEM_2[14:10]

16.19.5.118 INTERNAL_DMA_CH_ENABLE

Table 16.638. INTERNAL_DMA_CH_ENABLE Register Description

Bit	Access	Function	Default Value	Description
31	R/W	Internal_DMA_Enable	0x0	When Set, Internal DMA will be used for reading ADC samples from ADC FIFO and writing them to ULP SRAM Memories.
30:16				

Bit	Access	Function	Default Value	Description
[15:0]	R/W	Per_channel Enable	0x0	Enable bit for Each channel.Bit 15 :Enable for Channel 16Bit 14 :Enable for Channel 15Bit 13 :Enable for Channel 14Bit 12 :Enable for Channel 13Bit 11 :Enable for Channel 12Bit 10 :Enable for Channel 11Bit 9 :Enable for Channel 10Bit 8 :Enable for Channel 9Bit 7 :Enable for Channel 8Bit 6 :Enable for Channel 7Bit 5 :Enable for Channel 6Bit 4 :Enable for Channel 5Bit 3 :Enable for Channel 4Bit 2 :Enable for Channel 3Bit 1 :Enable for Channel 2Bit 0 :Enable for Channel 1

16.19.5.119 TS_PTAT_ENABLE

Table 16.639. TS_PTAT_ENABLE Register Description

Bit	Access	Function	Default Value	Description
31:1				
0	R/W	TS_PTAT_EN	0x0	BJT based Temperature sensor enable1 : Enable0 : Disable

16.19.5.120 ADC_FIFO_THRESHOLD

Table 16.640. ADC_FIFO_THRESHOLD Register Description

Bit	Access	Function	Default Value	Description
[31:8]	-	Reserved	-	-
7:4	W	adc_fifo_afull_threshold	0x0	FIFO almost full threshold for ADC
3:0	W	adc_fifo_aempty_threshold	0x0	FIFO almost empty threshold for ADC

16.19.5.121 BOD

Table 16.641. BOD Register Description

Bit	Access	Function	Default Value	Description
0	R/W	en_bod_test_mux	0	1 - To enable test mux
2:1	R/W	bod_test_sel	0	Select bits for test mux
3	R/W	refbuf_enable	0	1 - To enable reference buffer
7:4	R/W	lvl_sel	0	To select output of ref scaler in reference buffer.
8	R/W	bod_res_enable	0	0 - To disable
				1 - To enable resistor bank, but enables only when cmp is enabled
13:09	R/W	bod_threshold	31	Programmability for resistor bank

16.19.5.122 COMPARATOR

Table 16.642. BOD Register Description

Bit	Access	Function	Default Value	Description
0	R/W	cmp1_en	0	1 - To enable comparator 1
1	R/W	cmp1_en_filter	0	1 - To enable filter for comparator 1
3:02	R/W	cmp1_hyst	0	Programmability to control hysteresis of comparator1
7:04	R/W	cmp1_mux_sel_p	0	Select for positive input of comparator 1
11:08	R/W	cmp1_mux_sel_n	0	Select for negative input of comparator 1
12	R/W	cmp2_en	0	1 - To enable comparator 2
13	R/W	cmp2_en_filter	0	1 - To enable filter for comparator 2
15:14	R/W	cmp2_hyst	0	Programmability to control hysteresis of comparator2
19:16	R/W	cmp2_mux_sel_p	0	Select for positive input of comparator 2
23:20	R/W	cmp2_mux_sel_n	0	Select for negative input of comparator 2
24	R/W	com_dyn_en	0	Dynamic enable for registers

16.19.5.123 AUXADC_CONFIG_2

Table 16.643. AUXADC_CONFIG_2 Register Description

Bit	Access	Function	Default value	Description
31:12				
11	R/W	AUXADC_DYN_ENABLE	0	Aux ADC Dynamic Enable
10:0				

16.19.5.124 AUXDAC_CONFIG_1

Table 16.644. AUXDAC_CONIG_1 Register Description

Bit	Access	Function	Default value	Description
31:15				
14	R/W	AUXDAC_DYN_EN	0	Dynamic Enable 0: Data will be outputted for the pads 1: Data will be outputted for dynamic/static mode
13:4	R/W	AUXDAC_DATA_S	0	DAC data out to pads
3				

Bit	Access	Function	Default value	Description
2	R/W	AUXDAC_OUT_MUX_SEL	0	0 – DAC Output is connected to AGPIO4 1 – DAC Output is connected to AGPIO15 Note: PAD should be configured to Analog mode
1	R/W	AUXDAC_OUT_MUX_EN	0	0 – DAC output is not connected to PAD 1 – DAC output is connected to PAD
0	R/W	AUXDAC_EN_S	0	Enable signal DAC

16.19.5.125 OPAMP_1

Table 16.645. OPAMP_1 Register Description

Bit	Access	Function	Default value	Description
0	R/W	opamp1_enable	0	0 - disable
1	R/W	opamp1_lp_mode	0	0 – normal mode
3:02	R/W	opamp1_R1_sel	1	Programmability to select resister bank R1
6:04	R/W	opamp1_R2_sel	0	Programmability to select resister bank R2
7	R/W	opamp1_en_res_bank	0	0 – disable
10:08	R/W	opamp1_res_mux_sel	0	Selecting input for resistor bank
11	R/W	opamp1_res_to_out_vdd	0	0 – connect resbank to out
12	R/W	opamp1_out_mux_en	0	1 - To connect opamp1 output to pad
15:13	R/W	opamp1_inn_sel	0	Selecting -ve input of opamp
19:16	R/W	opamp1_inp_sel	0	Selecting +ve input of opamp
20	R/W	opamp1_out_mux_en	0	Out Mux Enable
21	R/W	mems_res_bank_en	0	Enable Mems Res Bank
25:22	R/W	vref_mux_en	0	VRef Mux Enable
26	R/W	mux_en	0	Mux Enable
30:27	R/W	vref_mux_sel	0	Vref Mux Sel
31	R/W	opamp1_dyn_en	0	Dynamic Enable For Opamp1 signals

16.19.5.126 OPAMP_2

Table 16.646. OPAMP_2 Register Description

Bit	Access	Function	Default value	Description
0	R/W	opamp1_enable	0	0 - disable
1	R/W	opamp1_lp_mode	0	0 – normal mode
3:02	R/W	opamp1_R1_sel	1	Programmability to select resister bank R1
6:04	R/W	opamp1_R2_sel	0	Programmability to select resister bank R2
7	R/W	opamp1_en_res_bank	0	0 – disable
10:08	R/W	opamp1_res_mux_sel	0	Selecting input for resistor bank
11	R/W	opamp1_res_to_out_vdd	0	0 – connect resbank to out
12	R/W	opamp1_out_mux_en	0	1 - To connect opamp1 output to pad
15:13	R/W	opamp1_inn_sel	0	Selecting -ve input of opamp
19:16	R/W	opamp1_inp_sel	0	Selecting +ve input of opamp
20	R/W	opamp1_out_mux_en	0	Out Mux Enable
21	R/W	mems_res_bank_en	0	Enable Mems Res Bank
25:22	R/W	vref_mux_en	0	VRef Mux Enable
26	R/W	mux_en	0	Mux Enable
30:27	R/W	vref_mux_sel	0	Vref Mux Sel
31	R/W	opamp1_dyn_en	0	Dynamic Enable For Opamp1 signals

16.19.5.127 OPAMP_3

Table 16.647. OPAMP_3 Register Description

Bit	Access	Function	Default value	Description
0	R/W	opamp3_enable	0	0 - disable 1 - to enable opamp 3
1	R/W	opamp3_lp_mode	0	0 – normal mode
				1 – low power mode
3:02	R/W	opamp3_R1_sel	1	Programmability to select resister bank R1
6:04	R/W	opamp3_R2_sel	0	Programmability to select resister bank R2
7	R/W	opamp3_en_res_bank	0	0 – disable 1 – enable resistor bank
10:08	R/W	opamp3_res_mux_sel	0	Selecting input for resistor bank

Bit	Access	Function	Default value	Description
11	R/W	opamp3_res_to_out_vdd	0	0 – connect resbank to out 1 – connect resbank to vdd
				1 – connect resbank to vdd
12	R/W	opamp3_out_mux_en	0	1 - To connect opamp3 output to pad
14:13	R/W	opamp3_inn_sel	0	Selecting -ve input of opamp
17:15	R/W	opamp3_inp_sel	0	Selecting +ve input of opamp
18	R/W	opamp3_dyn_en	0	Dynamic Enable For Opamp3 signals
31:19	-	Reserved	-	

16.19.5.128 AUX_LDO

Table 16.648. AUX_LDO Register Description

Bit	Access	Function	Default Value	Description
31:8	-	Reserved	-	
7	R/W	Dyn_en	0	Dynamic Enable
6	R/W	ENABLE_LDO	1	To turn on ldo
5	R/W	BYPASS_LDO	0	To enable bypass mode
4	R/W	LDO_DEFAULT_MODE	1	0 : normal mode 1:default mode (1.8V)
3:0	R/W	LDO_Crtl	3	Word to set the output voltage

16.20 ULP Timers

16.20.1 General Description

The MCU ULP Timer block supports four 32-bit timers, which can be used to generate various timing events for the software. Each of the four timers can be independently programmed to work in periodic or one-shot mode and can be configured either as a microsecond timer or as a counter.

16.20.2 Features

- Supports 4 independent timers
- Supports per timer enable and disable.
- Option to configure each timer as a 32-bit counter or 32-bit microsecond timer.
- Supports 1µs mode and 256µs modes per timer.
- Support for programming 1µs and 256µs time unit values. Accounts for integral and fractional value of the time units programmed.
- Microsecond timer supports two modes:
  - 1 Microsecond mode : The time unit is 1µs. Number of microseconds required to be counted has to be programmed.
  - 256microsecond mode : The time unit is 256µs. Number of 256µs units required to be counted has to be programmed. This is useful when the timer is being used for counting large time values and microsecond based tracking not required.
- Supports one shot and periodic modes per timer.
- Option to interrupt the processor on timeout. Supports per timer interrupt enable and disable.
- Can run synchronous or asynchronous to SoC clock.

16.20.3 Block Diagram

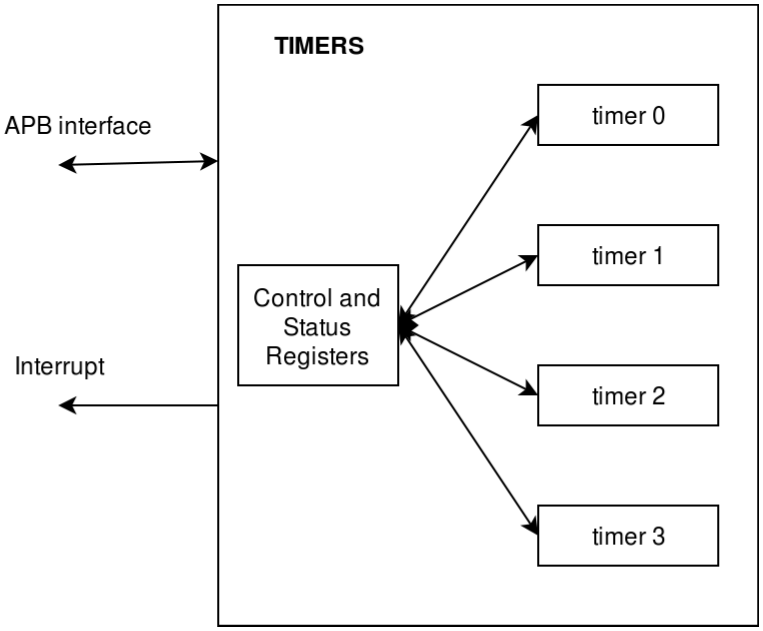


Figure 16.46. Timers BSlock Diagram

16.20.4 Functional Description

16.20.4.1 Basic Timer Operation

To operate the nth timer the count till which the counter should count is loaded in the MCUULP_TMRn_MATCH register and the start bit MCUULP_TMRn_CNTRL[TMR_START] is set to start the timer. When the timer reaches the timeout value, a time out indication can be read in the MCUULP_TMR_INTR_STAT register. If the interrupt is enabled in the MCUULP_TMRn_CNTRL register the timeout condition will generate an interrupt to the processor. The timer can also be stopped any time in between its operation before the time out condition by setting the MCUULP_TMRn_CNTRL[TMR_STOP] bit. After stopping the timer new parameters can be programmed into the registers.

16.20.4.2 One-Shot & Periodic Operation

In one shot operation, the timer counts till the timeout and then generates a single interrupt after which it returns to idle state. Where as in periodic operation the timer when reaches the timeout value generates an interrupt and starts counting again from the originally set value.
Timers are by default in one shot mode;Periodic operation can be enabled by setting the MCUULP_TMRn_CNTRL[TMR_TYPE] bit.

16.20.4.3 Microsecond Timer Operation

Microsecond timer supports two modes:

- 1µs mode and 256µs mode.
- In 1µs mode, the time unit is 1µs. Number of microseconds required to be counted has to be programmed. The maximum value that can be counted is (2^{32}-1)µs, i.e 1.2 hours.
- In 256µs mode, the time unit is 256µs. Number of 256µs units required to be counted has to be programmed. This is useful when the timer is being used for counting large time values and microsecond based tracking not required. The maximum value that can be counted is (2^{32}-1)*256µs, i.e. nearly equal to 13 days.

One microsecond mode is enabled by setting the MCUULP_TMRn_CNTRL[TMR_MODE] to '1' . In this mode of operation, the value programmed in MCUULP_TMRn_MATCH register is considered in microseconds.
To operate in this mode, the MCUULP_TMR_US_PERIOD_INT register should be programmed with integer part of number of clock cycles per microsecond and MCUULP_TMR_US_PERIOD_FRAC is programmed with the fractional part of number of clock cycles per microsecond depending on the system clock being used(clock period of the system clock used in microseconds). Only the lower significant 8-bits of MCUULP_TMR_US_PERIOD_FRAC are considered. In fractional representation, the n^{th} bit (n= 0..7) has the value of $2^{(n-8)}$
256 µs mode is enabled by setting the MCUULP_TMRn_CNTRL[TMR_MODE] to '2' . In this mode of operation, the value programmed in MCUULP_TMRn_MATCH register is to be in terms of TUs, where 1TU = 256us.
To operate in this mode, the MCUULP_TMR_MS_PERIOD_INT register should be programmed with integer part of number of clock cycles per TU and MCUULP_TMR_MS_PERIOD_FRAC is programmed with the fractional part of number of clock cycles per TU.

16.20.5 Register Summary

Base Address: 0x2404_2000

Table 16.649. Timers Register Summary Table

Register Name	Offset	Description
Section 16.20.6.1 MATCH_CTRLn	0x00	Timer 0 Match Register
Section 16.20.6.2 MCUULP_TMR_n__CNTRL	0x04	Timer 0 Control Register
Section 16.20.6.1 MATCH_CTRLn	0x08	Timer 1 Match Register
Section 16.20.6.2 MCUULP_TMR_n__CNTRL	0x0C	Timer 1 Control Register
Section 16.20.6.1 MATCH_CTRLn	0x10	Timer 2 Match Register
Section 16.20.6.2 MCUULP_TMR_n__CNTRL	0x14	Timer 2 Control Register
Section 16.20.6.1 MATCH_CTRLn	0x18	Timer 3 Match Register
Section 16.20.6.2 MCUULP_TMR_n__CNTRL	0x1C	Timer 3 Control Register
Reserved	0x20-0x7C	
Section 16.20.6.3 MCUULP_TMR_INTR_STAT	0x80	Timer Interrupt Status Register
Section 16.20.6.4 MCUULP_TMR_US_PERIOD_INT	0x84	Micro-second Timer Period Integer Part Register
Section 16.20.6.5 MCUULP_TMR_US_PERIOD_FRAC	0x88	Micro-second Timer Period Fractional Part Register
Section 16.20.6.6 MCUULP_TMR_MS_PERIOD_INT	0x8C	Milli-second Timer Period Integer Part Register
Section 16.20.6.7 MCUULP_TMR_ MS_PERIOD_FRAC	0x90	Milli-second Timer Period Fractional Part Register
Section 16.20.6.8 MCUULP_TMR_ACTIVE_STATUS	0x9C	Timer Active Status Register

16.20.6 Register Description

Legend:

R = Read-only, W = Write-only, R/W = Read/Write, R/WC = Read/Clear on Write

16.20.6.1 MATCH_CTRLn

n = 0 to 3

Table 16.650. MCUULP_TMRn_MATCH Register Description

Bit	Access	Function	Reset Value	Description
[31:0]	R/W	TMR_MATCH	0xffff_ffff	These bits are used to program the timer time out value in microseconds or in number of system clocks. Upon reading, these bits indicate time remaining before timeout. (read as 32'hFFFF_FFFF initially) If counter_up is set these bits directly gives out the up-running counter/timer value.

16.20.6.2 MCUULP_TMR_n__CNTRL

n = 0 to 3

Table 16.651. MCUULP_TMR_CNTRL Register Description

Bit	Access	Function	Reset Value	Description
[31:8]	R/W	Reserved	0x0	Reserved for future use.
7	R/W	counter_up	0x0	For reading/tracking counter in up-counting this bit has to be set. By setting this bit down-counting reading will be lost.
6	W	TMR_STOP	0x0	This bit is used to stop the active timer. '1'- stops the timer, if timer is active (This bit is self clearing bit)
5	R/W	TMR_MODE	0x0	This bit is used to select the mode of working of timer '1'- Periodic timer '0'- One shot time In periodic mode, timer gets reset when timeout occurs and starts again automatically.
4:3	R/W	TMR_TYPE	0x0	These bits are used to select the type of the timer '2'- 256 µs mode '1'-1µs mode '0'- Countdown timer
2	R/W	TMR_INTR_ENABLE	0x0	This bit is used to enable the timeout interrupt. '1'- Interrupt enabled '0'- Interrupt disabled
1	W	TMR_INTR_CLR	0x0	This bit is used to clear the interrupt '1'- Clear interrupt (This bit is self clearing bit)
0	W	TMR_START	0x0	This bit is used to start the timer. Timer gets reset upon setting this bit. '1'- Timer start (This bit is self clearing bit)

16.20.6.3 MCUULP_TMR_INTR_STAT

Table 16.652. MCUULP_TMR_STAT Register Description

Bit	Access	Function	Reset Value	Description
[31:4]	R	Reserved	0x0	Reserved
3	R/WC	TMR3_INTR_STATUS	0x0	This bit indicates the status of the interrupt generated by timer3. '1'- Interrupt present '0'- No interrupt present
2	R/WC	TMR2_INTR_STATUS	0x0	This bit indicates the status of the interrupt generated by timer2. '1'- Interrupt present '0'- No interrupt present
1	R/WC	TMR1_INTR_STATUS	0x0	This bit indicates the status of the interrupt generated by timer1. '1'- Interrupt present '0'- No interrupt present
0	R/WC	TMR0_INTR_STATUS	0x0	This bit indicates the status of the interrupt generated by timer0. '1'- Interrupt present '0'- No interrupt present

16.20.6.4 MCUULP_TMR_US_PERIOD_INT

Table 16.653. MCUULP_TMR_US_PERIOD_INT Register Description

Bit	Access	Function	Reset Value	Description
[31:16]	R	Reserved	0x0	Reserved
[15:0]	R/W	TMR_US_PERIOD_INT	0xffff	These bits are used to program the integer part of number of clock cycles per microsecond of the system clock being used.

16.20.6.5 MCUULP_TMR_US_PERIOD_FRAC

Table 16.654. MCUULP_TMR_US_PERIOD_FRAC Register Description

Bit	Access	Function	Reset Value	Description
[31:8]	R	Reserved	0x0	Reserved
[7:0]	R/W	TMR_US_PERIOD_FRAC	0xff	These bits are use to program the fractional part of clock cycles per microsecond of the system clock being used.

16.20.6.6 MCUULP_TMR_MS_PERIOD_INT

Table 16.655. MCUULP_TMR_MS_PERIOD_INT Register Description

Bit	Access	Function	Reset Value	Description
[31:16]	R	Reserved	0x0	Reserved
[15:0]	R/W	TMR_MS_PERIOD_INT	0xffff	These bits are used to program the integer part of number of clock cycles per 256 microsecond of the system clock being used.

16.20.6.7 MCUULP_TMR_ MS_PERIOD_FRAC

Table 16.656. MCUULP_TMR_MS_PERIOD_FRAC Register Description

Bit	Access	Function	Reset Value	Description
[31:8]	R	Reserved	0x0	Reserved
[7:0]	R/W	TMR_MS_PERIOD_FRAC	0xff	These bits are used to program the fractional part of clock cycles per 256 microsecond of the system clock being used.

16.20.6.8 MCUULP_TMR_ACTIVE_STATUS

Table 16.657. MCUULP_TMR_ACTIVE_STATUS Register Description

Bit	Access	Function	Reset Value	Description
[31:4]	R	Reserved	0x0	Reserved
[3:0]	R	Timer_active	0x0	Timer active status for each timer. LSB bit specifies the status for 0th timer and so on. For each bit, 1 – Corresponding timer is active 0 – Corresponding timer is inactive

16.20.7 Programming Sequence

1. Enable Timer module power domain in ULP_Peripheral_Power_Control_SET register as described in Section 9. Power Architecture.
2. Configure Timer module clock using MCUULP_CLK_EN_REG1 and MCUULP_TIMER_CLK_CONFIG as described in Section 6.13 MCU ULP Clock Architecture.
3. To function as Microsecond timer MCUULP_TMRn_CNTRL[TMR_TYPE] should be set accordingly and to function as Counter timer the MCUULP_TMRn_CNTRL[TMR_TYPE] bit should be cleared.
4. Program the maximum count value of the counter in MCUULP_TMRn_MATCH register if counter mode is selected.
5. If the one microsecond mode is selected, program the MCUULP_TMR_US_PERIOD_INT and MCUULP_TMR_US_PERIOD_FRAC registers with integral and fractional part of time period (in number of clocks) of the system clock being used and program the time out value of the timer in microseconds in MCUULP_TMRn_MATCH register.
6. If the 256 micro-second mode is selected, program the MCUULP_TMR_MS_PERIOD_INT and MCUULP_TMR_MS_PERIOD_FRAC registers with integral and fractional part of time period required for 1 TU (i.e.256 microseconds) and program the time out value of the timer in terms of TUs in the MCUULP_TMRn_MATCH register.
7. To enable the interrupt on timeout, set the MCUULP_TMRn_CNTRL[TMR_INTR_ENABLE] bit.
8. Timers are by default in one shot mode;Periodic operation can be enabled by setting the MCUULP_TMRn_CNTRL[TMR_TYPE] bit.
9. To start the timer set the MCUULP_TMRn_CNTRL[TMR_START] bit.
10. Wait for the timer interrupt, if the timeout interrupt is enabled.
11. Set MCUULP_TMRn_CNTRL[TMR_INTR_CLR], to clear the interrupt generated by timeout.
12. MCUULP_TMR_INTR_STAT[TMRn_INTR_STATUS] can also be monitored to check the timeout status.

16.21 Bandgap Top

16.21.1 General Description

- This Bandgap Reference provides a voltage reference of 1.2V which is independent of PVT variations and PTAT currents of 20nA and 50nA.
- There is a V2I which provides constant currents of 5nA, 10nA, 20nA.
- It also has reference scaling circuits which provides reference voltages of 0.8-1.1V, 0.75-1.05V, 0.8-1.05 and 0.55-0.8V for ULP_DCDC, 0.85-1.2V for pmu.
- It has a low power mode/sampling mode in which its current consumption is ~15nA.

16.21.2 Features

- Provides reference of 1.2V for other analog blocks. It can be trimmed using BG_R and BG_R_PTAT to optimize temperature coefficient.
- It provides 20nA, 50nA and 100nA currents to other analog blocks. These currents can be switched on/off using bod_clks_ptat_en and an_perif_ptat_en for ULP and analog_peripherals respectively.
- It provides programmable references of 0.8, 0.9, 0.95, 1, 1.05, 1.1V to SCDC, can be programmed using ref_sel_dcdc.
- It provides programmable reference of 0.9-1.15V in steps of 50mV to HP and LP Ldos in SCDC, can be programmed using ref_sel_lp_dcdc.
- It provides programmable reference of 0.55-0.8V to 0.6V LDO, can be programmed using LDO_0P6_CTRL.
- It provides programmable reference of 0.85-1.2V to pmu, it will be used as reference in PFM mode of buck, can be programmed using ref_sel_PMU.
- It provides 5nA, 10nA and 20nA constant currents to pmu, high freq RO and Temp sensor respectively.
- LP mode (low power mode or sampled mode of bandgap) will be enabled using sampling_en. And will be controlled using bg_en and sh_en clks. These clks are generated using 32KRO/32KRC clocks in ipmu_digital_top.

Note: The voltages can only be programmed using SPI writes, so changing reference voltages is not possible in LP mode.

16.21.3 Functional Description

- UULP_VBATT_2 is the power supply for bandgap. All the control signals will be on UULP_VOUTSCDC and are level shifted to UULP_VBATT_2.
- This is enabled by default. bg_en and sh_en are always low in HP mode.
- Vbg = Vbe + (ΔVbe/Rptat)*R, Iptat = ΔVbe/Rptat
- BG_R_PTAT trims PTAT current and thereby trims Vbg also. Increasing BG_R_PTAT increases PTAT current and Vbg. Increasing BG_R_PTAT creates positive gradient in Vbg across temperature.
- BG_R trims Vbg. Increasing BG_R reduces Vbg and also create negative gradiant in Vbg across temperature.
- If Vbg reduces(increases) with increase in temperature, increasing(decreasing) BG_R_PTAT or decreasing(increasing) BG_R helps.

  The bandgap can enter low power mode only when all high frequency clocks are off.

16.21.3.1 Low Power Mode

The bandgap can enter low power mode only when all high frequency clocks are off.

Low power mode has two states ACTIVE and SLEEP. During ACTIVE state bg_en is low(so bandgap will be on) and during SLEEP state bg_en and sh_en are high.

ACTIVE time and SLEEP time can be programmed using bgs_active_timer_sel and bgs_sleep_timer_sel respectively.

If the bandgap output decays faster in SLEEP state, reduce SLEEP time.

If the ACTIVE time is not sufficient (in this case, out will be much lower than 1.2V), increase ACTIVE time using bgs_active_timer_sel.

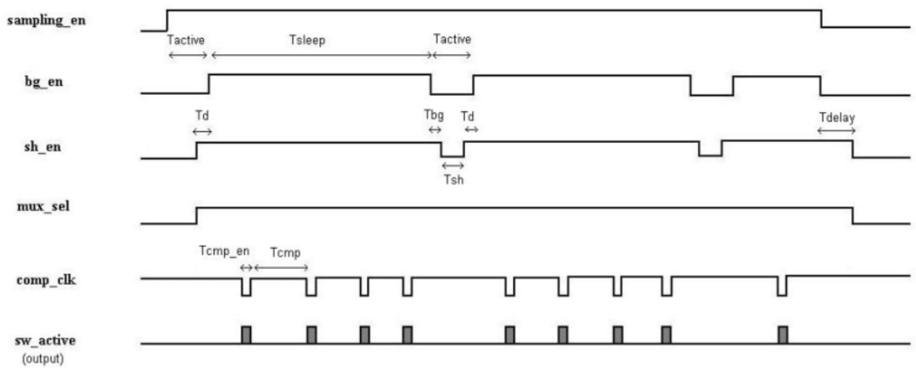


Figure 16.47. Low Power Mode

16.21.4 Register Summary

Base Address: 0x2405A400

Table 16.658. Register Summary

Register Name	Offset
Section 16.21.5.1 BG_SLEEP_TIMER_REG	0x94
Section 16.21.5.2 SCDC_CTRL_REG_0	0x98
Section 16.21.5.3 BG_SCDC_PROG_REG_1	0x9C
Section 16.21.5.4 BG_SCDC_PROG_REG_2	0xA0
Section 16.21.5.5 BG_LDO_REG1	0xA4
Section 16.21.5.6 BG_SCDC_READ_BACK	0xA8
Section 16.21.5.7 BLACKOUT_MON_EN_REG	0xAC

16.21.5 Register Description

16.21.5.1 BG_SLEEP_TIMER_REG

Table 16.659. Band-Gap Sleep Timer Register Description

Bit	Access	Function	Reset Value	Description
31:22	-	Reserved	-	It is recommended to write these bits to 0.
21:20	RW	bgs_sleep_timer_sel	2	sleep timer count is 2'd0: active_timer_count * 2^6 2'd1: active timer_count * 2^7 2'd2: active_timer_count * 2^8 2'd3: active_timer_count * 2^9
19	R	bgs_active_timer_sel[0]	1	taking one bit of bgs_active_timer_sel[17:16]
18	RW	mask_sw_active	0	1: disable comp clock in between sleep duration
17:16	bit 16 - W bit 17 - RW	bgs_active_timer_sel	1	Active mode time: 2'd0: 4, 2'd1: 8, 2'd2: 16, 2'd3: 32 cycles of 32KHz clock time after which sh_en goes low after bg_en goes low = sh_en_delay (50% of active time) time for which sh_en is low in active state = sh_en_width (50% of active time) active mode time value will write in the [17:16] bit of BG_SLEEP_TIMER_REG while read operation of these value done from [17] and [19] bit of BG_SLEEP_TIMER_REG.
15:4	-	Reserved	-	It is recommended to write these bits to 0.
3	RW	bg_ctrl_auto	1	1: bg_en and bg_sh_en are automatically controlled 0: bg_en and sh_en take values from SPI.
2	RW	bypass_pwrgating_combi	0	Powergating is disabled for combi logic . It will always be ON.
1	RW	bg_sampling_spi_sel	0	enable bandgap sampling through spi / pin 1 = spi ; 0 = pin(sleep_en)
0	RW	bgs_clk_en	0	bandgap sampling enable through spi

16.21.5.2 SCDC_CTRL_REG_0

Table 16.660. SCDC Control Register Description

Bit	Access	Function	Reset Value	Description
31:22	RW	Reserved	-	It is recommended to write these bits to 0.
21	RW	ext_cap_en	1'b0	To change current trim bits to high or low through spi, based on high power or low power mode. When 0, curr prog value is 0.

Bit	Access	Function	Reset Value	Description
20:17	RW	fixed_curr_prog_high	4'd15	Current prog value to take when ext cap en is high and sel_high freq_ext_b is 0
16:13	RW	fixed_curr_prog_low	4'd0	Current prog value to take when ext cap en is high and sel_high freq_ext_b is 1
12	RW	bypass_trim_ro	1'b0	To program the trim value manually, irrespective of the fsm
11:7	RW	fixed_trim_ro	5'd0	Manual trim word
6	RW	fixed_mode	1'b0	fixed mode
5:4	RW	max_mode	2'd2	maximum mode it can go to
3:0	RW	count_reset	4'hF	Count reset value, count threshold will be doubler this value

16.21.5.3 BG_SCDC_PROG_REG_1

Table 16.661. Band-Gap SCDC Control-1 Register Description

Bit	Access	Function	Reset Value	Description
31:22	RW	Reserved	-	It is recommended to write these bits to 0.
21:19	RW	bg_r_ptat	2	Bandgap voltage programming
18:16	RW	bg_r	0	Bandgap voltage programming
15	RW	bg_en	0	bg_en from spi
14	RW	bg_sh_en	0	bg_sh_en from spi
13	-	Reserved	-	It is recommended to write these bits to 0.
12:10	RW	ref_sel_dcdc	1	DCDC output programming vref_1p1/vref_1p05 3'd0 - 1.15/1.1 3'd1 - 1.1/1.05 3'd2 - 1.05/1.0 3'd3 - 1.0/0.95 3'd4 - 0.95/0.9 3'd5 - 0.9/0.85
9:7	RW	ref_sel_lp_dcdc	1	DCDC output programming in LDO high/low power mode 3'd0 -1.1 3'd1 - 1.15 3'd2 - 1.05 3'd3 - 1.0 3'd4 - 0.95 3'd5 - 0.9
6:5	-	Reserved	-	It is recommended to write these bits to 0.
4	RW	bod_clks_ptat_en	1	1 - To enable ptat currents to clocks and bod(cmp_npss)
3	RW	an_perif_ptat_en	1	1 - To enable ptat currents to analog peripherals
2:0	RW	ref_sel_PMU	0	3'd0 - 1.2V 3'd1 - 1.15V 3'd2 - 1.1V 3'd3 - 1.05V 3'd4 - 1.0V 3'd5 - 0.95V 3'd6 - 0.9V 3'd7 - 0.85V

16.21.5.4 BG_SCDC_PROG_REG_2

Table 16.662. Band-Gap SCDC Control-2 Register Description

Bit	Access	Function	Reset Value	Description
31:22	-	Reserved	-	It is recommended to write these bits to 0.

Bit	Access	Function	Reset Value	Description
21	RW	scdcdc_sel	0	To switch to SCDCDC mode from LDO mode. 1 - SCDC mode 0 - LDO mode
20	RW	testmode_0_en	0	Enable for output on to BG_TESTMODE0
19:18	RW	testmode_0_sel	0	2'd0: bg_sw_active 2'd1: scdcdc_sown 2'd2: scdcdc_lp_mode (sel_high_freq_ext_b) 2'd3: scdcdc_sel (To select ldo - scdcdc)
17	RW	testmode_1_en	0	To enable test mux for BG_TESTMODE1
16:15	RW	testmode_1_sel	0	2'd0: bg_sh_en 2'd1: scdcdc_up 2'd2: scdcdc_en (Enable for scdcdc block) 2'd3: scdcdc_lp_en (enable for 10uA LDO)
14	RW	testmode_2_en	0	To enable testmux for BG_TESTMODE2
13:11	RW	testmode_2_sel	0	3'd0: bg_en 3'd1: bg_comp_clk 3'd2: en_ldo_5m_b 3'd3: comp_clk 3'd4: scdcdc_conv_1b1 3'd5: scdcdc_conv_1b2 3'd6: scdcdc_conv_1b3 3'd7: 0
10:6	RW	trim_clamp_lp	1	trim value lower clamp value when sel high freq_b is 1
5:1	RW	trim_clamp_hp	16	trim value lower clamp value when sel high freq_b is 0
0	RW	scdcdc_soft_reset	0	soft reset signal for scdcdc fsm

16.21.5.5 BG_LDO_REG1

Table 16.663. Band-Gap LDO Control Register Description

Bit	Access	Function	Reset Value	Description
31:22	-	Reserved	-	It is recommended to write these bits to 0.
21	RW	LDO_0P6_BYPASS	1'b0	bypass signal for DCDC1p1_lp_500uA
20:18	RW	LDO_0P6_CTRL	3'd2	vref for DCDC1p1_lp_500uA 3'd0 - 0.8V 3'd1 - 0.75V 3'd2 - 0.7V 3'd3 - 0.65V 3'd4 - 0.6V 3'd5 - 0.55V
17	-	Reserved	-	Reserved
16	RW	LDO_0P6_LP_MODE	1'b0	enable low power mode, otherwise in high power mode

Bit	Access	Function	Reset Value	Description
15	RW	LDO_0P6_ENABLE	1'b1	enable digital LDO
14:5	-	Reserved	-	It is recommended to write these bits to 0.
4	RW	test_amux_en	1'b0	Enable analog mux to test reference voltages
3:1	RW	test_amux_sel	3'd0	Select for analog mux 3'd0: Vbg_core 3'd1: vref_1p05 3'd2: vref_ulp 3'd3: vbg_lp_buff
0	-	Reserved	-	It is recommended to write these bits to 0.

16.21.5.6 BG_SCDC_READ_BACK

Table 16.664. Band-Gap Read-Back Register Description

Bit	Access	Function	Reset Value	Description
31:22	-	Reserved	-	-
21:18	R	scdcdc_curr_prog	0	Scdcdc curr prog read back
17:13	R	trim_4mhz_ro	16	Trim value for scdcdc ring oscillator
12:1	-	Reserved	-	-
0	R	sync_reset_read	0	Read back for sync reset with ro clock

16.21.5.7 BLACKOUT_MON_EN_REG

Table 16.665. Blackout monitor Enable Register Description

Bit	Access	Function	Reset Value	Description
31:22	-	Reserved	-	-
21:6	-	Reserved	0	Reserved
5	RW	blackout_en	1	Control signal for blackout monitor from SPI
4:0	RW	bg_r	16	Bandgap Voltage Programming

16.22 BOD

16.22.1 General Description

The Nano Power BOD consists of a comparator, reference buffer, scaler and a resistor bank.

The comparator compares inputs p and n to produce an output, cmp_out.

p > n, cmp_out = 1

p < n, cmp_out = 0

16.22.2 Features

The following cases of comparison are possible

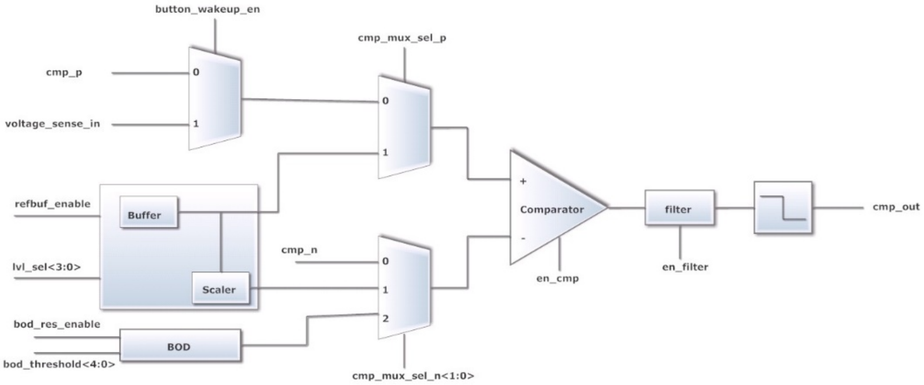
1. Compare external pin inputs
2. Compare external pin input to internal voltages.
3. Compare internal voltages.

The comparator has to be enabled to enable reference buffer and resistor bank.

The reference buffer buffers the bandgap reference voltage and also has a voltage divider which gives programmable 100mV to 1.1V output.

The resistor bank is used to detect battery voltage from 1.75V to 3.65V with a 50mV step. Since each bod_threshold value refers to a particular battery voltage, the battery voltage is found by comparing resistor bank output with reference buffer output using comparator for different bod_threshold values.

16.22.3 Block Diagram



16.22.4 GPIO Pins

Pin	GPIO
cmp_p	UULP_VBAT_GPIO_3
cmp_n	UULP_VBAT_GPIO_4
voltage_sense_in	UULP_VBAT_GPIO_2

Note: UULP_VBATT_AD_GPIO_x has to be in mode 7

16.22.5 Functional Description

- UULP_VOUTSCDC and UULP_VBATT_1/2   supplies should be available for the comparators to work.
- Enable reference buffer and resistor bank if required.
- Select the inputs of comparator using input mux selects and then enable the comparator. By default comparator is disabled.

16.22.6 Voltage Scaler

The reference buffer uses 1.2V from ULP_BG as its reference. And the scaler takes this buffered 1.2V as its input.

Scaler is configured using lvl_sel<3:0>. The output of scaler for different scale factors are in the table below

Bandgap_scale_factor	Scaler output	Units
0	0.1	V
1	0.2	V
2	0.3	V
3	0.4	V
4	0.5	V
5	0.6	V
6	0.7	V
7	0.8	V
8	0.9	V
9	1	V
10	1.1	V

16.22.7 Resistor Bank (BOD)

BOD is configured using bod_threshold<5:0>. If the battery voltage goes lower than the referred voltage, the comparator output becomes high. The referred voltages for 0 to 38 are shown in the table below. Words 39 to 63 are mapped to voltage corresponding to word 38.

Resbank output = VBATT*Resbank_output_fraction = VBATT*(200/(290+bod_threshold*8.33))

bod threshold	Resbank_output_fraction	Referred voltage	Unit
0	0.69	1.75	V
1	0.67	1.8	V
2	0.652	1.85	V
3	0.635	1.9	V
4	0.618	1.95	V
5	0.603	2	V
6	0.588	2.05	V
7	0.574	2.1	V
8	0.561	2.15	V
9	0.548	2.2	V
10	0.536	2.25	V
11	0.524	2.3	V
12	0.513	2.35	V
13	0.502	2.4	V
14	0.492	2.45	V
15	0.482	2.5	V
16	0.472	2.55	V
17	0.463	2.6	V
18	0.455	2.65	V
19	0.446	2.7	V
20	0.438	2.75	V
21	0.43	2.8	V
22	0.423	2.85	V
23	0.415	2.9	V
24	0.408	2.95	V
25	0.401	3	V
26	0.395	3.05	V
27	0.388	3.1	V
28	0.382	3.15	V
29	0.376	3.2	V
30	0.37	3.25	V
31	0.365	3.3	V
32	0.359	3.35	V

bod threshold	Resbank_output_fraction	Referred voltage	Unit
33	0.354	3.4	V
34	0.349	3.45	V
35	0.344	3.5	V
36	0.339	3.55	V
37	0.334	3.6	V
38	0.33	3.65	V

16.22.8 Input Modes

Mode	Mode Name	Input 1	Input 2	Enable Bit	Interrupt Generated	manual_cmp_mux_sel (if in manual mode)
1		cmp_p	cmp_n	cmp1_en	cmp_intr_1	3'd0
2		cmp_p	v1p2_scaled	cmp2_en	cmp_intr_2	3'd1
3		cmp_p	VBATT_scaled	cmp3_en	cmp_intr_3	3'd2
4		v1p2_buffered	cmp_n	cmp4_en	cmp_intr_4	3'd3
5	BOD	v1p2_buffered	VBATT_scaled	cmp5_en	cmp_intr_5	3'd4
6	Button Wakeup	voltage_sense_in	VBATT_scaled	button_wakeup_en	cmp_intr_6	3'd5

16.22.9 Comparison Modes

It has two modes, Auto comparison mode and Manual comparison mode.

16.22.9.1 Auto Comparison Mode

In this mode all the 6 comparisons are made in 6 consecutive clock cycles in a slot in regular intervals if all the comparisons are enabled, i.e., cmp_[1-5]_en and button_wakeup_en (or whatever comparisons are enabled).

16.22.9.2 Manual Comparison mode

In manual mode only one particular comparison keeps happening. Select which comparison is to be made using manual_cmp_mux_sel. Also enable the respective cmp_en signal.

16.22.10 Button wakeup

Upto 3 buttons can be detected. When the voltage output from button press falls into any of the following regions (defined using Resbank_Output_Fraction in Resistor Bank section), button detection occurs.

Region for button 1 = {button1_min_value: button1_min_value + button_max_value};

Region for button 2 = {button2_min_value: button2_min_value + button_max_value};

Region for button 3 = {button3_min_value: button3_min_value + button_max_value};

Conditions to avoid false detection:

- Regions should be non-overlapping.
- Region should not contain values in the extremes near 0 and the highest values.
- The min and max values must be programmed based on the resistor values used around buttons.

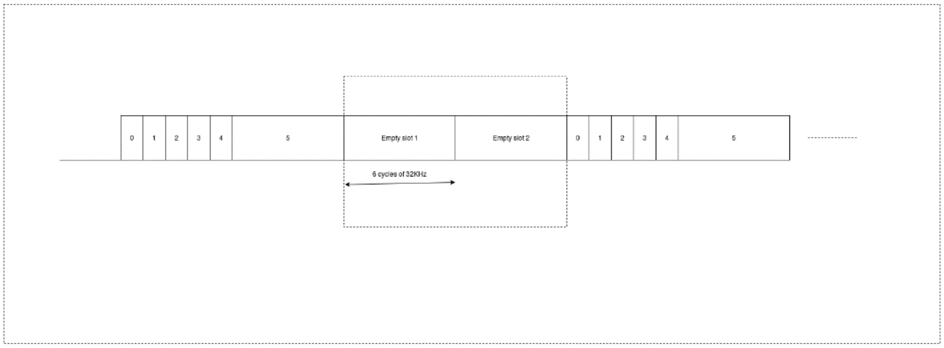
In manual mode, Button wakeup is checked for continuously using a FSM. When a region is detected, cmp_6_intr is raised and sent to FSM and further action is taken. In order to check which button is detected read spi register 1E3 for button1_wakeup, button2_wakeup and button3_wakeup. The corresponding bit will be high.

16.22.11 Slotting

By programming cmp_slot_value, number of empty slots in between 2 slots can be programmed. Each empty slot consists of 6 cycles of 32KHz (ulp_fsm_clk).

In the figure below, 0 to 5 are the 6 modes of comparator and cmp_slot_value is programmed as 2.

- mode 5 is button wakeup, its length is 6 cycles when button wakeup mode is enabled and 1 when disabled.



16.22.12 Register Summary

Table 16.666. Register Summary

Register Name	Offset	Reset Value
BOD_COMP_MODE_REG	1E0	0x00
BOD_COMP_SEL_REG	1E1	0x00
BOD_BUTTON_REG	1E2	0x00
BOD_TEST,PG&VBATT_STATUS_REG	1E3	0x00

16.22.13 Register Description

NANO POWER BOD
BOD_COMP_MODE_REG
BOD : Address : 1E0

NANO POWER BOD

Access	Bits	Register	Default value	Description
R/W	21	auto_cmp_mode_en	1'b0	1: Enable auto slotting of comparator inputs auto comparison mode enable 0: Check for manual mode
R/W	20	manual_cmp_mode_en	1'b0	Manual comparison mode en ; 1: Manually select the inputs for comparator comparision 0: Manual mode disabled
R/W	19:17	manual_cmp_mux_sel	3'd0	Selecting and fixing the inputs of comparator when in manual mode.
R/W	16:1	cmp_slot_value	16'd0	Slot value after which comparator outputs are sampled in auto mode.
R/W	0	cmp_op_filter_en	1'b0	Enable signal for filter at comparator output.
BOD_COMP_SEL_REG				
BOD : Address : 1E1				
Access	Bits	Register	Default value	Description
	Combinational	cmp_en_reg_wr	1'b0	Whenever any of the cmp_ens change or this register of spi is written cmp_en_reg_wr = 1;
R/W	21	button_wakuep_en	1'b0	Enable button wakeup
R/W	20	cmp_1_en	1'b0	Enables comparision 1
R/W	19	cmp_2_en	1'b0	Enables comparision 2
R/W	18	cmp_3_en	1'b0	Enables comparision 3
R/W	17	cmp_4_en	1'b0	Enable signal for comparision 4
R/W	16	cmp_5_en	1'b0	Enable signal for bod detection
R/W	15	cmp_1_polarity	1'b0	1: polarity of comparator is reversed 0: Same op of comparator is taken
R/W	14	cmp_2_polarity	1'b0	1: polarity of comparator is reversed 0: Same op of comparator is taken
R/W	13	cmp_3_polarity	1'b0	1: polarity of comparator is reversed 0: Same op of comparator is taken

NANO POWER BOD				
R/W	12	cmp_4_polarity	1'b0	1: polarity of comparator is reversed 0: Same op of comparator is taken
R/W	11	cmp_5_polarity	1'b0	1: polarity of comparator is reversed 0: Same op of comparator is taken
R/W	10:7	bandgap_scale_factor	4'd0	Programmability for scaling bandgap v1p2
R/W	6:1	batt_scale_factor	6'd0	Programmability for scaling vbatt
R	0	Reserved	1'd0	Reserved
BOD_BUTTON_REG				
BOD : Address : 1E2				
Access	Bits	Register	Default value	Description
R	21	sync_reset_read	1'b0	read back synced reset with 32KHz fsm clock
R/W	20 : 17	button_max_value	4'd0	MAximum range for each button wakeup detect
R/W	16 : 12	button1_min_value	5'd0	Minimum threshold value to detect button 1 wakeup
R/W	11 : 7	button2_min_value	5'd0	Minimum value for button 2 detect
R/W	6 : 2	button3_min_value	5'd0	Minimum value for button 3 detect (calib word lies in the range of min3_value to min3_value + button_max_value)
R/W	1:0	comparator hysterisis	2'd0	comparator hysterisis
BOD_TEST,PG&VBATT_STATUS_REG				
BOD : Address : 1E3				
Access	Bits	Register	Default value	Description
R/W	21	en_bod_test_mux	1'd0	To enable test mux to connect to GPIO pad
R/W	20:19	bod_test_sel	2'd0	Select bits for test mux
R	18	button1_wakeup (read)	1'b0	1 => button 1 detected for wakeup 0 => button 1 not detected for wakeup
R	17	button2_wakeup (read)	1'b0	button 2 wakeup status
R	16	button3_wakeup (read)	1'b0	button 3 wakeup status

NANO POWER BOD				
R/W	15	bod_pwrgate_en_n_ulp_button_calib	1'b1	powergate enable signal for button calib and vbatt status checking block
R/W	14	blackout_en	1'b0	Unused
R/W	13:9	brown_out_interrupt_threshold	5'd2	Threshold for brown out detection beyond which interrupt is not given to NPSS
R/W	8	periodic_trigger_en	1'b0	Periodic checking for battery status enable
only W	7	check_vbatt_status	1'b0	pulse signal, combinational logic. to check battery status
R	6	vbatt_status_valid	1'b0	Valid signal for reading vbatt status
R	5:0	vbatt_status	1'b0	Status of battery, 31 -> full , 0 -> low

Table 16.667. Register Description

BOD BUTTON READ				
BOD : Address : 1E4				
Access	Bits	Register	Default value	Description
R/W	21:16	read_button_word	6'd0	Word captured when button pressed
-	15:0	reserved	0	Reserved

16.23 Calendar

16.23.1 General Description

Calendar block acts a RTC with time in seconds, minutes, hours, days, months, years and centuries. The real time can also be read through APB with accuracy less than a second by reading the milli second count value and further less also by reading the number of counts of APB clock in 1 milli second of RTC clock. Accuracy is high.

16.23.2 Features

- Calendar block can provide a seconds trigger and also a msec trigger.
- Calendar block takes care of number of days in each month and also leap years. It can count up to 4 centuries.
- Real time is readable through APB and also programmable through APB.
- Option to choose either RC clock RO clock as calendar clock.

16.23.3 Functional Description

Calendar block counts time based on the 24 bit time_period measured. Calendar block has separate counters for milliseconds (7 bits: 0 - 124), 1/8th seconds (3 bits: 0 - 7), seconds (6 bits: 0 - 59), minutes (6 bits: 0 - 59), hours (5 bits: 0 - 23), days(5 bits: 1 - 31), months (4 bits: 1 - 12), years (7 bits: 0 - 99) and centuries(2 bits: 0 - 3). All these counters can be programmed to a required value through APB by programming prog_time_trig bit to 1.

Every 4th year, year[1:0] == 00 is considered a leap year. The clock input to this block can be selected through APB by rtc_clk_sel bit. rtc_clk_sel = 0 : RC clock and rtc_clk_sel = 1 : RO clock.

16.23.4 Register Summary

Base Address: 0x2404_8200

Table 16.668. Register Summary

Register Name	Offset
Section 16.23.5.1 MCU CAL ALARM PROG 1	0x1C
Section 16.23.5.2 MCU CAL ALARM PROG 2	0x20
Section 16.23.5.3 MCU CAL POWERGATE REG	0x24
Section 16.23.5.4 MCU CAL PROG TIME 1	0x28
Section 16.23.5.5 MCU CAL PROG TIME 2	0x2C
Section 16.23.5.6 MCU CAL READ TIME MSB	0x30
Section 16.23.5.7 MCU CAL READ TIME LSB	0x34
Section 16.23.5.9 MCU CAL READ COUNT TIMER	0x38
Section 16.23.5.8 MCU CAL SLEEP CLK COUNTERS	0x3C
Section 16.23.5.10 MCU CAL KEY EANBLE	0x40

16.23.5 Register Description

16.23.5.1 MCU CAL ALARM PROG 1

Table 16.669. MCU CAL ALARM PROG 1 Register Description

Bit	Access	Function	Reset Value	Description
31:27				
26:22	R/W	prog_alarm_hour	0	hours value of alarm time
21:16	R/W	prog_alarm_min	0	mins value of alarm time
15:10	R/W	prog_alarm_sec	0	seconds value of alarm time
9:0	R/W	prog_alarm_msec	0	milli seconds value of alarm time

16.23.5.2 MCU CAL ALARM PROG 2

Table 16.670. MCU CAL ALARM PROG 2 Register Description

Bit	Access	Function	Default Value	Description
31	R/W	alarm_en	0	1-alarm function enable for calendar alarm interrupt is generated when real time matches alarm time
30:25	-	Reserved	-	-
24:23	R/W	prog_alarm_century	0	century count in alarm time
22:16	R/W	prog_alarm_year	0	year count in alarm time 0 - 99
15:12	--	--	0	--

Bit	Access	Function	Default Value	Description
11:8	R/W	prog_alarm_month	0	month count in alarm time
7:5	--	Reserved	0	Reserved
4:0	R/W	prog_alarm_day	0	day count in alarm time 1-31

16.23.5.3 MCU CAL POWERGATE REG

Table 16.671. MCU CAL POWERGATE REG Register Description

Bit	Access	Function	Default Value	Description
31:4	-	Reserved	-	-
3	R/W	static_combi_rtc_pg_en	1'b0	Enable static combo RTC power gate
2	R/W	disable_combi_dyn_pwrgate_en	1'b0	Disable option for dynamic combo RTC power gate
1	R/W	enable_calendar_combi	1'b0	Enable calendar combitional logic block
0	R/W	pg_en_calendar	1'b1	Start calendar block

Note

Only Accessible if RTC_KEY is Enabled

16.23.5.4 MCU CAL PROG TIME 1

Table 16.672. MCU CAL PROG TIME 1 Register Description

Bit	Access	Function	Default Value	Description
31:27	-	Reserved	-	-
26:22	R/W	prog_hour	5'd0	hours value to be programmed to real time in calendar when pro_time_trig is 1 0 - 23
21:16	R/W	prog_min	6'd0	minutes value to be programmed to real time in calendar when pro_time_trig is 1 0- 59
15:10	R/W	prog_sec	6'd0	seconds value to be programmed to real time in calendar when pro_time_trig is 1 0 - 59
9:0	R/W	prog_msec	10'd0	Milli seconds value to be programmed to real time in calendar when pro_time_trig is 1 0-999

Note:

Only Accessible if RTC_KEY is Enabled

16.23.5.5 MCU CAL PROG TIME 2

Table 16.673. MCU CAL PROG TIME 2 Register Description

Bit	Access	Function	Default Value	Description
31	W	prog_time_trig	0	1 - load the programmed to the real time in calendar block
30:25	-	Reserved	-	-
24:23	R/W	prog_century	0	century value to be programmed to real time in calendar when pro_time_trig is 1 0 - 3
22:16	R/W	prog_year	0	year value to be programmed to real time in calendar when pro_time_trig is 1 0 - 99
15:12	-	Reserved	-	-
11:8	R/W	prog_month	0	month value to be programmed to real time in calendar when pro_time_trig is 1 1-12
7:5	R/W	prog_week_day	0	program which week day it is
4:0	R/W	prog_day	5'd0	day count value to be programmed to real time in calendar when pro_time_trig is 1 1- 31

Note:

Only Accessible if RTC_KEY is Enabled

16.23.5.6 MCU CAL READ TIME MSB

Table 16.674. MCU CAL READ TIME MSB Register Description

Bit	Access	Function	Reset Value	Description
31:16	-	Reserved	-	-
15:0	R	Real_time[47:32]	--	Read value of current time in calendar block real time = {century_count , year_count, months, week days,days,hours, mins, secs, milliseconds} real_time[47:46] = century count real_time[45:39] = year_count real_time[38:35] = months_count real_time[34:32] = week day

16.23.5.7 MCU CAL READ TIME LSB

Table 16.675. MCU CAL READ TIME LSB Register Description

Bit	Access	Function	Reset Value	Description
31:0	R	Real_time[31:0]	--	Read value of current time in calendar block real time = {century_count , year_count, months, week days,days,hours, mins, secs, milliseconds} real_time[31:27] = days_count real_time[26:22] = hours_count real_time[21:16] = mins_count real_time[15:10] = seconds_count real_time[9:0] = milliseconds count

16.23.5.8 MCU CAL SLEEP CLK COUNTERS

Table 16.676. MCU CAL SLEEP CLK COUNTERS Register Description

Bit	Access	Function	Reset Value	Description
31:28	-	Reserved	-	-
27:16	R	pclk_count_wrt_sleep_clk	--	no. of APB clks in 1 sleep clock duration
15:12	-	Reserved	-	-
11:0	R	sleep_clk_duration	--	No of sleep clks with respect to APB clock so far from the posedge of sleep clk

16.23.5.9 MCU CAL READ COUNT TIMER

Table 16.677. MCU CAL READ COUNT TIMER Register Description

Bit	Access	Function	Reset Value	Description
31:27	-	Reserved	-	-
26:0	R	read_count_timer	--	Read timer which increments by time period value to reach to count milliseconds

16.23.5.10 MCU CAL KEY EANBLE

Table 16.678. MCU CAL KEY EANBLE Register Description

Bit	Access	Function	Reset Value	Description
31:0	W	rtc_key	0x55555555	Program the below key to enable access to program Watch dog registers 0x8D845F4C Program the below key to disable access to program Watch dog registers 0x55555555

16.24 GPIO Timestamp[2]

16.24.1 General Description

The Block is used for capturing the Timestamp of GPIO signal going high from SLEEP to Active state.

16.24.2 Features

- Option to choose 1 GPIO out 4 UULP GPIO's for time stamping application.
- Uses a 32MHz RC clock for time stamping.

16.24.3 Programming Sequence

16.24.3.1 Configuring

- GPIO Timesamping is available only during Sleep state.
- Choose one GPIO on which Time stamping is required.
  - program 'timestamping_on_gpio0' register in address **MCU_GPIO_TIMESTAMP_CONFIG** to get GPIO timestamp of UULP_GPIO_0.
  - program 'timestamping_on_gpio1' register in address **MCU_GPIO_TIMESTAMP_CONFIG** to get GPIO timestamp of UULP_GPIO_1.
  - program 'timestamping_on_gpio2' register in address **MCU_GPIO_TIMESTAMP_CONFIG** to get GPIO timestamp of UULP_GPIO_2.
  - program 'timestamping_on_gpio3' register in address **MCU_GPIO_TIMESTAMP_CONFIG** to get GPIO timestamp of UULP_GPIO_3.
- Program 'enable_gpio_timestamping' register in address **MCU_GPIO_TIMESTAMP_CONFIG** for enabling time stamping application.

16.24.3.2 Reading

- Poll register 'timestamping_done' in address **MCU_GPIO_TIMESTAMP_CONFIG**. When the signal goes high will indicated the timestamp values is ready for reading.
- Read register **MCU_GPIO_TIMESTAMP_READ** to get the GPIO TimeStamp.
  - Value of 'gpio_event_count_partial' will indicating number for 32MHz clock present in 1 Sleep clock.
  - Value of 'gpio_event_count_full' will indicating the duration from GPIO going high to first Sleep clock posedge from GPIO going high with respect to 32MHz clock.

16.24.3.3 Re-Initiating

- Once timestamp value is read. Write to address **MCU_GPIO_TIMESTAMP_READ** with value 0 to clear timestamp values so that is can get timestamp of next GPIO event.

16.24.4 Register Summary

Base Address: 0x2404_8100

Table 16.679. Register Summary

Register Name	Offset	Description
Section 16.24.5.1 MCU_GPIO_TIMESTAMP_CONFIG	0x28	
Section 16.24.5.2 MCU_GPIO_TIMESTAMP_READ	0x2C	

16.24.5 Register Description

16.24.5.1 MCU_GPIO_TIMESTAMP_CONFIG

Table 16.680. MCU_GPIO_TIMESTAMP_CONFIG Register

Bit	Access	Function Name	Reset Value	Description
31:16	-	Reserved	-	
15	R/W	timestamping_done	0	This signal indicated Timestamp of GPIO is ready for reading.
14:5	-	Reserved	-	
4	R/W	timestamping_on_gpio3	0	Enable GPIO time stamping on GPIO3
3	R/W	timestamping_on_gpio2	0	Enable GPIO time stamping on GPIO2
2	R/W	timestamping_on_gpio1	0	Enable GPIO time stamping on GPIO1
1	R/W	timestamping_on_gpio0	0	Enable GPIO time stamping on GPIO0
0	R/W	enable_gpio_timestamping	0	Enable GPIO time stamping Feature. This will enable measurement of GPIO high duration from SLEEP to wakeup

Note:

The GPIO used for doing timespaming application are UULP GPIO's

16.24.5.2 MCU_GPIO_TIMESTAMP_READ

Table 16.681. MCU_GPIO_TIMESTAMP_READ Register

Bit	Access	Function Name	Reset Value	Description
31:27	-	Reserved	-	
26:16	R	gpio_event_count_partial	0	Counter value indicating number for 32MHz clock present in 1 Sleep clock (MCU FSM Clock)
15:11	-	Reserved	-	
10:0	R	gpio_event_count_full	0	Counter value indicating the duration from GPIO going high to first Sleep clock(MCU FSM Clock) posedge from GPIO going high with respect to 32MHz clock.

16.25 Secure Storage

16.25.1 General Description

The Block is used for storing configuration values with data protection feature.

16.25.2 Features

- MCU has 3 set's for storage block
  - First chunk is 64 bits.
  - Second chunk is 64 bits &
  - Third Chunks is 128 bits
- Each chunk is a power domain.
- Secure mode is available for first and second Chunk.
- Storage space can be used for storing Configuration values

16.25.3 Functional Description

16.25.3.1 Block Diagram

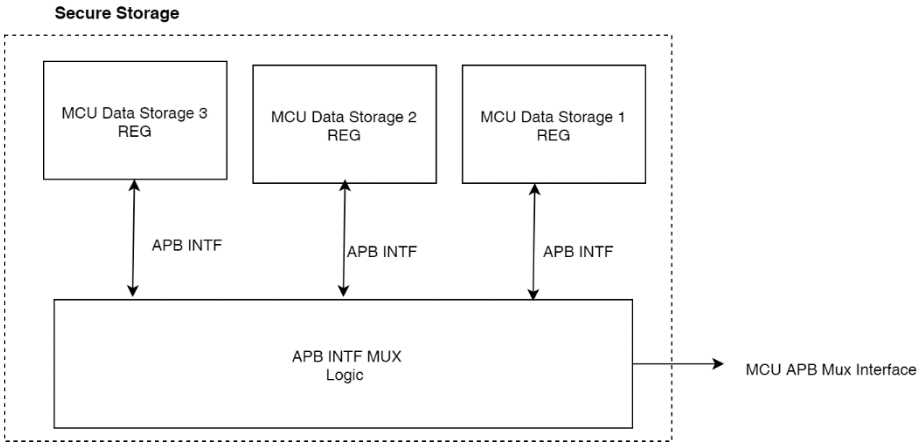


Figure 16.48. Block Diagram of Secure Storage

MCU_STORAGE_REG0 and MCU_STORAGE_REG1, 32-bit each, form the MCU data storage 1 (64-bit chunk).

MCU_STORAGE_REG2 and MCU_STORAGE_REG3, 32-bit each, form the MCU data storage 2 (64-bit chunk).

MCU_STORAGE_REG4, MCU_STORAGE_REG5, MCU_STORAGE_REG6 and MCU_STORAGE_REG7, 32-bit each, form the MCU data storage 1 (128-bit chunk).

16.25.4 Programming Sequence

16.25.4.1 Writing to MCU Storage Domain

1. Enable Write Access to NWP Storage domain by programming to MCU_STORAGE_WRITE_KEY register. Please see register description.
2. Write to register MCU_STORAGE_REG0 with required value.
3. Write to register MCU_STORAGE_REG1 with required value.
4. Write to register MCU_STORAGE_REG2 with required value.
5. Write to register MCU_STORAGE_REG3 with required value.
6. Write to register MCU_STORAGE_REG4 with required value.
7. Write to register MCU_STORAGE_REG5 with required value.
8. Write to register MCU_STORAGE_REG6 with required value.
9. Write to register MCU_STORAGE_REG7 with required value.
10. Diasble Write Access to NWP Storage domain by programming to MCU_STORAGE_WRITE_KEY register. Please see register description.

16.25.4.2 Reading from MCU Storage Domain

1. Read from register MCU_STORAGE_REG0 to fetch required value.
2. Read from register MCU_STORAGE_REG1 to fetch required value.
3. Read from register MCU_STORAGE_REG2 to fetch required value.
4. Read from register MCU_STORAGE_REG3 to fetch required value.
5. Read from register MCU_STORAGE_REG4 to fetch required value.
6. Read from register MCU_STORAGE_REG5 to fetch required value.
7. Read from register MCU_STORAGE_REG6 to fetch required value
8. Read from register MCU_STORAGE_REG7 to fetch required value.

16.25.5 Secure Mode

There are twp option available:

1. Reset Protection
2. Power Domain Protection from accidental turn-off of power domain controls of Storage domain.

The feature will enabled by Boot load code.

16.25.5.1 Reset Protection

When Bit[2] is Set in register **NWPAON_POR_CTRL_BITS**. Storage domain's will be immune to Reset from Pin, WDT Reset and Host Reset Request.

Note:

Once the Bit is set, it can not be cleared.

16.25.5.2 Power Domain Protection

When **Write_protect/Bit[4]** is set in register **NWPAON_POR_CTRL_BITS**, storage domains are protected from accidental turn-off Power-Supply to these blocks, and once data is written to the protected registers is can not be over-written again

Note:

Once the Bit is set, it can not be cleared.

16.25.6 Register Summary

Base Address: 0x2404_8500

Table 16.682. Base Address: 0x2404_8500

Register Name	Offset	Description
Section 16.25.7.1 MCU_STORAGE_REG0	0x80	This register Can be used to storing 32bits of Data.
Section 16.25.7.2 MCU_STORAGE_REG1	0x84	This register Can be used to storing 32bits of Data.
Section 16.25.7.3 MCU_STORAGE_REG2	0x88	This register Can be used to storing 32bits of Data.
Section 16.25.7.4 MCU_STORAGE_REG3	0x8C	This register Can be used to storing 32bits of Data.
Section 16.25.7.5 MCU_STORAGE_REG4	0x90	This register Can be used to storing Data
Section 16.25.7.6 MCU_STORAGE_REG5	0x94	This register Can be used to storing Data
Section 16.25.7.7 MCU_STORAGE_REG6	0x98	This register Can be used to storing Data
Section 16.25.7.8 MCU_STORAGE_REG7	0x9C	This register Can be used to storing Data

Base Address: 0x2404_8700

Table 16.683. Base Address: 0x2404_8700

Register Name	Offset	Description
Section 16.25.7.9 MCU_STORAGE_WRITE_KEY	0x00	Programming the key will enable or disable access to program MCU storage register

16.25.7 Register Description

16.25.7.1 MCU_STORAGE_REG0

Table 16.684. MCU_STORAGE_REG0 Register Description

Bit	Access	Function Name	Reset Value	Description
31:0]	R/W	MCU_STORAGE_WORD_0	0	This register Can be used to storing 32bits of Data. If Write_protect is set and a value is written to register, then data will not be overwritten

16.25.7.2 MCU_STORAGE_REG1

Table 16.685. MCU_STORAGE_REG1 Register Description

Bit	Access	Reset Value	Reset Value	Description
31:0	R/W	MCU_STORAGE_WORD_1	0	This register Can be used to storing 32bits of Data. If Write_protect is set and a value is written to register, then data will not be overwritten

16.25.7.3 MCU_STORAGE_REG2

Table 16.686. MCU_STORAGE_REG2 Register Description

Bit	Access	Function Name	Reset Value	Description
31:0	R/W	MCU_STORAGE_WORD_2	0	This register Can be used to storing 32bits of Data. If Write_protect is set and a value is written to register, then data will not be overwritten

16.25.7.4 MCU_STORAGE_REG3

Table 16.687. MCU_STORAGE_REG3 Register Description

Bit	Access	Function Name	Reset Value	Description
31:0	R/W	MCU_STORAGE_WORD_3	0	This register Can be used to storing 32bits of Data. If Write_protect is set and a value is written to register, then data will not be overwritten

16.25.7.5 MCU_STORAGE_REG4

Table 16.688. MCU_STORAGE_REG4 Register Description

Bit	Access	Function Name	Reset Value	Description
31:0	R/W	MCU_STORAGE_WORD_4	0	This register Can be used to storing Data

16.25.7.6 MCU_STORAGE_REG5

Table 16.689. MCU_STORAGE_REG5 Register Description

Bit	Access	Function Name	Reset Value	Description
31:0	R/W	MCU_STORAGE_WORD_5	0	This register Can be used to storing Data

16.25.7.7 MCU_STORAGE_REG6

Table 16.690. MCU_STORAGE_REG6 Register Description

Bit	Access	Function Name	Reset Value	Description
31:0	R/W	MCU_STORAGE_WORD_6	0	This register Can be used to storing Data

16.25.7.8 MCU_STORAGE_REG7

Table 16.691. MCU_STORAGE_REG7 Register Description

Bit	Access	Function Name	Reset Value	Description
31:0	R/W	MCU_STORAGE_WORD_7	0	This register Can be used to storing Data

16.25.7.9 MCU_STORAGE_WRITE_KEY

Table 16.692. MCU_STORAGE_WRITE_KEY Register Description

Bit	Access	Function Name	Description	Description
31:0	W	MCU_STORAGE_KEY	By default the Access to MCU storage Register is enabled	Program the below key to enable access to program MCU storage register 0x91437B2B Program the below key to disable access to program MCU storage register 0xCCCCCCCC

16.26 Sleep Clock Calibrator

16.26.1 General Description

In this block, the time periods of 32KHz RC clock, 32KHz RO clock and 32KHz XTAL clock can be calibrated. Apart from this, there is a block to generate periodic triggers to recalibrate time periods for temperature changes and periodic changes. Also there is another block which gives seconds trigger approximately with every 2^10 clocks of FSM clock.

16.26.2 Features

- 32KHz RC clock time period calibration is done using time period of known XTAL 40MHz clock programmed through APB.
- 32KHz RO calibration uses time period of RC 32KHz clock as reference.
- The frequency of RO calibration is programmable and calibration can happen at a maximum rate of 8 times per second.
- RC calibration can be triggered every 5 secs or every 10 secs or every 15 secs or every 30secs or temperature based, when temperature change is more than the given maximum temperature change acceptable.
- Temperature sensor is triggered to measure the temperature change every 1 sec, 2sec, 4 sec or 5 seconds.
- RTC time period can be can be made either of clocks time periods based on the clock being used as fsm clk.

16.26.3 Functional Description

16.26.3.1 RC Time Period Calibration

Calendar block provides rc_calibration trigger in regular intervals for time period calibration based on rc_trigger_time_sel.

2'b00 : Every 30 seconds (default)

2'b01 : Every 15 seconds

2'b10 : Every 10 seconds

2'b00 : Every 5 seconds

After every trigger, 40mhz XTAL clock is enabled and waits untill the clock gets settled. The settling time is programmable through APB, default value is 7'd64. Time periods used and measured in this block are assumed to have a **granularity of 10ps for the LSB**. Time-period of reference clock 13 bits must be programmed in the register before the trigger.

There are 2 counters. One running on RC 32KHz clock and the other on Reference 40mhz clock. Counter 1 counts for 2^no_of_rc_clocks = N_rc (can be 1, 2, 4 or 8) .In the mean time counter 2 (28 bit) accumulates the time period of ref clock every posedge, say the counter encounters N_ref clock cycles, then the counter 2 value will be N_ref * T_ref.

Total time = N_ref * T_ref = N_rc * T_rc_inst where T_rc_inst is the instantaneous time period of RC 32KHz clock.

Therefore, **T_rc_inst = (N_ref * T_ref) / N_rc** == Counter_2_value << no_of_rc_clocks

The instantaneous time period measured is averaged with the previously measured value.

T_RC = T_RC_PREV* (1 –alpha_rc) + alpha_rc * T_RC_INST

Where alpha_rc is a fraction can be 1 , 1/2 , 1/4 , 1/8 , 1/16, 1/32, 1/64, 1/128 programmed through APB.

16.26.3.2 RO Time Period Calibration

RO timeperiod calibration block is triggered very frequently as RO clock is not stable based on the ro_trigger_time_sel .

2'b00 : Every second (default)

2'b01 : Every 1/2 second

2'b10 : Every 1/4 second

2'b11 : Every 1/8 second

RC 32KHz clock acts as reference clock for this calibration. The procedure and theory is similar to that of RC calibration. But as both the clocks are of relatively equal frequencies, higher no.of clock cycles are counted for to notice the difference.

Here the counter 1 counts for 2^no_of_ro_clocks (can be a 1, 2, 4,........... 2^16, default is 2^10). Counter 2 is 40 bits to account for so much time. With each posedge od 32KHz RC clock counter 1 increments by timeperiod of RC clock. With each posedge of RO clock counter 1 increments by 1 to total N_ro.

T_ro_inst = (N_rc * T_rc) / N_ro

T_ro = T_ro_PREV * (1 –alpha) + alpha * T_ro_inst

Where alpha_ro is a fraction, which can be different from alpha rc programmed through APB.

16.26.3.3 Temperature Change Detector

Calendar block triggers temperature sensor at regular intervals of time based on the temp_trigger_time_sel value

2'b00 : Every second (default)

2'b01 : Every 2 seconds

2'b10 : Every 4 seconds

2'b11 : Every 5 seconds

This block keeps a note of temperature at which last calibration of RC time period happened (Temp_prev). After every trigger of temperature sensor It checks for the temperature measurement done signal and reads the temperature value after done.

If this temperature value is beyond the maximum temperature change acceptable, It triggers the RC calibration and copies this value to T_prev. Maximum temperature change acceptable can be changed through APB default (5'd5).

Everytime RC calibration block is triggered, this is also triggered to note the Temp_prev value.

16.26.4 Register Summary

Base Address: 0x2404_8200

Table 16.693. Base Address: 0x2404_8200

Register Name	Offset
Section 16.26.5.1 MCU CAL RO TIMEPERIOD READ	0x00
Section 16.26.5.2 MCU CAL TIMER CLOCK PERIOD	0x04
Section 16.26.5.3 MCU CAL TEMP PROG REG	0x08
Section 16.26.5.4 MCU CAL START REG	0x0C
Section 16.26.5.5 MCU CAL REF CLK SETTLE REG	0x10
Section 16.26.5.6 MCU CAL RC TIMEPERIOD READ	0x14
Section 16.26.5.7 MCU CAL REF CLK TIMEPERIOD REG	0x18

16.26.5 Register Description

16.26.5.1 MCU CAL RO TIMEPERIOD READ

Table 16.694. MCU CAL RO TIMEPERIOD READ Register Description

Bit	Access	Function	Reset Value	Description
31:25	-	Reserved	-	
24:0	R	timeperiod_ro	0	Calibrated RO timeperiod

16.26.5.2 MCU CAL TIMER CLOCK PERIOD

Table 16.695. MCU CAL TIMER CLOCK PERIOD Register Description

Bit	Access	Function	Reset Value	Description
31	R	spi_rtc_timer_clk_period_applied	0	Indicated SOC programmed rtc_timer clock period is applied at KHz clock domain. 1 - Programmed period applied 0 - Programmed period is not applied yet
30:25	-	Reserved	-	
24:0	W/R	rtc_timer_clk_period	0	RTC timer clock period programmed by SOC MS 8 bit are for Integer part & LS 17bit are for Fractional part Ex: 32Khz clock = 31.25us ==> 31.25*2^17 = 4096000 = 0x3E8000

16.26.5.3 MCU CAL TEMP PROG REG

Table 16.696. MCU CAL TEMP PROG REG Register Description

Bit	Access	Function	Reset Value	Description
31:25	-	Reserved	-	
24	R/W	rtc_timer_period_mux_sel	0	1- calibrated value is taken 0- SPI value is taken
23	R/W	periodic_temp_calib_en	0	Enable periodic checking of temperature
22:21	R/W	temp_trigger_time_sel	0	2'd0: Every second 2'd1: every 2 seconds 2'd2: every 4 seconds 2'd3: every 5 seconds
20:16	R/W	max_temp_change	5	maximum temperature change after which rc calibration must be trigger
15:1	-	Reserved	-	
0	R/W	bypass_calib_pg	0	To bypass power gating and keep all the blocks always on

16.26.5.4 MCU CAL START REG

Table 16.697. MCU CAL START REG Register Description

Bit	Access	Function	Reset Value	Description
31:30	-	Reserved	-	
29:27	R/W	vbatt_trigger_time_sel	1	trigger to ipmu block for checking vbatt status periodicaly 3'd6 : Every 2 minutes 3'd5: Every minute 3'd4: Every 30 secs 3'd3 : every 15 secs, 3'd2: every 10 secs, 3'd1: every 5 secs 3'd0: every second
26	R/W	low_power_trigger_sel	0	1 - seperate counter runs based 2^15 clocks of 32KHz clock = 1sec 0 - calendar runs and triggers are generated based on calendar
25	R/W	rc_xtal_mux_sel	0	0 - RC clock calibration happens 1 - XTAL 32khz clock timeperiod calibration occurs with reference clock as 40mhz xtal This should not be changed in the middle of process. Must be changed only once.
24	W	start_calib_rc	0	to initiate RC calibration
23	W	start_calib_ro	0	to initiate RO calibration
22	R/W	periodic_rc_calib_en	0	periodically calibrate RC timeperiod based rc trigger time sel
21	R/W	periodic_ro_calib_en	0	periodically calibrate RO timeperiod based ro trigger time sel
20:18	R/W	rc_trigger_time_sel	0	3'd0 : Every 5secs 3'd1 : every 10 secs, 3'd2: every 15 secs, 3'd3: every 30 secs 3'd4: every minute 3'd5: Every 2 minutes
17:16	R/W	ro_trigger_time_sel	0	2'd3 : 8 times in a second 2'd2 : 4 times in a second 2'd1 : 2 times in a second 2'd0 : 1 time in a second

Bit	Access	Function	Reset Value	Description
15:13	R/W	rc_settle_time	5	no of clocks of RO for the RC clk to settle when enabled
12:10	R/W	no_of_rc_clks	3	2^no_of_rc_clocks = no of rc clocks used in calibration
9:6	R/W	no_of_ro_clks	10	2^no_of_ro_clks no of clocks of ro clock counts for no of rc clocks in that time to measure timeperiod
5:3	R/W	alpha_rc	2	alpha = 1/2^alpha_rc , averaging factor of RC timeperiod T = alpha(t_inst) + (1- alpha)t_prev
2:0	R/W	alpha_ro	2	alpha = 1/2^alpha_ro , averaging factor of RO timeperiod T = alpha(t_inst) + (1- alpha)t_prev

16.26.5.5 MCU CAL REF CLK SETTLE REG

Table 16.698. MCU CAL REF CLK SETTLE REG Register Description

Bit	Access	Function	Reset Value	Description
31:18	-	Reserved	-	
17	R	valid_ro_timeperiod	0	Valid signal for reading RO timeperiod
16	R	valid_rc_timeperiod	0	Valid signal for reading RC timeperiod calibrated
15:7	-	Reserved	-	
6:0	R/W	xtal_settle	64	no of 32khz clocks for xtal 40mhz clk to settle

16.26.5.6 MCU CAL RC TIMEPERIOD READ

Table 16.699. MCU CAL RC TIMEPERIOD READ Register Description

Bit	Access	Function	Reset Value	Description
31:25	-	Reserved	-	
24:0	R	timeperiod_rc	0	Calibrated RC timeperiod

16.26.5.7 MCU CAL REF CLK TIMEPERIOD REG

Table 16.700. MCU CAL REF CLK TIMEPERIOD REG Register Description

Bit	Access	Function	Reset Value	Description
31:24	-	Reserved	-	

Bit	Access	Function	Reset Value	Description
23:0	R/W	timeperiod_ref_clk	0x33_3333	timeperiod of reference clk with each bit corresponding to granularity of 2^27 = 1us ex: 40Mhz Period is 25ns. (25ns/1us = 25e-3us = 25e-3*2^27 = 24’d3355443 = 0x33_3333 ex: 36Mhz Period is 27.778ns. (27.778ns/1us = 27.8e-3us = 27.8e-3*2^27 = 24’d3728270 = 0x38_E38E

16.26.6 Programming Sequence

16.26.6.1 RC Time Period Calibration

1. Write the timeperiod of ref clock and settling time for reference clock being used (npss_ref_clk from IPMU) into **MCU CAL REF CLK TIMEPERIOD REG**.
   Default values are for xtal_settle(40mhz clock with settling time) in register **MCU CAL REF CLK SETTLE REG** is of 64 clocks of 32khz clock = 64 * 31.25us = 2ms.
2. In **MCU CAL START REG** program alpha_rc, no_of_rc_clocks based on the requirement. **rc_xtal_mux_sel** = 0 and also wirte to **start_calib_rc** = 1;
3. If *periodic* calibration of RC is required, Enable bit 21, of MCU CAL START REG, periodic_RC_calib_en , and select rc_trigger_time_sel bits based on the rate at with calibration needs to happen.
4. If *temperature* based calibration has to happen along with periodic calibration, also enable, 23:16 bits of **MCU CAL TEMP PROG REG** based on the description. max_temp_change is the value increase in the temperature after which RC has to be recalibrated.
5. Look for 24th bit in **MCU CAL RC TIMEPERIOD READ** if 1, RC timeperiod is valid, Already the timeperiod calibration has occurred once.Inorder to read the timeperiod of RC, read 23:0 bits of same register.

16.26.6.2 XTAL 32K Time Period Calibration

1. Write the timeperiod of ref clock and settling time for reference clock being used (npss_ref_clk from IPMU) into **MCU CAL REF CLK REG**.
   Default values are for 40mhz clock with settling time of 64 clocks of 32khz clock = 64 * 31.25us = 2ms.
2. In **MCU CAL START REG** program alpha_rc, no_of_rc_clocks based on the requirement. **rc_xtal_mux_sel** = 1 and also wirte to **start_calib_rc** = 1;
3. If *periodic* calibration of RC is required, enable bit 21, of **MCU CAL START REG**, **periodic_RC_calib_en** , and select **rc_trigger_time_sel** bits based on the rate at with calibration needs to happen.
4. If *temperature* based calibration has to happen along with periodic calibration, also enable, 23:16 bits of **MCU CAL TEMP PROG REG** based on the description. max_temp_change is the value increase in the temperature after which RC has to be recalibrated.
5. Look for 24th bit in **MCU CAL RC TIMEPERIOD READ** if 1, RC timeperiod is valid, Already the timeperiod calibration has occurred once.Inorder to read the timeperiod of RC, read 23:0 bits of same register.

16.26.6.3 R0 Time Period Calibration

1. A valid RC timeperiod non zero value must be available to start RO calibration. Also if the reference clock is 32k xtal clock The 32KHz xtal clock must be enabled manually in IPMU. For rc ref clock - rc_xtal_mux_sel = 0, for xtal 32khz ref clock rc_xtal_mux_sel = 1.
2. In **MCU CAL START REG** program alpha_ro, no_of_ro_clocks, rc_settle_time based on the requirement. wirte to **start_calib_ro** = 1;
3. If *periodic* calibration of RO is required, enable in register **MCU CAL START REG**, **periodic_RO_calib_en** , and select **ro_trigger_time_sel** bits based on the rate at with calibration needs to happen.
4. Look for 24th bit in **MCU CAL RO TIMEPERIOD READ** if 1, RO timeperiod is valid, Already the timeperiod calibration has occurred once. Inorder to read the timeperiod of RO, read 23:0 bits of same register.

16.27 System RTC

16.27.1 Overview

The SYSRTC (System Real Time Clock) is a highly configurable RTC capable of serving multiple cores. It contains up to 2 groups, where the number of compare- and capture-channels within each group is parameterized individually. Each group has it's own interrupt- and configuration-registers. The main idea is to save power by letting all groups share a single counter.

16.27.2 Features

- 32-bit counter
- 32 kHz / 1kHz intended operation
- Low energy mode and wake-up
- Up to 2 groups
- Each group has either 1-2 compare channels
- Each grou can have 1 or no capture channel
- Optional debug halting
- Software Reset

16.27.3 Functional Description

16.27.3.1 Counter

The counter is shared between all groups. It can be started/stopped by writing to START/STOP fields in CMD register. RUNNING field in STATUS register indicates if counter is running or not. By default, counter will halt when core is halted during debug. RUNNING is not affected by halting. If DEBUGRUN in CFG register is set, counter will not halt when core is halted. The count value can be accessed via CNT register even when it is not running. When CNT is written, count value will be updated on the next clock edge. When the counter reaches the maximum value of 0xFFFFFFFF, it will overflow to 0x00000000 on the next clock edge. All OVFIF interrupt flags are set when this happens.

16.27.3.2 Compare Channel

When count value matches CMPnVALUE, and CMPnEN in CTRL register is set, the CMPnIF interrupt flag is set. At the same time, PRS output is updated according to CMPnCMOA value in CTRL register. CTRL and CMPnVALUE can be written at any time and will take effect immediately.

16.27.3.3 Capture Channel

When CAPnEN in CTRL register is set, the count value will be captured into CAPnVALUE based on PRS input edges. CAPnEDGE in CTRL register controls which edges that will result in capture. When count value is captured, CAPnIF interrupt flag is set.

A capture event is generated whenever RUNNING status set, the corresponding GRP_CTRL_CAPEN register setting set and the desired PRS input edge occurs according to the GRP_CTRL_CAPEDGE register setting. This event is followed by GRP_IF_CAPIF being set after up to 3 cycles. At the same time when the corresponding flag is set the GRP_CAPVALUE register captures the current counter value. Note that PRS input edges should not occur more frequently than once in 3 cycles. If the counter is being started/stopped or GRP_CTRL_CAPEN/GRP_CTRL_CAPEDGE being reprogrammed close to the PRS input edge, please account for the race condition.

16.27.3.4 Secure Time

One of the groups can be parameterized as a ROOT-group. This group will contain Failure Detection and Tamper Detection to allow ROOT to manage Secure Time. When FAILDETEN in FAILDET register is set, Failure Detection compares the low frequency (~32 kHz) peripheral clock with an ultra low frequency reference clock (~1 kHz). The number of peripheral clock cycles per reference clock cycle is counted and compared to maximum and minimum limits. The limits are configured in FAILCNTHI and FAILCNTLO in FAILDET register. If the counted value is outside the limits, FAILDETIF interrupt flag is set. Tamper Detection is continuously monitoring the counter. TAM-PERIF interrupt flag is set whenever;

- CNT is written
- Module is disabled (EN=0)
- Counter is stopped
- Counter is halted
- Soft Reset is asserted

All registers within the ROOT-group and the FAILDET registers can be accessed by ROOT only. If registers are accessed by any other primary, register reads will return 0 and register writes will be ignored.

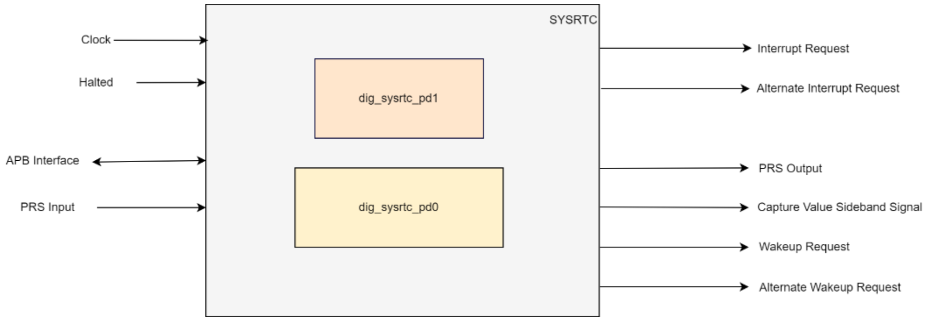
16.27.3.5 Alternate Interrupts/Wakeups

The number of interrupt flags in a group can be doubled based on parameter setting. The name of the alternate interrupt flags will have an ALT prefix. The alternate interrupt flags share the same hardware events as the regular interrupts, but can be set and cleared individually by software. Note that a ROOT-group is not allowed to have alternate interrupts, hence there will never be any ALTFAILDETIF nor ALTTAMPERIF interrupt flags. The alternate interrupt flags are used to form an alternate interrupt line in addition to the regular interrupt line. Similarly, the alternate interrupt flags are used to form an alternate wake-up signal in addition to the regular wake-up signal.

16.27.3.6 Wakeup

All interrupt flags, except for (ALT) CAPnIF, are used for wake-up generation. When at least one of the enabled interrupts are set, wakeup is requested.

16.27.3.7 Block Diagram



16.27.3.8 Module Parameters

Parameter Name	Parameter Description	Encoding	Restriction	Parameter Values
ROOTDIS	ROOT group disable	Set to 1 if there are no ROOT groups	This bit must be set when GROUP_ROOTDIS are all 1	1
GROUP_DIS[7:0]	Group Disable	If bit N is set, group N will not be implemented		{1,1,1,1,1,1,0,0}
GROUP_CMP1DIS[7:0]	Group compare channel 1 disable	If bit N is set, group N only has one compare channel		{1,1,1,1,1,1,0,0}

Parameter Name	Parameter Description	Encoding	Restriction	Parameter Values
GROUP_CAPDIS[7:0]	Group capture disable	If bit N is set, group N does not have a capture channel		{1,1,1,1,1,1,0,0}
GROUP_ALTIRQDIS[7:0]	Group alternate IRO disable	If bit N is set, group N does not have alternate IRQ implemented	The ROOT group should not have alternate IRQs, meaning that if GROUP_ROOTDIS[N]= 0, GROUP_ALTIRQDIS[N] must be 1	{1,1,1,1,1,1,1,1}
GROUP_ROOTDIS[7:0]	Group ROOT disable	If bit N is set, group N will not be the ROOT group		{1,1,1,1,1,1,1,1}

16.27.3.9 Description

Group2 to Group7 is disabled, so only Group0 and Group1 are enabled. Accordingly the parameters are set as mentioned in the table.

For SYSRTC clock, we are using dynamic muxing between 4 clocks (32khz RC, 32khz RO, 32khz XTAL and 1khz). Support for asynchronous clock is present.

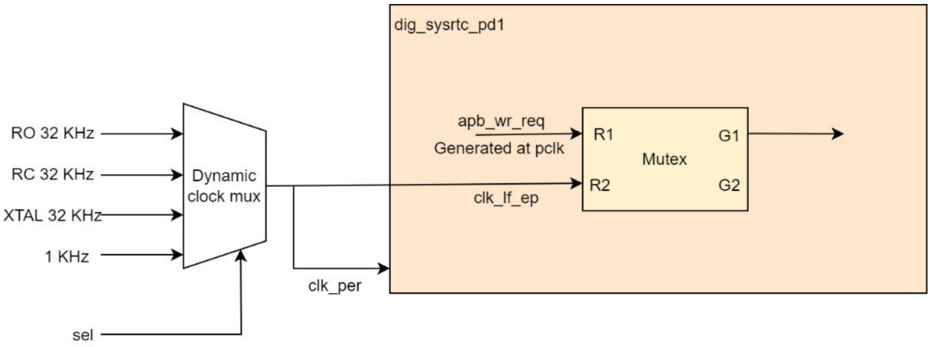
1KHz clock is generated using 6 bit divider, whose input is muxed between 3 clocks (32khz RC, 32khz RO and 32khz XTAL). Out of these 3 clocks, XTAL clock may not be exactly 32KHz, so for that, we can not get exactly 1 KHz clock whereas for other clocks (RC and RO), we will be getting 1 KHz clock using 6 bit divider. By setting division factor as 6'b010000 in the divider, we can generate 1KHz clock as mentioned in Section 6.13 MCU ULP Clock Architecture.

By setting division factor as 6'b010000 in the divider, we can generate 1KHz clock.

Two interrupts are generated for 2 groups , which are ORed internally and 1 interrupt output signal is going out.

16.27.3.9.1 Mutex Requirements

Parameter	Min	Typ	Max	Unit	Comment
Operating voltage	0.9	1.0	1.1	V	1.0V +/- 10%
Operating temp.	-40	25	125		
Clock Frequency fR1		32	100	kHz	
Clock Frequency fR2		10		MHz	Low speed peripherals will only use the 20MHz clock. Since we never expect 2 back-to-back requests: we can conclude that the worst-case scenario for the request access signal is 10MHz.
Settling time ts			10	ns	Also called resolution time
Metastability window tw				fs	Minimize it as much as possible
Transition Current		TBD		mA	Refers to peak transient current
Power down current (R1=R2=0)		TBD		nA	
Cell height		TBD		mm	
Cell Width		TBD		mm	



16.27.3.10 Pin Interface

PD1 Pin List:

Pin Name	Description	Direction
apb	APB Secondary interface	in/out
hvpb	HVPB Primary interface	in/out
clk_lf_ep	Low frequency clock early pulse used for HVQUICKLFWSYNC registers	in
data_lf_ep	Low frequency clock early pulse used for HVLF(R)WSYNC registers	in
halted	Indicates that core is halted during debug	in
irq[7:0]	Interrupt Request	out
altirq[7:0]	Alternate interrupt request	out
sysrtc_pd0_to_pd1	Internal SYSRTC signals from PD0	in
sysrtc_pd1_to_pd0	Internal SYSRTC signals to PD0	out

PD0 Pin List:

Pin Name	Description	Direction
hvpb	HVPB Primary interface	in/out
clkrst	clk_lf interface	in/out
clkrst_faildet	clk_faildet interface	in/out
prs_in[7:0]	PRS input signals used by capture channels	in
prs_out[7:0][1:0]	PRS output signals generated by compare channels	out
cap0value[7:0][31:0]	Capture value sideband signal	out
wu[7:0]	Wakeup Request	out
altwu[7:0]	Alternate wakeup request	out
sysrtc_pd1_to_pd0	Internal SYSRTC signals from PD1	in
sysrtc_pd0_to_pd1	Internal SYSRTC signals to PD1	out

16.27.4 Register Access

Unless otherwise specified, all registers can be written using a bit-masked write in addition to the normal 32-bit write. It allows writing a register in different ways, as described below:

Base Address	Alias	Description
2404_8C00	none	Normal write with no masking applied
2404_8D00	SET	Treat write data as a bit mask and set all bits that are 1 in the mask
2404_8E00	CLR	Treat write data as a bit mask and clear all bits that are 1 in the mask
2404_8F00	TGL	Treat write data as a bit mask and toggle all bits that are 1 in the mask.

Reads always just read the value as normal, regardless of which alias is used.

Note: Bit Masked Register writes is not accessed by MCUSYSRTC_REG1

16.27.5 SYSRTC Register Map

Base Address: 2404_8C00/2404_8D00/2404_8E00/2404_8F00

Register Name	Offset	Description
Section 16.27.5.1.2 SYSRTC_IPVERSION	0x000	IP version
Section 16.27.5.1.3 SYSRTC_EN	0x004	Module Enable Register
Section 16.27.5.1.4 SYSRTC_SWRST	0x008	Software Reset Register
Section 16.27.5.1.5 SYSRTC_CFG	0x00C	Configuration Register
Section 16.27.5.1.6 SYSRTC_CMD	0x010	SYSRTC start or stop Command Register
Section 16.27.5.1.7 SYSRTC_STATUS	0x014	SYSRTC lock or running Status Register
Section 16.27.5.1.8 SYSRTC_CNT	0x018	Counter Value Register
Section 16.27.5.1.9 SYSRTC_SYNCBUSYSection	0x01C	Synchronization Busy Register
Section 16.27.5.1.10 SYSRTC_LOCK	0x020	Configuration Lock Register
Section 16.27.5.1.11 SYSRTC_GRP0_IF	0x040	Group-0 interrupt flag register
Section 16.27.5.1.12 SYSRTC_GRP0_IE	0x044	Group-0 Interrupt Enables
Section 16.27.5.1.13 SYSRTC_GRP0_CTRL	0x048	Group-0 Control Register
Section 16.27.5.1.14 SYSRTC_GRP0_CMP0VALUE	0x04C	Group-0 Compare 0 Value Register
Section 16.27.5.1.15 SYSRTC_GRP0_CMP1VALUE	0x050	Group-0 Compare 1 Value Register
Section 16.27.5.1.16 SYSRTC_GRP0_CAP0VALUE	0x054	Group-0 Capture 0 Value Register
Section 16.27.5.1.17 SYSRTC_GRP0_SYNCBUSY	0x058	Group-0 Synchronization Busy Register
Section 16.27.5.1.18 SYSRTC_GRP1_IF	0x060	Group-1 Interrupt Flags
Section 16.27.5.1.19 SYSRTC_GRP1_IE	0x064	Group-1 Interrupt Enables
Section 16.27.5.1.20 SYSRTC_GRP1_CTRL	0x068	Group-1 Control Register
Section 16.27.5.1.21 SYSRTC_GRP1_CMP0VALUE	0x06C	Group-1 Compare 0 Value Register
Section 16.27.5.1.22 SYSRTC_GRP1_CMP1VALUE	0x070	Group-1 Compare 1 Value Register
Section 16.27.5.1.23 SYSRTC_GRP1_CAP0VALUE	0x074	Group-1 Capture 0 Value Register
Section 16.27.5.1.24 SYSRTC_GRP1_SYNCBUSY	0x078	Group-1 Synchronization Busy Register

Base Address: 2404_8C00

Offset	Name	Type	Label
0x3FC	Section 16.27.5.1.1 MCUSYSRTC_REG1	RW	Input/Output Register (Always accessed with base address 0x2404_8C00)

Note: For GRPn_IF registers, writes will be done throught SET/CLR set of registers.

16.27.5.1 Register Description

16.27.5.1.1 MCUSYSRTC_REG1

Table 16.701. MCUSYSRTC_REG1 Register

Offset address: 0x3FC				
Bit	Access	Function	Default Value	Description
31:7	-	Reserved	-	-
6:3	R	prs_out		Output from SYSRTC module
2:1	W	prs_in	0	Input to SYSRTC module
0	R/W	prs_select	0	'0': Selects PRS input and output from GPIOs '1': Selects PRS input and output from register bits

16.27.5.1.2 SYSRTC_IPVERSION

Offset address: 0x000				
Bit	Access	Function	Default Value	Description
31:0	R	IP_VERSION	0x1	IP Version

16.27.5.1.3 SYSRTC_EN

Offset address: 0x004				
Bit	Access	Function	Default Value	Description
31:2	-	Reserved	-	-
1	R	DISABLING	0x0	Disablement busy status. Set when EN cleared and cleared when the peripheral core reset is finished
0	R/W	ENABLE	0x0	SYSRTC Enable. Enable the SYSRTC by requesting Clock

16.27.5.1.4 SYSRTC_SWRST

Offset address: 0x008				
Bit	Access	Function	Default Value	Description
31:2	-	Reserved	-	-
1	R	RESETTING	0x0	Software reset busy status
0	W	SWRST	0x0	Software reset command

16.27.5.1.5 SYSRTC_CFG

Offset address: **0x00C**				
Bit	Access	Function	Default Value	Description
31:1	-	Reserved	-	-
0	R/W	DEBUG_RUN	0x0	1 - Debug Mode run Enable 0- Debug Mode run Disable

16.27.5.1.6 SYSRTC_CMD

Offset address: **0x010**				
Bit	Access	Function	Default Value	Description
31:2	-	Reserved	-	-
1	W	STOP	0x0	Stop SYSRTC
0	W	START	0x0	Start SYSRTC

16.27.5.1.7 SYSRTC_STATUS

Offset address: **0x014**				
Bit	Access	Function	Default Value	Description
31:2	-	Reserved	-	-
1	R	LOCK_STATUS	0x0	Lock status
0	R	RUNNING	0x0	SYSRTC running status

16.27.5.1.8 SYSRTC_CNT

Offset address: **0x018**				
Bit	Access	Function	Default Value	Description
31:0	R/W	CNT	0x0	Counter value

16.27.5.1.9 SYSRTC_SYNCBUSY

Offset address: **0x01C**				
Bit	Access	Function	Default Value	Description
31:3	-	Reserved	-	
2	R	CNT	0x0	Sync busy for CNT bitfield
1	R	STOP	0x0	Sync busy for STOP bitfield
0	R	START	0x0	Sync busy for START bitfield

16.27.5.1.10 SYSRTC_LOCK

Offset address: 0x020				
Bit	Access	Function	Default Value	Description
31:16	-	Reserved	-	
15:0	W	LOCK_KEY	0x0	Configuration Lock Key

16.27.5.1.11 SYSRTC_GRP0_IF

Offset address: 0x040				
Bit	Access	Function	Default Value	Description
31:4	-	Reserved	-	
3	R/W	CAP_0_IF	0x0	Capture 0 Interrupt Flag
2	R/W	CMP_1_IF	0x0	Compare 1 Interrupt Flag
1	R/W	CMP_0_IF	0x0	Compare 0 Interrupt Flag
0	R/W	OVFI_IF	0x0	Overflow Interrupt Flag

16.27.5.1.12 SYSRTC_GRP0_IE

Offset address: 0x044				
Bit	Access	Function	Default Value	Description
31:4	-	Reserved	-	
3	R/W	CAP_0_EN	0x0	Capture 0 Interrupt Enable
2	R/W	CMP_1_EN	0x0	Compare 1 Interrupt Enable
1	R/W	CMP_0_EN	0x0	Compare 0 Interrupt Enable
0	R/W	OVFI_EN	0x0	Overflow Interrupt Enable

16.27.5.1.13 SYSRTC_GRP0_CTRL

Offset address: 0x048				
Bit	Access	Function	Default Value	Description
31:11	-	Reserved	-	
10:9	R/W	CAP_0_EDGE	0x0	Capture 0 Edge Select
8:6	R/W	CMP_1_CM_OA	0x0	Compare 1 Compare Match Output Action
5:3	R/W	CMP_0_CM_OA	0x0	Compare 0 Compare Match Output Action
2	R/W	CAP_0_EN	0x0	Capture 0 Enable
1	R/W	CMP_1_EN	0x0	Compare 1 Enable
0	R/W	CMP_0_EN	0x0	Compare 0 Enable

16.27.5.1.14 SYSRTC_GRP0_CMP0VALUE

Offset address: 0x04C				
Bit	Access	Function	Default Value	Description
31:0	R/W	CMP_0_VALUE	0x0	Compare 0 Value

16.27.5.1.15 SYSRTC_GRP0_CMP1VALUE

Offset address: 0x050				
Bit	Access	Function	Default Value	Description
31:0	R/W	CMP_1_VALUE	0x0	Compare 1 Value

16.27.5.1.16 SYSRTC_GRP0_CAP0VALUE

Offset address: 0x054				
Bit	Access	Function	Default Value	Description
31:0	R	CAP_0_VALUE	0x0	Capture 0 Value

16.27.5.1.17 SYSRTC_GRP0_SYNCBUSY

Offset address: 0x058				
Bit	Access	Function	Default Value	Description
31:3	-	Reserved	-	-
2	R	CMP_1_VALUE	0x0	Sync busy for CMP 1 VALUE register
1	R	CMP_0_VALUE	0x0	Sync busy for CMP 0 VALUE register
0	R	CTRL	0x0	Sync busy for CTRL register

16.27.5.1.18 SYSRTC_GRP1_IF

Offset address: 0x060				
Bit	Access	Function	Default Value	Description
31:8	-	Reserved	-	-
7	R/W	ALTCAP_0_IF	0x0	Alternate Capture 0 interrupt Flag
6	R/W	ALTCMP_1_IF	0x0	Alternate Compare 1 interrupt Flag
5	R/W	ALTCMP_0_IF	0x0	Alternate Compare 0 interrupt Flag
4	R/W	ALTOVF_IF	0x0	Alternate Overflow Interrupt Flag
3	R/W	CAP_0_IF	0x0	Capture 0 Interrupt Flag
2	R/W	CMP_1_IF	0x0	Compare 1 Interrupt Flag
1	R/W	CMP_0_IF	0x0	Compare 0 Interrupt Flag

Offset address: **0x060**				
0	R/W	OVF_IF	0x0	Overflow Interrupt Flag

16.27.5.1.19 SYSRTC_GRP1_IE

Offset address: **0x064**				
Bit	Access	Function	Default Value	Description
31:8	-	Reserved	-	-
7	R/W	ALTCAP_0_IF	0x0	Alternate Capture 0 interrupt enable
6	R/W	ALTCMP_1_IF	0x0	Alternate Compare 1 interrupt enable
5	R/W	ALTCMP_0_IF	0x0	Alternate Compare 0 interrupt enable
4	R/W	ALTOVF_IF	0x0	Alternate Overflow Interrupt flag
3	R/W	CAP_0_IF	0x0	Capture 0 Interrupt enable
2	R/W	CMP_1_IF	0x0	Compare 1 Interrupt enable
1	R/W	CMP_0_IF	0x0	Compare 0 Interrupt enable
0	R/W	OVF_IF	0x0	Overflow Interrupt Flag

16.27.5.1.20 SYSRTC_GRP1_CTRL

Offset address: **0x068**				
Bit	Access	Function	Default Value	Description
31:11	-	Reserved	-	-
10:9	R/W	CAP_0_EDGE	0x0	Capture 0 Edge Select
8:6	R/W	CMP_1_CM_OA	0x0	Compare 1 Compare Match Output Action
5:3	R/W	CMP_0_CM_OA	0x0	Compare 0 Compare Match Output Action
2	R/W	CAP_0_EN	0x0	Capture 0 Enable
1	R/W	CMP_1_EN	0x0	Compare 1 Enable
0	R/W	CMP_0_EN	0x0	Compare 0 Enable

16.27.5.1.21 SYSRTC_GRP1_CMP0VALUE

Offset address: **0x06C**				
Bit	Access	Function	Default Value	Description
31:0	R/W	CMP_0_VALUE	0x0	Compare 0 Value

16.27.5.1.22 SYSRTC_GRP1_CMP1VALUE

Offset address: 0x070				
Bit	Access	Function	Default Value	Description
31:0	R/W	CMP_1_VALUE	0x0	Compare 1 Value

16.27.5.1.23 SYSRTC_GRP1_CAP0VALUE

Offset address: 0x074				
Bit	Access	Function	Default Value	Description
31:0	R	CAP_0_VALUE	0x0	Capture 0 Value

16.27.5.1.24 SYSRTC_GRP1_SYNCBUSY

Offset address: 0x078				
Bit	Access	Function	Default Value	Description
31:3	-	Reserved	-	-
2	R	CMP_1_VALUE	0x0	Sync busy for CMP_1_VALUE register
1	R	CMP_0_VALUE	0x0	Sync busy for CMP_0_VALUE register
0	R	CTRL	0x0	Sync busy for CTRL register

16.28 WatchDog Timer (WDT)

16.28.1 General Description

The WatchDog Timer is used generate an interrupt on timeout and a reset in case of system failure which can be caused by an external event like ESD pulse or due to a software failure. Also the Interrupt can be used as a Wakeup source for transitioning from SLEEP/ STANDBY to ACTIVE states.

16.28.2 Features

- Independent watchdog timer.
- Interrupt is generated before the system reset is applied which can be used as a wakeup source.
- Configurable low frequency clock. The generation of this clock is described in Section 6.14 MCU ULP VBAT Clock Architecture.
  - Low-Frequency RC clock (RC_32KHZ_CLK).
  - Low-Frequency RO clock (RO_32KHZ_CLK).
  - External 32KHz XTAL clock (XTAL_32KHZ_CLK)
- Configurable timeout period.
- Able to operate when CPU is in SLEEP state (as defined in Section 9. Power Architecture) during power-save applications*
- APB Interface for accesses from CPU.
- Individually controllable power domain for low-power applications.

16.28.3 Functional Description

WatchDog Timer will generate an Interrupt and Reset at different time instants as configured.

The Processor needs to restart the Timer upon WDT Interrupt if it the timer is not intended to hit the reset threshold.

The Timer will be inactive upon reaching SLEEP state (as defined in Section 9. Power Architecture) and resets itself upon wakeup to ACTIVE state. However, the Timer can be configured to be running during sleep to avoid system failure during SLEEP state*. Also the WDT Interrupt can be used for switching from SLEEP to ACTIVE state.

16.28.3.1 State Machine

The figure below depicts the functional flow for the WatchDog Timer

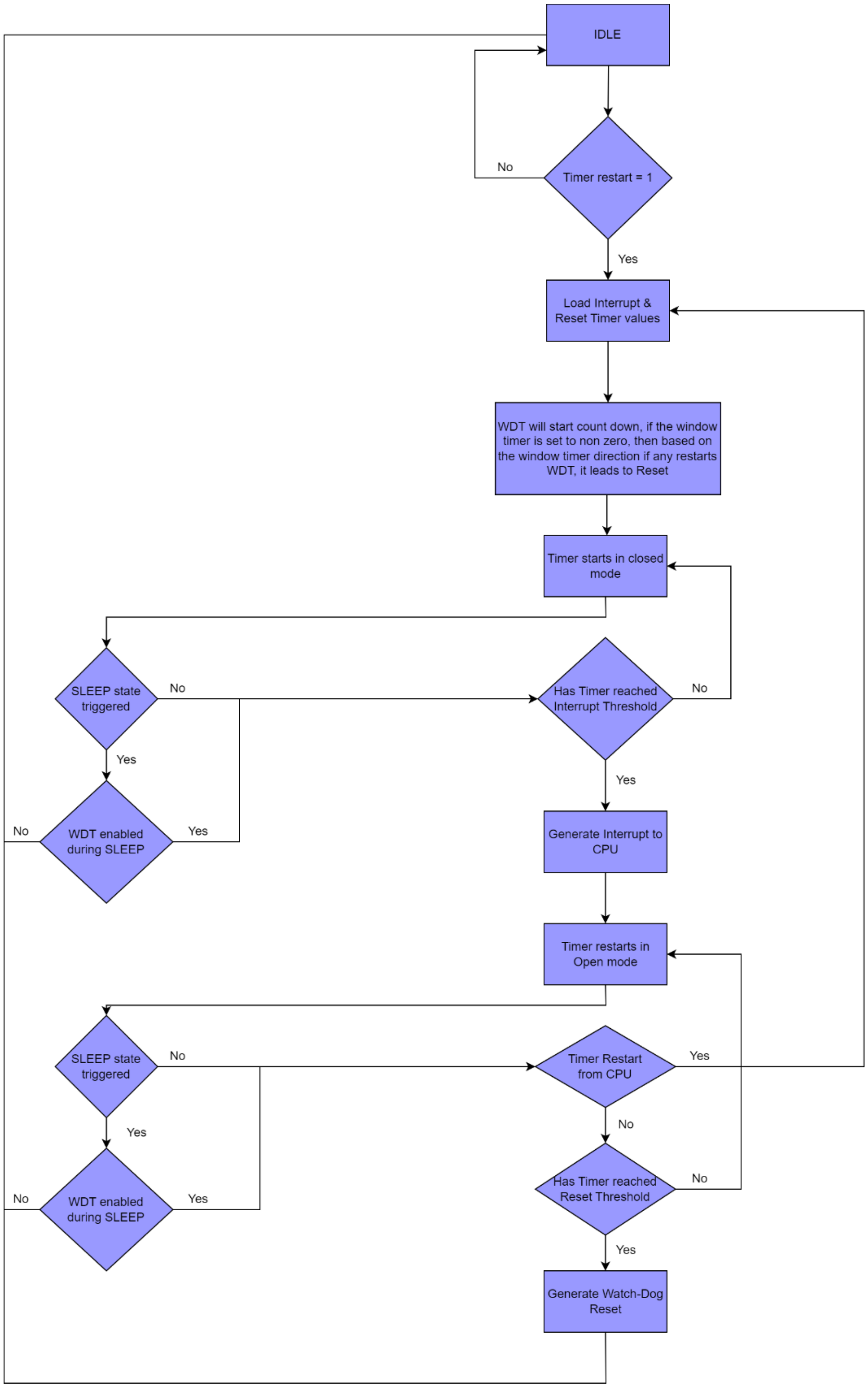

IDLE
Timer restart = 1
No
Yes
Load Interrupt & Reset Timer values
WDT will start count down, if the window timer is set to non zero, then based on the window timer direction if any restarts WDT, it leads to Reset
Timer starts in closed mode
SLEEP state triggered
No
Has Timer reached Interrupt Threshold
No
Yes
WDT enabled during SLEEP
No
Yes
Yes
Generate Interrupt to CPU
Timer restarts in Open mode
SLEEP state triggered
No
Timer Restart from CPU
Yes
Yes
No
WDT enabled during SLEEP
No
Yes
Has Timer reached Reset Threshold
No
Yes
Generate Watch-Dog Reset

Figure 16.49. WatchDog Timer Functional Flow Diagram

Note : Window feature not supported

16.28.3.2 Programming Sequence

The steps below shows the sequence of programming to be done for the WatchDog functionality

1. Power-Up the WatchDog Domain.
   a. Refer to Section 9. Power Architecture for programming details
2. Enable the WatchDog Timer
   a. This can be configured through "wwd_timer_en" in Register 16.28.5.4 MCU_WWD_TIMER_ENABLE.
3. Load the Timer Values for Interrupt and Reset Generation
   a. system reset duration can be configured through "wwd_system_reset_timer" in Register 16.28.5.2 MCU_WWD_SYSTEM_RESET_TIMER.
   b. Interrupt duration can be configured through "wwd_interrupt_timer" in Register 16.28.5.1 MCU_WWD_INTERRUPT_TIMER.
4. Start the Timer. The same configuration needs to be done for restarting the Timer in case of interrupt.
   a. This can be configured through "wwd_timer_rstart" in Register 16.28.5.4 MCU_WWD_TIMER_ENABLE.
5. The Timer can be configured to be running during SLEEP state
   a. Refer FSM_CTRL_POWER_DOMAINS in Section 9. Power Architecture.
6. The WDT key needs to be configured to readback the parameters programmed to WDT
   a. This can be configured through "wwd_key_enable" in Register 16.28.5.5 MCU_WWD_KEY_ENABLE.

16.28.4 Register Summary

Base Address: 0x2404_8300

Table 16.702. Register Summary

Register Name	Offset	Description
Section 16.28.5.1 MCU_WWD_INTERRUPT_TIMER	0x00	Interrupt Configuration Register
Section 16.28.5.2 MCU_WWD_SYSTEM_RESET_TIMER	0x04	Reset Configuration Register
Section 16.28.5.3 MCU_WWD_WINDOW_TIMER	0x08	Window timer Register
Section 16.28.5.4 MCU_WWD_TIMER_ENABLE	0x10	WDT Enable Register
Section 16.28.5.5 MCU_WWD_KEY_ENABLE	0x18	WDT Key Register

16.28.5 Register Description

16.28.5.1 MCU_WWD_INTERRUPT_TIMER

Table 16.703. Watch-Dog Interrupt Timer Register Description

Bit	Access	Function	Reset Value	Description
31:8	-	Reserved	-	It is recommended to write these bit to 0
7:0	RW	wwd_interrupt_timer	0	Indicates the time duration for generation of System Reset This is specified in terms of number of clock(LOW-FREQ clock used for WDT) pulses Number of clock pulses = 2^(wwd_interrupt_timer)

16.28.5.2 MCU_WWD_SYSTEM_RESET_TIMER

Table 16.704. Watch-Dog System Reset Timer Register Description

Bit	Access	Function	Reset Value	Description
31:8	-	Reserved	-	It is recommended to write these bit to 0
7:0	RW	wwd_system_reset_timer	0	Indicates the time duration for generation of System Reset This is specified in terms of number of clock(LOW-FREQ clock used for WDT) pulses Number of clock pulses = 2^(wwd_system_reset_timer)

16.28.5.3 MCU_WWD_WINDOW_TIMER

Table 16.705. WWD_WINDOW_TIMER Register Description

Bit	Access	Function	Reset Value	Description
[31:4]	--	Reserved	0	Reserved
[3:0]	R/W	window_timer	0	watchdog window timer Total duration = 2^window_timer FSM clocks

16.28.5.4 MCU_WWD_TIMER_ENABLE

Table 16.706. Watch-Dog Mode Enable Register Description

Bit	Access	Function	Reset Value	Description
31:24	-	Reserved	-	It is recommended to write these bit to 0
23:16	W	wwd_timer_en	0	0xAA – Enables the WatchDog Timer 0xF0 – Disables the WatchDog Timer
15:1	-	Reserved	-	It is recommended to write these bit to 0
0	W	wwd_timer_rstart	0	Writing 1 to this restarts the WatchDog Timer Writing 0 this has no effect

16.28.5.5 MCU_WWD_KEY_ENABLE

Table 16.707. Watch-Dog Key Enable Register Description

Bit	Access	Function	Reset Value	Description
31:0	W	wwd_key_enable	0x877F38E9	Specifies the key to read back the WDT Registers described above. Writing 0x877F38E9 to this enables Read Access Writing 0x0AAAAAAA to this disables Read Access

16.29 Analog Comparators

16.29.1 General Description

Analog comparators peripheral consists of two analog comparators, a reference buffer, a scaler and a resistor bank.

The comparator compares analog inputs p and n to produce a digital output, cmp_out according to:

p > n, cmp_out = 1

p < n, cmp_out = 0

16.29.2 Features

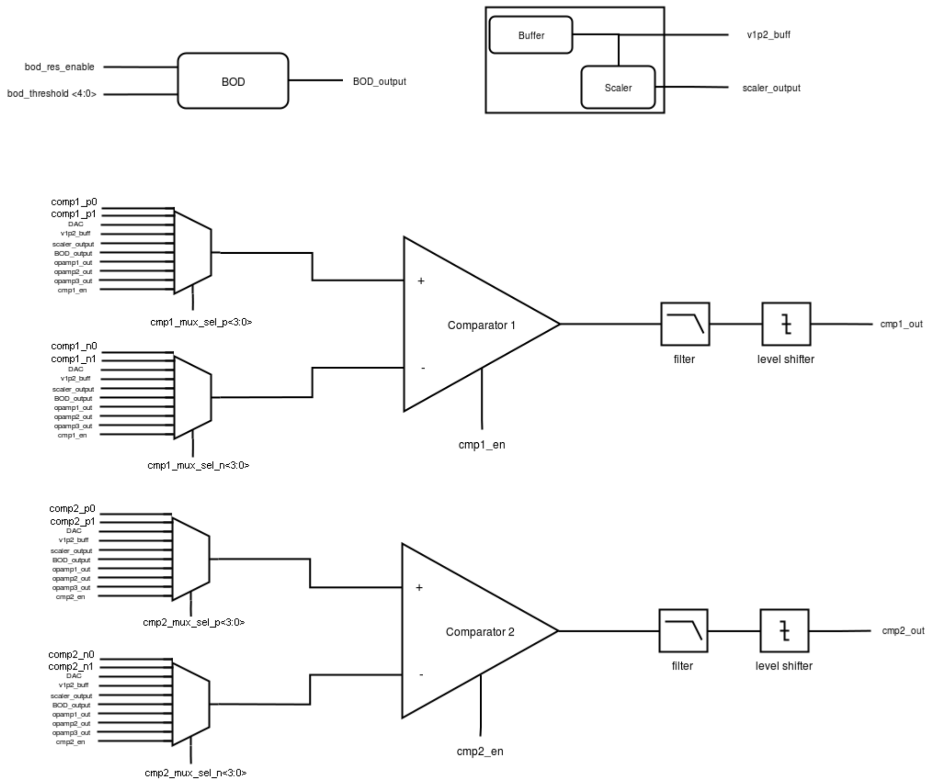
Both comparators can take inputs from GPIOs.

There are 9 different inputs for each pin of comparator, and 2 of the 9 are external pin inputs (GPIOs).
The following cases of comparison are possible

1. Compare external pin inputs
2. Compare external pin input to internal voltages.
3. Compare internal voltages.

The inputs of 2 comparators can be programmed independently. The reference buffer, scaler and resistor bank are shared between the two comparators and can be enabled only when atleast one of the comparators is enabled.

16.29.3 Block Diagram



16.29.4 Functional Description

- Enable reference buffer and resistor bank if required.
- Select the inputs of comparator using input mux selects and then enable the comparator. By default comparators are disabled.

16.29.5 Input Selection

Each comparator has Vinp mux to select "p," and Vinn mux to select "n" input of comparator.

16.29.5.1 cmp_p mux

cmp_mux_sel_p<3:0>	0	1	2	3	4	5	6	7	8
comparator 1	comp1_p0	comp1_p1	DAC	reference buffer out	reference scaler out	resistor bank out	opamp1_out	opamp2_out	opamp3_out
comparator 2	comp2_p0	comp2_p1	DAC	reference buffer out	reference scaler out	resistor bank out	opamp1_out	opamp2_out	opamp3_out

16.29.5.2 cmp_n mux

cmp_mux_sel_n<3:0>	0	1	2	3	4	5	6	7	8
comparator 1	comp1_n0	comp1_n1	DAC	reference buffer out	reference scaler out	resistor bank out	opamp1_out	opamp2_out	opamp3_out
comparator 2	comp2_n0	comp2_n1	DAC	reference buffer out	reference scaler out	resistor bank out	opamp1_out	opamp2_out	opamp3_out

16.29.6 Voltage Scaler

The reference buffer uses 1.2V from ULP_BG as its reference. And the scaler takes this buffered 1.2V as its input.

Scaler is configured using REFBUF_VOLT_SEL<3:0>. The output of scaler for different scale factors are in the table below

Bandgap_scale_factor	Scaler output	Units
0	0.1	V
1	0.2	V
2	0.3	V
3	0.4	V
4	0.5	V
5	0.6	V
6	0.7	V
7	0.8	V
8	0.9	V
9	1	V
10	1.1	V

16.29.7 Resistor Bank (BOD)

Resbank output = VBATT*(200/(300+bod_threshold*8.33))

bod threshold	Referred voltage	Unit
0	1.8	V
1	1.85	V
2	1.9	V
3	1.95	V
4	2	V
5	2.05	V
6	2.1	V
7	2.15	V
8	2.2	V
9	2.25	V
10	2.3	V
11	2.35	V
12	2.4	V
13	2.45	V
14	2.5	V
15	2.55	V
16	2.6	V
17	2.65	V
18	2.7	V
19	2.75	V
20	2.8	V
21	2.85	V
22	2.9	V
23	2.95	V
24	3	V
25	3.05	V
26	3.1	V
27	3.15	V
28	3.2	V
29	3.25	V
30	3.3	V
31	3.35	V

16.29.8 Register Summary

Base Address: 0x24043800

Table 16.708.

Register Name	Offset	Reset Value	Description
Section 16.29.9.1 Comparator 1	0x204	0x 0000 0000	Programs comparators 1&2
Section 16.29.9.2 BOD	0x200	0x 0000 3E00	Programs resistor bank, reference buffer and scaler

16.29.9 Register Description

16.29.9.1 Comparator 1

Bit	Access	Function	Default Value	Description
31:25	R	Reserved	-	Reserved
24	R/W	com_dyn_en	0	Dynamic enable for registers
23:20	R/W	cmp2_mux_sel_n	0	Select for negative input of comparator 2
19:16	R/W	cmp2_mux_sel_p	0	Select for positive input of comparator 2
15:14	R/W	cmp2_hyst	0	Programmability to control hysteresis of comparator2
13	R/W	cmp2_en_filter	0	1 - To enable filter for comparator 2
12	R/W	cmp2_en	0	1 - To enable comparator 2
11:08	R/W	cmp1_mux_sel_n	0	Select for negative input of comparator 1
7:04	R/W	cmp1_mux_sel_p	0	Select for positive input of comparator 1
3:02	R/W	cmp1_hyst	0	Programmability to control hysteresis of comparator1
1	R/W	cmp1_en_filter	0	1 - To enable filter for comparator 1
0	R/W	cmp1_en	0	1 - To enable comparator 1

16.29.9.2 BOD

Table 16.709. Register Description

Bit	Access	Function	Default Value	Description
31:14	R	Reserved	-	Reserved
13:9	R/W	BOD_THRESHOLD	0	Programmability for resistor bank
8	R/W	BOD_RES_ENABLE	0	Configuration for Resistor Bank 0 – Disable 1 – Enable
7:4	R/W	LVL_SEL	0	Please refer to Voltage Scalar section
3	R/W	REFBUF_EN	0	Reference Buffer Configuration 0 – Disable 1 – Enable

Bit	Access	Function	Default Value	Description
2:0	R	Reserved	-	Reserved

16.30 Analog to Digital Converter

16.30.1 General Description

The Analog to Digital Convertor Peripheral (AUXADC) converts analog input to 12 bit digital output.

16.30.2 Features

The AUXADC can take analog inputs in single ended or differential mode. The output is 12 bit digital which can be given out with or without noise averaging. The Aux VRef can be connected directly to Vbat (Aux LDO bypass mode) or to the Aux LDO output

The AUXADC has five modes of operation:

1)Single ended input with noise averaging,

2)Single ended input without noise averaging,

3)Differential input with noise averaging,

4)Differential input without noise averaging and

5)Shutdown mode.

16.30.3 Functional Description

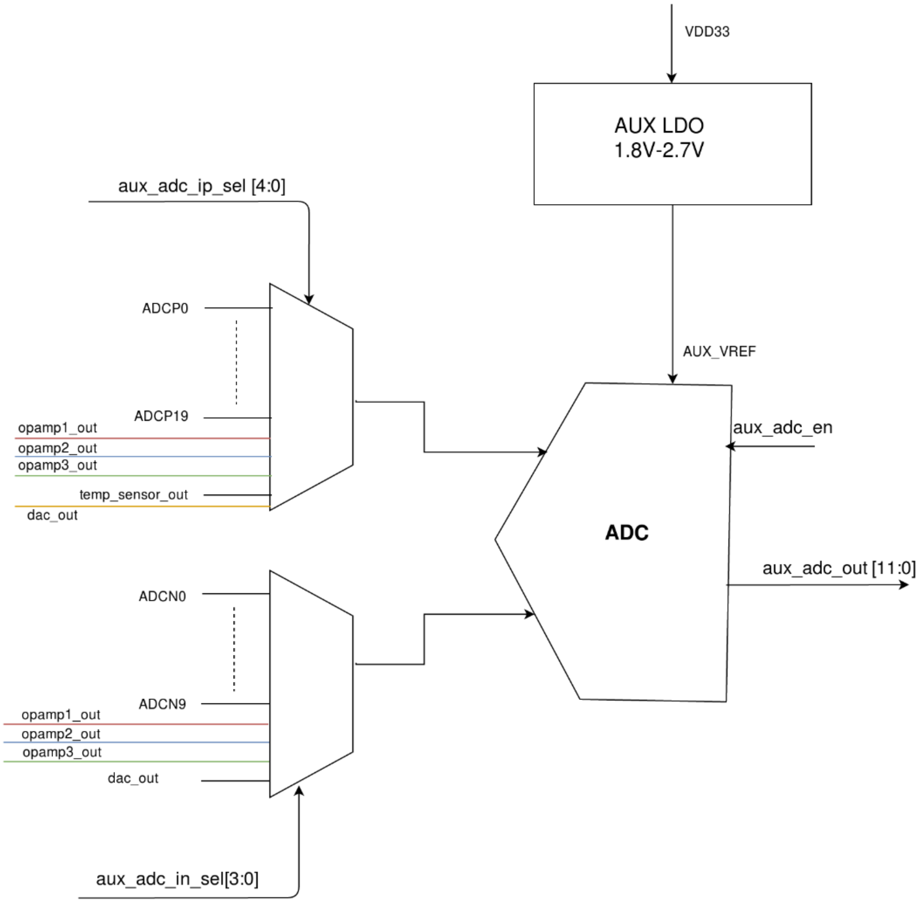


Figure 16.50. Block Diagram of Analog to Digital Converter

1)Single ended input with noise averaging: Here make sure that following conditions met

- reset_n must be high
- enadc_in must be high
- auxadc_pg_en_b must be 1
- auxadc_bypass_iso_gen must be 0
- auxadc_isolatio_enable must be 0
- clk_in must be there
- differential_in should be 0
- bypass_noiseavg must be 0.

Here the analog signal is expected with voltage swing of vdd peak to peak for single ended inputs, supporting common mode voltage of vdd/2. There is no reference supply voltage for the AUXADC .It operates from a supply ranging from 1.8V to 3.6V. The sampling frequency here can be less than or equal to 10MHz for 3.3V operationo and 5MHz for 1.8V.

The outputs will come with noise averaged at 2nd posedge of clock with combinational delay in noise averaging. This data is registered at 3rd posedge of output clock.

2)Single ended input without noise averaging: Here make sure that following conditions met

- reset_n must be high
- enadc_in must be high
- auxadc_pg_en_b must be 1
- auxadc_bypass_iso_gen must be 0
- auxadc_isolatio_enable must be 0
- clk_in must be there
- differential_in should be 0

- bypass_noiseavg must be 1.

Here the analog signal is expected with voltage swing of vdd peak to peak for single ended inputs, supporting common mode voltage of vdd/2. There is no reference supply voltage for the AUXADC .It operates from a supply ranging from 1.8V to 3.6V. The sampling frequency here can be less than or equal to 10MHz for 3.3V and 5MHz for 1.8V.

Here the outputs will come without noise averaging asynchronously during hold phase of the current sample. This data is registered at 2nd posedge of output clock.

3)Differential input with noise averaging: Here make sure that following conditions met

- reset_n must be high
- enadc_in must be high
- auxadc_pg_en_b must be 1
- auxadc_bypass_iso_gen must be 0
- auxadc_isolatio_enable must be 0
- clk_in must be there
- differential_in should be 1
- bypass_noiseavg must be 0.

Here the analog signal is expected with voltage swing of vdd/2 peak to peak for single ended inputs, supporting common mode voltage of vdd/4. There is no reference supply voltage for the AUXADC .It operates from a supply ranging from 1.8V to 3.6V. The sampling frequency here can be less than or equal to 10MHz for 3.3V and 5MHz for 1.8V.

Here the outputs will come noise averaged at 2nd posedge of clock with combinational delay in noise averaging. This data is registered at 3rd posedge of output clock.

4)Differential input without noise averaging: Here make sure that following conditions met

- reset_n must be high
- enadc_in must be high
- auxadc_pg_en_b must be 1
- auxadc_bypass_iso_gen must be 0
- auxadc_isolatio_enable must be 0
- clk_in must be there
- differential_in should be 1
- bypass_noiseavg must be 1.

Here the analog signal is expected with voltage swing of vdd/2 peak to peak for single ended inputs, supporting common mode voltages of vdd/4. There is no reference supply voltage for the AUXADC .It operates from a supply ranging from 1.8V to 3.6V. The sampling frequency here can be less than or equal to 10MHz for 3.3V and 5MHz for 1.8V.

Here the outputs will come without noise averaging asynchronously during hold phase of the current samlple. This data is registered at 2nd posedge of output clock.

5)Shutdown mode: Here make sure that following conditions met

- reset_n must be high
- enadc_in must be low
- auxadc_pg_en_b must be 0
- auxadc_bypass_iso_gen must be 0
- auxadc_isolatio_enable must be 0

16.30.4 ADC Channel Select Mode

Here we need to select channel using the following register

aux_adc_ip_sel<4:0>	inp	aux_adc_in_sel<3:0>	inn
00000	ULP_GPIO_0	0000	ULP_GPIO_1
00001	ULP_GPIO_2	0001	ULP_GPIO_3
00010	ULP_GPIO_4	0010	ULP_GPIO_5
00011	ULP_GPIO_6	0011	ULP_GPIO_11
00100	ULP_GPIO_8	0100	ULP_GPIO_9
00101	ULP_GPIO_10	0101	ULP_GPIO_7
00110	GPIO_25	0110	GPIO_26
00111	GPIO_27	0110	GPIO_28
01000	GPIO_29	1000	GPIO_30
01001		1001	
01010	ULP_GPIO_1	1010	opamp1_out
01011	ULP_GPIO_3	1011	opamp2_out
01100	ULP_GPIO_5	1100	opamp3_out
01101	ULP_GPIO_11	1101	dac_out_1
01110	ULP_GPIO_9		
01111	ULP_GPIO_7		
10000	GPIO_26		
10001	GPIO_28		
10010	GPIO_30		
10011			
10100	opamp1_out		
10101	opamp2_out		
10110	opamp3_out		
10111	temp_sensor_out		
11000	dac_out_1		

Here channel selection can be done by setting [21:17] bits as shown in the following register

Register	SPI/ APB	read/ write	Address	Data	address	Wait time (in us)	set/clear bit information	comments
ADC_SINGLE_CH_CTRL	APB	write	0x24043800+(113*4)	0x0000_0000			clear all bits	an_perif_adc_ip_sel =ch_index[21:17] Give adc positive input at AGPIO[0] (default auxadc_ip_sel is 0 which gives AGPIO[0] data to auxadc_ip) an_perif_adc_ip_sel =20 => auxadc_ip from opamp1 an_perif_adc_ip_sel =21 => auxadc_ip from opamp2 an_perif_adc_ip_sel =22 => auxadc_ip from opamp3 an_perif_adc_ip_sel =23 => auxadc_ip from temp_sensor an_perif_adc_ip_sel =24 => auxadc_ip from auxdacout

16.30.5 ADC Calibration Mode

ADC comparator and caparray needs to be calibrated for the desired performance. Procedure for the calibration is as follows

Pre requisite to enable AUXADC is

1)reset_n must be high

2)enadc_in must be high

3)auxadc_pg_en_b must be 1

4)auxadc_bypass_iso_gen must be 0

5)auxadc_isolatio_enable must be 0

6)clk_in must be there

7)differential_in may be 1 or 0

Note:

This has to be done only for first time bootup

1)Calibration enable

2)Calib word read back

3)Calib write manually

Note:

Here calibration will happen for 32 calib_clk cycles(10 for calib_cmp, 10 for calib_cap_p, 10 for calib_cap_n, 2 clocks in between calibration switching) i.e., 32*4*clk_adc cycles(each calib clk is divided by 4 with clk_in frequency with default values) and also we need to wait for 16 clock cycles for reset_n to sync with calib_clk(32*4+16=144 clk_in cycles).

Register	SPI/ APB	read/ write	Address	Data	32'h ad-dress	Wait time (in us)	set/clear bit information	comments
POWERGATE REG WRITE	SPI	write	0x142	0x00C800	5080C800	100	set bit11, clear bit10,9,8	Note: It is a spi register auxadc_pg_en_b=1, auxadc_bypass_iso_gen=0 auxadc_isolatio_enable=0 auxdac_pg_enb=0
ULP_DYN_CLK_CTRL_DISABLE	APB	write	0x24041400+ 0xA0	0X0006_3800			set bit11,12,13,17, 18	Note: It is APB write Aux mem en=1 Aux clk en=1 aux pclk en=1 udma_clk_enable=1;
MISC_CONFIG_REG	APB	write	0x24041400+ 0x00	0x3007_FFE0			set bit 5 to 18,28,29	
ULP_TA_CLK_GEN_REG	APB	write	0x24041400+ 0x14	0x0000_0001			set bit0 clear all other bits	NWP ref clk selection

Register	SPI/ APB	read/ write	Address	Data	32'h ad-dress	Wait time (in us)	set/clear bit information	comments
ULP_AUX_CLK_GEN	APB	write	0x24041400+ 0x34	0x0000_ 0001			set bit0 clear all other bits	ulp aux clk enable
ADC_SIN-GLE_CH_CTRL	APB	write	0x24043800+ (113*4)	0x0000_ 0000			clear all bits	an_perif_adc_ip_sel =ch_in-dex[21:17] Give adc positive input at AGPIO[0] (default auxadc_ip_sel is 0 which gives AGPIO[0] data to auxadc_ip) an_perif_adc_ip_sel =20 => aux-adc_ip from opamp1 an_perif_adc_ip_sel =21 => aux-adc_ip from opamp2 an_perif_adc_ip_sel =22 => aux-adc_ip from opamp3 an_perif_adc_ip_sel =23 => aux-adc_ip from temp_sensor an_perif_adc_ip_sel =24 => aux-adc_ip from auxdacout
ADC_SIN-GLE_CH_CTRL	APB	write	0x24043800+ (114*4)	0x0000_ 0000			clear all bits	an_perif_adc_in_sel =ch_in-dex[25:22]
ADC_SEQ_CTRL	APB	write	0x24043800+ (115*4)	0x0001_ 0001			set bit0,16	To enable disable per channel ping-pong operation (One-hot coding) for channel 1
ADC_INT_MEM_1	APB	write	0x24043800+ (117*4)	0x24062 040				address 0 -ping memory address for channel 1
ADC_INT_MEM_2	APB	write	0x24043800+ (118*4)	0x00008 020				address 0- number of values 32+memory write enable 1
ADC_INT_MEM_2	APB	write	0x24043800+ (118*4)	0x00000 020				address 0 -number of values 32+memory write enable 0
ADC_INT_MEM_1	APB	write	0x24043800+ (117*4)	0x24062 440				address 1- pong memory address for channel 1
ADC_INT_MEM_2	APB	write	0x24043800+ (118*4)	0x00008 420				address 1- number of values 32+memory write enable 1
ADC_INT_MEM_2	APB	write	0x24043800+ (118*4)	0x00000 420				address 1-number of values 32+memory write enable 0
ADC_CH_OFFSET	APB	write	0x24043800+ (78*4)	0x0000_ 0000			clear all bits	Offset value for this particular channel. This offset value describes the number of clock phases (corresponding to this particular channel) after which this particular channel should be sampled

Register	SPI/ APB	read/ write	Address	Data	32'h address	Wait time (in us)	set/clear bit information	comments
ADC_CH_FREQ_OFFSET	APB	write	0x24043800+ (94*4)	0x0000_ 0001			set bit0 clear all other bits	Sampling frequency value for this particular channel. This sampling frequency value specifies the frequency of phases (corresponding to this particular channel) on which this particular channel is sampled
ADC_CTRL_REG	APB	write	0x24043800+ (1*4)	0x0800_ 0C3D			set bit0,2,3,4,5,10, 11,27 clear all other bits	adcen, adc sw,clear diff
ADC clk div	APB	write	0x24043800+ (3*4)	0x0000_ 0004			set bit2 clear all other bits	These bits control the adc clock division factor clock_freq = input_clock_freq/(2*division factor)
channel enable reg	APB	write	0x24043800+ (119*4)	0x8000_ 0001			set bit31,0	internal DMA enable[31] and channel 1 enable
AUXADC	SPI	write	0x24043800+ (512+8)	0x0000_ 0400			set bit10 clear all bits	aux_adc_en = 1 aux_adc_dyn_en = 0
AUXADCREG0	SPI	write	10'h110	0x040A 00			set bit11	cal_en=1. Here cal_en set to 1, here we have to wait for 144 clk_in cycles(144*0.2us=30us for clk_in 5MHz)
SPAREREG2	SPI	write	10'h1C1				Check/Poll if Bit[0] is 1 then Check/ Poll if Bit[0] is 0	
AUXADCREG2	SPI	read	10'h312				read back data and store it in variable AUXADC_CALDATA	Calib word read back. Read back calibrated values

Register	SPI/ APB	read/ write	Address	Data	32'h address	Wait time (in us)	set/clear bit information	comments
AUXADCREG1	SPI	write	10'h111	enter the stored cal data			set bit0,7 write stored calibrated values as mentioned in beside coloumn.	Calib write manually. Manual calibration register programming(AUXADCREG1[17:13]=AUXADC_CALDATA[15:11], AUXADCREG1[12:8]=AUXADC_CALDATA[10:6], AUXADCREG1[7]=1'b1, AUXADCREG1[6:2]=AUXADC_CALDATA[5:0], AUXADCREG1[0]=1'b1 Here output will be available after 3 clocks of clk_in(3*0.2us=0.6us)
AUXADC	SPI	write	0x24043800+(512+8)	0x0000_0C00			set bit10,11	aux_adc_en = 1 aux_adc_dyn_en = 1
INTR_STATUS_REG	APB	read	0x24043800+(10*4)				poll for bit7	
Repeat STEPS 9, 10, 11	APB	write						
INTR_CLEAR_REG	APB	write	0x24043820	0x00000100			set bit8 clear all bits	Clear first_mem_switch

16.30.6 Channel selection

aux_adc_ip_sel<4:0>	inp	aux_adc_in_sel<3:0>	inn
00000	ULP_GPIO_0	0000	ULP_GPIO_1
00001	ULP_GPIO_2	0001	ULP_GPIO_3
00010	ULP_GPIO_4	0010	ULP_GPIO_5
00011	ULP_GPIO_6	0011	ULP_GPIO_11
00100	ULP_GPIO_8	0100	ULP_GPIO_9
00101	ULP_GPIO_10	0101	ULP_GPIO_7
00110	GPIO_25	0110	GPIO_26
00111	GPIO_27	0110	GPIO_28
01000	GPIO_29	1000	GPIO_30
01001		1001	
01010	ULP_GPIO_1	1010	opamp1_out
01011	ULP_GPIO_3	1011	opamp2_out
01100	ULP_GPIO_5	1100	opamp3_out
01101	ULP_GPIO_11	1101	dac_out_1
01110	ULP_GPIO_9		
01111	ULP_GPIO_7		
10000	GPIO_26		
10001	GPIO_28		
10010	GPIO_30		
10011			
10100	opamp1_out		
10101	opamp2_out		
10110	opamp3_out		
10111	temp_sensor_out		
11000	dac_out_1		

Here channel selection can be done by setting [21:17] bits as shown in the following register

Register	SPI/ APB	read/ write	Address	Data	address	Wait time (in us)	set/clear bit information	comments
ADC_SINGLE_CH_CTRL	APB	write	0x24043800+(113*4)	0x0000_0000			clear all bits	an_perif_adc_ip_sel =ch_index[21:17] Give adc positive input at AGPIO[0] (default auxadc_ip_sel is 0 which gives AGPIO[0] data to auxadc_ip) an_perif_adc_ip_sel =20 => auxadc_ip from opamp1 an_perif_adc_ip_sel =21 => auxadc_ip from opamp2 an_perif_adc_ip_sel =22 => auxadc_ip from opamp3 an_perif_adc_ip_sel =23 => auxadc_ip from temp_sensor an_perif_adc_ip_sel =24 => auxadc_ip from auxdacout
Note: We need to write following registers whenever we are coming back from deep sleep to wakeup mode. This enables manual calibration write								

Register	SPI/ APB	read/ write	Address	Data	32'h address	Wait time (in us)	set/clear bit information	comments
POWERGATE REG WRITE	SPI	write	0x142	0x00C800	5080C800	100	set bit11, clear bit10,9,8	Note: It is a spi register auxadc_pg_en_b=1, auxadc_bypass_iso_gen=0 auxadc_isolatio_enable=0 auxdac_pg_enb=0

Register	SPI/ APB	read/ write	Address	Data	32'h ad-dress	W ait ti me (in us)	set/clear bit information	comments
See 16.30.7.1.2 AUXADCREG1	SPI	write	10'h111	enter the stored cal data			set bit0,7 write stored calibrated values as mentioned in beside colo-umn.	Calib write manually. Manual calibration register programming(AUXADCREG1[17:13]=AUXADCREG2[15:11], AUXADCREG1[12:8]=AUXADCREG2[10:6], AUXADCREG1[7]=1'b1, AUXADCREG1[6:2]=AUXADCREG2[5:0], AUXADCREG1[0]=1'b1) Here output will be available after 3 clocks of clk_in(3*0.2us=0.6us)
ULP_DYN_CLK_CTRL_DISABLE	APB	write	0x24041400+ 0xA0	0X0006_ 3800			set bit11,12,13,17,18	Note: It is APB write Aux mem en=1 Aux clk en=1 aux pclk en=1 udma_clk_enable=1;
MISC_CONFIG_REG	APB	write	0x24041400+ 0x00	0x3007_ FFE0			set bit 5 to 18,28,29	
ULP_TA_CLK_GEN_REG	APB	write	0x24041400+ 0x14	0x0000_ 0001			set bit0 clear all other bits	NWP ref clk selection
ULP_AUX_CLK_GEN	APB	write	0x24041400+ 0x34	0x0000_ 0001			set bit0 clear all other bits	ulp aux clk enable
ADC_SINGLE_CH_CTRL	APB	write	0x24043800+(113*4)	0x0000_ 0000			clear all bits	an_perif_adc_ip_sel =ch_index[21:17] Give adc positive input at AGPIO[0](default auxadc_ip_sel is 0 which gives AGPIO[0] data to auxadc_ip) an_perif_adc_ip_sel =20 => auxadc_ip from opamp1 an_perif_adc_ip_sel =21 => auxadc_ip from opamp2 an_perif_adc_ip_sel =22 => auxadc_ip from opamp3 an_perif_adc_ip_sel =23 => auxadc_ip from temp_sensor an_perif_adc_ip_sel =24 => auxadc_ip from auxdacout

Register	SPI/ APB	read/ write	Address	Data	32'h address	Wait time (in us)	set/clear bit information	comments
ADC_SINGLE_CH_CTRL	APB	write	0x24043800+(114*4)	0x0000_0000			clear all bits	an_perif_adc_in_sel =ch_index[25:22]
16.19.5.114 ADC_SEQ_CTRL	APB	write	0x24043800+(115*4)	0x0001_0001			set bit0,16	To enable disable per channel ping-pong operation (One-hot coding) for channel 1
16.19.5.116 ADC_INT_MEM_1	APB	write	0x24043800+(117*4)	0x24062040				address 0 -ping memory address for channel 1
16.19.5.117 ADC_INT_MEM_2	APB	write	0x24043800+(118*4)	0x00008020				address 0- number of values 32+memory write enable 1
16.19.5.116 ADC_INT_MEM_1	APB	write	0x24043800+(118*4)	0x00000020				address 0 -number of values 32+memory write enable 0
16.19.5.116 ADC_INT_MEM_1	APB	write	0x24043800+(117*4)	0x24062440				address 1- pong memory address for channel 1
16.19.5.117 ADC_INT_MEM_2	APB	write	0x24043800+(118*4)	0x00008420				address 1- number of values 32+memory write enable 1
16.19.5.117 ADC_INT_MEM_2	APB	write	0x24043800+(118*4)	0x00000420				address 1-number of values 32+memory write enable 0
ADC_CH_OFFSET	APB	write	0x24043800+(78*4)	0x0000_0000			clear all bits	Offset value for this particular channel. This offset value describes the number of clock phases (corresponding to this particular channel) after which this particular channel should be sampled
ADC_CH_FREQ_OFFSET	APB	write	0x24043800+(94*4)	0x0000_0001			set bit0 clear all other bits	Sampling frequency value for this particular channel. This sampling frequency value specifies the frequency of phases (corresponding to this particular channel) on which this particular channel is sampled
ADC_CTRL_REG	APB	write	0x24043800+(1*4)	0x0800_0C3D			set bit0,2,3,4,5,10,11,27 clear all other bits	adcen, adc sw,clear diff
ADC clk div	APB	write	0x24043800+(3*4)	0x0000_0004			set bit2 clear all other bits	These bits control the adc clock division factor clock_freq = input_clock_freq/(2*division factor)
channel enable reg	APB	write	0x24043800+(119*4)	0x8000_0001			set bit31,0	internal DMA enable[31] and channel 1 enable
AUXADC	SPI	write	0x24043800+(512+8)	0x0000_0C00			set bit10,11	aux_adc_en = 1 aux_adc_dyn_en = 1
16.19.5.11 INTR_STATUS_REG	APB	read	0x24043800+(10*4)				poll for bit7	

Register	SPI/ APB	read/ write	Address	Data	32'h address	Wait time (in us)	set/clear bit information	comments
Repeat STEPS 10, 11, 12	APB	write						
16.19.5.9 INTR_CLEAR_REG	APB	write	0x24043820	0x00000100			set bit8 clear all bits	Clear first_mem_switch
Note: After writing calibration bits manually, we need to wait for 3 clk_in cycles to get the digital_op<11:0>								

If POC is 0 and all power supplies vddulp, dvdd, avdd, vref are there then AUXADC will give the output bits as

vip=[0 vref]; %input voltage ranging from [0 vref]

vin=((78+8*shift_gain)/(143+8*shift_gain))*avdd; %adjacent value in single ended mode and vin ranging from [0 avdd/2]in differential mode

vdiff=vip-vin;

vref=(64/(89+8*gain))*avdd;

%% Modelled output of adc is given by the following equation

ADCOUT=2047.5+(2047.5*vdiff/vref); % this is in binary offset mode

ADCOUT=(2047.5*vdiff/vref); % this is in 2's complement mode, which we are using here.

16.30.7 Register Summary

Address	Register Name	Register Description
10'h 110	Section16.30.7.1.1 AUXADCREG0	AUXADC config register
10'h 111	Section 16.30.7.1.2 AUXADCREG1	Manual mode settings for comparator and caparray calibration
10'h 112	Section 16.30.7.1.3 AUXADCREG2	Comparator and caparray calib read back

16.30.7.1 Register Description

16.30.7.1.1 AUXADCREG0

Address: 10'h 110				
Bit	Access	Function	Default Value	Description
[21:19]	R/W	reserved	0	Reserved bits
18	R/W	sel_pin_en	1	0 – Enable is given from SPI 1 – Enable is controlled from SOC
17	R/W	bypass_noiseavg	0	1 to bypass noise average 0 to enable noise averaging

Address: 10'h 110				
16	R/W	enadc_r	0	1 to adc enable from register 0 to enable adc from pin
[15:14]	R/W	adc_clk_sel	0	auxadc clock select 00 => clk_out=clk_in 01 => clk_out=clk_in/2 10 => clk_out=clk_in/4 11 => clk_out=clk_in/8
[13:12]	R/W	delay_sel	0	to select sam_en_n on time in single ended mode 00 => sam_en_n_ontime=tdelay of buffer 01 => sam_en_n_ontime=2*tdelay of buffer 10 => sam_en_n_ontime=3*tdelay of buffer 11 => sam_en_n_ontime=4*tdelay of buffer
11	R/W	cal_en	0	0 to disable calibration at bootup 1 to enable calibration at bootup
[10:9]	R/W	clk_div_sel	1	to clock division select of calibration clock 00 => clk_calib=clk_in/2 01 => clk_calib=clk_in/4 10 => clk_calib=clk_in/8 11 => clk_calib=clk_in/16

[8:6]	R/W	calib_clk_delay_sel	0	to select delay between calib_clk_cmp and calib_clk_caparr
				3'd0: calib_clk_caparr = clk_calib_dly[24]; // 5.25n delay
				3'd1: calib_clk_caparr = clk_calib_dly[29]; // 6.3n
				3'd2: calib_clk_caparr = clk_calib_dly[49]; // 10.5n
				3'd3: calib_clk_caparr = clk_calib_dly[57]; // 11.8n
				3'd4: calib_clk_caparr = clk_calib_dly[109]; // 23.5n
				3'd5: calib_clk_caparr = clk_calib_dly[115]; // 24.7n
				3'd6: calib_clk_caparr = clk_calib_dly[224]; // 47n
				3'd7: calib_clk_caparr = clk_calib_dly[234]; // 49.9n

[5:4]	R/W	capcal_avg_duration	0	avg duration between caparr calibration
				case(capcal_avg_duration)
				2'd0: average_length = 4'd0; // 1 -clocks
				2'd1: average_length = 4'd1; // 2 -clocks
				2'd2: average_length = 4'd3; // 4 -clocks
				2'd3: average_length = 4'd7; // 8 -clocks
				endcase

Address: 10'h 110				
[3:2]	R/W	gain	0	to control gain error of auxadc
[1:0]	R/W	shift_gain	0	to control offset error of auxadc

16.30.7.1.2 AUXADCREG1

Address: 10'h 111				
Bit	Access	Function	Default Value	Description
[21:19]	R/W	reserved	0	Reserved bits
18	R/W	manual_clk_select	0	1 to select caparray clock from clk_in 0 to select clock from internal one shot generator
[17:13]	R/W	manual_cap_pbits	0	manual caparray calibration bits
[12:8]	R/W	manual_cap_nbits	0	manual caparray calibration bits
7	R/W	manual_cap_cal_en	0	manual caparray calibration select
[6:2]	R/W	manual_cmp_bits	0	manual comparatot calibration bits
1	--	Reserved	0	Reserved
0	R/W	manual_cmp_cal_en	0	manual comparatot calibration enable

16.30.7.1.3 AUXADCREG2

Address: 10'h 112				
Bit	Access	Function	Default Value	Description
[21:15]	R	reserved	7'd0	reserved
[14:10]	R	caparr_calib_n	0	read n-caparray calibration bits
[9:5]	R	caparr_calib_p	0	read p-caparray calibration bits
[4:0]	R	cmp_calib	0	read cmp calibration bits

16.31 AUX_LDO

16.31.1 General Description

This LDO gives 1.6V to 2.8V(steps of 80mV) Output voltage with a maximum load current of 25mA with a dropout voltage of 300mV. It is external compensated LDO which is stable for load current ranging from 0 to 25mA, with the load cap 1uF.

The LDO reference voltage is 1.2V.

AUX_AVDD is the output of this ldo. Analog Peripherals like opamps, comparators, DAC and ADC will work on AUX_AVDD.

This ldo can be disabled and AUX_AVDD can be connected to an external supply for Ultra-low-power direct battery peripheral operation to save the LDO quiescent current.

16.31.2 Functional Description

- It uses 1.2V from ULP_BG as reference.
- Select the control depending on required output voltage. By default this ldo is enabled.
- Make BYPASS_LDO high to bypass ldo, if required. In bypass mode, output is equal to ULP_IO_VDD.

16.31.3 Output Programming

Vout = vref*(4/3 + ctrl/15) = 1.6 + 0.08*ctrl

LDO_CTRL<3:0>	Vout (V)
0	1.6
1	1.68
2	1.76
3	1.84
4	1.92
5	2
6	2.08
7	2.16
8	2.24
9	2.32
10	2.4
11	2.48
12	2.56
13	2.64
14	2.72
15	2.8

16.31.4 Register Summary

Base Address: 0x24043800

Table 16.710. Register Summary

Register Name	Offset	Reset Value	Description
Section 16.31.5.1 LDO	0x210	0x 0000 0053	Programs LDO CTRL, ENABLE, etc

16.31.5 Register Description

16.31.5.1 LDO

Table 16.711. Register Description

Bit	Access	Function	Default Value	Description	Dynamic controllable
3:0	R/W	LDO_Ctrl	3	Word to set the output voltage	No

Bit	Access	Function	Default Value	Description	Dynamic controllable
4	R/W	LDO_DEFAULT_MODE	1	0 : normal mode	No
				1 : default mode (1.8V)	No
5	R/W	BYPASS_LDO	0	1 - To enable bypass mode	Yes
6	R/W	ENABLE	1	1 - To turn on ldo	Yes
7	R/W	Dyn_en	0	Dynamic Enable	No

16.32 Digital to Analog Converter

16.32.1 General Description

The AUXDAC takes 10 bit inputs and generates corresponding buffered analog voltage output.

16.32.2 Features

The AUXDAC can take 10 bit digital inputs and convert them into analog voltage within range 5*vdd/36 to 31*vdd/36. Vdd can vary from 1.8 volts to 3.6 volts.

The output of the Aux DAC can be multiplexed onto one of two pins.

The AUXDAC has two modes: Operational mode and Shutdown mode.

16.32.3 Functional Description

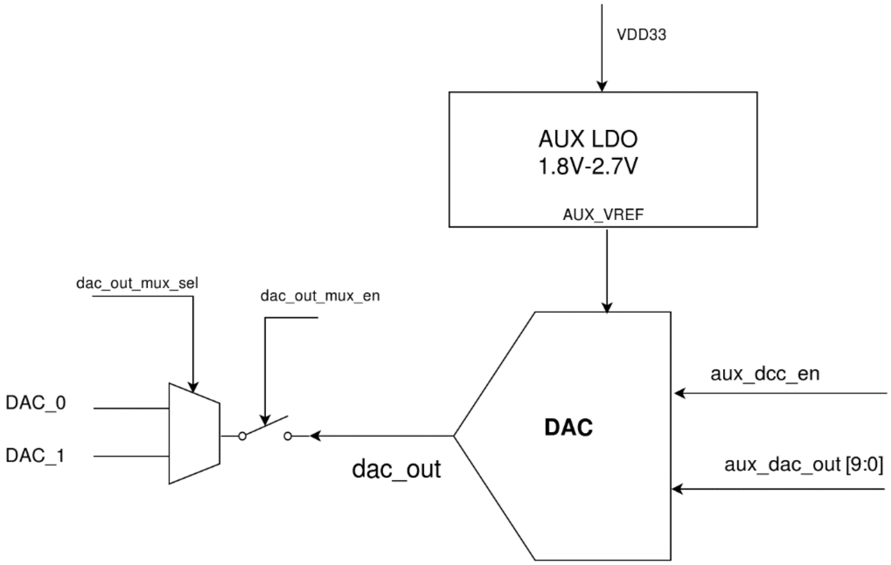


Figure 16.51. Block Diagram of Digital to Analog Converter

Operation mode:

For DAC operation the following conditions should be met:

- vdd, vref, and avdd pins should have stable voltages and reset_n should be high .
- Pins endac_in and auxdac_pg_en_b need to be high
- Clock provided.

Shutdown mode:

For shutdown:

- endac_in and auxdac_pg_en_b pins need to be low.

16.32.4 Register Summary

Address	Register Name	Description
11A	See 16.32.5.1 AUXDACREG0	To enable AUXDAC and rrbs

16.32.5 Register Description

16.32.5.1 AUXDACREG0

Table 16.712. AUXDACREG0 Register

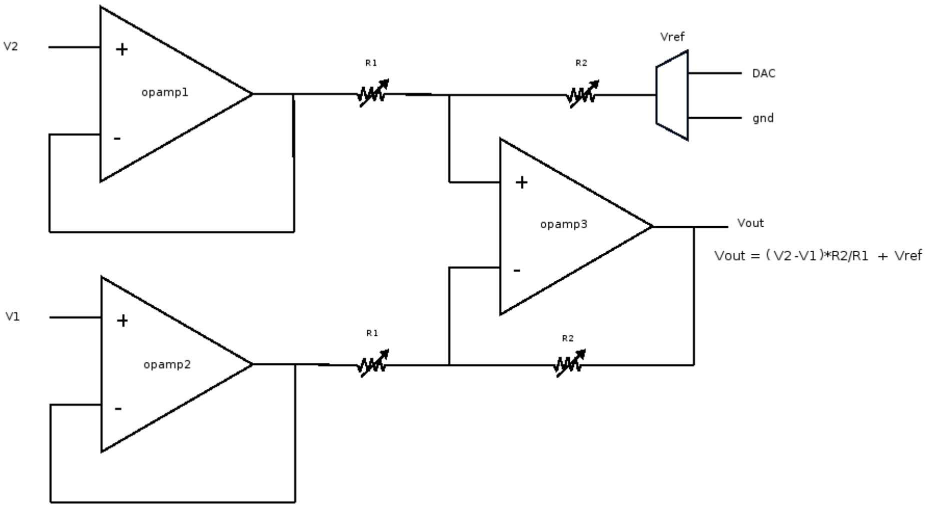
10'h11A				
Bit	Access	Functin	Default	Description
[13:6]	R/W	preset_rrbs<7:0>	8'd170	7:4 for ulsb rotation bits programming 3:0 for msb rotation bits programming
5	R/W	prbs_start	1	1 to select rrbs rotation 0 to disable rrbs rotation

10'h11A				
4:2	--	Reserved	0	Reserved
1	R/W	sel_pin_eni	1	1 to sel pin enable 0 to select from register
0	R/W	endaci_r	0	1 to enable dac 0 to disable dac

16.32.6 AUXDAC OUTPUT MUX

auxdac_out_mux_sel	0	1
GPIO	DAC0	DAC1

16.33 OPAMPS



16.33.1 General Description

The opamps top consists of 3 general purpose Operational Amplifiers (OPAMP) offering rail-to-rail inputs and outputs.

Each of the three opamps has 2 inputs (inp, inn) and 1 output.

16.33.2 Features

All the opamps can take inputs from GPIOs and their outputs can be seen on GPIOs.

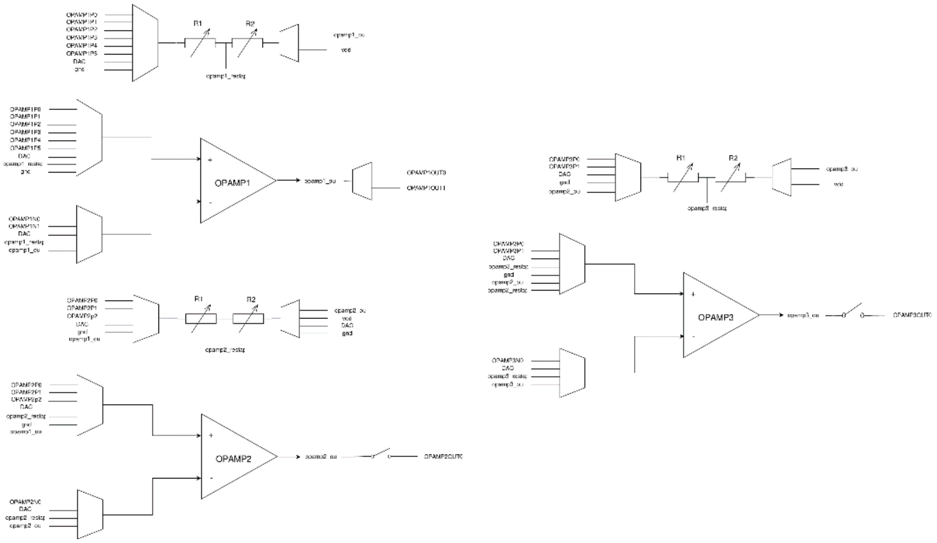
An opamp can be configured in either low power mode or high power mode. Make lp_mode bit high to enable low power mode.

The opamps top consists of 3 general purpose Operational Amplifiers (OPAMP) offering rail-to-rail inputs and outputs.
The opamps can be configured as follows:

1. Section 16.33.8.1 Unity Gain Buffer or Voltage Follower
2. Section 16.33.8.2 Trans-Impedance Amplifier (TIA)
3. Section 16.33.8.4 Non-Inverting PGA
4. Section 16.33.8.3 Inverting PGA
5. Section 16.33.8.9 Non-Inverting Comparator with Programmable Hysteresis
6. Section 16.33.8.8 Inverting Comparator with Programmable Hysteresis
7. Section 16.33.8.6 Cascaded Non-Inverting PGA
8. Section 16.33.8.5 Cascaded Inverting PGA
9. Section 16.33.8.7 Two Opamps Differential Amplifier
10. Section 16.33.8.10 Instrumentation Amplifier

- 7,8,9 are configured by cascading 2 opamps.
- 10 is configured by cascading 3 opamps

16.33.3 Block Diagram



16.33.4 Functional Description

Following is needed for OpAmp operation:

- Configured power mode (normal mode / low power mode using lp_mode bit).
- Configure the opamp into one of the possible configurations (as described earlier) and then enable the opamp. By default opamps are disabled.
- Opamp's gain (R2/R1 or 1+R2/R1) can be configured using Resistor Banks
- To see an opamp's output on its respective GPIO, make out_mux_en high

16.33.5 Input Selection

Every opamp will have Vinp mux to select "inp", Vinn mux to select "inn," and Resistor mux for feedback.

16.33.5.1 Vinp Mux Selection

inp_sel	0	1	2	3	4	5	6	7	8
Opamp1	OPAMP1P0 (GPIO_27)	OPAMP1P1(ULP_GPIO_7)	OPAMP1P2 (ULP_GPIO_0)	OPAMP1P3 (ULP_GPIO_2)	OPAMP1P4 (ULP_GPIO_6)	OPAMP1P5 (ULP_GPIO_8)	DAC	res_tap	gnd
Opamp2	OPAMP2P0 (ULP_GPIO_11)	OPAMP2P1 (ULP_GPIO_5)	opamp2_p2	DAC	res_tap	gnd	OPAMP1_out	--	--
Opamp3	OPAMP3P0 (ULP_GPIO_10)	OPAMP3P1 (GPIO_29)	DAC	res_tap	gnd	OPAMP2_out	OPAMP2_restap	--	--

16.33.5.2 Vinn Mux Selection

inn_sel	0	1	2	3	4
Opamp1	OPAMP1N0 (GPIO_27)	OPAMP1N1 (ULP_GPIO_7)	DAC	res_tap	out
Opamp2	OPAMP2N0 (ULP_GPIO_11)	DAC	res_tap	out	--
Opamp3	OPAMP3N0 (ULP_GPIO_10)	DAC	res_tap	out	--

16.33.5.3 Resistor Mux Selection

res_mux_sel	0	1	2	3	4	5	6	7
Opamp1	OPAMP1P0 (GPIO_27)	OPAMP1P1 (ULP_GPIO_7)	OPAMP1P2 (ULP_GPIO_0)	OPAMP1P3 (ULP_GPIO_2)	OPAMP1P4 (ULP_GPIO_6)	OPAMP1P5 (ULP_GPIO_8)	DAC	gnd
Opamp2	OPAMP2P0 (ULP_GPIO_11)	OPAMP2P1 (ULP_GPIO_5)	opamp2_p2	DAC	gnd	OPAMP1_out	--	--
Opamp3	OPAMP3P0 (ULP_GPIO_10)	OPAMP3P1 (GPIO_29)	DAC	gnd	OPAMP2_out	--	--	--

16.33.6 Configuring the Opamps

Each Opamp can be configured by its respective controls.

- Select the positive input using inp_sel
- Select the negative input using inn_sel
- Select en_res_bank if resistor bank is used.
- Select R1 and R2 using R1_sel and R2_sel respectively.
- Use res_to_out_vdd to connect resistor bank either to output or vdd.
- Select lp_mode to enable low power mode

There will be more than one option for some controls, and they have to be programmed depending on the input that need to be selected.

16.33.7 Standalone Mode

Each Opamp can be used as standalone amplifier. In this mode, positive input, negative input and the output are routed from/to external I/Os and uses external feedback. Opamps can also be cascaded to support some configurations.

	inp_sel	inn_sel	en_res_bank	res_mux_sel	R1_sel	R2_sel	res_to_out_vdd
opamp1	4'd0-4'd5	3'd0 - 3'd1	1'd0	-	-	-	-
opamp2	3'd1	2'd0	1'd0	-	-	-	-
opamp3	3'd1	2'd0	1'd0	-	-	-	-

16.33.8 Built-In Modes

16.33.8.1 Unity Gain Buffer or Voltage Follower

In the unity gain buffer configuration the output is connected to inverting input internally and the input has to be applied to non-inverting input. It has a 3dB bandwidth greater than 6MHz.

	inp_sel	inn_sel	en_res_bank	res_mux_sel	R1_sel	R2_sel	res_to_out_vdd
opamp1	4'd0-4'd5	3'd4	1'b0	-	-	-	-
opamp2	3'd0-3'd1	2'd3	1'b0	-	-	-	-
opamp3	3'd0-3'd1	2'd3	1'b0	-	-	-	-

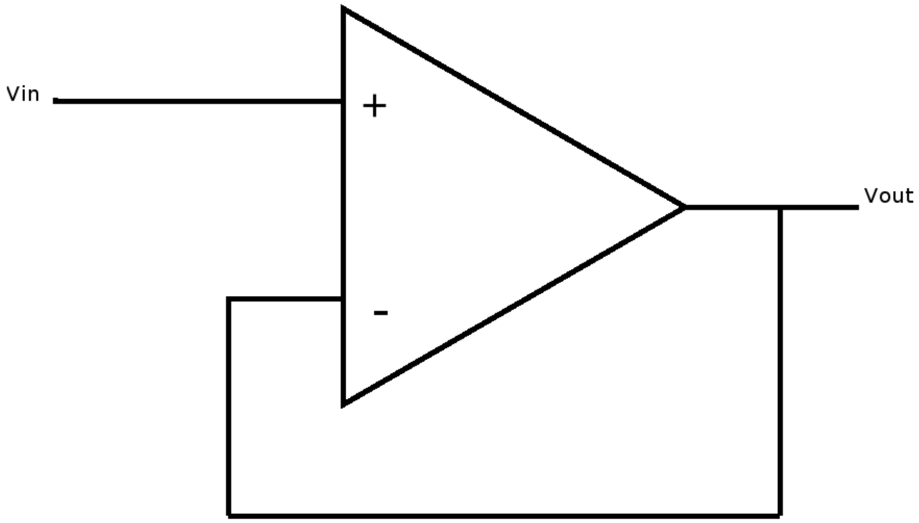
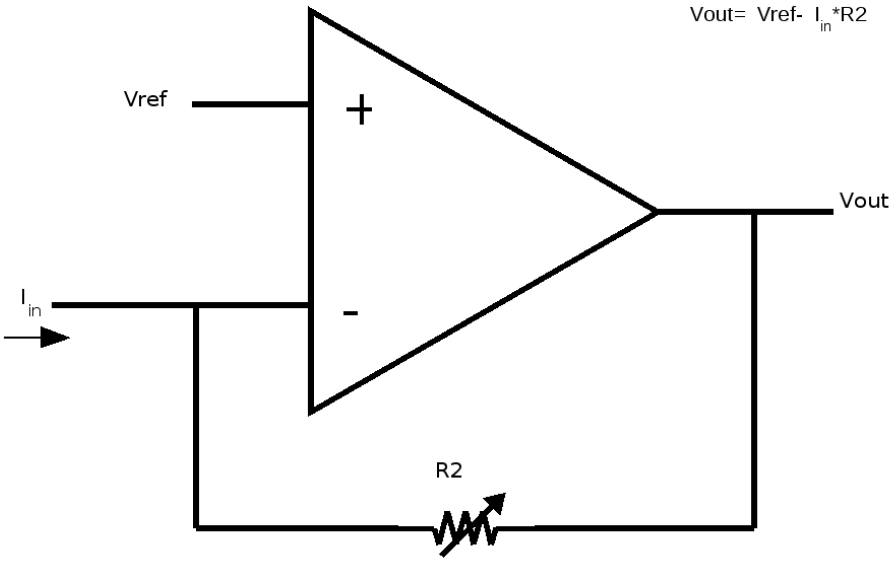


Figure 16.52. offset = Voff = vout - vin

16.33.8.2 Trans-Impedance Amplifier (TIA)

The TIA converts an internal or external current to an output voltage. It uses an internal resistor to convert input current to output voltage. The resistor is connected from inverting pin to output. The resistor value can be programmed from 20KΩ to 1MΩ. A reference voltage has to be applied to non inverting input of opamp. Then the output of opamp is Vref – Iin*R.

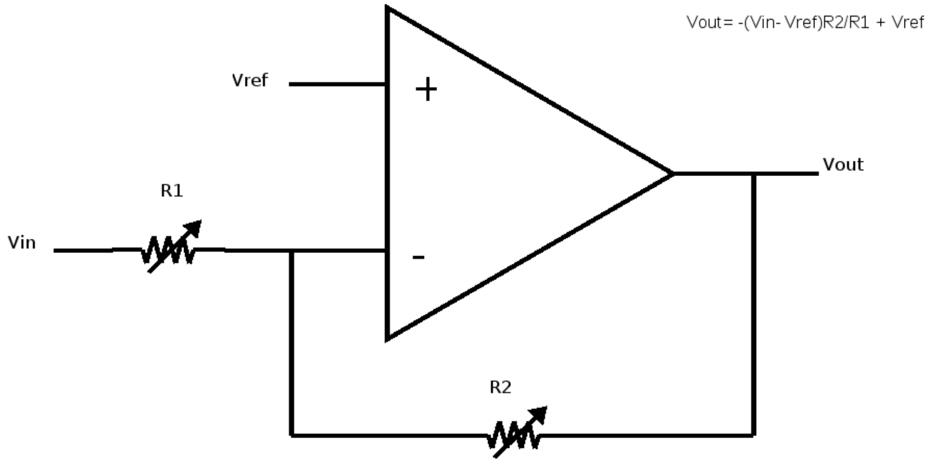
	inp_sel	inn_sel	en_res_bank	res_mux_sel	R1_sel<1:0>	R2_sel<2:0>	res_to_out_vdd
opamp1	4'd0 - 4'd6	3'd3	1	3'd0-3'd5	0	0-7	1'd0
opamp2	3'd0, 3'd1, 3'd3	2'd2	1	3'd0-3'd1	0	0-7	2'd0
opamp3	3'd0 - 3'd2	2'd2	1	3'd0-3'd1	0	0-7	1'd0



16.33.8.3 Inverting PGA

This mode amplifies an internal or external signal. In inverting amp configuration gain is –R2/R1, where R1, R2 are selected from R1_sel, R2_sel controls.

	inp_sel	inn_sel	en_res_bank	res_mux_sel	R1_sel<1:0>	R2_sel<2:0>	res_to_out_vdd
opamp1	4'd0-4'd6	3'd3	1	3'd0-3'd5	1-3	0-7	1'b0
opamp2	3'd0, 3'd1, 3'd3	2'd2	1	3'd0-3'd1	1-3	0-7	2'd0
opamp3	3'd0 - 3'd2	2'd2	1	3'd0-3'd1	1-3	0-7	1'b0



16.33.8.4 Non-Inverting PGA

This mode amplifies an internal or external signal. In non inverting amp configuration gain is 1+R2/R1, where R1, R2 are selected from R1_sel, R2_sel controls.

	inp_sel	inn_sel	en_res_bank	res_mux_sel	R1_sel<1:0>	R2_sel<2:0>	res_to_out_vdd
opamp1	4'd0-4'd5	3'd3	1	3'd0-3'd6	1-3	0-7	1'b0
opamp2	3'd0-3'd1	2'd2	1	3'd0,3'd1,3'd3	1-3	0-7	2'd0
opamp3	3'd0-3'd1	2'd2	1	3'd0-3'd2	1-3	0-7	1'b0

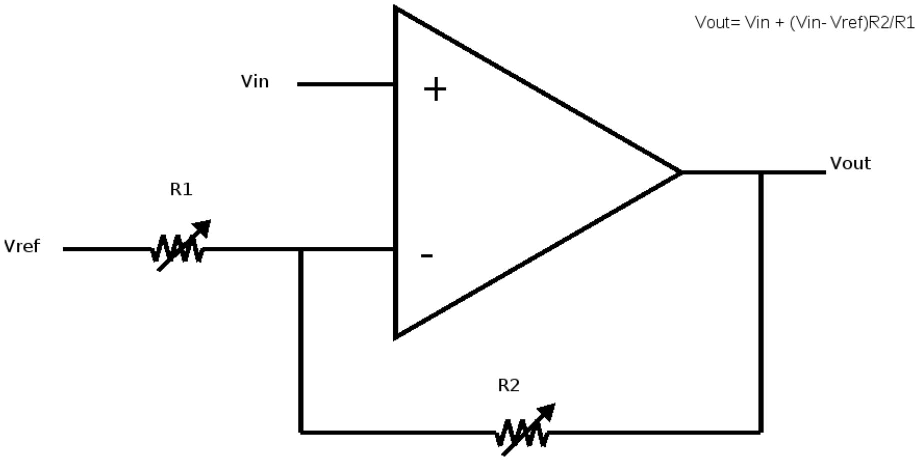


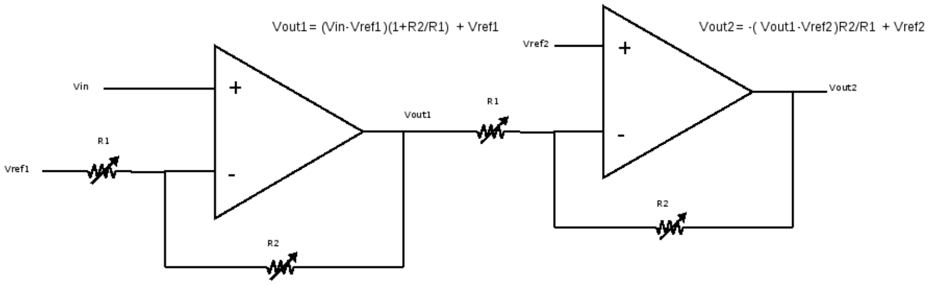
Figure 16.53. Vout = Vin + (Vin-Vref)R2/R1 + Voff(1+R2/R1)

16.33.8.5 Cascaded Inverting PGA

The 2 opamps, one in non inverting and the other in inverting PGA configuration can be cascaded to get cascaded inverting PGA using following settings.

	inp_sel	inn_sel	en_res_bank	res_mux_sel	R1_sel<1:0>	R2_sel<2:0>	res_to_out_vdd
opamp1	4'd0-4'd5	3'd3	1	3'd0-3'd6	1-3	0-7	1'b0
opamp2	3'd0, 3'd1,3'd3	2'd2	1	3'd5	1-3	0-7	2'd0

This can be implemented using opamp2 and opamp3 also in the same way.

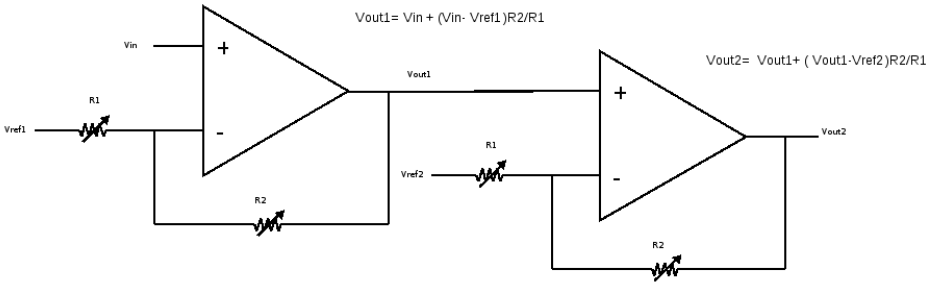

Vout1 = (Vin-Vref1)(1+R2/R1) + Vref1
Vout2 = -(Vout1-Vref2)R2/R1 + Vref2
Vref2
+
Vin
+
Vout2
Vout1
R1
R1
Vref1
-
-
R2
R2

16.33.8.6 Cascaded Non-Inverting PGA

The 2 opamps each in non inverting PGA configuration can be cascaded to get cascaded non inverting PGA using following settings.

	inp_sel	inn_sel	en_res_bank	res_mux_sel	R1_sel<1:0>	R2_sel<2:0>	res_to_out_vdd
opamp1	4'd0-4'd5	3'd3	1	3'd0-3'd6	1-3	0-7	1'b0
opamp2	3'd6	2'd2	1	3'd0,3'd1,3'd3	1-3	0-7	2'd0

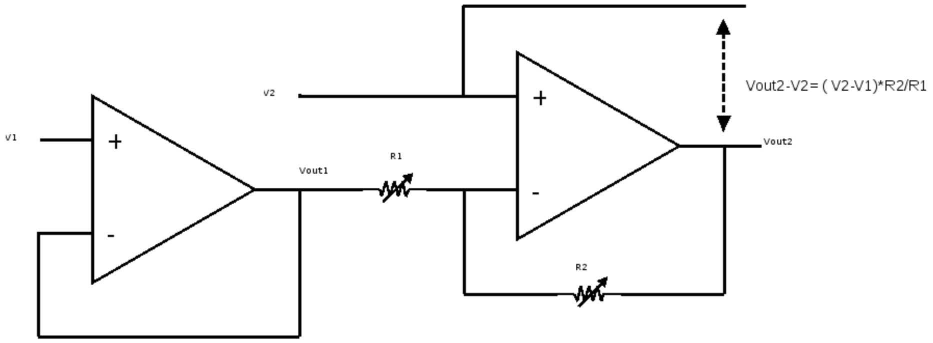
This can be implemented using opamp2 and opamp3 also in the same way.


Vout1= Vin + (Vin- Vref1)R2/R1
Vin
+
-
R1
Vref1
Vout1
R2
Vout2= Vout1+ (Vout1-Vref2)R2/R1
+
-
Vout2
R1
Vref2
R2

16.33.8.7 Two Opamps Differential Amplifier

This configuration can be achieved by following settings. The 1st Opamp is in UGB mode and 2nd Opamp is in PGA mode. The differential output is taken between output and non inverting input of 2nd Opamp.

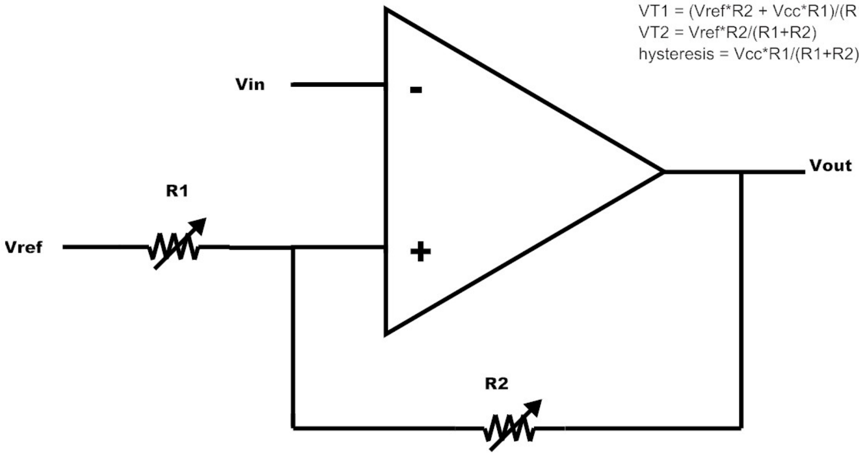
	inp_sel	inn_sel	en_res_bank	res_mux_sel	R1_sel<1:0>	R2_sel<2:0>	res_to_out_vdd
opamp1	4'd0-4'd5	3'd4	0	--	--	--	--
opamp2	3'd0-3'd1	2'd2	1	3'd5	1-3	0-7	2'd0



16.33.8.8 Inverting Comparator with Programmable Hysteresis

In both inverting and non-inverting comparator configurations, the non inverting input of Opamp is connected to resbank.
The Opamp can be configured as inverting comparator using these settings.

	inp_sel	inn_sel	en_res_bank	res_mux_sel	R1_sel<1:0>	R2_sel<2:0>	res_to_out_vdd
opamp1	4'd7	3'd0-3'd1	1	3'd0-3'd6	1-3	0-7	1'b0
opamp2	3'd4	2'd0	1	3'd0,3'd1,3'd3	1-3	0-7	2'd0
opamp3	3'd3	2'd0	1	3'd0-3'd2	1-3	0-7	1'b0

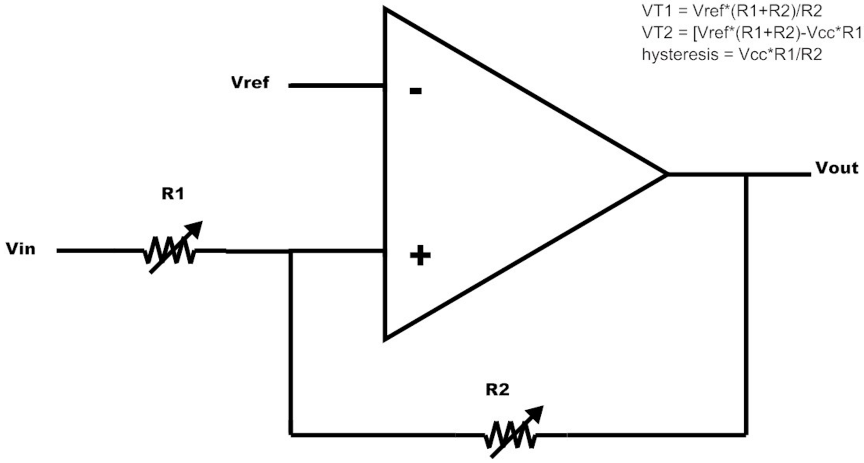


VT1 = (Vref*R2 + Vcc*R1)/(R1+R2)
VT2 = Vref*R2/(R1+R2)
hysteresis = Vcc*R1/(R1+R2)

16.33.8.9 Non-Inverting Comparator with Programmable Hysteresis

The Opamp can be configured as non inverting comparator using following settings.

	inp_sel	inn_sel	en_res_bank	res_mux_sel	R1_sel<1:0>	R2_sel<2:0>	res_to_out_vdd
opamp1	4'd7	3'd0-3'd2	1	3'd0-3'd5	1-3	0-7	1'b0
opamp2	3'd4	2'd0-2'd1	1	3'd0-3'd1	1-3	0-7	2'd0
opamp3	3'd3	2'd0-2'd1	1	3'd0-3'd1	1-3	0-7	1'b0



VT1 = Vref*(R1+R2)/R2
VT2 = [Vref*(R1+R2)-Vcc*R1]/R2
hysteresis = Vcc*R1/R2

16.33.8.10 Instrumentation Amplifier

In this mode opamp1 and opamp2 are configured as voltage follower and opamp3 will amplify the differential voltage of opamp1 and opamp2's outputs.

	inp_sel	inn_sel	en_res_bank	res_mux_sel	R1_sel<1:0>	R2_sel<2:0>	res_to_out_vdd
opamp1	4'd0-4'd5	3'd4	0	--	--	--	--
opamp2	3'd0-3'd1	2'd3	1	3'd5	1-3	0-7	2'd2
opamp3	3'd6	2'd2	1	3'd4	1-3	0-7	1'd0

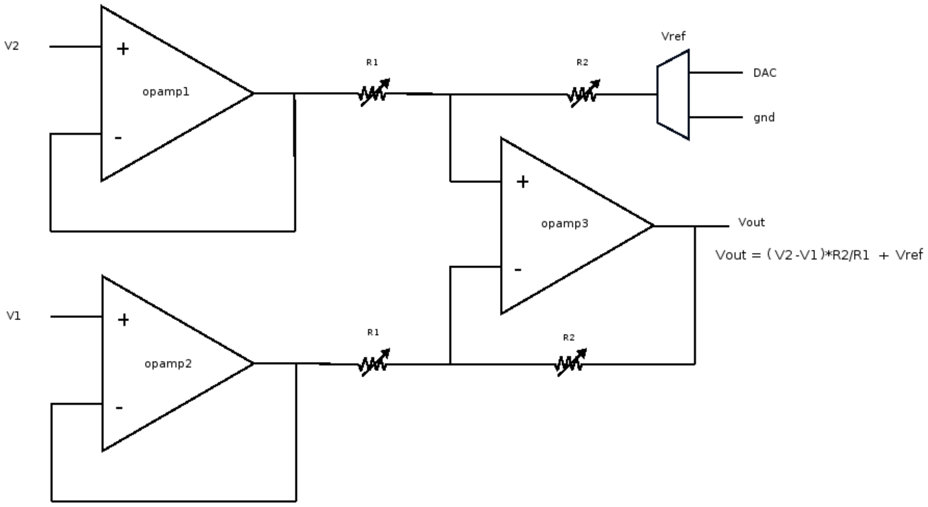


Figure 16.54.

16.33.9 Resistor Banks

There are two sets of resistor banks, one for R1 and other for R2. R1 can be programmed using R1_sel<1:0> and R2 can be programmed using R2_sel<2:0>.

R1_sel<1:0>	R1 (KΩ)
0	0 (short)
1	20
2	60
3	140
R2_sel<2:0>	**R2 (KΩ)**
0	20
1	30
2	40
3	60
4	120
5	250
6	500
7	1000

16.33.10 Opamp's output on GPIO

opamp1_out	- opamp1_out_mux_en =1, opamp1_out_mux_sel =0	- ULP_GPIO_4
opamp1_out	- opamp1_out_mux_en =1, opamp1_out_mux_sel =1	- GPIO_30
opamp2_out	- opamp2_out_mux_en =1	- ULP_GPIO_9
opamp3_out	- opamp3_out_mux_en =1	- GPIO_27

16.33.11 Guidelines for 'Ton' Selection

Ton represents the time required for opamp output to settle to 12 bit accuracy from enable signal.

Configuration	Gain	HP/LP mode	TYP	MAX	Units
Non Inverting Amplifier	2	HP	285n	846n	s
		LP	933n	2.1u	s
	50	HP	9.1u	12.62u	s
		LP	14.53u	21.1u	s
Inverting Amplifier	-1	HP	293n	846n	s
		LP	959n	2.09u	s
	-50	HP	9.09u	12.64u	s
		LP	14.49u	21.15u	s
Instrumentation Amplifier	1	HP	285n	1.01u	s
		LP	774n	1.75u	s
	50	HP	9.16u	12.91u	s
		LP	14.98u	21.81u	s
Cascaded Non Inverting Amplifier	4	HP	380n	1.471u	s
		LP	900n	1.86u	s
	2601	HP	18u	25u	s
		LP	29u	42u	s

16.33.12 Register Summary

Base Address: 0x24043800

Table 16.713. Register Summary

Register Name	Offset	Reset Value	Description
OPAMP1 1	0x214	0x 0000 0004	Programs opamp1
OPAMP2 1	0x218	0x 0000 0004	Programs opamp2
OPAMP3 1	0x21C	0x 0000 0004	Programs opamp3

16.33.13 Register Description

16.33.13.1 OPAMP_1

Table 16.714. Register Description

Bit	Access	Function	Default value	Description
31	R/W	opamp1_dyn_en	0	Dynamic Enable For Opamp1 signals
30:27	R/W	vref_mux_sel	0	Vref Mux Sel
26	R/W	mux_en	0	Mux Enable
25:22	R/W	vref_mux_en	0	VRef Mux Enable
21	R/W	mems_res_bank_en	0	Enable Mems Res Bank
20	R/W	opamp1_out_mux_sel	0	1 - To connect opamp1 output to pad
19:16	R/W	opamp1_inp_sel	0	Selecting +ve input of opamp
15:13	R/W	opamp1_inn_sel	0	Selecting -ve input of opamp
12	R/W	opamp1_out_mux_en	0	Out Mux Enable
11	R/W	opamp1_res_to_out_vdd	0	0 – connect resbank to out 1 – connect resbank to vdd
10:8	R/W	opamp1_res_mux_sel	0	Selecting input for resistor bank
7	R/W	opamp1_en_res_bank	0	0 – disable 1 – enable resistor bank
6:4	R/W	opamp1_R2_sel	0	Programmability to select resister bank R2
3:2	R/W	opamp1_R1_sel	1	Programmability to select resister bank R1
1	R/W	opamp1_lp_mode	0	0 – normal mode 1 – low power mode
0	R/W	opamp1_enable	0	0 - disable 1 - to enable opamp 1

16.33.13.2 OPAMP_2

Table 16.715. Register Description

Bit	Access	Function	Default value	Description
31:20	R	Reserved	-	Reserved
19	R/W	opamp2_dyn_en	0	Dynamic Enable For Opamp2 signals
18:16	R/W	opamp2_inp_sel	0	Selecting +ve input of opamp
15:14	R/W	opamp2_inn_sel	0	Selecting -ve input of opamp
13	R/W	opamp2_out_mux_en	0	Out Mux Enable

Bit	Access	Function	Default value	Description
12:11	R/W	opamp2_res_to_out_vdd	0	0 – connect resbank to out 1 – connect resbank to vdd 2 – connect resbank to DAC 3 – connect resbank to gnd
10:8	R/W	opamp2_res_mux_sel	0	Selecting input for resistor bank
7	R/W	opamp2_en_res_bank	0	0 – disable 1 – enable resistor bank
6:4	R/W	opamp2_R2_sel	0	Programmability to select resister bank R2
3:2	R/W	opamp2_R1_sel	1	Programmability to select resister bank R1
1	R/W	opamp2_lp_mode	0	0 – normal mode 1 – low power mode
0	R/W	opamp2_enable	0	0 - disable 1 - to enable opamp 2

16.33.13.3 OPAMP_3

Table 16.716. Register Description

Bit	Access	Function	Default value	Description
31:19	R	Reserved	-	-
18	R/W	opamp3_dyn_en	0	Dynamic Enable For Opamp3 signals
17:15	R/W	opamp3_inp_sel	0	Selecting +ve input of opamp
14:13	R/W	opamp3_inn_sel	0	Selecting -ve input of opamp
12	R/W	opamp3_out_mux_en	0	Out Mux Enable
11	R/W	opamp3_res_to_out_vdd	0	0 – connect resbank to out 1 – connect resbank to vdd
10:8	R/W	opamp3_res_mux_sel	0	Selecting input for resistor bank
7	R/W	opamp3_en_res_bank	0	0 – disable 1 – enable resistor bank
6:4	R/W	opamp3_R2_sel	0	Programmability to select resister bank R2
3:2	R/W	opamp3_R1_sel	1	Programmability to select resister bank R1
1	R/W	opamp3_lp_mode	0	0 – normal mode 1 – low power mode
0	R/W	opamp3_enable	0	0 - disable 1 - to enable opamp 3

16.34 Temperature Sensor: BJT Based

16.34.1 General Description

BJT based sensor works for temperature range from -40° to 125° and voltage variation from 1.8V to 3.6V. It outputs the digital word having resolution of nearly 1 degree C. The conversion time is 2 clock cycles of ADC after turning ON the temperature sensor.

16.34.2 BJT Temperature Sensor

16.34.2.1 Register Summary

Address	Register Name
base address	0x24043800
1E0	TS_PTAT_ENABLE

16.34.2.2 Register Description

TS_PTAT_ENABLE				
Offset Address: 1E0				
Bit	Access	Function	Default Value	Description
[31:1]	R	Reserved	0x0	Reserved
[0]	R/W	ts_ptat_enable	0x0	BJT based Temperature sensor enable 1 : Enable 0 : Disable

17. Security Architecture

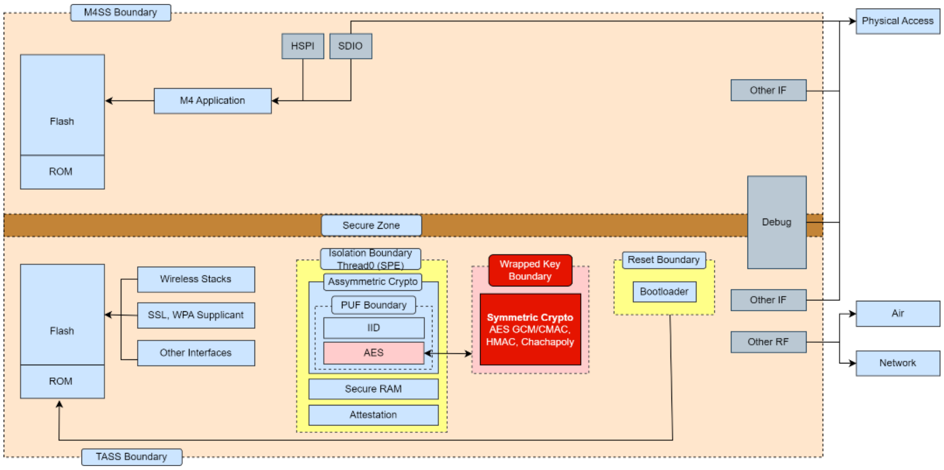
17.1 General Description

A rich set security features are provided to differentiate the end product.

17.2 Features

- GlobalPlatform® - Trusted Execution Environment compatible architecture with separate processor for secure applications.
- Hardware device identity and key storage with PUF based secure Roots-of-trust (RoT).
- True Random Number Generator.
- High Performance Security Accelerators:
  - Hardware accelerators - AES128/256, SHA256/384/512, CRC, SHA3, AES-GCM/CMAC, ChaCha-poly, TRNG.
  - Software Implementation - ECC, RSA
- FIPS140-2 certifiable
- Secure XIP from Flash with inline AES engine
- Secure Boot loading performed by the secure processor.
- Secure Firmware upgrade

17.3 Security Overview



17.4 Register Summary

There are no registers in this section.

17.5 Register Description

There are no registers in this section.

18. In-System Programming (ISP)

18.1 General Description

In System Programming (ISP) is programming or reprogramming of the flash through boot loader. This can be done after the part is integrated on end user board.

18.2 Features

ISP can done through any of the following interfaces

Interface	Pins
UART	Rx: GPIO_8 Tx: GPIO_9
HSPI	GPIO_25 - GPIO_30
SDIO	GPIO_25 - GPIO_30

18.3 Functional Description

Boot loader can be requested to boot in ISP mode by pulling down the GPIO_34 pin with 4.7 K ohms of resistor. This pin has to be left unconnected during reset for the boot loader to bypass ISP and execute the code that is present in flash. ISP mode can be used to reprogram the flash, if the application codes uses JTAG pins for functional use. On boot up, if the application code goes into a state where JTAG interface is not functioning, ISP mode can be used to gain the control and to reprogram the flash.

18.4 Register Summary

There are no registers in this module.

18.5 Register Description

There are no registers in this module.

19. Bootloader

19.1 General Description

The Bootloader controls the initial operation of the device after any form of reset. The Bootloader supports Flash programming and initial startup of the application code.

19.2 Features

- Two Bootloaders - Security Bootloader and Application Bootloader
- Support for ISP (In-System Programming) through multiple interfaces - UART, SPI, and SDIO
- Auto-detection of ISP interface. The host interfaces are the external peripheral interfaces over which Bootloader can receive commands or firmware when in ISP mode. The Bootloader supports UART, SPI and SDIO interfaces. Bootloader in ISP mode waits for data on any of these interfaces and can automatically detect which interface the data is being received.
- Support for secure boot
- Support for secure firmware upgrade using PUF based Roots-of-Trust (RoT)
- Anti-rollback protection. This feature prevents the firmware version from being downgraded. A new firmware is allowed to be upgraded only if it is be equal to or greater than the current firmware.
- Secure Key Management and Protection
- Support for different flash protection levels and write-protected Flash
- Secure XIP from Flash
- Support for multiple isolated images and selection
- Fail-proof migration of current active firmware to new (update) firmware
- Public key cryptography (digital signature) based authentication

19.3 Functional Description

The SiWx91x includes two Bootloaders - Security Bootloader and Application Bootloader. The Security Bootloader runs on the Network Wireless Processor and the Application Bootloader runs on the Cortex M4 processor. On any reset, execution will always start in Security Bootloader, which is responsible for all security features, ISP and firmware upgrades. Once the Security Bootloader finishes its tasks, it enables the Application Bootloader. The Application bootloader will load and execute the application and also execute wakeup sequence on wakeup from sleep.

The following are the sources which can trigger the Bootloader:

- Primary reset (RESET_N_PAD)
- Power on reset (POC_IN)
- Watchdog reset
- Black out monitor
- Reset request through SYSRESETREQn bit in the Cortex-M4 processor
- Wake-up from Sleep

19.4 Secure Boot

Key Features

- Ensures the device runs authentic code in the boot and OTA update to eliminate malware insertion threats
- Secure Immutable Bootloader in ROM.
- Authenticates signatures of all other SW using public keys.
- NWP and M4 flash images can be encrypted with separate keys.

On reset, the Security Bootloader configures the module hardware based on the configuration present in the eFuse. It also passes the required information from the eFuse to the Application Bootloader. The Security Bootloader validates the integrity and authenticity of the firmware in the Flash and invokes the Application Bootloader. It detects and prevents execution of unauthorized software during the boot sequence. The Bootloader uses public & private key based digital signatures to recognize authentic software. The Security Bootloader provides provision for inline execution (XIP) of encrypted firmware from Flash. The Bootloader provides 3 flash protection levels which can used to secure different sections of the Flash for different purposes:

- Protection level 1: Stored at manufacturing, not allowed to modify by the Security Bootloader
- Protection level 2: Allowed to modify by the Bootloader only, usually used to maintain secure information used/consumed by Bootloader
- Protection level 3: Allowed to modify by the Bootloader only, usually used to maintain protected firmware images.

The protection levels are written to Flash during the manufacturing process. The write-protection feature prevents the application program from changing the Flash protection levels. The Bootloader supports multiple isolated firmware and provision to select the firmware to execute on bootup.

The Security Bootloader is enabled or disabled during the manufacturing process.

19.5 Secure Firmware Upgrade (ISP)

The secure firmware upgrade feature of the Bootloader checks the authenticity of the new firmware image along with its integrity. The Bootloader supports the following interfaces to upgrade the firmware:

- - UART
  - SPI
  - SDIO

The Bootloader automatically detects the host interface in use and configures the host interface hardware accordingly. The Bootloader updates the image only after successfully validating the authenticity and integrity of the image. It prevents downgrade to a lower version of firmware using the anti-rollback feature, if it is enabled. The Bootloader also supports transparent migration to a wirelessly updated image and protection against failures by providing recovery mechanisms.

Secure OTA :

- Secure OTA update to eliminate malware insertion threats
- Wireless and Application image transfer over the air
- Network Wireless Processor authenticates the signatures of OTA image using public keys
- Bootloader copies the OTA image to primary firmware location upon successful authentication

19.6 Secure Key management

The Bootloader supports robust key management and secure key upgradation. The Bootloader's key management section maintains the following types of keys:

Key Type	Description
Master key	The Master key is used to validate the integrity of other keys and can be used to store the other keys in encrypted form. It is a unique key generated using PUF, specific to a device to protect against cloning.
Firmware key	The Firmware key is used to validate the integrity of images and can be used to store the images in encrypted form. It is a unique key generated using PUF, specific to a device to protect the firmware image
Firmware Upgrade key	The Bootloader uses the Firmware Upgrade key to decrypt (if encrypted) and check the integrity of new firmware image.
Public key	The Bootloader uses the Public key to validate the signature of the image during upgradation and secure bootup
PUF keys	The PUF keys are a table of keycode, which is used to retrieve the keys stored in PUF during the manufacturing process. The Master key and the Firmware key can be PUF keys.
User keys	The Bootloader provides provision to maintain keys used by the user application.

19.7 Secure Zone

Secure Zone Key Features

- Secure Zone is a hardware enforced isolation between the NWP core from the Application (M4) core.
- No access to NWP processor, NWP address space memory (except misc-registers that are used for bootloader command exchanges) and NWP HW registers from external peripherals and host interfaces
- Secure zone protects NWP internal keys (for example, secure boot keys), customer application keys, and certificates residing in the NWP by restricting direct access from the application M4 core.
- Secure zone can be enabled during the manufacturing process, at the user's discretion, by setting the secure_isolation_enabled OTP/MBR bit.

When Secure Zone is enabled, the Host Processor cannot reconfigure interfaces that are controlled by the NWP by default. For instance, ISP interfaces such as UART, SPI, and SDIO will be mutually exclusive between the NWP and M4 when secure zone is enabled. This is because secure zone will not allow Host processor to program the registers (in NWP space) to get control of those interfaces. In such cases, configurations in MBR defines the default ownership of the interface.

19.8 Bootloader Flowchart

The following diagram shows the top level flow for the Bootloader.

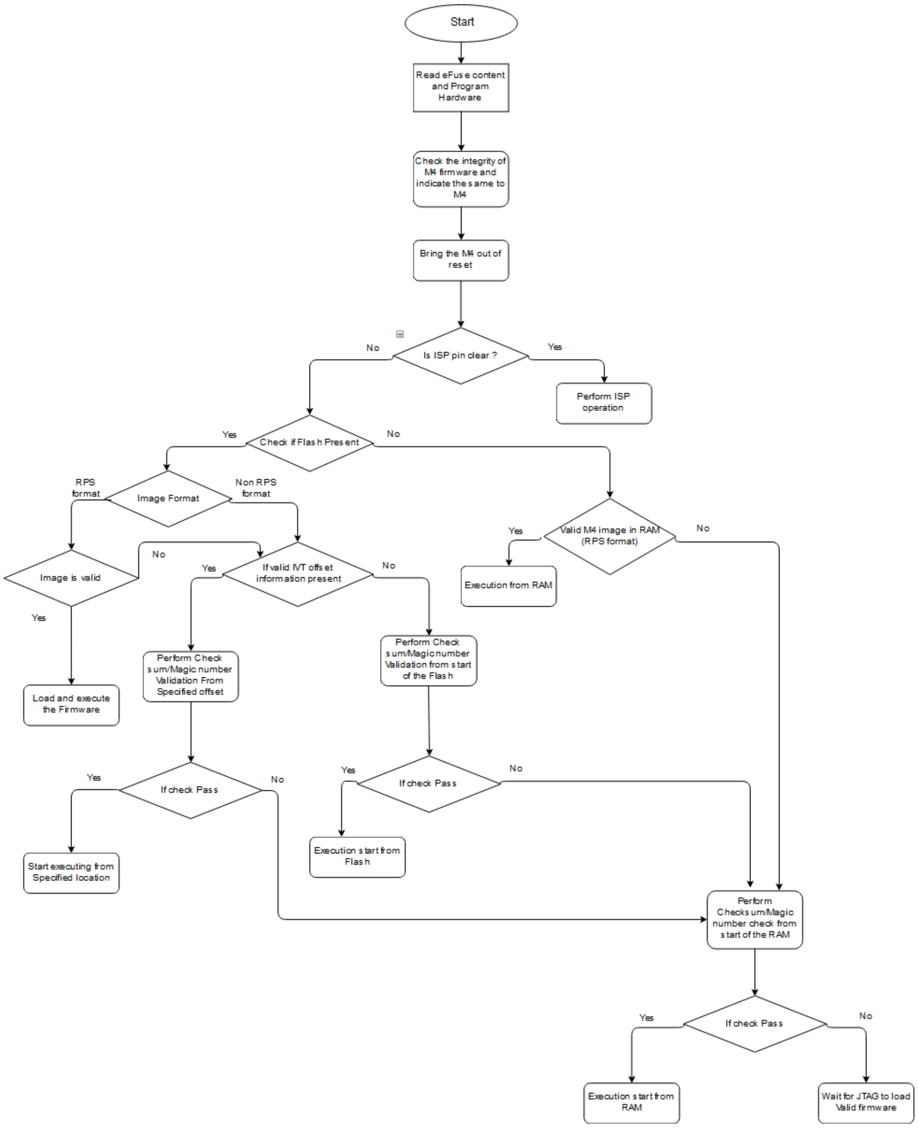

Start
Read eFuse content and Program Hardware
Check the integrity of M4 firmware and indicate the same to M4
Bring the M4 out of reset
No
Is ISP pin clear ?
Yes
Perform ISP operation
Yes
Check if Flash Present
No
RPS format
Image Format
Non RPS format
Yes
Valid M4 image in RAM (RPS format)
No
Execution from RAM
No
Image is valid
Yes
If valid IVT offset information present
No
Yes
Perform Check sum/Magic number Validation From Specified offset
Perform Check sum/Magic number Validation from start of the Flash
Load and execute the Firmware
Yes
If check Pass
No
Yes
If check Pass
No
Execution start from Flash
Start executing from Specified location
Perform Checksum/Magic number check from start of the RAM
Yes
If check Pass
No
Execution start from RAM
Wait for JTAG to load Valid firmware

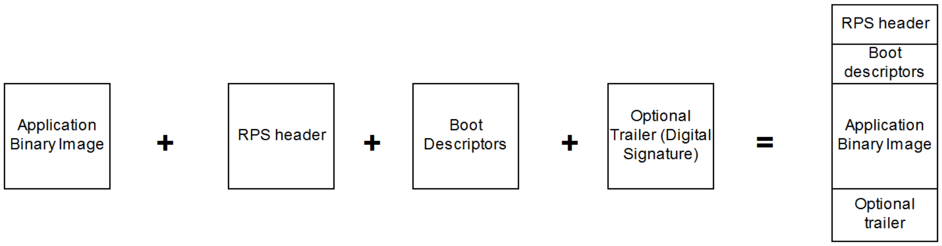
Figure 19.1. Bootloader Flowchart

The Boot loader supports RPS Format for the firmware image.

19.8.1 RPS Format

The RPS Format is a binary executable format understood by the Boot loader to perform the required integrity and authenticity checks and load and execute the application.

The Firmware Image in RPS format includes an RPS header, Boot descriptors, Application's binary image and an optional trailer (digital signature).



RPS Header:

This information is used by the Bootloader to process the configuration related to the firmware image.

Field	Size	Description
Control flags	16 bit	Bit map which Indicates image information BIT(0) : 0 - NWP processor image 1 - MCU image BIT(1): 0 - Image is not encrypted 1 - Image is encrypted BIT(2) : 0 - CRC based integrity check 1 - MIC based integrity check BIT(3) : 0 - Digitally not signed 1 - Digitally Signed (in this case the Digital signature is attached in the Trailer section of the image)
sha_type	16 bit	Represents the SHA size used to compute the digest for the digital signature 1 - SHA_256 2 - SHA_384 3 - SHA_512
Reserved	32 bit	
image_size	32 bit	Size of the image
fw_version	32 bit	Firmware version number
flash_address	32 bit	Address location in flash to store the image
crc	32 bit	CRC of the image (polynomial to use can be decided at the time of manufacturing)
mic	128 bit	MIC of the image

Field	Size	Description
public key reference	32 bit	Number to match with public key present in the device to validate the digital signature
Reserved	16 bit	

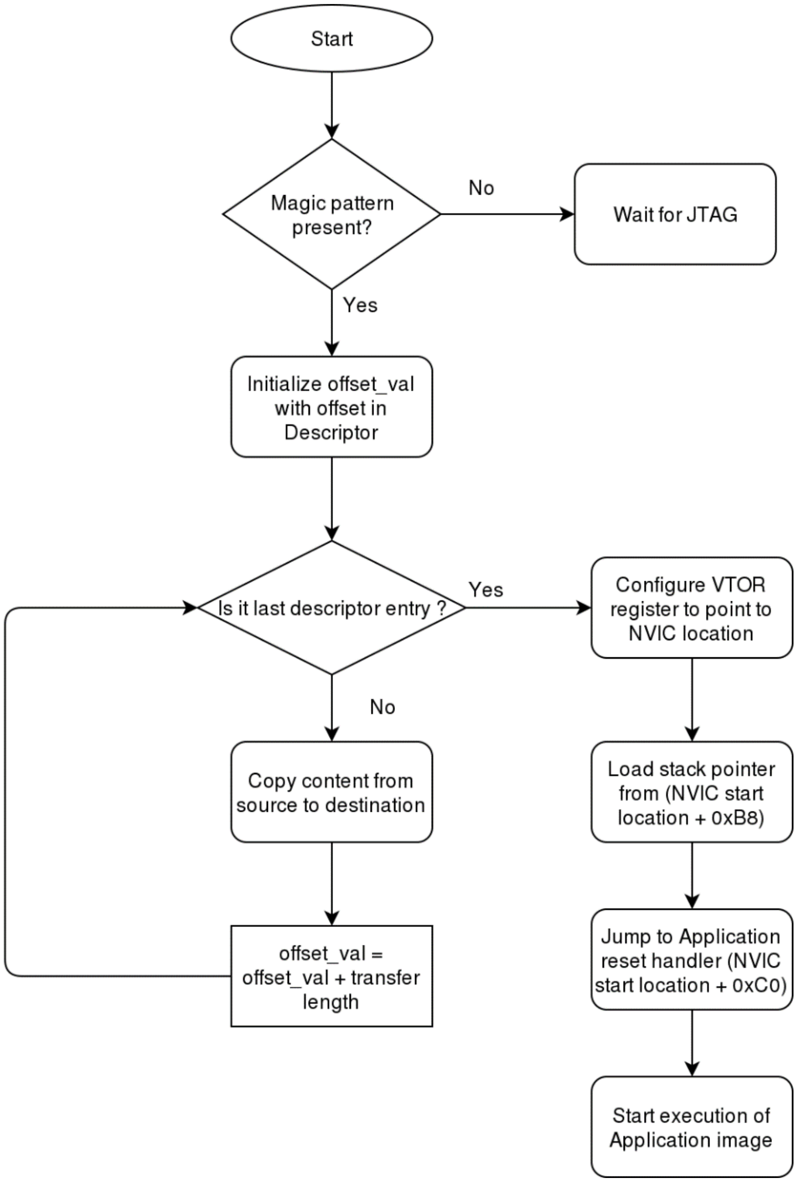
Boot Descriptors

The Boot Descriptors are executed by the Boot loader to load the firmware. By using these boot descriptors, the application can instruct the Boot loader to load the image (usually initialized data section) to different parts of the on-chip RAM before start of execution.

Field	Size	Description
magic_pattern	16 bit	Pattern for identification of a valid Flash content (0x5aa5)
offset	16 bit	Offset of the binary image where the transfer should start from flash to RAM
IVT offset	32 bit	Value to program VTOR register
bootload_entries	56 bytes	Boot loader descriptor entries which are executed by the Boot loader while loading firmware - see table below for more details.

Bootloader Descriptor Entries' Format:

Field	Size	Description
length	24 bit	Length of transfer to destination
reserved	7 bit	
last entry	1 bit	If set indicate it is last boot descriptor entry
dst_addr	32 bit	destination address


Start
Magic pattern present?
No
Wait for JTAG
Yes
Initialize offset_val with offset in Descriptor
Is it last descriptor entry ?
Yes
Configure VTOR register to point to NVIC location
No
Copy content from source to destination
Load stack pointer from (NVIC start location + 0xB8)
offset_val = offset_val + transfer length
Jump to Application reset handler (NVIC start location + 0xC0)
Start execution of Application image

19.9 Register Summary

There are no registers in this module.

19.10 Register Description

There are no registers in this module.

20. Trace and Debug

20.1 General Description

Trace and Debug functions are integrated into the ARM Cortex-M4. The Cortex-M4 TPIU supports two output modes:

- Clocked mode, using up to 4-bit parallel data output ports
- SWV mode, using single-bit SWV output

20.1.1 Serial Wire Viewer (SWV) Support

The SWV provides real-time data trace information from various sources within a Cortex-M4 device. It is transmitted via the SWO pin while your system processor continues to run at full speed. Information is available from the ITM and DWT units. SWV data trace is available via the SWO pin.

Setup: Configure GPIO_6 pin in GPIO Mode 13 (M4SS_TRACE_CLKIN) to loop back to GPIO_12 pin in Mode 8 (MCU_CLK_OUT). The MCU_CLK_OUT has programmable divider option from mcu clock. The MCU_CLK_OUT frequency must be less than 40Mhz to use the SWO function. By following the configurations data trace can be observed with the supporting debug probes.

MCU_CLK_OUT register details are present in Section 6.12.18.9 CLK_CONFIG_REG3.

Refer below steps as an example.

1. Loop back GPIO_6 pin in GPIO Mode 13 (M4SS_TRACE_CLKIN) to GPIO_12 pin in Mode 8 (MCU_CLK_OUT).
2. Program the Debug Exception and Monitor Control Register(DEMCR) to enable trace, Async Clock Prescaler Register for buad rate,ITM Trace Enable Register(TER) to enable tracing on stimulus ports and ITM Trace Control Register(TCR) to enable ITM,sync packets for TPIU, SWV behavior, ATB ID for CoreSight system.
3. Write data to the required ITM channel stimulus port and then will receive the same data to Debug(printf)Viewer of Keil IDE serial windows.

20.1.2 Embedded Trace Macrocell (ETM) Support

The ETM provides high bandwidth instruction trace via four dedicated trace pins accessible on the ***20-pin Cortex Debug + ETM*** connector. The MCU_CLK_OUT frequency must be in the range of 40Mhz to 90MHz to Instruction trace using ETM component.

Setup: Configure Trace pins (GPIO_52 to GPIO_57) in Mode 6 and GPIO_12 pin in Mode 8 (MCU_CLK_OUT). By default on power up these GPIO pins are mapped to Cortex Debug+ETM connector(20-pin). These pin mapping can be changed using the NC pin TRACE_ETM_DIS available on Rev1.2 GPIO Peripheral Card. When TRACE_ETM_DIS is high GPIO_53 to GPIO_57 pins are mapped to GPIO peripheral card and can be used as normal GPIOs. By following the configurations Instruction trace can be observed with the supporting debug probes.

Refer below steps as an example

1. Configure GPIO_53 to GPIO_57 pins as trace pins in ouput direction and GPIO_52 as trace pin in input direction before doing ETM enable.
2. External or internal PLL clock sources are feeded as mentioned above to GPIO_52(M4SS_TRACE_CLKIN).
3. Program TPIU,DWT and ETM registers to enable and control ETM for tracing.
4. Execute FreeRTOS blinky code by loading to RAM memory and we can observe Instruction tracing,Exception tracing,Code coverage features of ETM in the Keil IDE.
5. Trace pins need to be programmed before start of debugger. Now trace pins are programming using Initialization file(.ini) in Keil IDE and this need to be leveraged for all supporting IDEs.

   Note : Free running clock should be present on MCU_CLK_OUT(GPIO_12), before programming Trace pins otherwise IDE will popup **Trace HW not present** error.

21. MVP - Matrix Vector Processor

21.1 Introduction

The Matrix Vector Processor (MVP) is designed to offload the major computationally intensive floating point operations, particularly matrixed complex floating point multiplications and additions. The MVP hardware supports the acceleration of the key Angle-of-Arrival (AoA) MUSIC (MUltiple SIgnal Classification) algorithm computations, as well as other heavily floating-point computational problems such as Machine Learning (ML) or linear algebra.

21.2 Features

- Instruction Set Architecture (ISA)
    - General purpose instruction set tailored towards algorithms built out of ALU, loop, and load/store instructions
    - Enables many high-level array functions, e.g.:
        - Matrix multiplication
        - Element-wise matrix multiplication
        - Matrix addition
        - Power series generation
        - Convolution
    - Program flexibility allows efficient iteration over N-dimensional array elements, including in-place processing of special ,atrix views:
        - Element-wise negate / conjugate
        - Transpose / adjoint / reverse
        - Matrix blocks (i.e., rectangular parts of matrix)
        - Matrix slices (i.e., taking rows, columns, or elements uniformly spaced within a matrix)
        - Row-major or column-major ordering
- Arithmetic Logic Unit (ALU)
    - Support for floating point real and complex numbers
        - Partial integer input support
            - Floating-point output operands, interpreted as 16-bit real or 32-bit complex number (16-bit real and 16-bit imaginary)
            - Register bank to hold all input/output operands
        - Includes 8 registers for temporary storage and/or accumulation
    - Hardware to support 1 complex floating point multiply-accumulate (MAC) per cycle
        - Four single-precision floating-point multipliers
        - Four single-precision floating-point adders
        - 6x performance of Cortex M33 FMAC operations
    - Operations supported at a rate of one operation per cycle:
        - Complex addition, multiplication, and MAC operations
        - Parallel real multiplication and MAC
        - Parallel real addition
        - Sum of 4 reals
        - Squared-magnitude of complex/real
        - Integer-to-float conversion
        - Conditional computation
    - Input transformations (per real/complex part of each input)
        - Negation (complex conjugate)
        - Zero-masking (real/imaginary part decomposition)
- Load/Store Unit (LSU)
    - Controls data streaming from memory-to-ALU and vice versa
    - Pipelined architecture to support two simultaneous 32-bit memory reads and one 32-bit memory write per cycle
    - Supports signed / unsigned 8-bit integer conversion for both load and store operations
    - First-party DMA ports
        - Used by load / store unit for handling accesses to external (system) memory addresses
        - Three independent 32-bit AHB manager ports for supporting 2 read channels and 1 write channel simultaneously
- Sequencer
    - Coordinates all MVP blocks to execute a sequence of instructions provided via the programming interface
    - Handles array iteration according to instruction sequence and static array configuration
    - Handles loop iteration according to instruction sequence and static loop configuration

- Programming interface
    - Control registers for starting / stopping engine
    - Status registers about ongoing and finished instruction sequences
        - Fault status
        - Useful information for debug
    - Breakpoint and stepping controls for debug
    - Interrupts and faults
        - Instruction sequence completion
        - Bus faults
        - Loop faults
        - Array faults
    - Array configuration registers
    - Loop configuration registers
    - Instruction queue registers
        - Array iteration
        - ALU operations
        - Looping

21.3 Functional Description

The Matrix Vector Processor (MVP) is a peripheral processor that can be used to accelerate the processing of floating point operations while offloading the primary CPU. At a high level, it consists of:

- A register interface for programming and controlling operations
- A sequencer (which includes the loop controllers) that manages execution of the program
- Array and bus controllers that manage addressing, loading, and storing of data in arrays stored in system memory
- The pipeline controller, ALU, and ALU register bank for processing data

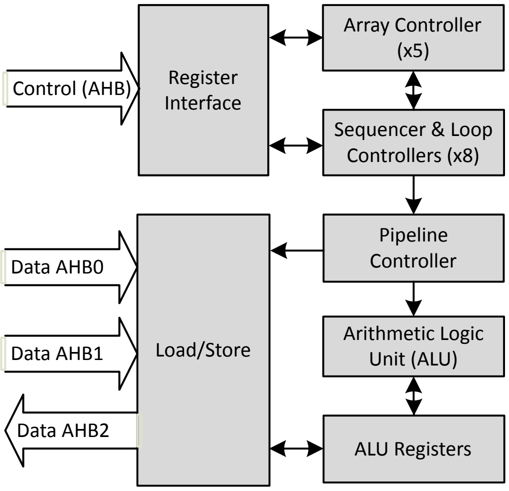


Figure 21.1. MVP Block Diagram

For most operations, software will program the MVP to address matrix (array) data in system memory and then process this data to perform a useful computation. The MVP provides three primary resources for controlling the operations that are fed through the ALU:

- Eight instructions, each of which encodes an ALU operation, load/store controls for the cycle, array increment controls, and loop controls
- Eight loop controllers, which can be used to form loops around a single or multiple instructions and can be nested to form complex sequences of ALU operations
- Five array controllers, each of which configure and control access to an independent matrix of data in system memory. The MVP supports arrays of up to 3 dimensions and each dimension can be independently incremented by the load/store streams, instruction, or loop controller on any given cycle

All of the MVP's control registers reside within its own register space, including the eight instructions (each of which is three 32-bit words). The MVP's register space has been organized such that DMA can be used to program all the configuration registers sequentially and initiate an operation with minimal CPU intervention. Once the program has successfully completed (by reaching an instruction with ENDPROG while completing all outstanding loops), the MVP interrupts the CPU with the PROGDONE interrupt. In the case of a fatal error (usually misprogramming), the MVP issues an error interrupt and terminate processing immediately.

22. Revision History

Revision 1.1

August, 2025

- Removed Programming Sequence and Register Descriptions to align with the data sheet.
- Updated features on front page.
- Updated System block diagram.
- Removed MPU (Memory Protection Unit).
- Removed Memory trap generation and its programming sequence.
- Updated Common flash description.
- Updated 6.10 Clock Distribution diagrams.
- Updated Clock Configuration register descriptions.
- Updated Power states diagram to align with the data sheet.
- Updated list of wake up sources to align with the data sheet.
- Updated Configurable Timers features to align with the data sheet.
- Updates I2S-PCM features.
- Updated UART features.
- Updated SDC features.

Revision 1.0

March, 2025

- Removed capacitive touch sensor feature.
- Updated hardware block diagram with missing blocks and corrected power state availability key.
- A 32.768 kHz crystal is mandatory for all applications requiring accurate timing or low-power Wi-Fi, BLE, and Coex sleep.
- UDMA section updated with architecture diagram , descriptor structure, channel priority register information.

Revision 0.7

September, 2024

- Replaced references to "ThreadArch" and "TA" with "Network Wireless Processor" and / or "NWP"
- Changed AI/ML mentions to more directly refer to the MVP block capabilities
- Updated peripheral and signal names for consistency with software libraries
- Removed unsupported features: SIO, IrDA, RO temperature sensor.
- Removed Pin-mux section as it is already part of the datasheet.
- Removed Flash/PSRAM Supply connections as it is already present in the datasheet.
- Presentation and formatting changes throughout document, including figure and table title assignments, units, cross-references, specification table formats, etc.
- Moved all the peripherals (AHB, APB, ULP, Analog, etc..) under single peripheral section.

Revision 0.3

January, 2024

Converted to DITA

Revision 0.2

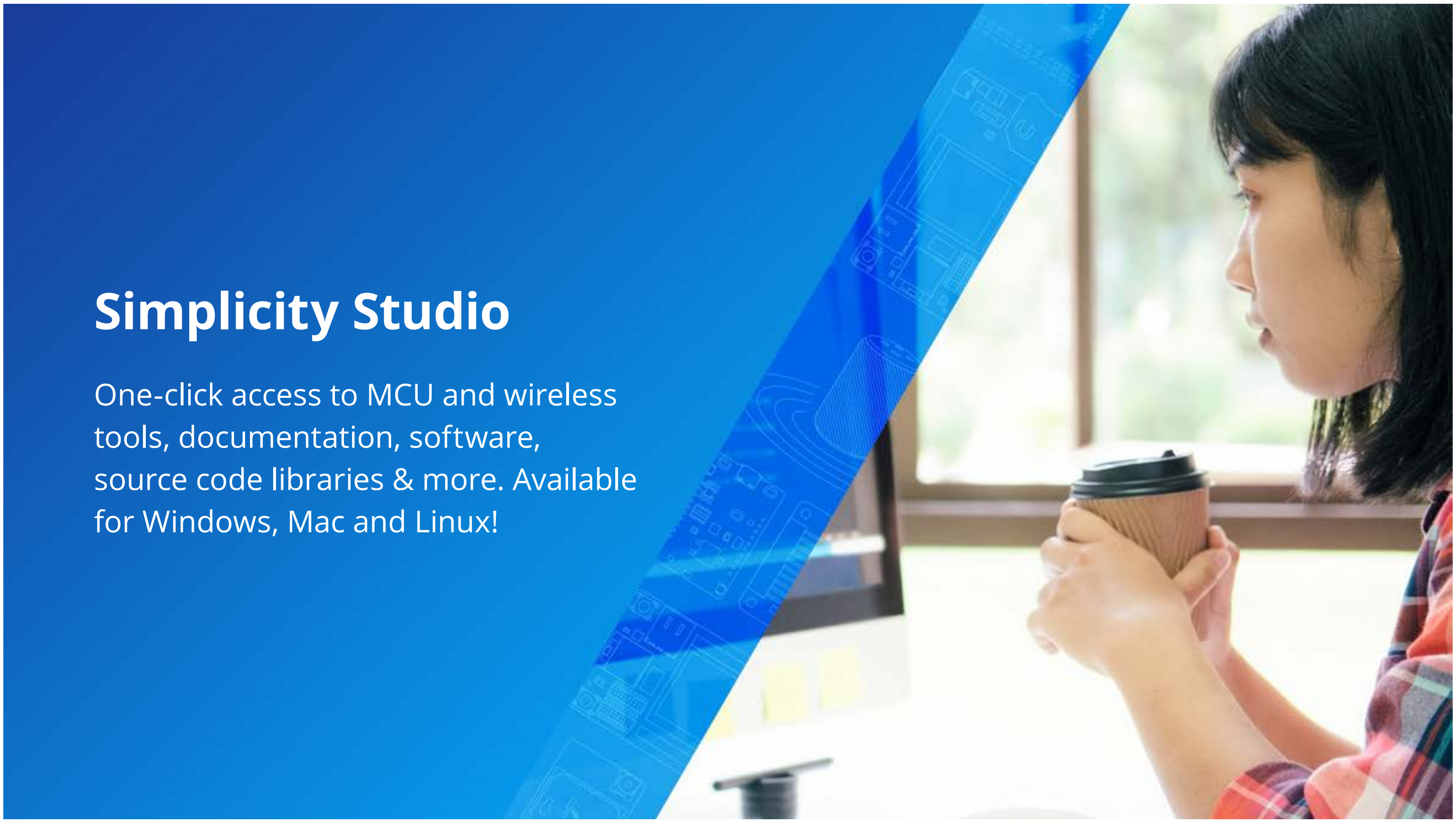
November, 2023

Converted to DITA

Revision 0.1

March, 2023

Initial Version



Simplicity Studio

One-click access to MCU and wireless tools, documentation, software, source code libraries & more. Available for Windows, Mac and Linux!

IoT Portfolio
www.silabs.com/IoT

SW/HW
www.silabs.com/simplicity

Quality
www.silabs.com/quality

Support & Community
www.silabs.com/community

Disclaimer
Silicon Labs intends to provide customers with the latest, accurate, and in-depth documentation of all peripherals and modules available for system and software implementers using or intending to use the Silicon Labs products. Characterization data, available modules and peripherals, memory sizes and memory addresses refer to each specific device, and "Typical" parameters provided can and do vary in different applications. Application examples described herein are for illustrative purposes only. Silicon Labs reserves the right to make changes without further notice to the product information, specifications, and descriptions herein, and does not give warranties as to the accuracy or completeness of the included information. Without prior notification, Silicon Labs may update product firmware during the manufacturing process for security or reliability reasons. Such changes will not alter the specifications or the performance of the product. Silicon Labs shall have no liability for the consequences of use of the information supplied in this document. This document does not imply or expressly grant any license to design or fabricate any integrated circuits. The products are not designed or authorized to be used within any FDA Class III devices, applications for which FDA premarket approval is required or Life Support Systems without the specific written consent of Silicon Labs. A "Life Support System" is any product or system intended to support or sustain life and/or health, which, if it fails, can be reasonably expected to result in significant personal injury or death. Silicon Labs products are not designed or authorized for military applications. Silicon Labs products shall under no circumstances be used in weapons of mass destruction including (but not limited to) nuclear, biological or chemical weapons, or missiles capable of delivering such weapons. Silicon Labs disclaims all express and implied warranties and shall not be responsible or liable for any injuries or damages related to use of a Silicon Labs product in such unauthorized applications.

Trademark Information
Silicon Laboratories Inc.®, Silicon Laboratories®, Silicon Labs®, SiLabs® and the Silicon Labs logo®, Bluegiga®, Bluegiga Logo®, EFM®, EFM32®, EFR, Ember®, Energy Micro, Energy Micro logo and combinations thereof, "the world's most energy friendly microcontrollers", Redpine Signals®, WiSeConnect , n-Link, EZLink®, EZRadio®, EZRadioPRO®, Gecko®, Gecko OS, Gecko OS Studio, Precision32®, Simplicity Studio®, Telegesis, the Telegesis Logo®, USBXpress® , Zentri, the Zentri logo and Zentri DMS, Z-Wave®, and others are trademarks or registered trademarks of Silicon Labs. ARM, CORTEX, Cortex-M3 and THUMB are trademarks or registered trademarks of ARM Holdings. Keil is a registered trademark of ARM Limited. Wi-Fi is a registered trademark of the Wi-Fi Alliance. All other products or brand names mentioned herein are trademarks of their respective holders.

SILICON LABS

Silicon Laboratories Inc.
400 West Cesar Chavez
Austin, TX 78701
USA

www.silabs.com