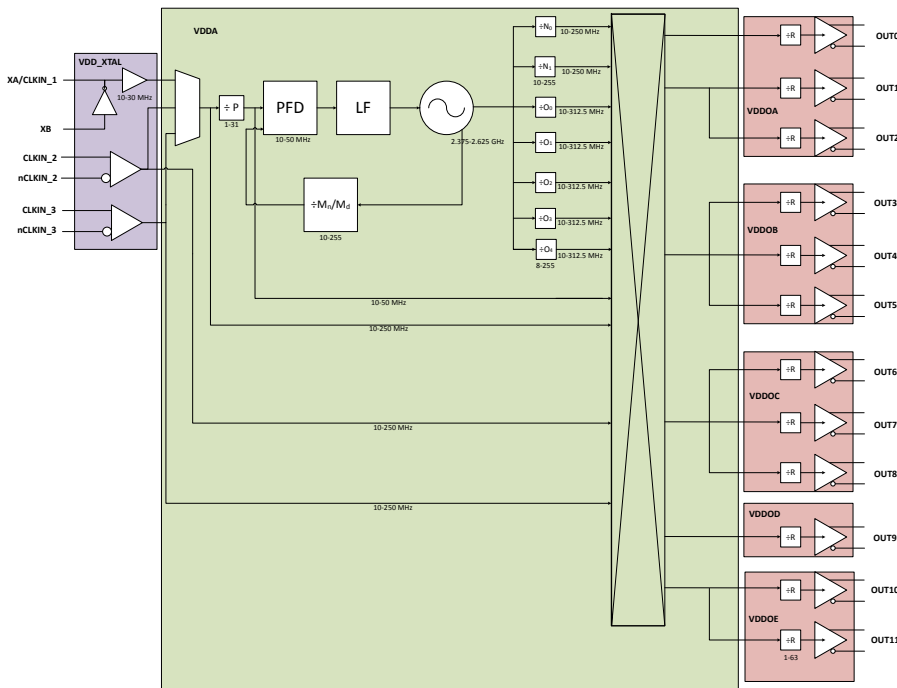


Si5332-AM1/2/3 Automotive Grade Device Reference Manual

The Si5332-AM1/2/3 is a high-performance, low-jitter clock generator capable of synthesizing five independent banks of user-programmable clock frequencies up to 333.33 MHz, while providing up to 12 differential or 24 single-ended output clocks. The Si5332 supports free run operation using an external crystal as well as lock to an external clock signal. The output drivers are configurable to support common signal formats, such as LVPECL, LVDS, HCSL, and LVCMOS. Separate output supply pins allow supply voltages of 3.3 V, 2.5 V, 1.8 V and 1.5 V (CMOS only) to power the multi-format output drivers. The core voltage supply (VDD) accepts 3.3 V, 2.5 V, or 1.8 V and is independent from the output supplies (VDDOs). Using its two-stage synthesis architecture and patented high-resolution Multisynth technology, the Si5332 can generate three fully independent/non-harmonically-related bank frequencies from a single input frequency.



RELATED DOCUMENTS

- Any-Frequency 6/8/12-output programmable clock generators
- Offered in three different package sizes, supporting different combinations of output clocks and user configurable hardware input pins
 - 32-pin QFN, up to 6 outputs
 - 40-pin QFN, up to 8 outputs
 - 48-pin QFN, up to 12 outputs (planned future device)
- Multisynth technology enables any frequency synthesis on any output up to 250 MHz using N dividers.
- Output frequencies up to 333.33 MHz using O dividers.
- Highly configurable output path featuring a cross point mux
- Up to three independent fractional synthesis output paths
- Up to five independent integer dividers
- Down and center spread spectrum
- Input frequency range:
 - External crystal: 16 MHz to 50 MHz
 - Differential clock: 10 MHz to 250 MHz
 - LVCMOS clock: 10 MHz to 170 MHz
- Output frequency range:
 - Differential: 5 MHz to 333.33 MHz
 - LVCMOS: 5 MHz to 170 MHz
- User-configurable clock output signal format per output: LVDS, LVPECL, HCSL, LVCMOS
- Easy device configuration using our [ClockBuilder Pro™](#) (CBPro™) software tool available for download from our web site
- Temperature range: -40 to +105 °C
- Pb-free, RoHS-6 compliant
- For more information, refer to the [Si5332 Automotive Grade Data Sheet](#)

Table of Contents

1. Overview	3
2. Power Supply Sequencing	4
3. Input Clocks	5
3.1 Input Clock Terminations	5
3.1.1 External Crystal	5
3.1.2 External Input Clock on XA Input	5
3.1.3 External Input Clock on CLKIN_x/CLKIN_x#	6
3.2 Calculating Crystal Loading Capacitance	8
4. GPIO	10
5. Output Clock Terminations	11
5.1 DC-Coupled Output Clock Terminations	12
5.2 AC-Coupled Output Clock Terminations	16
6. Programming the Volatile Memory (Registers)	17
6.1 Programming the PLL	18
6.2 Programming the Clock Path	21
6.3 Programming the Output Clock Frequency	23
6.4 Programming the Output Clock Format	25
6.5 Programming for Frequency Select Operations	27
6.6 Programming Spread Spectrum	28
7. Si5332 Pinout and Package Variant	30
8. Recommended Schematic and Layout Practices	32
9. Register Map	33
10. Revision History	54

1. Overview

In addition to clock generation, the input clocks can bypass the synthesis stage, enabling the Si5332 to be used as a high-performance clock buffer or a combination of a buffer and a generator. The Multisynth dividers have two sets of divide ratio registers, an A set and a B set. The active in-use divide ratio can be switched between the A set or B set via external input pin or register control. This feature allows for dynamic frequency shifting at ppb accuracy for applications such as frequency margining. Similar A set and B set divider ratios are available for the integer dividers, but the ratios must be integer related. CBPro supports the configuration and use of A and B divider sets. Spread spectrum is available for any clock output sourced from the Multisynth dividers and is available for use in EMI-sensitive applications, such as PCI Express. The 2-wire I²C bus can be used to control and configure the Si5332. Alternatively, some device features can be controlled by an external pin via CBPro configuration of one of more General Purpose I/O (GPIO) pins. Examples of GPIO pin assigned control are clock input selection, frequency A/B select, spread spectrum enable, output enable, or I²C address select.

2. Power Supply Sequencing

The Si5332 VDD_core voltages are VDD_DIG, VDD_XTAL, and VDDA. These 3 VDD_core pins must all use the **same** voltage. Power supply sequencing between VDD_core and any VDDOx pin is allowed in any order. However, if desiring to minimize the “bring-up” time, it is recommended that VDD_core is powered up first; this ensures that the NVM download is completed first and the I²C master can communicate with the Si5332 slave. The figure below shows the Si5332 device power-up sequencing and expected device behavior. Note that a blank (unconfigured) part will stop and wait to be configured with outputs disabled.

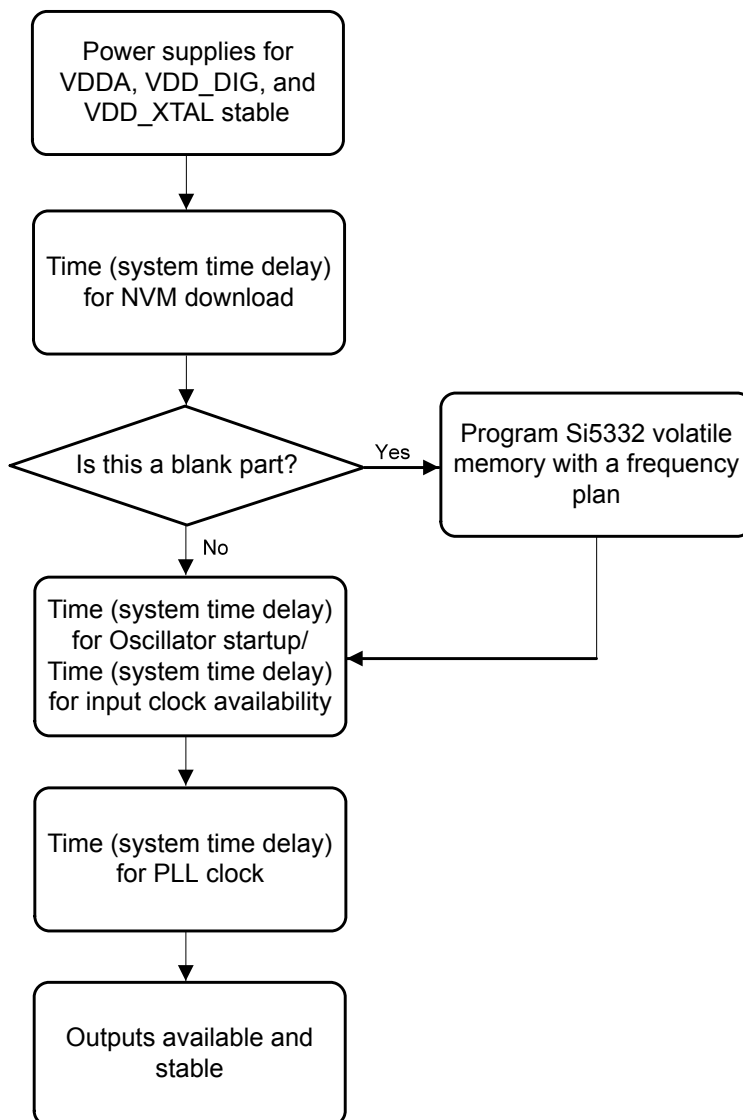


Figure 2.1. Power Supply Sequencing for Si5332

3. Input Clocks

The Si5332 has three input clock nodes: the XA/XB pair, the CLKIN_2/CLKIN_2# pair, and the CLKIN_3/CLKIN_3# pair.

XA/XB supports a crystal input or an external clock input whereas the CLKIN_x/CLKIN_x# pairs support ONLY external clock inputs. The GPIO pins can be set to select the active input clock for the PLL (or the user can set the active input via register writes).

3.1 Input Clock Terminations

Supported input clock sources for the Si5332 are:

1. External crystal attached to the Si5332 XA/XB inputs.
2. External single-ended clock attached to XA (XA/CLKIN1 pin).
3. Externally supplied clock attached to available CLKIN_x/CLKINx# inputs.

3.1.1 External Crystal

An external crystal can be connected to the Si5332's XA/XB inputs as shown below. See Table 5.4 in the Si5332 Data Sheet for crystal specifications when selecting a crystal. Note that the external crystal specifications in Si5332 Data Sheet must be met. A list of recommended AEC-Q200 qualified crystals for the Si5332 can be found in the Silicon Labs document "Recommended Crystal Reference Manual for Si5332, Si5357, and Si5225x Automotive Grade Clock Generators."

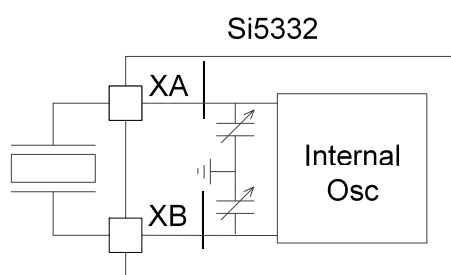


Figure 3.1. External Crystal Connection

3.1.2 External Input Clock on XA Input

The XA input (XA/CLKIN1 pin) can accept an externally supplied, AC-coupled clock with maximum voltage swing of 1 Vpp. See figure below for connection details. The XB pin must be left open with nothing connected. If using this input clock mode, it is suggested to zero out the internal crystal loading capacitance (CL) for best performance."

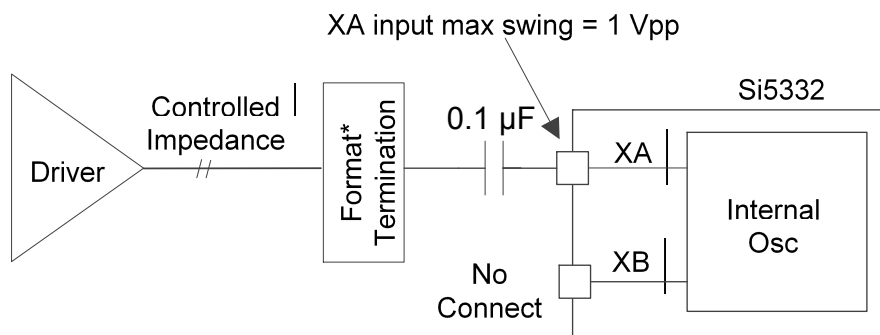


Figure 3.2. External Input Clock on XA Input

3.1.3 External Input Clock on CLKIN_x/CLKIN_x#

When supplying differential input clocks into the CLKIN_x/CLKIN_x# inputs, AC or DC coupling can be used. The figures below show the AC and DC coupled differential input clock connection to the Si5332 clock inputs. (There are some restrictions to observe when using DC coupled input clocks as described further below.) The input clock Format Termination shown in below figures is dependent on the driver's termination requirements. The Si5332 clock inputs are high impedance inputs and the clock driven into the Si5332 must meet the Si5332 Data Sheet's specified electrical requirements. When using differential input clocks, the respective Si5332 input must be configured as a differential input using CBPro.

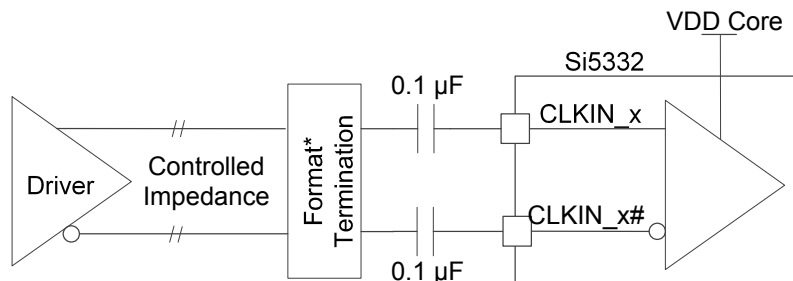


Figure 3.3. AC-Coupled Differential Input Clock (LVDS, LVPECL, HCSL, CML, etc.)

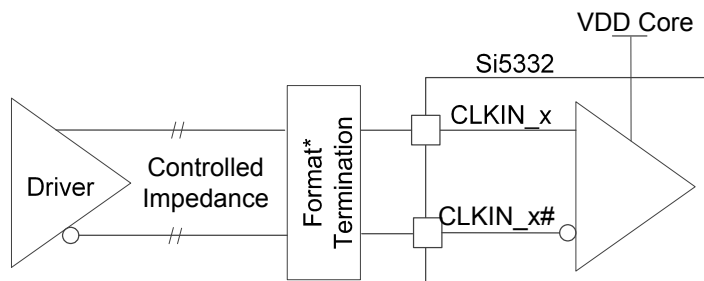


Figure 3.4. DC-Coupled Differential Input Clock

To determine if a specific DC-coupled differential input clock arrangement is supported, refer to the table below.

Table 3.1. Si5332 Input Clock Coupling Restrictions (AC or DC)

Format	VDD_Core		
	3.3 V	2.5 V	1.8 V
LVDS 3.3 V/2.5 V	AC or DC	AC only	AC only
LVDS 1.8 V	AC or DC	AC only	AC only
LVPECL 3.3 V/2.5 V	AC or DC	AC only	AC only
HCSL	AC or DC	AC or DC	AC only
CML	AC only	AC only	AC only
LVC MOS	AC only	AC only	AC only

Note:

1. For DC-coupled, input clock peak voltage must not exceed VDD_Core and minimum voltage must not be below GND.
2. For AC-coupled, peak swing must not exceed VDD_Core.

The figure below shows how to connect single-ended input clocks, such as LVCMOS. The single-ended clock must be connected to the positive CLKIN input as shown below.

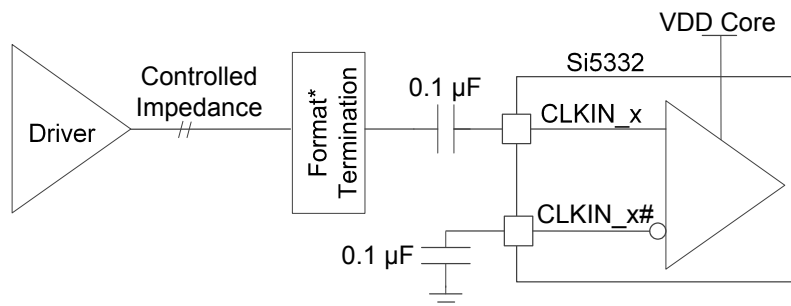


Figure 3.5. AC-Coupled Single-Ended Input Clock (LVCMOS)

For AC-coupled single-ended input clocks (such as LVCMOS) the V_{swing} of the clock must be limited to the maximum VDD_Core voltage. (VDD_Core is defined as the following group of VDD supply pins: VDD_DIG , $VDDA$, and VDD_XTAL .) The Input clock format termination is dependent on the driver format used and is usually specified by the driving device and/or industry standard clock format specification.

For example, in the case of using a LVCMOS input clock, the driving device may recommend a series termination resistor. When using LVCMOS input clocks the Si5332 input must be configured in LVCMOS mode in CBPro. The single-ended CLKIN input of Si5332 is a high impedance input.

3.2 Calculating Crystal Loading Capacitance

Crystals will resonate at their specified frequency (i.e., be “on-frequency”) if the capacitive loading across the crystal’s terminals is the same as specified by the crystal manufacturer’s loading capacitance (CL) specification. The total loading capacitance presented to the crystal must factor in all capacitance sources such as parasitic “stray” capacitance as well as added loading capacitance. Stray capacitance comes from sources like PCB traces, capacitive coupling to nearby components, as well as any stray capacitance within the device itself. For “on-frequency” oscillator operation, all capacitance sources must be factored into determining the correct additional capacitance to be presented to the crystal to match its required CL.

The Si5332 contains variable **internal** loading capacitors (CL_{VAR}) to provide the necessary added crystal matching capacitance such that external matching capacitors are not needed. The figure below shows the Si5332’s internal variable capacitance and the two main sources of stray loading capacitance, PCB stray ($CL_{S_{EXT}}$) and Si5332 device stray capacitance on XA/XB input pins ($CL_{S_{INT}}$).

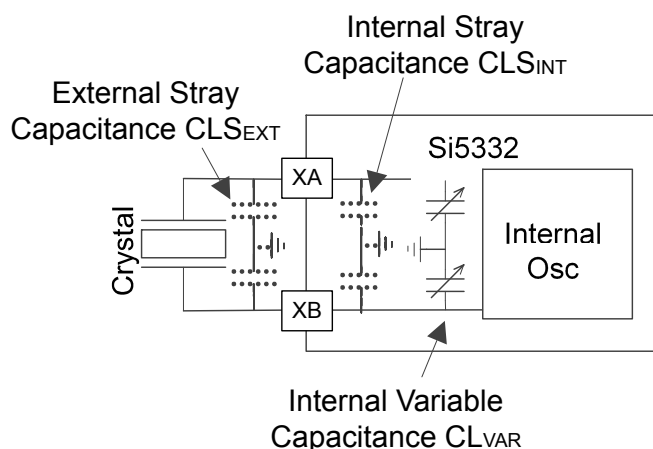


Figure 3.6. Sources of Crystal Loading Capacitance

Using the Si5332’s internal variable loading capacitors (CL_{VAR}), the crystal’s required CL can be matched by adding capacitance to the external stray ($CL_{S_{EXT}}$) and internal device capacitance ($CL_{S_{INT}}$) to match the crystal’s requirements. A value for CL_{VAR} must be selected such that:

$$\text{Required Crystal CL} = CL_{VAR} + CL_{S_{INT}} + CL_{S_{EXT}}$$

Or rearranged:

$$CL_{VAR} = \text{Crystal CL} - CL_{S_{INT}} - CL_{S_{EXT}} \quad \text{Equation 1}$$

Note: the required Crystal CL must be greater than or equal to the total stray capacitance quantity ($CL_{S_{INT}} + CL_{S_{EXT}}$) or the crystal can’t be used as CL_{VAR} is always ≥ 0 .

For the following example, a Crystal CL value of 10 pF will be used. The internal stray capacitance ($CL_{S_{INT}}$) of the Si5332 is 2.4 pF. External PCB stray capacitance ($CL_{S_{EXT}}$) is usually in the order of 2-3 pF given a reasonably compact layout. The Si5332 EVB external stray capacitance is ~ 2.75 pF. Given these example values, the required CL_{VAR} can be calculated as shown below, using Equation 1.

$$CL_{VAR} = 10 \text{ pF} - 2.4 \text{ pF} - 2.75 \text{ pF} = 4.85 \text{ pF} \quad \text{Using Equation 1}$$

As shown in the figure above, the internal variable loading capacitor (CL_{VAR}) is implemented by using **two** capacitors **in series**: one connected to the XA pin (CL_{XA}) and one to the XB pin (CL_{XB}) of the Si5332. The CL_{VAR} value is therefore determined by the values of CL_{XA} and CL_{XB} .

For capacitors in series, if we keep $CL_{XA} = CL_{XB}$, we can simply double the value of CL_{VAR} to arrive at the correct CL_{XA} and CL_{XB} value. (This is a technique commonly used for calculating crystal loading capacitors when external loading capacitors are used.) Now, CL_{XA} and CL_{XB} values are simply:

$$CL_{XA} = CL_{XB} = (2 \times CL_{VAR}) = 2 \times 4.85 \text{ pF} = 9.7 \text{ pF} \quad \text{Equation 2}$$

Substituting Equation 1 for CL_{VAR} into Equation 2 will solve for CL_{XA}/CL_{XB} in single equation form:

$$CL_{XA} = CL_{XB} = 2 \times (\text{Crystal } CL - CL_{int} - CL_{ext}) \quad \text{Equation 3}$$

Equation 3 can now be used to determine the CL_{XA} and CL_{XB} values needed to set internal CL_{XA} and CL_{XB} register values. The range of CL_{XA} and CL_{XB} values the Si5332 can support is 0 to 38.395 pF. CL_{XA} and CL_{XB} values less than 0 cannot be implemented and any values greater than 38.395 pF cannot be implemented using internal capacitors alone. Remember, CL_{XA} and CL_{XB} are not derived simply from the crystal's CL because internal and external stray capacitance must be factored into the calculations.

Once the required CL_{XA} and CL_{XB} have been determined using Equation 3, use the following set of formulas to calculate the required register values to implement the desired CL_{XA} and CL_{XB} .

If ($CL_{XA/XB} \leq 30.555$ pF, then:

- Register `xosc_cint_ena` = 0
- Register `xosc_ctrim_xin` = Round to nearest integer ($CL_{XA} / 0.485$)
- Register `xosc_ctrim_xout` = Round to nearest integer ($CL_{XB} / 0.485$)

If ($30.555 \text{ pF} < CL_{XA/XB} \leq 38.395 \text{ pF}$, then:

- Register `xosc_cint_ena` = 1
- Register `xosc_ctrim_xin` = Round to nearest integer ($(CL_{XA} - 7.84) / 0.485$)
- Register `xosc_ctrim_xout` = Round to nearest integer ($(CL_{XB} - 7.84) / 0.485$)

To summarize, use Equation 3 to calculate CL_{XA}/CL_{XB} , then use the above set of formulas to calculate register values to implement CL_{XA}/CL_{XB} in the Si5332. Note that external loading capacitors can be used to supplement the internal loading capacitors if necessary.

Note: Your unique PCB assembly's stray capacitance value plays a role in determining correct internal capacitor settings and, consequently, the crystal's frequency of oscillation. Small differences in actual board stray capacitance value from the value you use in the above equations will result in the crystal oscillating slightly off-frequency. Significant capacitance differences can result in significant frequency error.

4. GPIO

Si5332 features universal General Purpose Input/Output (GPIO) hardware pins whose functions **must** be programmed in NVM to assume a pre-defined function in ClockBuilder Pro during custom configuration file development.

Table 4.1. Available GPIO Pin Functions

Function Name	In/Out	Description
OEn (up to 7)	In	Output enable(s) for selected output(s)
SSEnN (up to 2)	In	Spread spectrum enable for outputs derived from N0 / N1 divider
FSn (up to 7)	In	Frequency select(s) for Dividers N0, N1, O0-O4
CLKIN_SEL [1:0]	In	Input clock select pins (requires use of 2 GPIOs)
I2C_ADDR0	In	Use to add 1 to I2C device base address but is logically Or'ed with I2C address bit 0. If device I2C base address is even, assertion will add 1 to base address. If device I2C base address is odd, assertion or de-assertion will not change base address.
LOS	Out	Loss of Input Clock Signal (Input clock or xtal), active low open drain output. Requires external pull-up resistor to VDD_CORE.
RESET	In	Device reset, active high input

Note: Creation of, or changes to, device GPIO pin functionality is only available when using customized Si5332 orderable part numbers (OPN) generated through CBPro and then either factory programmed, or field programmed using the CBPro Field Programming Dongle. GPIO pin functionality can be evaluated/tested on a Si5332 EVB by downloading a valid CBPro configuration into the EVB, and asserting the GPIO pins on the EVB. New GPIO configurations, or changes to existing GPIO configurations, are **not** available through Si5332 I²C register programming. Blank Si5332 devices have no GPIO configuration loaded and require burning a configuration into device NVM with the Field Programming Dongle to configure any GPIO functionality.

CBPro will allow user configuration of the available GPIOs according to the functionality restrictions of the function selected. Some functions can share a pin, some can not. For example, a single pin can control OE of one or more outputs. Some functions require two pins, such as CLKIN_SEL0 and CLKIN_SEL1. Some functions are mutually exclusive, for instance FS_x functions can't share a pin with OE. For all Si5332 devices, CBPro uses a default I²C base address of 6Ah unless the base address is changed by the user.

5. Output Clock Terminations

The Si5332 supports output formats of LVCMOS, LVDS, LVPECL, and HCSL with some additional format specific features. Each output driver is individually programmable to any of the supported formats by use of the following registers.

Table 5.1. Output Format Related Register Fields

outx_mode:	- Sets the mode of the driver. See Table 5.2 OUTx_Mode vs Output Formats on page 11 .
outx_cmos_inv:	- Sets the inversion state of each side of the dual output driver when in CMOS driver format.
outx_cmos_slew:	- Sets the slew rate of the CMOS driver.
outx_cmos_str:	- Sets the output impedance of the CMOS driver (25Ω / 50Ω).

Table 5.2. OUTx_Mode vs Output Formats

OUTx_MODE	Driver Mode
0	off
1	CMOS on positive output only
2	CMOS on negative output only
3	dual CMOS outputs
4	2.5V/3.3V LVDS
5	1.8V LVDS
6	2.5V/3.3V LVDS fast
7	1.8V LVDS fast
8	HCSL 50 Ω (external termination)
9	HCSL 50 Ω (internal termination)
10	HCSL 42.5 Ω (external termination)
11	HCSL 42.5 Ω (internal termination)
12	LVPECL
13	Reserved
14	Reserved
15	Reserved

The recommended termination for each output format is shown in these figures: [Figure 5.1 LVCMOS Termination, Option 1 on page 12](#) and [Figure 5.2 LVCMOS Termination, Option 2 on page 12](#).

5.1 DC-Coupled Output Clock Terminations

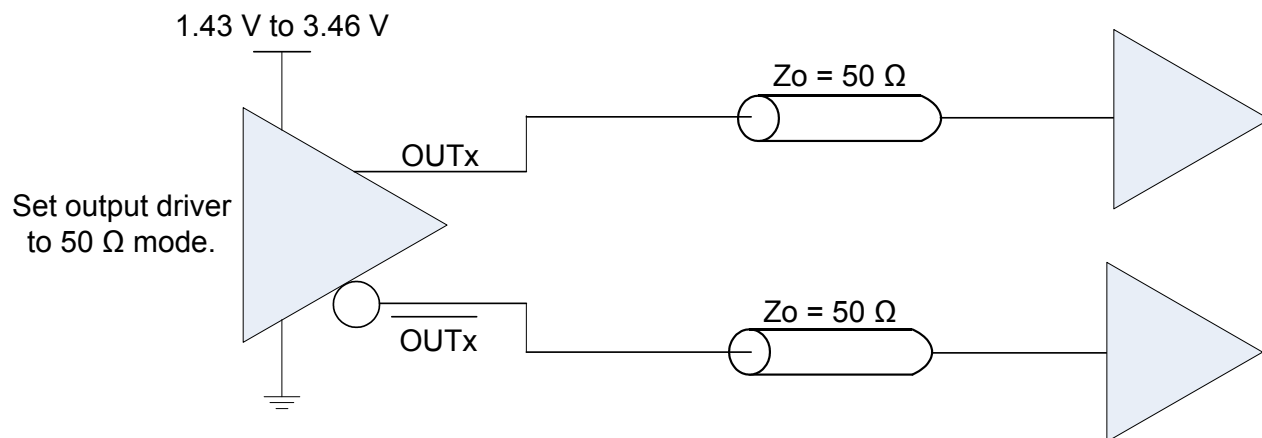


Figure 5.1. LVC MOS Termination, Option 1

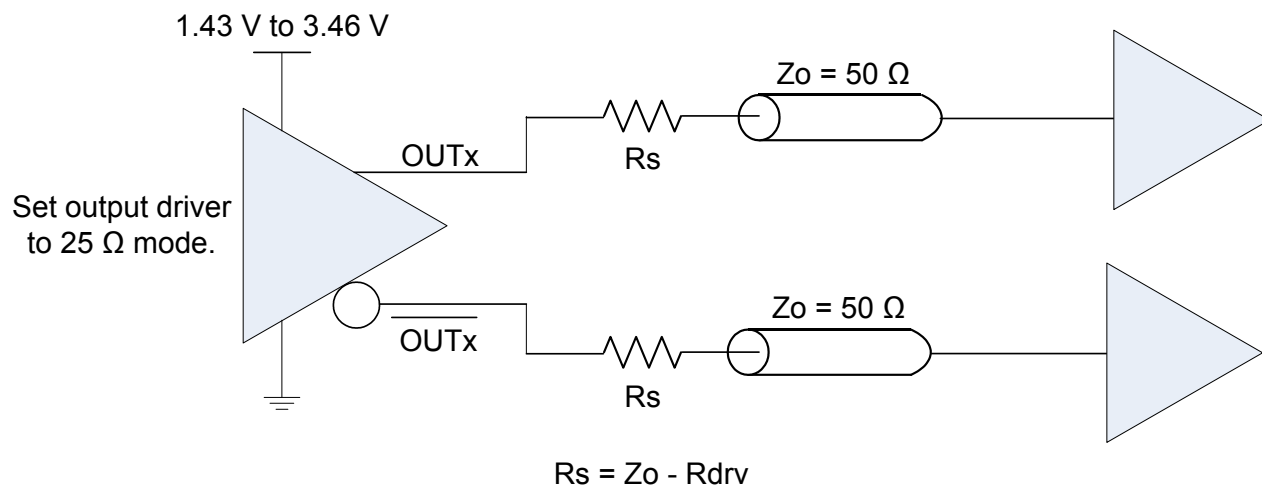


Figure 5.2. LVC MOS Termination, Option 2

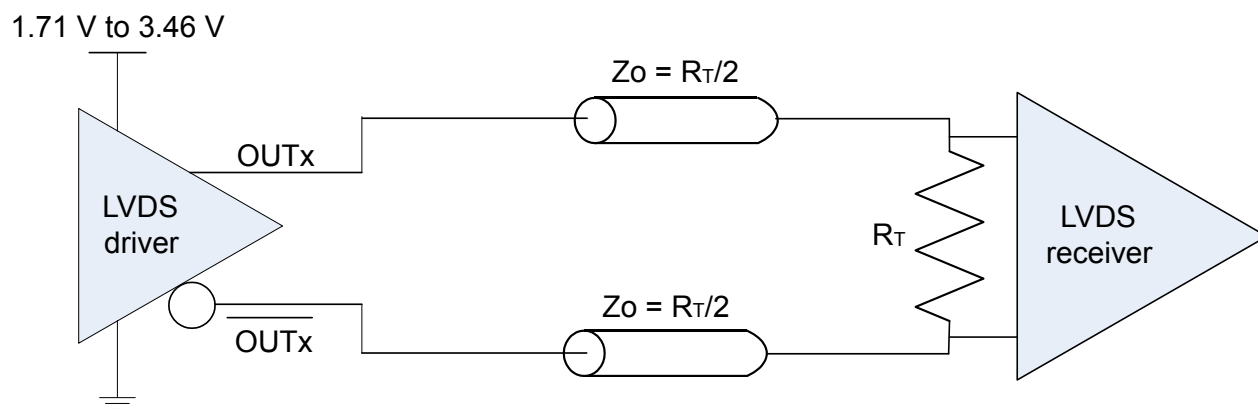


Figure 5.3. LVDS/LVDS Fast Termination, Option 1

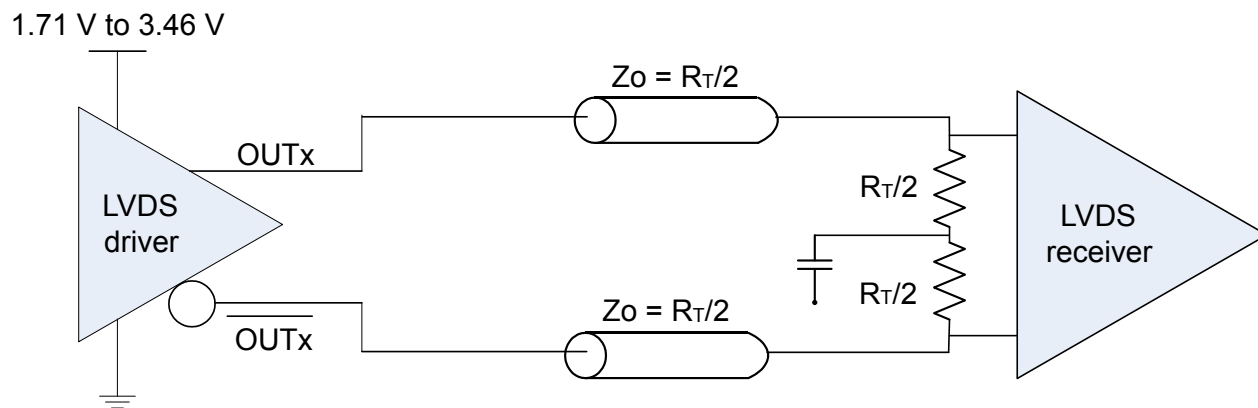


Figure 5.4. LVDS/LVDS Fast Termination, Option 2

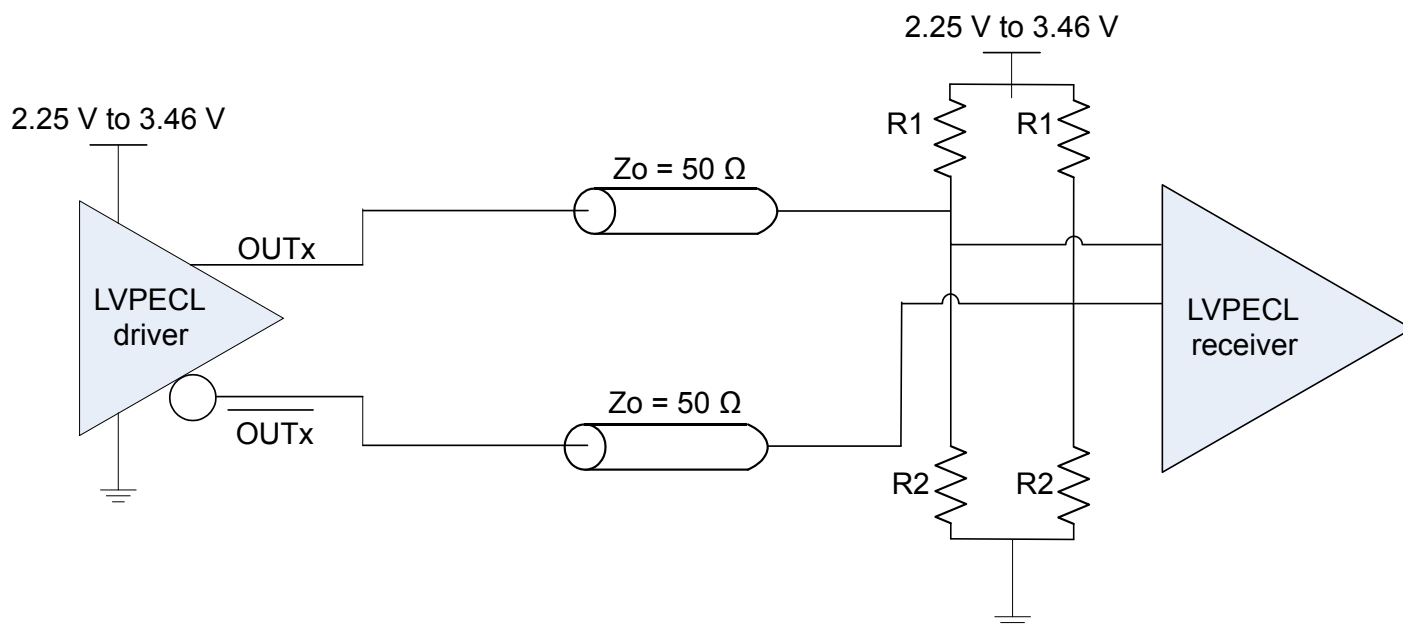


Figure 5.5. LVPECL Termination, Option 1

Table 5.3. LVPECL Termination, Option 1

VDD Standard	Resistance	Resistance Value
2.5	R1	250
	R2	62.5
3.3	R1	125
	R2	84

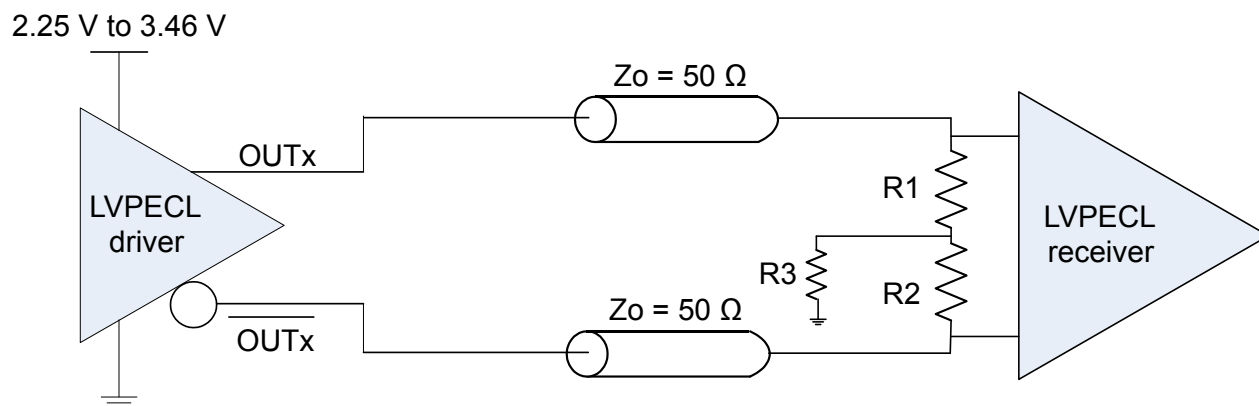


Figure 5.6. LVPECL Termination, Option 2

Table 5.4. LVPECL Termination, Option 2

VDD Standard	Resistance	Resistance Value
2.5	R1	50
	R2	50
	R3	29.5
3.3	R1	50
	R2	50
	R3	54 or 0

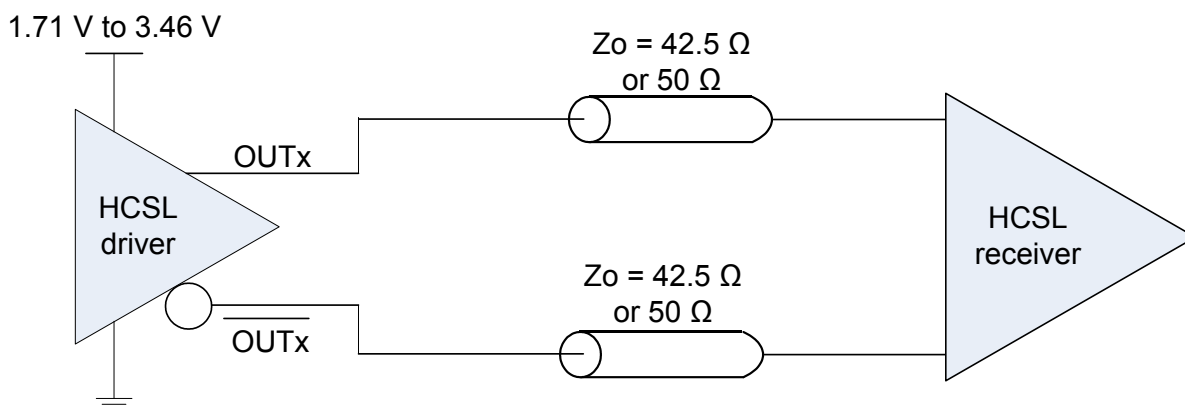


Figure 5.7. HCSL Internal Termination Mode

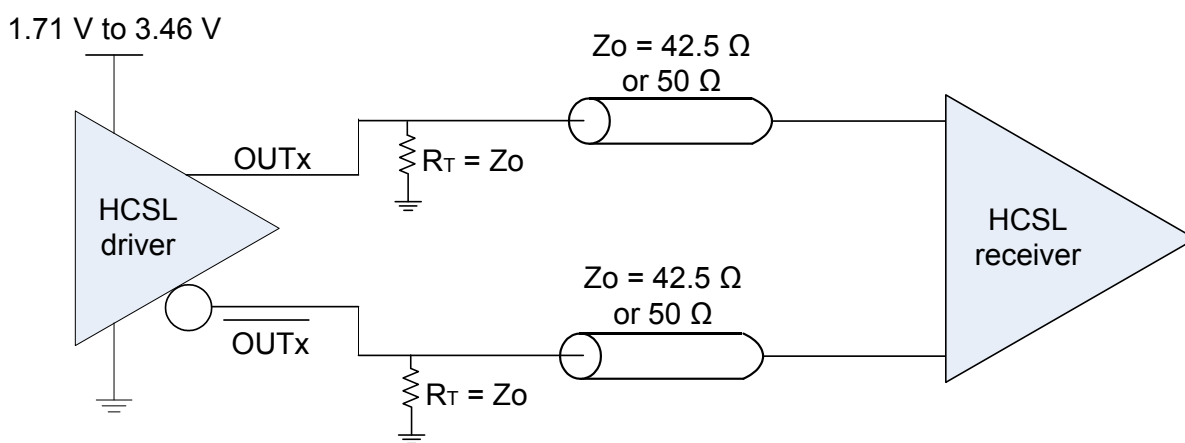


Figure 5.8. HCSL External Termination Mode

5.2 AC-Coupled Output Clock Terminations

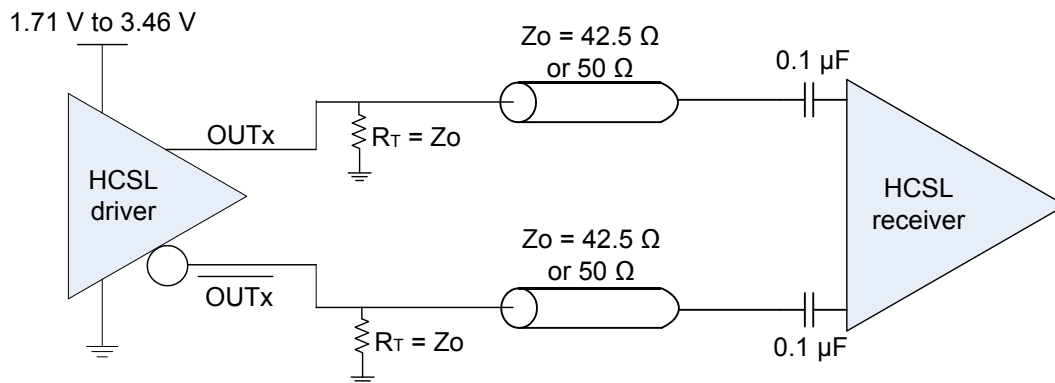


Figure 5.9. HCSL External Termination Mode

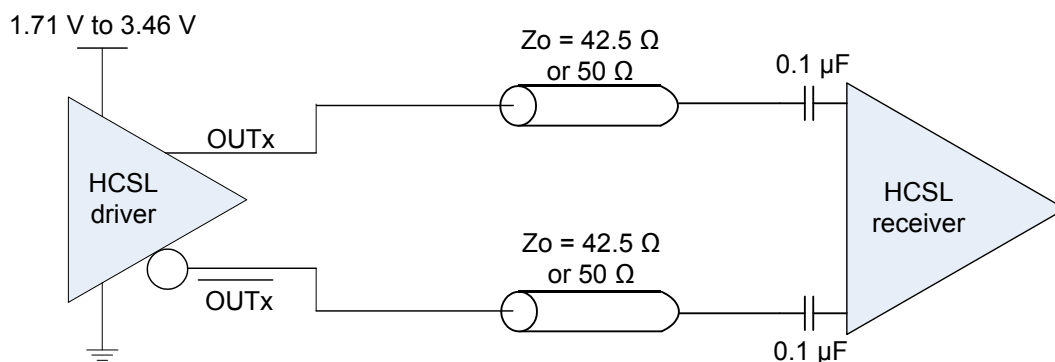


Figure 5.10. HCSL Internal Termination Mode

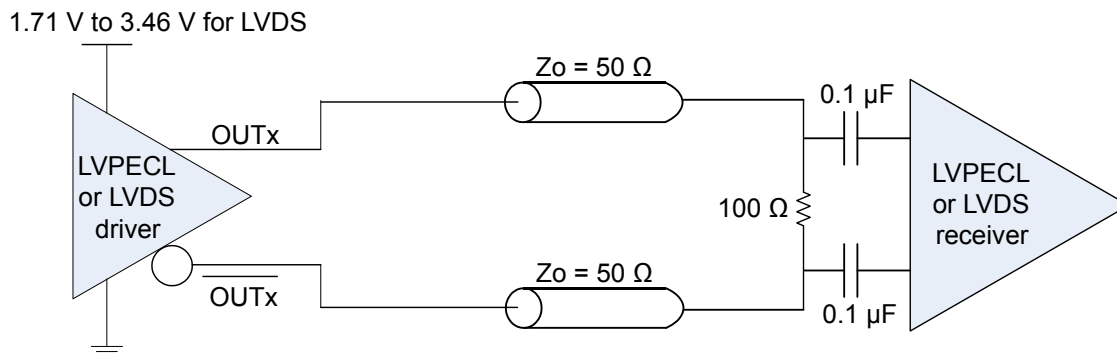


Figure 5.11. LVDS Termination

The terminations shown in [Figure 5.3 LVDS/LVDS Fast Termination, Option 1 on page 13](#) through [Figure 5.6 LVPECL Termination, Option 2 on page 14](#) can also be converted by adding DC-blocking capacitances right before the receiver pins. However, the recommendation shown in [Figure 5.11 LVDS Termination on page 16](#) is the simplest way to realize AC-coupling (i.e., the least number of components) and the recommended circuit for AC-coupled termination circuits.

6. Programming the Volatile Memory (Registers)

The process described in this section is very complex to implement and/or calculate. A device register solution is most easily calculated using CBPro and then exporting the resulting register file to simply download into the device. **Using CBPro to craft your register setting solution is the most highly recommended approach.** The following descriptions of register value calculations are for reference only.

The volatile memory can be programmed to set up the various functions necessary to realize a PLL function, a clock output to clock input relationship and can be used to monitor input clock that controls the PLL. The top-level block diagram is repeated here to refresh the various limits and possibilities that are necessary for the calculations below.

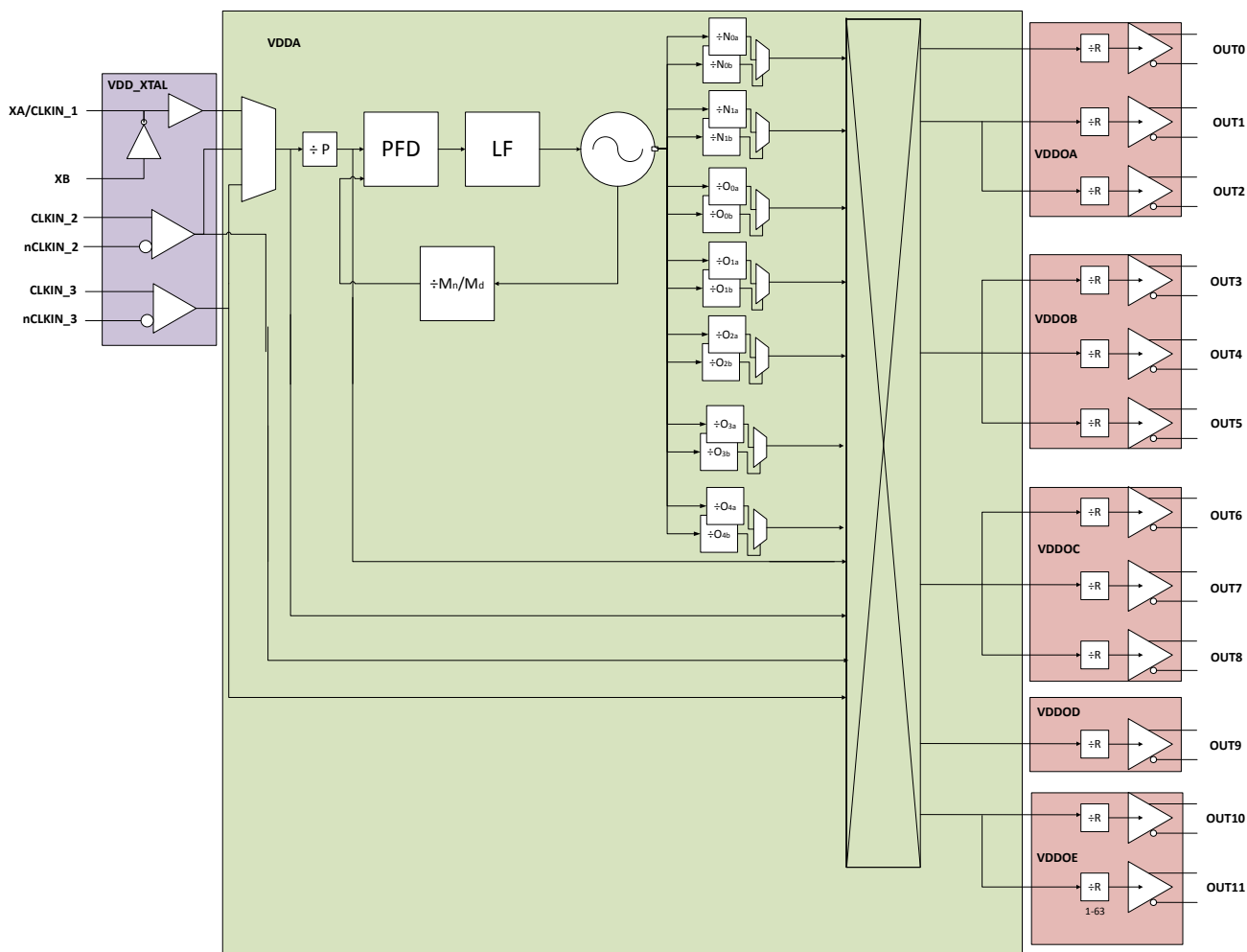


Figure 6.1. Top-Level Block Diagram

6.1 Programming the PLL

The PLL programming involves three distinct constraints:

1. The minimum and the maximum frequencies possible for the PFD (Phase Frequency Detector) at lock. This is set by the reference frequency that sets the input divider P and the active input clock as selected by the IN SEL pins or registers.
2. The VCO frequency that is set by feedback divider (Mn/Md) and the PFD frequency also has a limited range that is unique to Si5332.
3. The PLL closed loop transfer function characterized by its loop band width and peaking is set by programming the loop parameters.

The table below lists the constraints for the PLL reference frequency and the VCO frequency. The PLL reference frequency ($p11RefFreq$) and the VCO frequency ($vcoFreq$) are related by the equation below:

$$vcoFreq = p11RefFreq \times \left(\frac{Mn}{Md} \right)$$

For a given plan, the $p11RefFreq$ can be readily solved as it is derived from the input clock frequency. To get to this optimization, the “active” input to the PLL must be selected from the XA/XB, CLKIN_1, CLKIN_2, in 1 p/m input clocks using either the IMUX_SEL register field or the CLKIN_SEL pins (if CKIN_SEL pins are available in the custom part that was chosen to reprogrammed). $p11RefFreq$ is given by the In-Freq (active clock input frequency) and P as:

$$P11RefFreq = \frac{InFreq}{P}$$

Table 6.1. Constraints for PLL Reference Frequency and VCO Frequency

Field Name	Value	Description
$p11MinRefFreq$	10 MHz	The minimum reference frequency the PLL can tolerate
$p11MaxRefFreq$	50 MHz	The maximum reference frequency the PLL can tolerate
$vcoCenterFreq$	2.5 GHz	The center frequency of the VCO's tuning range
$vcoMinFreq$	2.375 GHz	The minimum frequency of the VCO's tuning range
$vcoMaxFreq$	2.625 GHz	The maximum frequency of the VCO's tuning range

List all required output frequencies, F_{xy} , in groups denoted by G_x , where $x = 0,1,2,3,4,5$ and $y = a,b,c$. This grouping is done such that frequencies related to each other by rational fractions of integers between 1 and 63 are in that group. For example, $100 \text{ MHz}/80 \text{ MHz} = 5/4$ is a rational fraction. Each group G_x is associated with a single output voltage supply driver inside Si5332 and is shown in [Table 6.2 Output Frequency Variables Grouping and Mapping to Actual Output Pins on page 18](#). The table also shows the output frequency symbol F_{xy} mapped to the output name in the Si5332 pin descriptions. The integer O-dividers are denoted by $hsdiv$. Each O_i divider maps to a $hdiv_i$ in the solver where i is an integer between 0 and 4. Similarly, the two Multisynth N-dividers, N_j map to ID_j and $j = 0$ or 1. The constraints for these divider values are listed in [Table 6.3 Constraints for \$hsdiv\$ and \$id\$ on page 19](#).

Table 6.2. Output Frequency Variables Grouping and Mapping to Actual Output Pins

Si5332 12-Output Part Output Pair (Future Device)	Si5332 8-Output Part Output Pair	Si5332 6-Output Part Output Pair	Output Frequency Variable for Solver	The Output Frequency Group
OUT0	OUT0	OUT0	F_{0A}	G_0
OUT1	OUT1	OUT1	F_{1A}	G_1
OUT2			F_{1B}	G_1
OUT3	OUT2	OUT2	F_{2A}	G_2
OUT4	OUT3		F_{2B}	G_2
OUT5			F_{2C}	G_2
OUT6	OUT4	OUT3	F_{3A}	G_3

Si5332 12-Output Part Output Pair (Future Device)	Si5332 8-Output Part Output Pair	Si5332 6-Output Part Output Pair	Output Frequency Variable for Solver	The Output Frequency Group
OUT7	OUT5		F _{3B}	G ₃
OUT8			F _{3C}	G ₃
OUT9	OUT6	OUT4	F _{4A}	G ₄
OUT10	OUT7	OUT5	F _{5A}	G ₅
OUT11			F _{5B}	G ₅

Table 6.3. Constraints for `hSDiv` and `id`

Field Name	Value	Description
<code>hSDivMinDiv</code>	8	The minimum divide value that the HSDIV can support
<code>hSDivMaxDiv</code>	255	The maximum divide value that the HSDIV can support
<code>idMinDiv</code>	10	The minimum divide value that the ID can support
<code>idMaxDiv</code>	255	The maximum divide value that the ID can support

Each output frequency F_{outxy} is given by:

For the integer dividers (O0 – O4):

$$F_{outxy} = \frac{vcoFreq}{hSDiv_j \times R_{xy}}$$

For the fractional dividers (N0, N1):

$$F_{outxy} = \frac{vcoFreq}{N_{xj} \times R_{xy}}$$

An `hSDiv` or `N` divider is common for output frequencies grouped in a given G_x . Given these constraints, the solver must first choose a $PLLRefFreq$ that satisfies the constraints in [Table 6.4 Loop BW Options on page 20](#). The search for $vcoFreq$ can be broken down into the following steps:

- From the output frequency set, form a set of “M” non-equal frequencies. Group the (N-M) equal frequencies into the same “x” in F_{outxy} grouping
- Now form $M C_2$ groups of {M-2} output frequencies. Find the LCM of each group and find an integer “l” that can such that:
 - $vcoFreq = l * LCM$ can meet the constraint for $vcoFreq$ in [Table 6.1 Constraints for PLL Reference Frequency and VCO Frequency](#).
 - List the “L” groups that provide a legal $vcoFreq$, i.e., a $vcoFreq$ that satisfies the condition in step a.
 - Choose the $vcoFreq$ that has most number of performance critical clocks that do not need “spread spectrum” clocking as part of the “M-2” output clocks.

Given that $vcoFreq$, calculate the fractional feedback divider as:

$$\frac{Mn}{Md} = \frac{vcoFreq}{pllRefFreq}$$

The Mn/Md fraction is represented in register fields `IDPA_INTG`, `IDPA_RES` and `IDPA_DEN`

$$IDPA_INTG = \text{floor} \left(\frac{128 \times vcoFreq}{pllRefFreq} \right)$$

$$\frac{IDPA_RES}{IDPA_DEN} = \left(\frac{128 \times vcoFreq}{pllRefFreq} \right) - IDPA_INTG$$

As can be seen from the above equations, the ratio `IDPA_RES/IDPA_DEN` will always be less than 1.

Note: All these register fields are 15 bits wide. Therefore, the fraction will need to truncate up to this precision. This section fully determines the VCO frequency, the P-divider and the feedback divider for this plan given the choice of using O-dividers {HSDIV} for M-2 output clocks and N-dividers {ID} for two output clocks.

The next step will be to determine the closed loop response that is required from the PLL. The table below lists the different loop BW settings possible and the register field value that will enable that loop BW setting:

Table 6.4. Loop BW Options

PLL_MODE	Loop Bandwidth (kHz)	PLL. Ref. Freq. Min (MHz)	PLL. Ref. Freq. Max. (MHz)
0		ILLEGAL IF PLL MODE IS ENABLED	
1	350	10	15
2	250	10	15
3	175	10	15
4	500	15	30
5	350	15	30
6	250	15	30
7	175	15	30
8	500	30	50
9	350	30	50
10	250	30	50
11	175	30	50

This algorithm will result in a final solution for a VCO frequency, vcoFreq, that can then be used to calculate the O-divider, N-divider, and R-divider values needed to derive each output frequency, Foutxy.

6.2 Programming the Clock Path

Given a valid VCO frequency for the M unique frequencies, segregate the N-M equal frequencies into outputs from each group Gx in [Table 6.2 Output Frequency Variables Grouping and Mapping to Actual Output Pins on page 18](#). When arranging outputs, care must be taken to minimize crosstalk (without violating the constraints imposed from the grouping of output frequencies into the VDDO “banks”). Whenever several high frequencies, fast rise time, large amplitude signals are all close to one another, the laws of physics dictate that there will be some amount of crosstalk. The jitter of the Si5332 is low, and, therefore, crosstalk can become a significant portion of the final measured output jitter. Some of the source of the crosstalk will be the Si5332 and some will be introduced by the PCB. For extra fine tuning and optimization in addition to following the usual PCB layout guidelines, crosstalk can be minimized by modifying the arrangements of different output clocks:

1. Avoid adjacent frequency values that are close. A 155.52 MHz clock should not be next to a 156.25 MHz clock. If the jitter integration bandwidth goes up to 20 MHz, then keep adjacent frequencies at least 20 MHz apart.
2. Adjacent frequency values that are integer multiples of one another are okay and these outputs should be grouped accordingly.
3. Unused outputs can be used to separate clock outputs that might otherwise interfere with one another. If some outputs have tight jitter requirements while others are relatively loose, rearrange the clock outputs so that the critical outputs are the least susceptible to crosstalk. These guidelines typically only need to be followed by those applications that wish to achieve the highest possible levels of jitter performance. Because CMOS outputs have large pk-pk swings and do not present a balanced load to the VDDO supplies, CMOS outputs generate much more crosstalk than differential outputs. For this reason, CMOS outputs should be avoided whenever possible. When CMOS is unavoidable, even greater care must be taken with respect to the above guidelines.

An output multiplexer (output mux) or crosspoint mux needs to be programmed such that each group Gx is set to the correct O-divider, N-divider, or input clock (in the case of buffering). Each output, Foutxy, has this common divider or input clock reference that needs to be set. The multiplexer setting that routes the correct divider/clock source to the correct group is shown in the following table.

Table 6.5. Output Mux (Crosspoint Mux) Settings

Register Field	Description
omuxx_sel0	Selects output mux clock for output clocks in group Gx: 0 = PLL reference clock before pre-scaler 1 = PLL reference clock after pre-scaler 2 = Clock from input buffer 0 3 = Clock from input buffer 1
omuxx_sel1	Selects output mux clock for output clocks in group Gx: 0 = HSDIV0 1 = HSDIV1 2 = HSDIV2 3 = HSDIV3 4 = HSDIV4 5 = ID0 6 = ID1 7 = Clock from omux1_sel0

The final steps will be to program the *h*sd_{div} and *i*d (N) dividers. The equations below show the relationship between *h*sd_{div}, *i*d divider values with their associated output frequency. They also show the register fields that need to be programmed to set up the divider values correctly. The register field and the divider value are both denoted by:

$$h\text{sd}ivx_div = \frac{vcoFreq}{F_{outx} \times R_x}$$

The *i*d (i.e. N) dividers are calculated as below:

$$i\text{d}x = \frac{vcoFreq}{F_{outx} \times R_x}$$

The ida fraction is represented in register fields IDPA_INTG, IDPA_RES and IDPA_DEN:

$$IDxA_INTG = \text{floor} \left(\frac{128 \times vcoFreq}{Foutxa \times Rxa} \right)$$
$$\frac{IDxA_RES}{IDxA_DEN} = \left(\frac{128 \times vcoFreq}{Foutxa \times Rxa} \right) - IDxA_INTG$$

6.3 Programming the Output Clock Frequency

The R_{xy} register fields are programmed as shown in the table below. This last step completes the settings of all dividers that will result in the frequency plan. When a valid divider solution space cannot be determined, that frequency plan is not realizable in the Si5332.

Table 6.6. Rxy to Register Field Mapping

Divider Value	Register Field	Description
R0A	OUT0_DIV	Driver divider ratio. 0 = disabled 1–63 = divide value
R1A	OUT1_DIV	Driver divider ratio. 0 = disabled 1–63 = divide value
R1B	OUT2_DIV	Driver divider ratio. 0 = disabled 1–63 = divide value
R2A	OUT3_DIV	Driver divider ratio. 0 = disabled 1–63 = divide value
R2B	OUT4_DIV	Driver divider ratio. 0 = disabled 1–63 = divide value
R2C	OUT5_DIV	Driver divider ratio. 0 = disabled 1–63 = divide value
R3A	OUT6_DIV	Driver divider ratio. 0 = disabled 1–63 = divide value
R3B	OUT7_DIV	Driver divider ratio. 0 = disabled 1–63 = divide value
R3C	OUT8_DIV	Driver divider ratio. 0 = disabled 1–63 = divide value
R4A	OUT9_DIV	Driver divider ratio. 0 = disabled 1–63 = divide value

Divider Value	Register Field	Description
R5A	OUT10_DIV	Driver divider ratio. 0 = disabled 1–63 = divide value
R5B	OUT11_DIV	Driver divider ratio. 0 = disabled 1–63 = divide value

6.4 Programming the Output Clock Format

The following tables provide the method to fully define every driver.

Table 6.7. Driver Set Up Options

Driver	Register Field	Description
Driver for output OUTx	OUTx_mode	Software interpreted driver configuration. See Table 6.8 Driver Mode Options on page 25 .
	OUTx _skew	Skew control. Programmed as an unsigned integer. Can add delay of 35 ps/step up to 280 ps.
	OUTx _stop_highz	Driver output state when stopped. 0 = low-z 1 = high-z
	OUTx _cmos_inv	Sets the polarity of the two outputs in dual CMOS mode 0 = no inversion 1 = OUTx~ inverted
	OUTx _cmos_slew	Controls CMOS slew rate from fast to slow. 00 = fastest 01 = slow 10 = slower 11 = slowest
	OUTx _cmos_str	CMOS output impedance control. 0 = 50 Ω 1 = 25 Ω

Table 6.8. Driver Mode Options

drvxy_MODE	Driver Mode
0	off
1	CMOS on positive output only
2	CMOS on negative output only
3	dual CMOS outputs
4	2.5 V/3.3 V LVDS
5	1.8 V LVDS
6	2.5 V/3.3 V LVDS fast
7	1.8 V LVDS fast
8	HCSL 50 Ω (external termination)
9	HCSL 50 Ω (internal termination)
10	HCSL 42.5 Ω (external termination)
11	HCSL 42.5 Ω (internal termination)
12	LVPECL

drvxy_MODE	Driver Mode
13	Reserved
14	Reserved
15	Reserved

6.5 Programming for Frequency Select Operations

Every `hdiv` and `id` has a Bank A and a Bank B divider. The register field names that begin with `hdivxb` or `idxb` denote Bank B dividers. Any FS frequency will be:

$$F_{out}y_{FS} = \frac{vcoFreq}{idxb}$$

Or

$$F_{out}y_{FS} = \frac{vcoFreq}{hdivb}$$

Any output associated with either `idxa` or `hdivxa` can be switched into the above FS frequency. The control that selects the Bank B divider is as shown in table below.

Table 6.9. The Control Register Bit to Switch Frequencies

Register Field	Description
<code>hdivx_div_sel</code>	Selects bank A or bank B divider HSDIV0 settings. The HSDIV0 supports dynamic integer divider changes through this divider select control bit. 0 = bank A divider 1 = bank B divider
<code>idx_cfg_sel</code>	Output interpolative divider 0 configuration bank select. The interpolative divider supports dynamically switching between two complete configurations controlled by this bit. Reconfiguration should be done on the unselected bank. If <code>ID0_CFG=0</code> , running based off bank A, then bank B may be freely reconfigured and once ready all changes will be applied to the ID once <code>ID0_CFG=1</code> thus changing the ID from bank A to bank B. Spread spectrum enable fields <code>ID0A_SS_ENA</code> and <code>ID0B_SS_ENA</code> are the only exception and may be enabled/disabled while bank is selected. 0 = bank A 1 = bank B

In a factory-programmed part, a pin (the FS pin) can be used for the same purpose as the control registers. Once a control bit is set, the backup divider values control the output frequency and that is described by the equations below:

For O-Divider

$$hdivxb_div = \frac{vcoFreq}{F_{outxb} \times R_{xa}}$$

For N-Divider

$$idxb = \frac{vcoFreq}{F_{outxb} \times R_{xa}}$$

The `ida` fraction is represented in register fields `IDPB_INTG`, `IDPB_RES` and `IDPB_DEN`

$$IDxB_INTG = \text{floor} \left(\frac{128 \times vcoFreq}{F_{outxb} \times R_{xa}} \right)$$

$$\frac{IDxB_RES}{IDxB_DEN} = \left(\frac{128 \times vcoFreq}{F_{outxb} \times R_{xa}} \right) - IDxB_INTG$$

As can be seen from the equations above, the backup divider values limit the possible values for the output frequency in this backup mode. Another key feature is that the switch to a FS frequency is “glitchless”. Therefore, the recommended method for glitchless frequency updates is to program either divider a or b (when divider b or a is currently driving the output frequency), and then switch this divider.

6.6 Programming Spread Spectrum

Spread spectrum clocking (SSC) is available only on the Multisynth outputs. Each Multisynth can implement spread spectrum in either the main divider or the backup divider (the FS option). Therefore, the user can program a maximum of four different spread spectrum “profiles” from the same part, although only two profile are available on outputs at any given time. The amplitude of the SSC clock frequency (as illustrated in the figure below) is denoted by $ssc\%$. The variable, $Amod$, in the equation below is a real number representation of the $ssc\%$, which is a percentage value. The modulation rate (also illustrated in the figure below) is denoted by $Fmod$ in the equations below.

$$\text{For Down Spread: } Amod = \frac{ssc\%}{100}$$

$$\text{For Center Spread: } Amod = 2 \times \frac{ssc\%}{100}$$

$$idxy_ss_step_num = \frac{\frac{vcoFreq}{idxy}}{Fmod \times 4}$$

$$idxy_ss_step_res = \frac{Amod \times idxy_den \times idxy \times 128}{2 \times idxy_ss_step_num}$$

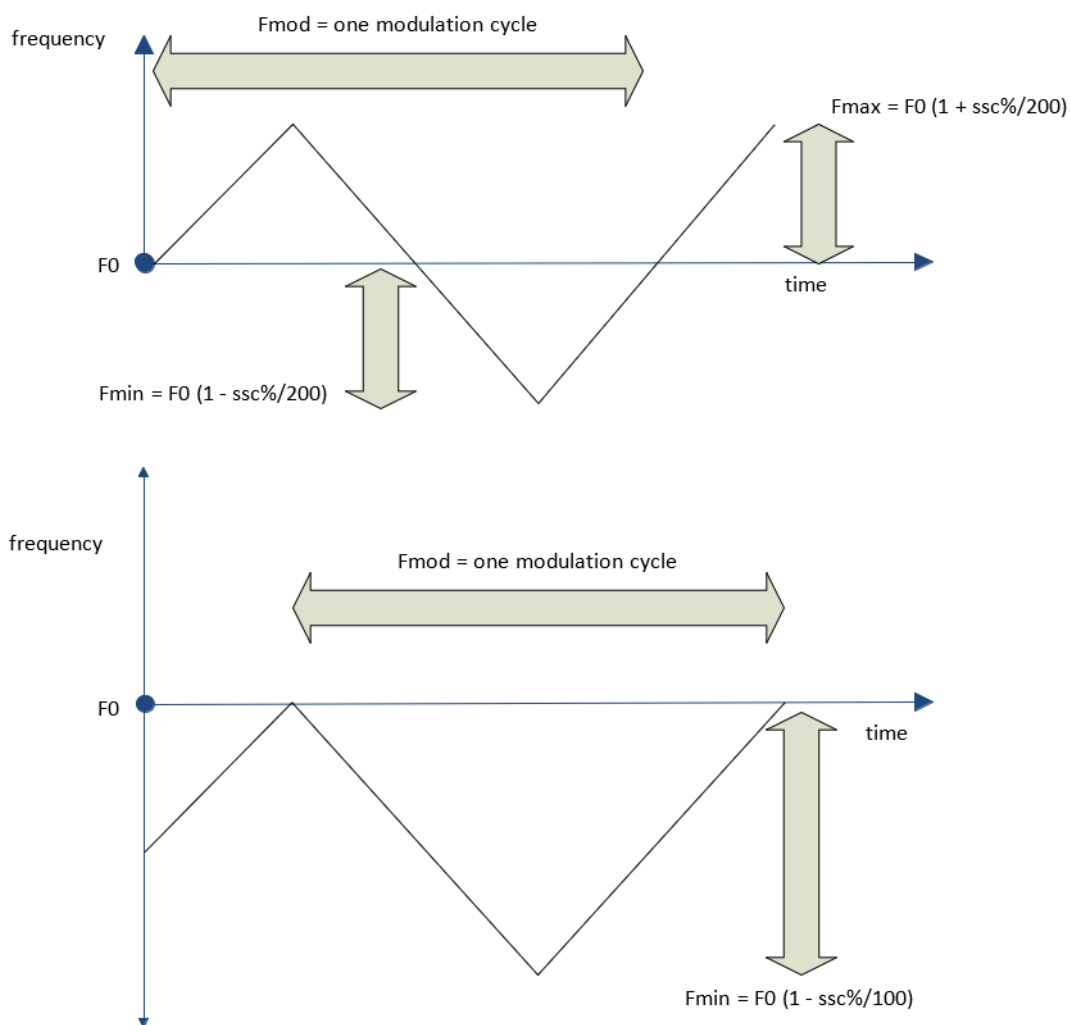


Figure 6.2. Center and Down Spread SSC Clocks as Frequency vs Time

The table below shows the register fields (and terms) $idxy_ss_step_num$ and $idxy_ss_step_res$. $idxy_ss_step_num$ is the number of frequency steps between the mean and the maximum/minimum frequencies in SSC clocking and $idxy_ss_step_res$ is the frequency resolution that is required in each step. The goal is to maximize the number of steps and minimize the resolution. However, the number of steps is set by the modulation rate (typically 30–33 kHz). The step resolution can be minimized by setting the largest value possible for $idxy_den$. $idxy_den$ is the denominator of the id divider and setting it as close as possible to $2^{15} - 1$ is desired.

Table 6.10. SCC Register Fields

idxy_ss_ena	Spread spectrum enable. This is the only bank configuration field which may be changed dynamically while the bank is selected as the active bank. Users may freely enable/disable spread spectrum. 0 = spread spectrum disabled 1 = spread spectrum enabled
idxy_ss_mode	Spread spectrum mode. 0 = disabled 1 = center 2 = invalid 3 = down
idxy_ss_clk_num	Number of output clocks for each frequency step.
idxy_ss_step_num	Number of frequency steps in one quarter SSC modulation period, allows for frequency step every output clock.
idxy_ss_step_intg	Divide ratio spread step size.
idxy_ss_step_res	Numerator of spread step size error term.
idxy_ss_step_den	Denominator of spread step size error term.

To enable SSC, `idxy_ss_ena` needs to be set and the right mode selected in `idxy_ss_mode`. The number of output clocks in each frequency step, `idxy_ss_clk_num`, needs to be set to 1 and `idxy_ss_step_den` is the same as `idxy_den` and `idxy_ss_step_intg` is always zero.

The following flow needs to be followed to program the registers into Si5332:

1. Write 0x01h to register 0x06h and put the Si5332 into the READY state.
2. Write all the relevant registers as calculated from the steps above.
3. Ensure that the valid input clocks are available for the Si5332 to attempt a PLL lock.
4. Write 0x02h to register 0x06h and put the Si5332 into the ACTIVE state.

7. Si5332 Pinout and Package Variant

There are three versions of the Si5332 available for customers. The pinout for each is shown in the figures below.

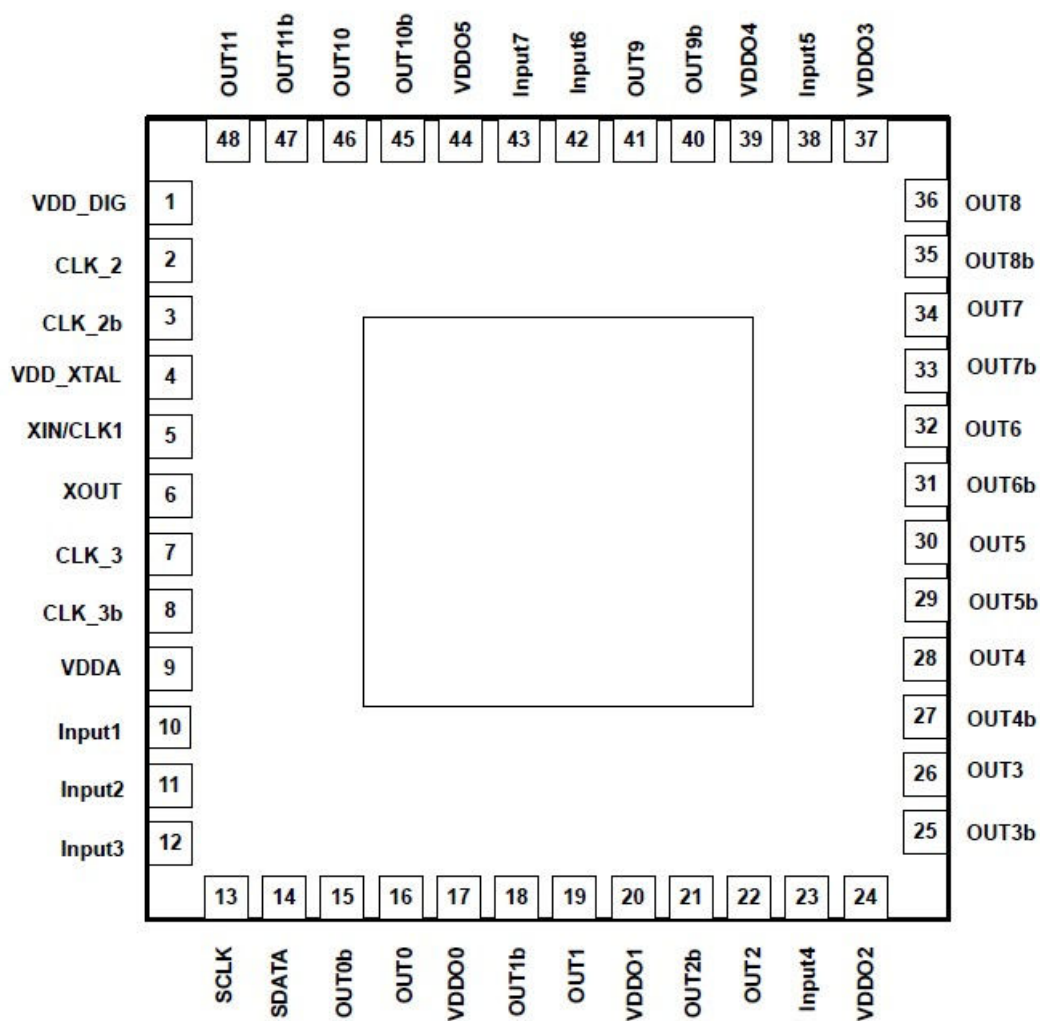


Figure 7.1. 12-Output Si5332 7x7 mm QFN Package

Note: Planned future product not yet released.

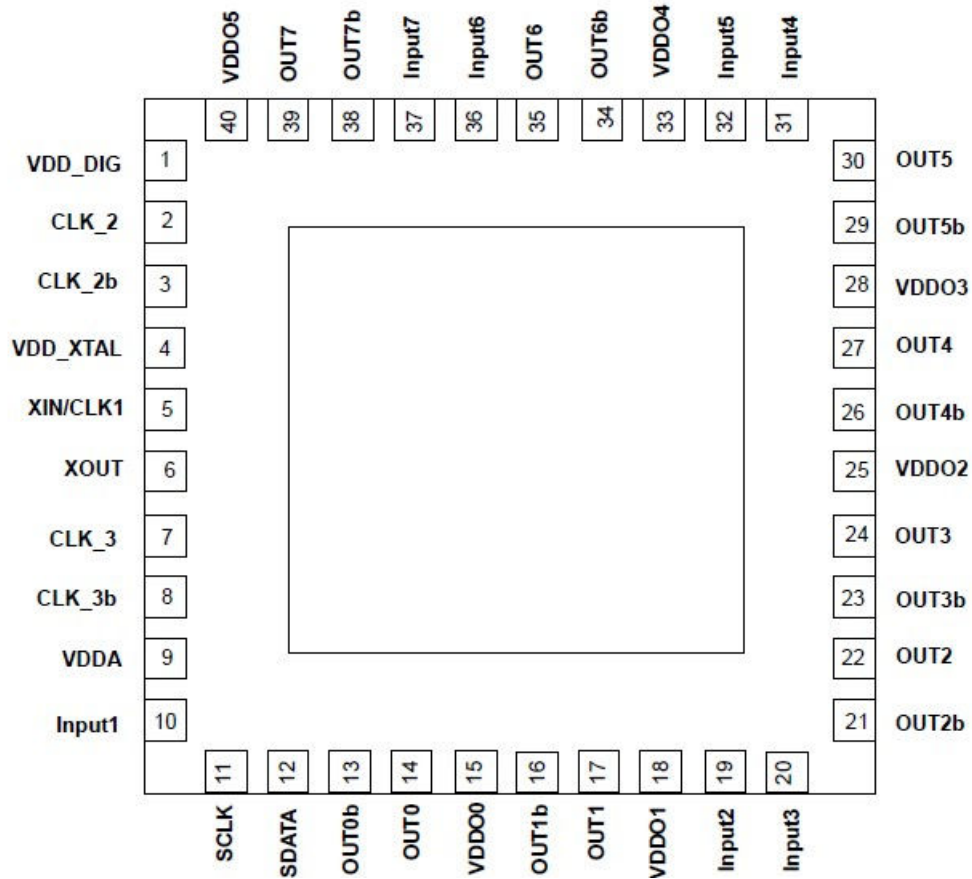


Figure 7.2. 8-Output Si5332 6x6 mm QFN Package

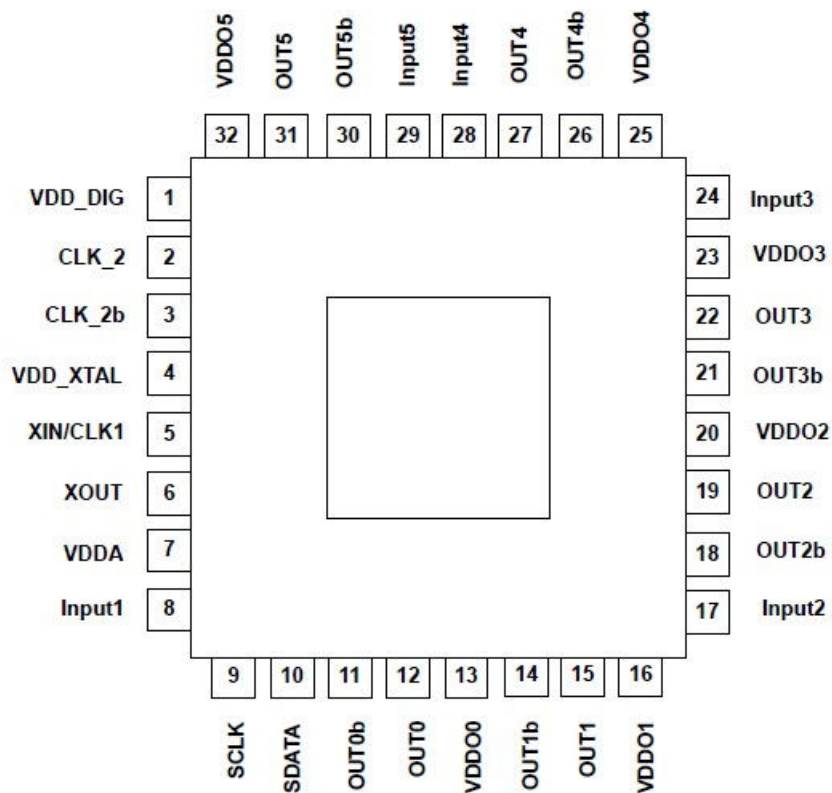


Figure 7.3. 6-Output Si5332 5x5 mm QFN Package

8. Recommended Schematic and Layout Practices

The Si5332 schematic and layout design can be referenced from the respective EVB design for Si5332. For each package, the User's Guide (links below) outlines the EVB design and provides links to schematic and layout references for each package type.

- [UG398: Si5332-8A-EVB User's Guide](#)
- [UG397: Si5332-6A-EVB User's Guide](#)

At the schematic/placement/layout design time, these are the following guidelines:

1. Power supply filtering:

- a. The Si5332 can tolerate up to 100 mV (+/-50 mV) of noise for each supply node. The Application Note, [AN1107: Si5332 Power Supply Noise Rejection](#), provides the performance to be expected with such a noise.
 - i. As can be seen, this noise can be from a switched mode power supply (which causes noise over a wide band of frequencies) or can be noise due to some oscillatory behavior from an LDO regulator.
 - ii. The only filtering needed on each supply node is a 1 μ F and a 0.1 μ F placed as close as possible to that node.
 - iii. The Si5332 EVBs have a much larger capacitance on the regulator end, mainly to compensate for the regulator loop so that there is no oscillatory behavior from the regulators regardless of the voltage supply value set for that regulator. The regulator supply design on the EVB is not required for Si5332 in system designs.

2. Crystal placement:

- a. The crystals should be placed as close as possible to the XA/XB pins. This placement ensures that the crystal oscillator traces do not cause undue delays and hence, cause either an unusually long crystal start up time or get susceptible to crosstalk and thereby increase jitter on the output clocks.

9. Register Map

All common registers are listed in the table below. The registers that are specific to the 32-QFN part are listed in [Table 9.2 Si5332 32-QFN Registers on page 43](#). The registers that are specific to the 40-QFN part are listed in [Table 9.3 Si5332 40-QFN Registers on page 48](#). The fields in these tables are the register field name, address, base, bit length, "R/W/RW", description, and device mode. Note that all registers hold values that are "big-endian", i.e., bit 7 is the MSB in an eight-bit field.

The definitions for these fields are:

1. *Register Field Name*: The name for the register field in this FRM as referenced in the tables below and in other sections in this FRM.
2. *Address*: The 8-bit register address to be used in the I²C transactions when the register field needs to be addressed.
3. *Base*: Every register field address addresses an 8-bit wide location. However, the register field may not occupy that entire location. In those cases, they may also not start at the LSB i.e. bit #0 of that location. Base provides the bit #i from which this register field begins in the addressed location.
4. *Bit Length*: Bit length indicates the "number of bits" that the register field occupies in the addressed location.
5. *R/W/RW*: This field indicates if the register field is Read only (R), Write only (W) or Read/Write (RW).
6. *Description*: Description is an explanation on the purpose and programmability offered by the register field.
7. *Device Mode*: Device mode is the mode of Si5332 in which the register field can be accessed. Si5332 has two modes of function "READY" where the Si5332 is ready for programming in which time there will no outputs from Si5332 and "ACTIVE" where the Si5332 is actively locked to an input and is providing outputs. Some register fields can be programmed in either READY or ACTIVE mode (READY/ACTIVE) whereas others can only be programmed in READY mode (READY). Device mode provides input on which mode applies to a register field a user intends to modify.

Table 9.1. Si5332 Register Map

Register Field Name	Address	Base	Bit Length	R/W/RW	Description	Device Mode
VDD_XTAL_OK	5	7	1	R	Flag that VDD_XTAL is greater than its minimum level, which is about 1.5 V.	READY/ ACTIVE
VDDO_OK	5	0	6	R	Flags that various VDDO supplies are greater than their minimum level, which is about 1.2 V.	
USYS_CTRL	6	0	8	RW	User system control. Write 0x01h to make the part READY. Write 0x02h to make the part ACTIVE	READY/ ACTIVE
USYS_STAT	7	0	8	R	User system status. This indicates the status of the application, and what state it is in, like READY, ACTIVE, etc. It is read only register for I ² C	READY/ ACTIVE
UDRV_OE_ENA	8	0	1	RW	User master output enable. Resets to 1. This bit controls simultaneously the driver start for all drivers.	READY/ ACTIVE
USER_SCRATCH0	9	0	8	RW	User scratch pad registers, freely R/W any time. This is just run time scratch area, not initialized from NVM. The reset value is 0x00 for all bytes. Can be I ² C read and written any time.	READY/ ACTIVE
USER_SCRATCH1	A	0	8	RW		
USER_SCRATCH2	B	0	8	RW		
USER_SCRATCH3	C	0	8	RW		
DEVICE_PN_BASE	D	0	8	R	Device part number	READY/ ACTIVE

Register Field Name	Address	Base	Bit Length	R/W/RW	Description	Device Mode
DEVICE_REV	E	0	8	R	Device revision	READY/ ACTIVE
DEVICE_GRADE	F	0	8	R	Device grade information	
FACTORY_OPN_ID0	10	0	4	R	The Orderable part number identification, OPN ID-0. Example: For Si5332AD98765-AM1, ID-0 = 5.	
FACTORY_OPN_ID1	10	4	4	R	The Orderable part number identification, OPN ID-1. Example: For Si5332AD98765-AM1, ID-1 = 6.	
FACTORY_OPN_ID2	11	4	4	R	The Orderable part number identification, OPN ID-2. Example: For Si5332AD98765-AM1, ID-2 = 7.	
FACTORY_OPN_ID3	11	0	4	R	The Orderable part number identification, OPN ID-3. Example: For Si5332AD98765-AM1, ID-3 = 8.	
FACTORY_OPN_ID4	12	0	4	R	The Orderable part number identification, OPN ID-4. Example: For Si5332AD98765-AM1, ID-4 = 9.	
FACTORY_OPN_REVISION	12	4	4	R	The Orderable part number's product revision number.	
DESIGN_ID0	17	0	8	R	Design identification set by user in CBPro project file	READY/ ACTIVE
DESIGN_ID1	18	0	8	R		
DESIGN_ID2	19	0	8	R		
I2C_ADDR	21	0	7	R	I2C mode device address. Reset value is 110_1010 binary.	
I2C_SCL_PUP_ENA	23	0	1	RW	Enable 50 kΩ pullup resistor on SCL pad.	READY/ ACTIVE
I2C_SDA_PUP_ENA	23	1	1	RW	Enable 50 kΩ pullup resistor on SDA pad.	READY/ ACTIVE
OMUX0_SEL0	25	0	2	RW	Selects output mux clock source for output clocks in group G0:OUT0 for AM1, AM2: 0 = PLL reference clock before pre-scaler 1 = PLL reference clock after pre-scaler 2 = Clock from input buffer CLKIN_2 3 = Clock from input buffer CLKIN_3	READY/ ACTIVE

Register Field Name	Address	Base	Bit Length	R/W/RW	Description	Device Mode
OMUX0_SEL1	25	4	3	RW	<p>Selects output mux clock source for output clocks in group G0:OUT0 for AM1, AM2:</p> <p>0 = HSDIV0 1 = HSDIV1 2 = HSDIV2 3 = HSDIV3 4 = HSDIV4 5 = ID0 6 = ID1 7 = Clock from OMUX0_SELO</p> <p>Note that the OMUX0_SEL1 value is forced to 7 whenever the PLL is disabled</p>	READY/ ACTIVE
OMUX1_SELO	26	0	2	RW	<p>Selects output mux clock source for output clocks in group G1: OUT1 for AM1, AM2:</p> <p>0 = PLL reference clock before pre-scaler 1 = PLL reference clock after pre-scaler 2 = Clock from input buffer CLKIN_2 3 = Clock from input buffer CLKIN_3</p>	READY/ ACTIVE
OMUX1_SEL1	26	4	3	RW	<p>Selects output mux clock source for output clocks in group G1: OUT1 for AM1, AM2:</p> <p>0 = HSDIV0 1 = HSDIV1 2 = HSDIV2 3 = HSDIV3 4 = HSDIV4 5 = ID0 6 = ID1 7 = Clock from OMUX1_SELO</p> <p>Note that the OMUX1_SEL1 value is forced to 7 whenever the PLL is disabled</p>	READY/ ACTIVE
OMUX2_SELO	27	0	2	RW	<p>Selects output mux clock source for output clocks in group G2: OUT2 for AM1; OUT2, OUT3 for AM2:</p> <p>0 = PLL reference clock before pre-scaler 1 = PLL reference clock after pre-scaler 2 = Clock from input buffer CLKIN_2 3 = Clock from input buffer CLKIN_3</p>	READY/ ACTIVE

Register Field Name	Address	Base	Bit Length	R/W/RW	Description	Device Mode
OMUX2_SEL1	27	4	3	RW	<p>Selects output mux clock source for output clocks in group G2: OUT2 for AM1; OUT2, OUT3 for AM2:</p> <p>0 = HSDIV0 1 = HSDIV1 2 = HSDIV2 3 = HSDIV3 4 = HSDIV4 5 = ID0 6 = ID1 7 = Clock from OMUX2_SELO</p> <p>Note that the OMUX2_SEL1 value is forced to 7 whenever the PLL is disabled</p>	READY/ ACTIVE
OMUX3_SELO	28	0	2	RW	<p>Selects output mux clock source for output clocks in group G3: OUT3 for AM1; OUT4, OUT5 for AM2:</p> <p>0 = PLL reference clock before pre-scaler 1 = PLL reference clock after pre-scaler 2 = Clock from input buffer CLKIN_2 3 = Clock from input buffer CLKIN_3</p>	READY/ ACTIVE
OMUX3_SEL1	28	4	3	RW	<p>Selects output mux clock source for output clocks in group G3: OUT3 for AM1; OUT4, OUT5 for AM2:</p> <p>0 = HSDIV0 1 = HSDIV1 2 = HSDIV2 3 = HSDIV3 4 = HSDIV4 5 = ID0 6 = ID1 7 = Clock from OMUX3_SELO</p> <p>Note that the OMUX3_SEL1 value is forced to 7 whenever the PLL is disabled</p>	READY/ ACTIVE
OMUX4_SELO	29	0	2	RW	<p>Selects output mux clock source for output clocks in group G4: OUT4 for AM1; OUT6 for AM2:</p> <p>0 = PLL reference clock before pre-scaler 1 = PLL reference clock after pre-scaler 2 = Clock from input buffer CLKIN_2 3 = Clock from input buffer CLKIN_3</p>	READY/ ACTIVE

Register Field Name	Address	Base	Bit Length	R/W/RW	Description	Device Mode
OMUX4_SEL1	29	4	3	RW	<p>Selects output mux clock source for output clocks in group G4: OUT4 for AM1; OUT6 for AM2:</p> <p>0 = HSDIV0 1 = HSDIV1 2 = HSDIV2 3 = HSDIV3 4 = HSDIV4 5 = ID0 6 = ID1 7 = Clock from OMUX4_SELO</p> <p>Note that the OMUX4_SEL1 value is forced to 7 whenever the PLL is disabled</p>	READY/ ACTIVE
OMUX5_SELO	2A	0	2	RW	<p>Selects output mux clock source for output clocks in group G5: OUT5 for AM1; OUT7 for AM2:</p> <p>0 = PLL reference clock before pre-scaler 1 = PLL reference clock after pre-scaler 2 = Clock from input buffer CLKIN_2 3 = Clock from input buffer CLKIN_3</p>	READY/ ACTIVE
OMUX5_SEL1	2A	4	3	RW	<p>Selects output mux clock source for output clocks in group G5: OUT5 for AM1; OUT7 for AM2:</p> <p>0 = HSDIV0 1 = HSDIV1 2 = HSDIV2 3 = HSDIV3 4 = HSDIV4 5 = ID0 6 = ID1 7 = Clock from OMUX5_SELO</p> <p>Note that the OMUX5_SEL1 value is forced to 7 whenever the PLL is disabled</p>	READY/ ACTIVE
HSDIV0A_DIV	2B	0	8	RW	O0 divider value	READY if divider is currently driving the output else, READY/ACTIVE
HSDIV0B_DIV	2C	0	8	RW	O0 divider value for bank A	
HSDIV1A_DIV	2D	0	8	RW	O1 divider value for bank A	
HSDIV1B_DIV	2E	0	8	RW	O1 divider value for bank B	
HSDIV2A_DIV	2F	0	8	RW	O2 divider value for bank A	

Register Field Name	Address	Base	Bit Length	R/W/RW	Description	Device Mode
HSDIV2B_DIV	30	0	8	RW	O2 divider value for bank B	
HSDIV3A_DIV	31	0	8	RW	O3 divider value for bank A	
HSDIV3B_DIV	32	0	8	RW	O3 divider value for bank B	
HSDIV4A_DIV	33	0	8	RW	O4 divider value for bank A	
HSDIV4B_DIV	34	0	8	RW	O4 divider value for bank B	
HSDIV3_DIV_SEL	35	3	1	RW	Selects bank A (0) or bank B (1) O3 divider settings. Same description applies as for HSDIV0_DIV_SEL.	READY/ ACTIVE
ID0_CFG_SEL	35	6	1	RW	N0 configuration bank select. The divider supports dynamically switching between two complete configurations controlled by this bit. Reconfiguration should be done on the unselected bank. If ID0_CFG=0, running based off bank A, then bank B may be freely reconfigured and once ready all changes will be applied to the ID once ID0_CFG=1 thus changing the ID from bank A to bank B. Spread spectrum enable fields ID0A_SS_ENA and ID0B_SS_ENA are the only exception and may be enabled/disabled while bank is selected. 0 = bank A 1 = bank B	READY/ ACTIVE
HSDIV4_DIV_SEL	35	4	1	RW	Selects bank A (0) or bank B (1) O4 divider settings. Same description applies as for HSDIV0_DIV_SEL.	READY/ ACTIVE
ID1_CFG_SEL	35	7	1	RW	N1 configuration bank select. Same description related to ID1 applies as in the ID0_CFG description. 0 = bank A 1 = bank B	READY/ ACTIVE
HSDIV2_DIV_SEL	35	2	1	RW	Selects bank A (0) or bank B (1) O2 divider settings. Same description applies as for HSDIV0_DIV_SEL.	READY/ ACTIVE
HSDIV0_DIV_SEL	35	0	1	RW	Selects bank A or bank B divider O0 settings. O0 supports dynamic integer divider changes through this divider select control bit. 0 = bank A divider 1 = bank B divider	READY/ ACTIVE
HSDIV1_DIV_SEL	35	1	1	RW	Selects bank A (0) or bank B (1) O1 divider settings. Same description applies as for HSDIV0_DIV_SEL.	READY/ ACTIVE

Register Field Name	Address	Base	Bit Length	R/W/RW	Description	Device Mode
ID0A_INTG	36	0	15	RW	The terms of an $a + b/c$ desired divider setting must be processed into ID0A_INTG, ID0A_RES, and ID0A_DEN register terms. $INTG = \text{floor}(((a*c+b)*128/c) - 512)$.	READY if divider is currently driving the output, else, READY/ACTIVE
ID0A_RES	38	0	15	RW	$res = \text{mod}(b*128, c)$	
ID0A_DEN	3A	0	15	RW	$den = c$	
ID0A_SS_ENA	3C	0	1	RW	Spread spectrum enable. This is the only bank configuration field which may be changed dynamically while the bank is selected as the active bank. Users may freely enable/disable spread spectrum. 0 = spread spectrum disabled 1 = spread spectrum enabled	READY if divider is currently driving the output, else, READY/ACTIVE
ID0A_SS_MODE	3C	1	2	RW	Spread spectrum mode. 0 = disabled 1 = center 2 = invalid 3 = down	
ID0A_SS_STEP_NUM	3D	0	12	RW	Number of frequency steps in one quarter SSC modulation period, allows for frequency step every output clock.	
ID0A_SS_STEP_INTG	3F	0	12	RW	Divide ratio spread step size.	
ID0A_SS_STEP_RES	40	0	15	RW	Numerator of spread step size error term.	

Register Field Name	Address	Base	Bit Length	R/W/RW	Description	Device Mode	
ID0B_INTG	42	0	15	RW	The terms of an $a + b/c$ desired divider setting must be processed into ID0B_INTG, ID0B_RES, and ID0B_DEN register terms. $INTG = \text{floor}(((a*c+b)*128/c) - 512)$.	READY if divider is currently driving the output, else, READY/ACTIVE	
ID0B_RES	44	0	15	RW	$\text{res} = \text{mod}(b*128, c)$		
ID0B_DEN	46	0	15	RW	$\text{den} = c$		
ID0B_SS_ENA	48	0	1	RW	Spread spectrum enable. This is the only bank configuration field which may be changed dynamically while the bank is selected as the active bank. Users may freely enable/disable spread spectrum. 0 = spread spectrum disabled 1 = spread spectrum enabled		
ID0B_SS_MODE	48	1	2	RW	Spread spectrum mode. 0 = disabled 1 = center 2 = invalid 3 = down		
ID0B_SS_STEP_NUM	49	0	12	RW	Number of frequency steps in one quarter SSC modulation period, allows for frequency step every output clock.		
ID0B_SS_STEP_INTG	4B	0	12	RW	Divide ratio spread step size.		
ID0B_SS_STEP_RES	4C	0	15	RW	Numerator of spread step size error term.		
ID1A_INTG	4E	0	15	RW	The terms of an $a + b/c$ desired interpolative divider setting must be processed into ID1A_INTG, ID1A_RES, and ID1A_DEN register terms. $INTG = \text{floor}(((a*c+b)*128/c) - 512)$.		READY if divider is currently driving the output, else, READY/ACTIVE
ID1A_RES	50	0	15	RW	$\text{res} = \text{mod}(b*128, c)$		
ID1A_DEN	52	0	15	RW	$\text{den} = c$		

Register Field Name	Address	Base	Bit Length	R/W/RW	Description	Device Mode
ID1A_SS_ENA	54	0	1	RW	Spread spectrum enable. This is the only bank configuration field which may be changed dynamically while the bank is selected as the active bank. Users may freely enable/disable spread spectrum. 0 = spread spectrum disabled 1 = spread spectrum enabled	READY if divider is currently driving the output, else, READY/ACTIVE
ID1A_SS_MODE	54	1	2	RW	Spread spectrum mode. 0 = disabled 1 = center 2 = invalid (up) 3 = down	
ID1A_SS_STEP_NUM	55	0	12	RW	Number of frequency steps in one quadrature, allows for frequency step every output clock.	
ID1A_SS_STEP_INTG	57	0	12	RW	Divide ratio spread step size.	
ID1A_SS_STEP_RES	58	0	15	RW	Numerator of spread step size error term.	
ID1B_INTG	5A	0	15	RW	The terms of an $a + b/c$ desired interpolative divider setting must be processed into ID1A_INTG, ID1A_RES, and ID1A_DEN register terms. $INTG = \text{floor}(((a*c+b)*128/c) - 512)$.	READY if divider is currently driving the output, else, READY/ACTIVE
ID1B_RES	5C	0	15	RW	$\text{res} = \text{mod}(b*128, c)$	
ID1B_DEN	5E	0	15	RW	$\text{den} = c$	
ID1B_SS_ENA	60	0	1	RW	Spread spectrum enable. This is the only bank configuration field which may be changed dynamically while the bank is selected as the active bank. Users may freely enable/disable spread spectrum. 0 = spread spectrum disabled 1 = spread spectrum enabled	
ID1B_SS_MODE	60	1	2	RW	Spread spectrum mode. 0 = disabled 1 = center 2 = invalid (up) 3 = down	
ID1B_SS_STEP_NUM	61	0	12	RW	Number of frequency steps in one quadrature, allows for frequency step every output clock.	
ID1B_SS_STEP_INTG	63	0	12	RW	Divide ratio spread step size.	
ID1B_SS_STEP_RES	64	0	15	RW	Numerator of spread step size error term.	

Register Field Name	Address	Base	Bit Length	R/W/RW	Description	Device Mode
IDPA_INTG	67	0	15	RW	The terms of an a + b/c desired divider setting must be processed into IDPA_INTG, IDPA_RES, and IDPA_DEN register terms. $INTG = \text{floor}(((a*c+b)*128/c) - 512)$.	READY
IDPA_RES	69	0	15	RW	$res = \text{mod}(b*128, c)$	READY
IDPA_DEN	6B	0	15	RW	$den = c$	READY
PDIV_DIV	75	0	5	RW	Chooses the PLL prescaler divide ratio.	READY
USYS_START	B8	0	8	RW	User defined application startup behavior. Flags for SW what to do at the startup, for example moving to ACTIVE on its own upon startup or waiting in READY state for a command. Used only upon startup, Initialized from NVM.	READY
PLL_MODE	BE	2	4	RW	Sets PLL BW. See Table 6.1 Constraints for PLL Reference Frequency and VCO Frequency on page 18 .	READY
XOSC_CINT_ENA	BF	7	1	RW	Enables a fixed 7.84 pf of internal loading capacitance to values set by XOSC_CTRIM_XA/XB registers. Refer to Section 3.2 Calculating Crystal Loading Capacitance for information on use of this register.	READY
XOSC_CTRIM_XA	C0	0	6	RW	Load capacitance trim on XA. Refer to Section 3.2 Calculating Crystal Loading Capacitance for information on use of this register.	READY
XOSC_CTRIM_XB	C1	0	6	RW	Load capacitance trim on XB. Refer to Section 3.2 Calculating Crystal Loading Capacitance for information on use of this register.	READY

Table 9.2. Si5332 32-QFN Registers

Register Field Name	Address	Base	Bit Length	R/W/RW	Description	Device Mode
OUT0_MODE	7A	0	4	RW	Software interpreted driver configuration. See Table 6.7 Driver Set Up Options on page 25 .	READY
OUT0_DIV	7B	0	6	RW	Driver divider ratio. 0 = disabled 1-63 = divide value	READY
OUT0_SKEW	7C	0	3	RW	Skew control. Programmed as an unsigned integer. Can add delay of 35 ps/step up to 280 ps.	READY
OUT0_STOP_HIGHZ	7D	0	1	RW	Driver output state when stopped. 0 = low-Z 1 = high-Z	READY
OUT0_CMOS_INV	7D	4	2	RW	Sets the polarity of the two outputs in dual CMOS mode. 0 = no inversion 1 = OUT0b inverted	READY
OUT0_CMOS_SLEW	7E	0	2	RW	Controls CMOS slew rate from fast to slow. 00 = fastest 01 = slow 10 = slower 11 = slowest	READY
OUT0_CMOS_STR	7E	2	1	RW	CMOS output impedance control. 0 = 50 Ω 1 = 25 Ω	READY
OUT1_MODE	7F	0	4	RW	Software interpreted driver configuration. See Table 6.7 Driver Set Up Options on page 25 .	READY
OUT1_DIV	80	0	6	RW	Driver divider ratio. 0 = disabled 1-63 = divide value	READY
OUT1_SKEW	81	0	3	RW	Skew control. Programmed as an unsigned integer. Can add delay of 35 ps/step up to 280 ps.	READY
OUT1_STOP_HIGHZ	82	0	2	RW	Driver output state when stopped. 0 = low-Z 1 = high-Z	READY

Register Field Name	Address	Base	Bit Length	R/W/RW	Description	Device Mode
OUT1_CMOS_INV	82	4	1	RW	Sets the polarity of the two outputs in dual CMOS mode. 0 = no inversion 1 = OUT1b inverted	READY
OUT1_CMOS_SLEW	83	0	1	RW	Controls CMOS slew rate from fast to slow. 00 = fastest 01 = slow 10 = slower 11 = slowest	READY
OUT1_CMOS_STR	83	2	1	RW	CMOS output impedance control. 0 = 50 Ω 1 = 25 Ω	READY
OUT2_MODE	89	0	4	RW	Software interpreted driver configuration. See Table 6.7 Driver Set Up Options on page 25 .	READY
OUT2_DIV	8A	0	6	RW	Driver divider ratio. 0 = disabled 1-63 = divide value	READY
OUT2_SKEW	8B	0	3	RW	Skew control. Programmed as an unsigned integer. Can add delay of 35 ps/step up to 280 ps.	READY
OUT2_STOP_HIGHZ	8C	0	2	RW	Driver output state when stopped. 0 = low-Z 1 = high-Z	READY
OUT2_CMOS_INV	8C	4	1	RW	Sets the polarity of the two outputs in dual CMOS mode. 0 = no inversion 1 = OUT2b inverted	READY
OUT2_CMOS_SLEW	8D	0	2	RW	Controls CMOS slew rate from fast to slow. 00 = fastest 01 = slow 10 = slower 11 = slowest	READY
OUT2_CMOS_STR	8D	2	1	RW	CMOS output impedance control. 0 = 50 Ω 1 = 25 Ω	READY

Register Field Name	Address	Base	Bit Length	R/W/RW	Description	Device Mode
OUT3_MODE	98	0	4	RW	Software interpreted driver configuration. See Table 6.7 Driver Set Up Options on page 25 .	READY
OUT3_DIV	99	0	6	RW	Driver divider ratio. 0 = disabled 1-63 = divide value	READY
OUT3_SKEW	9A	0	3	RW	Skew control. Programmed as an unsigned integer. Can add delay of 35 ps/step up to 280 ps.	READY
OUT3_STOP_HIGHZ	9B	0	1	RW	Driver output state when stopped. 0 = low-Z 1 = high-Z	READY
OUT3_CMOS_INV	9B	4	1	RW	Sets the polarity of the two outputs in dual CMOS mode. 0 = no inversion 1 = OUT3b inverted	READY
OUT3_CMOS_SLEW	9C	0	1	RW	Controls CMOS slew rate from fast to slow. 00 = fastest 01 = slow 10 = slower 11 = slowest	READY
OUT3_CMOS_STR	9C	2	1	RW	CMOS output impedance control. 0 = 50 Ω 1 = 25 Ω	READY
OUT4_MODE	A7	0	4	RW	Software interpreted driver configuration. See Table 6.7 Driver Set Up Options on page 25 .	READY
OUT4_DIV	A8	0	6	RW	Driver divider ratio. 0 = disabled 1-63 = divide value	READY
OUT4_SKEW	A9	0	3	RW	Skew control. Programmed as an unsigned integer. Can add delay of 35 ps/step up to 280 ps.	READY
OUT4_STOP_HIGHZ	AA	0	1	RW	Driver output state when stopped. 0 = low-Z 1 = high-Z	READY
OUT4_CMOS_INV	AA	4	1	RW	Sets the polarity of the two outputs in dual CMOS mode. 0 = no inversion 1 = OUT4b inverted	READY

Register Field Name	Address	Base	Bit Length	R/W/RW	Description	Device Mode
OUT4_CMOS_SLEW	AB	0	1	RW	Controls CMOS slew rate from fast to slow. 00 = fastest 01 = slow 10 = slower 11 = slowest	READY
OUT4_CMOS_STR	AB	2	1	RW	CMOS output impedance control. 0 = 50 Ω 1 = 25 Ω	READY
OUT5_MODE	AC	0	4	RW	Software interpreted driver configuration. See Table 6.7 Driver Set Up Options on page 25 .	READY
OUT5_DIV	AD	0	6	RW	Driver divider ratio. 0 = disabled 1–63 = divide value	READY
OUT5_SKEW	AE	0	3	RW	Skew control. Programmed as an unsigned integer. Can add delay of 35 ps/step up to 280 ps.	READY
OUT5_STOP_HIGHZ	AF	0	1	RW	Driver output state when stopped. 0 = low-Z 1 = high-Z	READY
OUT5_CMOS_INV	AF	4	1	RW	Sets the polarity of the two outputs in dual CMOS mode. 0 = no inversion 1 = OUT5b inverted	READY
OUT5_CMOS_SLEW	B0	0	1	RW	Controls CMOS slew rate from fast to slow. 00 = fastest 01 = slow 10 = slower 11 = slowest	READY
OUT5_CMOS_STR	B0	2	1	RW	CMOS output impedance control. 0 = 50 Ω 1 = 25 Ω	READY
OUT2_OE	B6	3	1	RW	Output enable control for OUT2	READY/ ACTIVE
OUT3_OE	B6	6	1	RW	Output enable control for OUT3	READY/ ACTIVE
OUT0_OE	B6	0	1	RW	Output enable control for OUT0	READY/ ACTIVE
OUT1_OE	B6	1	1	RW	Output enable control for OUT1	READY/ ACTIVE
OUT5_OE	B7	2	1	RW	Output enable control for OUT5	READY/ ACTIVE

Register Field Name	Address	Base	Bit Length	R/W/RW	Description	Device Mode
OUT4_OE	B7	1	1	RW	Output enable control for OUT4	READY/ ACTIVE
CLKIN_2_CLK_SEL	73	0	2	RW	0 = disabled 1 = differential 2 = CMOS DC 3 = CMOS AC	READY
IMUX_SEL	24	0	2	RW	Selects input mux clock source: 0 = Disabled 1 = XOSC 2 = CLKIN_2 3 = Disabled	READY

Table 9.3. Si5332 40-QFN Registers

Register Field Name	Address	Base	Bit Length	R/W/RW	Description	Device Mode
OUT0_MODE	7A	0	4	RW	Software interpreted driver configuration. See Table 6.7 Driver Set Up Options on page 25 .	READY
OUT0_DIV	7B	0	6	RW	Driver divider ratio. 0 = disabled 1-63 = divide value	READY
OUT0_SKEW	7C	0	3	RW	Skew control. Programmed as an unsigned integer. Can add delay of 35 ps/step up to 280 ps.	READY
OUT0_STOP_HIGHZ	7D	0	1	RW	Driver output state when stopped. 0 = low-Z 1 = high-Z	READY
OUT0_CMOS_INV	7D	4	2	RW	Sets the polarity of the two outputs in dual CMOS mode. 0 = no inversion 1 = OUT0b inverted	READY
OUT0_CMOS_SLEW	7E	0	2	RW	Controls CMOS slew rate from fast to slow. 00 = fastest 01 = slow 10 = slower 11 = slowest	READY
OUT0_CMOS_STR	7E	2	1	RW	CMOS output impedance control. 0 = 50 Ω 1 = 25 Ω	READY
OUT1_MODE	7F	0	4	RW	Software interpreted driver configuration. See Table 6.7 Driver Set Up Options on page 25 .	READY
OUT1_DIV	80	0	6	RW	Driver divider ratio. 0 = disabled 1-63 = divide value	READY
OUT1_SKEW	81	0	3	RW	Skew control. Programmed as an unsigned integer. Can add delay of 35 ps/step up to 280 ps.	READY
OUT1_STOP_HIGHZ	82	0	2	RW	Driver output state when stopped. 0 = low-Z 1 = high-Z	READY

Register Field Name	Address	Base	Bit Length	R/W/RW	Description	Device Mode
OUT1_CMOS_INV	82	4	1	RW	Sets the polarity of the two outputs in dual CMOS mode. 0 = no inversion 1 = OUT1b inverted	READY
OUT1_CMOS_SLEW	83	0	1	RW	Controls CMOS slew rate from fast to slow. 00 = fastest 01 = slow 10 = slower 11 = slowest	READY
OUT1_CMOS_STR	83	2	1	RW	CMOS output impedance control. 0 = 50 Ω 1 = 25 Ω	READY
OUT2_MODE	89	0	4	RW	Software interpreted driver configuration. See Table 6.7 Driver Set Up Options on page 25 .	READY
OUT2_DIV	8A	0	6	RW	Driver divider ratio. 0 = disabled 1-63 = divide value	READY
OUT2_SKEW	8B	0	3	RW	Skew control. Programmed as an unsigned integer. Can add delay of 35 ps/step up to 280 ps.	READY
OUT2_STOP_HIGHZ	8C	0	2	RW	Driver output state when stopped. 0 = low-Z 1 = high-Z	READY
OUT2_CMOS_INV	8C	4	1	RW	Sets the polarity of the two outputs in dual CMOS mode. 0 = no inversion 1 = OUT2b inverted	READY
OUT2_CMOS_SLEW	8D	0	2	RW	Controls CMOS slew rate from fast to slow. 00 = fastest 01 = slow 10 = slower 11 = slowest	READY
OUT2_CMOS_STR	8D	2	1	RW	CMOS output impedance control. 0 = 50 Ω 1 = 25 Ω	READY

Register Field Name	Address	Base	Bit Length	R/W/RW	Description	Device Mode
OUT3_MODE	8E	0	4	RW	Software interpreted driver configuration. See Table 6.7 Driver Set Up Options on page 25 .	READY
OUT3_DIV	8F	0	6	RW	Driver divider ratio. 0 = disabled 1-63 = divide value	READY
OUT3_SKEW	90	0	3	RW	Skew control. Programmed as an unsigned integer. Can add delay of 35 ps/step up to 280 ps.	READY
OUT3_STOP_HIGHZ	91	0	2	RW	Driver output state when stopped. 0 = low-Z 1 = high-Z	READY
OUT3_CMOS_INV	91	4	1	RW	Sets the polarity of the two outputs in dual CMOS mode. 0 = no inversion 1 = OUT3b inverted	READY
OUT3_CMOS_SLEW	92	0	1	RW	Controls CMOS slew rate from fast to slow. 00 = fastest 01 = slow 10 = slower 11 = slowest	READY
OUT3_CMOS_STR	92	2	1	RW	CMOS output impedance control. 0 = 50 Ω 1 = 25 Ω	READY
OUT4_MODE	98	0	4	RW	Software interpreted driver configuration. See Table 6.7 Driver Set Up Options on page 25 .	READY
OUT4_DIV	99	0	6	RW	Driver divider ratio. 0 = disabled 1-63 = divide value	READY
OUT4_SKEW	9A	0	3	RW	Skew control. Programmed as an unsigned integer. Can add delay of 35 ps/step up to 280 ps.	READY
OUT4_STOP_HIGHZ	9B	0	1	RW	Driver output state when stopped. 0 = low-Z 1 = high-Z	READY
OUT4_CMOS_INV	9B	4	1	RW	Sets the polarity of the two outputs in dual CMOS mode. 0 = no inversion 1 = OUT4b inverted	READY

Register Field Name	Address	Base	Bit Length	R/W/RW	Description	Device Mode
OUT4_CMOS_SLEW	9C	0	1	RW	Controls CMOS slew rate from fast to slow. 00 = fastest 01 = slow 10 = slower 11 = slowest	READY
OUT4_CMOS_STR	9C	2	1	RW	CMOS output impedance control. 0 = 50 Ω 1 = 25 Ω	READY
OUT5_MODE	9D	0	4	RW	Software interpreted driver configuration. See Table 6.7 Driver Set Up Options on page 25 .	READY
OUT5_DIV	9E	0	6	RW	Driver divider ratio. 0 = disabled 1-63 = divide value	READY
OUT5_SKEW	9F	0	3	RW	Skew control. Programmed as an unsigned integer. Can add delay of 35 ps/step up to 280 ps.	READY
OUT5_STOP_HIGHZ	A0	0	1	RW	Driver output state when stopped. 0 = low-Z 1 = high-Z	READY
OUT5_CMOS_INV	A0	4	1	RW	Sets the polarity of the two outputs in dual CMOS mode. 0 = no inversion 1 = OUT5b inverted	READY
OUT5_CMOS_SLEW	A1	0	1	RW	Controls CMOS slew rate from fast to slow. 00 = fastest 01 = slow 10 = slower 11 = slowest	READY
OUT5_CMOS_STR	A1	2	1	RW	CMOS output impedance control. 0 = 50 Ω 1 = 25 Ω	READY
OUT6_MODE	A7	0	4	RW	Software interpreted driver configuration. See Table 6.7 Driver Set Up Options on page 25 .	READY
OUT6_DIV	A8	0	6	RW	Driver divider ratio. 0 = disabled 1-63 = divide value	READY

Register Field Name	Address	Base	Bit Length	R/W/RW	Description	Device Mode
OUT6_SKEW	A9	0	3	RW	Skew control. Programmed as an unsigned integer. Can add delay of 35 ps/step up to 280 ps.	READY
OUT6_STOP_HIGHZ	AA	0	1	RW	Driver output state when stopped. 0 = low-Z 1 = high-Z	READY
OUT6_CMOS_INV	AA	4	1	RW	Sets the polarity of the two outputs in dual CMOS mode. 0 = no inversion 1 = OUT6b inverted	READY
OUT6_CMOS_SLEW	AB	0	1	RW	Controls CMOS slew rate from fast to slow. 00 = fastest 01 = slow 10 = slower 11 = slowest	READY
OUT6_CMOS_STR	AB	2	1	RW	CMOS output impedance control. 0 = 50 Ω 1 = 25 Ω	READY
OUT7_MODE	AC	0	4	RW	Software interpreted driver configuration. See Table 6.7 Driver Set Up Options on page 25 .	READY
OUT7_DIV	AD	0	6	RW	Driver divider ratio. 0 = disabled 1-63 = divide value	READY
OUT7_SKEW	AE	0	3	RW	Skew control. Programmed as an unsigned integer. Can add delay of 35 ps/step up to 280 ps.	READY
OUT7_STOP_HIGHZ	AF	0	1	RW	Driver output state when stopped. 0 = low-Z 1 = high-Z	READY
OUT7_CMOS_INV	AF	4	1	RW	Sets the polarity of the two outputs in dual CMOS mode. 0 = no inversion 1 = OUT7b inverted	READY

Register Field Name	Address	Base	Bit Length	R/W/RW	Description	Device Mode
OUT7_CMOS_SLEW	B0	0	1	RW	Controls CMOS slew rate from fast to slow. 00 = fastest 01 = slow 10 = slower 11 = slowest	READY
OUT7_CMOS_STR	B0	2	1	RW	CMOS output impedance control. 0 = 50 Ω 1 = 25 Ω	READY
OUT3_OE	B6	4	1	RW	Output enable control for OUT3	READY/ ACTIVE
OUT2_OE	B6	3	1	RW	Output enable control for OUT2	READY/ ACTIVE
OUT5_OE	B6	7	1	RW	Output enable control for OUT5	READY/ ACTIVE
OUT4_OE	B6	6	1	RW	Output enable control for OUT4	READY/ ACTIVE
OUT0_OE	B6	0	1	RW	Output enable control for OUT0	READY/ ACTIVE
OUT1_OE	B6	1	1	RW	Output enable control for OUT1	READY/ ACTIVE
OUT7_OE	B7	2	1	RW	Output enable control for OUT7	READY/ ACTIVE
OUT6_OE	B7	1	1	RW	Output enable control for OUT6	READY/ ACTIVE
CLKIN_2_CLK_SEL	73	0	2	RW	Select the CLKIN_2 input buffer mode. 0 = disabled 1 = differential 2 = CMOS DC 3 = CMOS AC	READY
CLKIN_3_CLK_SEL	74	0	2	RW	Select the CLKIN_3 input buffer mode. 0 = disabled 1 = differential 2 = CMOS DC 3 = CMOS AC	READY
IMUX_SEL	24	0	2	RW	Selects input mux clock source: 0 = Disabled 1 = XOSC 2 = CLKIN_2 3 = CLKIN_3	READY

10. Revision History

Revision 0.1

September 20, 2019

- Initial release.



ClockBuilder Pro

One-click access to Timing tools, documentation, software, source code libraries & more. Available for Windows and iOS (CBGo only).

www.silabs.com/CBPro



Timing Portfolio
www.silabs.com/timing



SW/HW
www.silabs.com/CBPro



Quality
www.silabs.com/quality



Support and Community
community.silabs.com

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®, ClockBuilder®, CMEMS®, DSPLL®, EFM®, EFM32®, EFR®, Ember®, Energy Micro, Energy Micro logo and combinations thereof, "the world's most energy friendly microcontrollers", Ember®, EZLink®, EZRadio®, EZRadioPRO®, Gecko®, Gecko OS, Gecko OS Studio, ISOModem®, Precision32®, ProSLIC®, Simplicity Studio®, SiPHY®, 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 Laboratories Inc.
400 West Cesar Chavez
Austin, TX 78701
USA

<http://www.silabs.com>