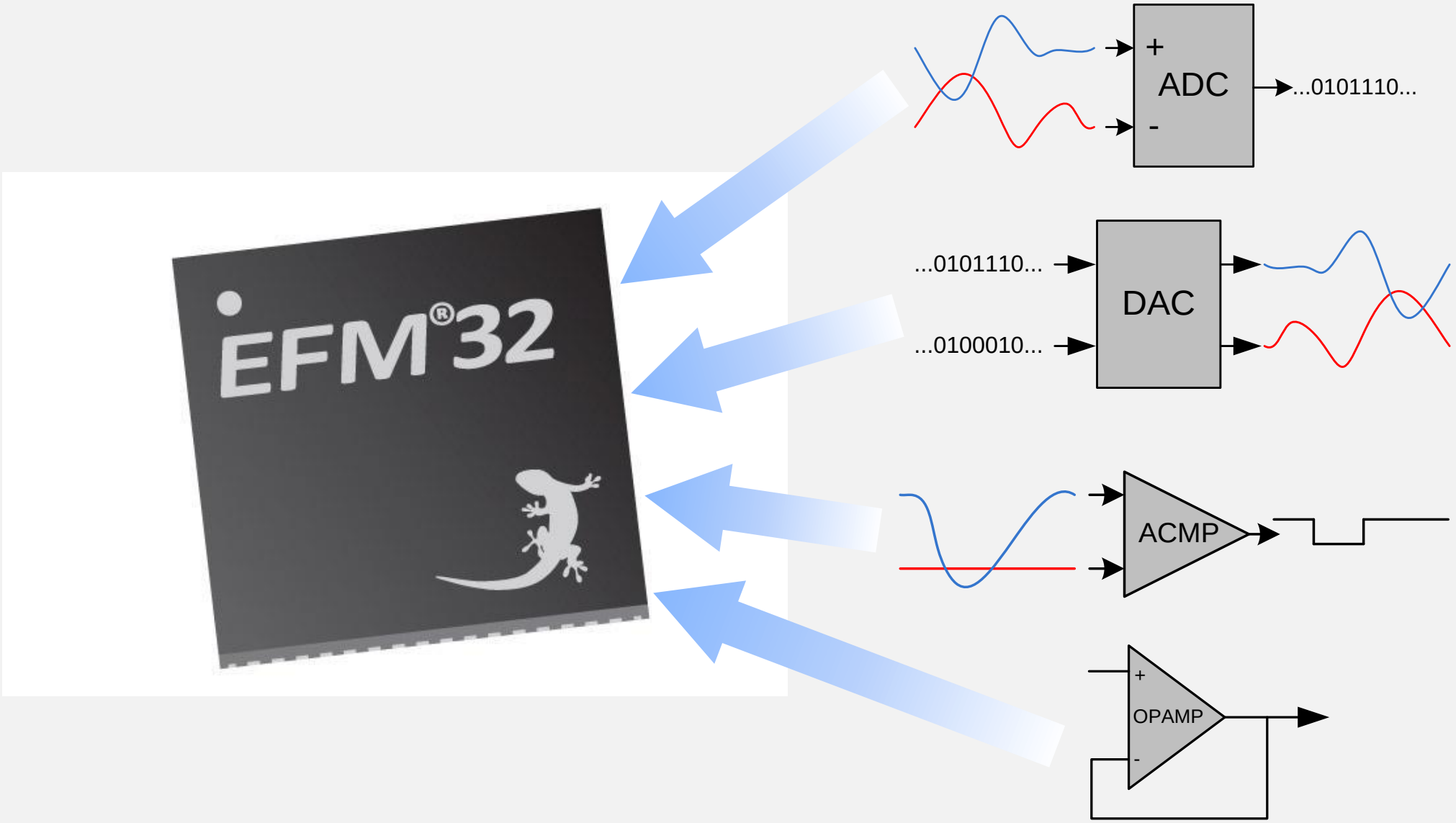
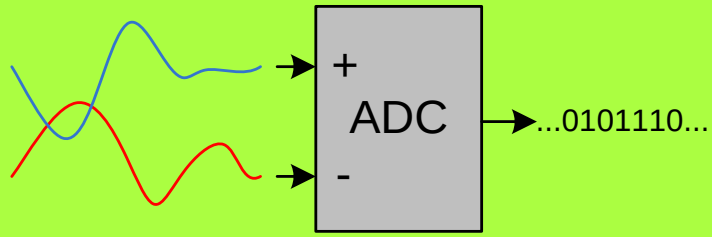
EFM32 Series 0: Analog Peripherals

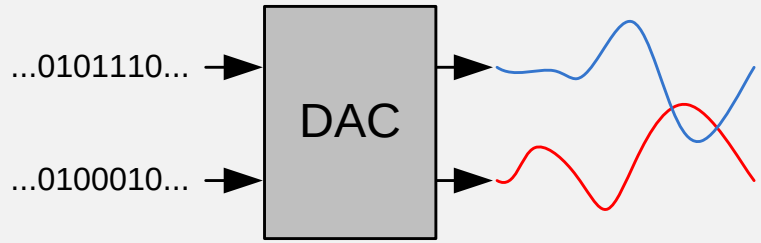
Analog Integration

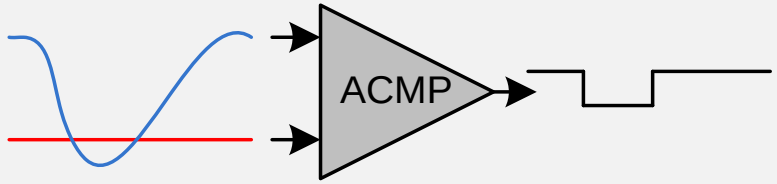


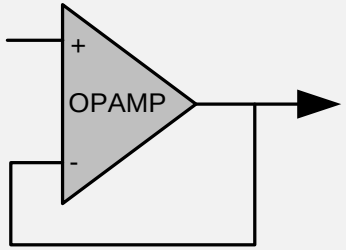
Analog-to-Digital Converter

- **12-bit @ 1 MSamples/s: 350 µA**
- **Up to 8 input channels**
  - **Integrated temperature sensor**
- **Up to 4096x oversampling in HW**
- **Internal/external references**
  - **5 µs settling**
- **Autonomous operation with DMA/PRS**
- **Separate single and scan mode configs**

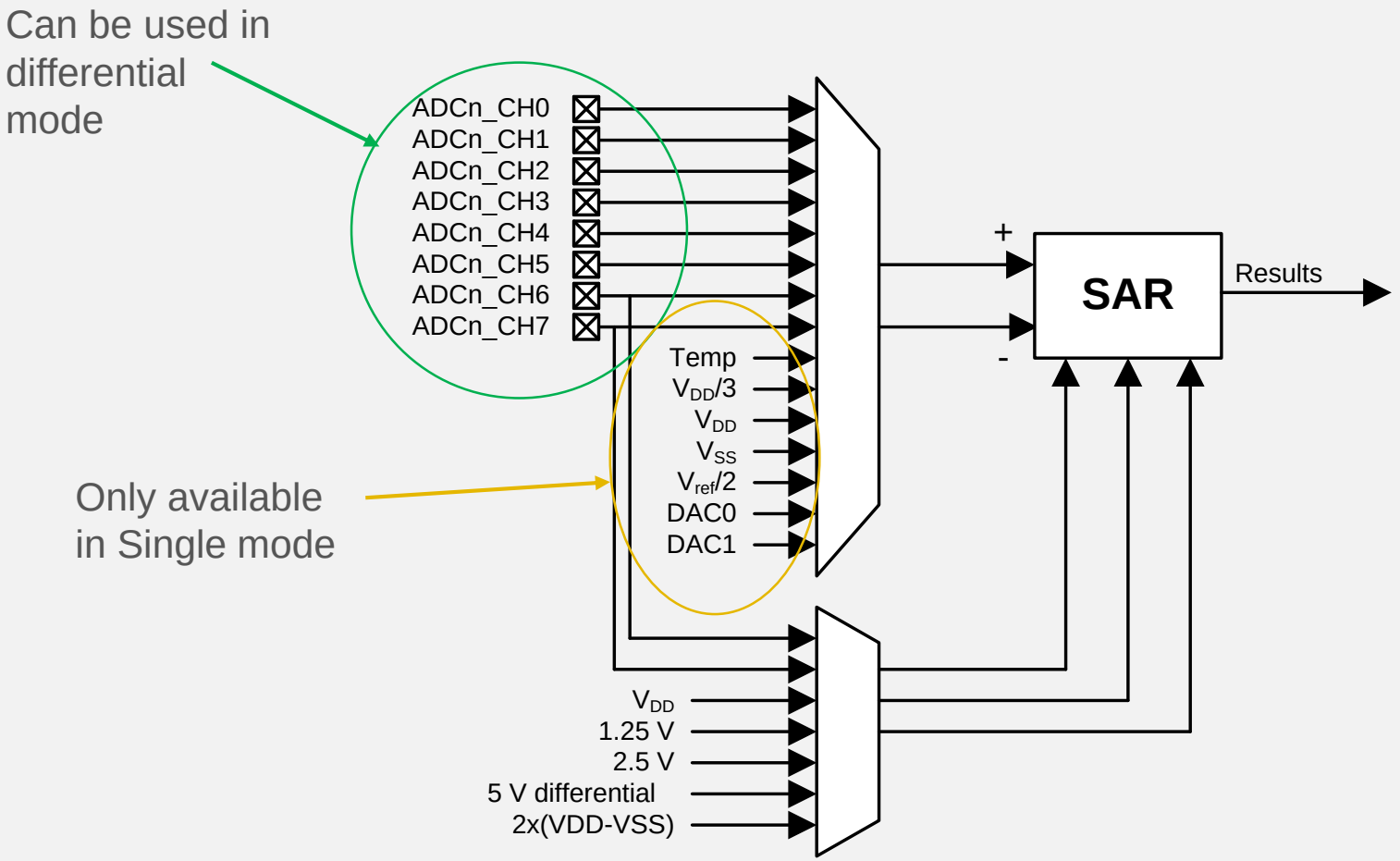








Input Options



Input and ref can be single ended or differential

Single vs Scan

Single conversion

INPUTSEL = CH3

Scan conversion

INPUTMASK = (CH0 | CH1 | CH4)

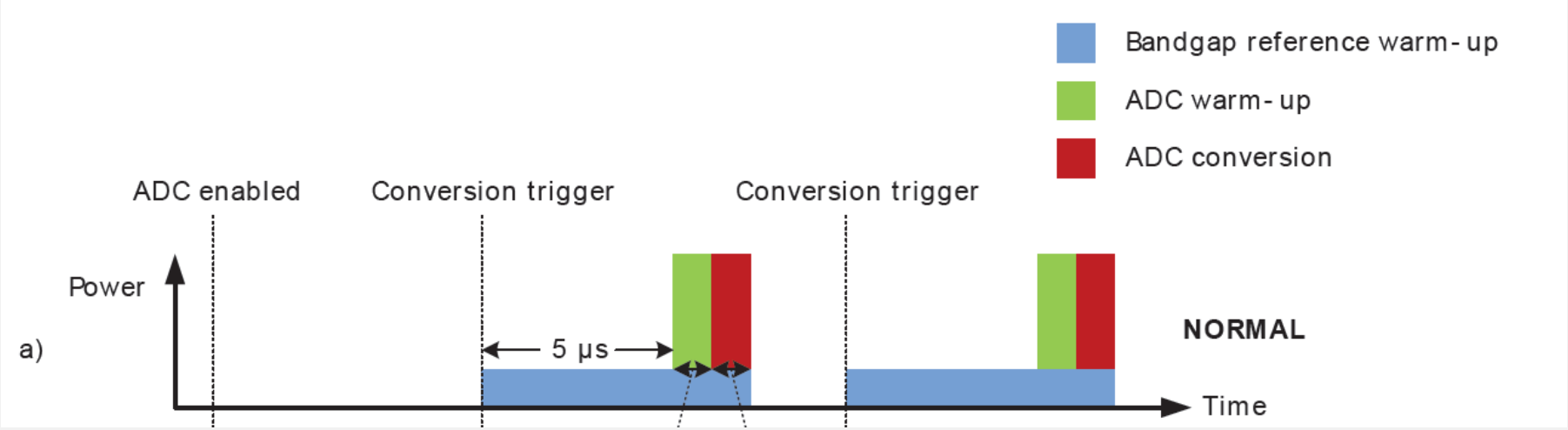
Warm Up

Mode	Function
NORMAL	ADC and references are shut off between samples
KEEPSCANREFWARM	Scan reference is kept warm
KEEPADCWARM	Both ADC and scan reference is kept warm
FASTBG	Bandgap warm-up is disabled

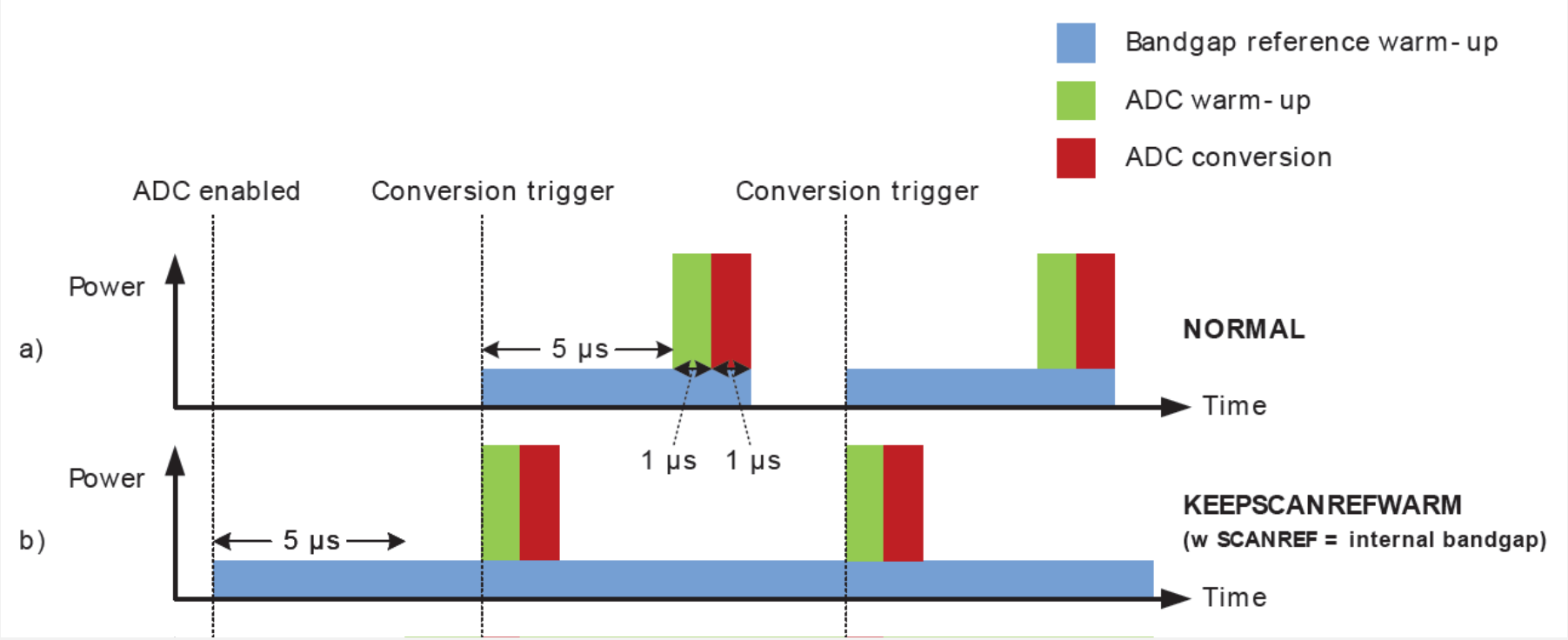
Reduces accuracy

Only applies to reference selected for Scan mode

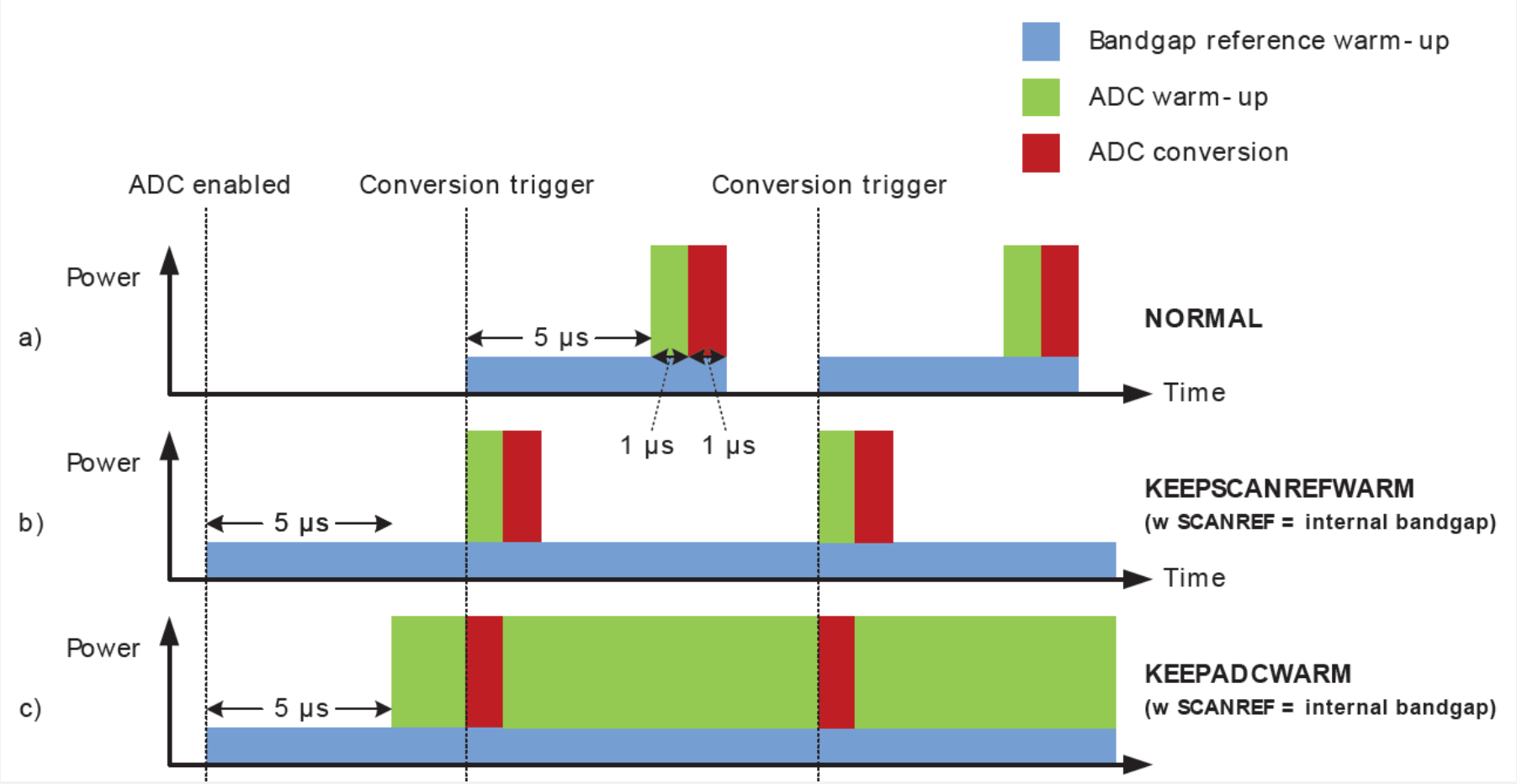
Warm Up


Bandgap reference warm-up
ADC warm-up
ADC conversion
ADC enabled
Conversion trigger
Conversion trigger
Power
a)
5 µs
NORMAL
Time

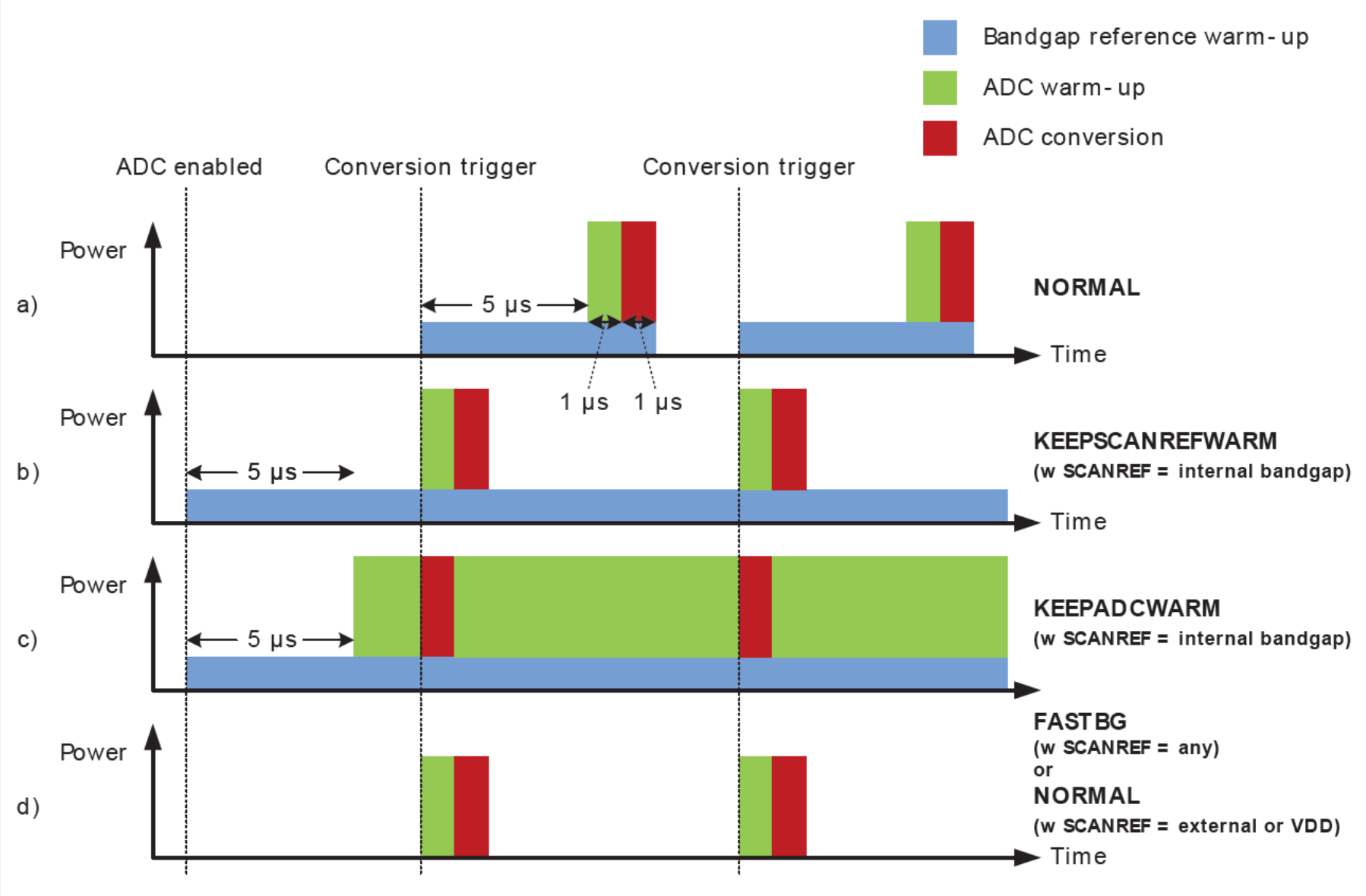
Warmp Up


Bandgap reference warm-up
ADC warm-up
ADC conversion
ADC enabled
Conversion trigger
Conversion trigger
Power
a)
5 µs
1 µs
1 µs
NORMAL
Time
Power
b)
5 µs
KEEPSCANREFWARM
(w SCANREF = internal bandgap)
Time

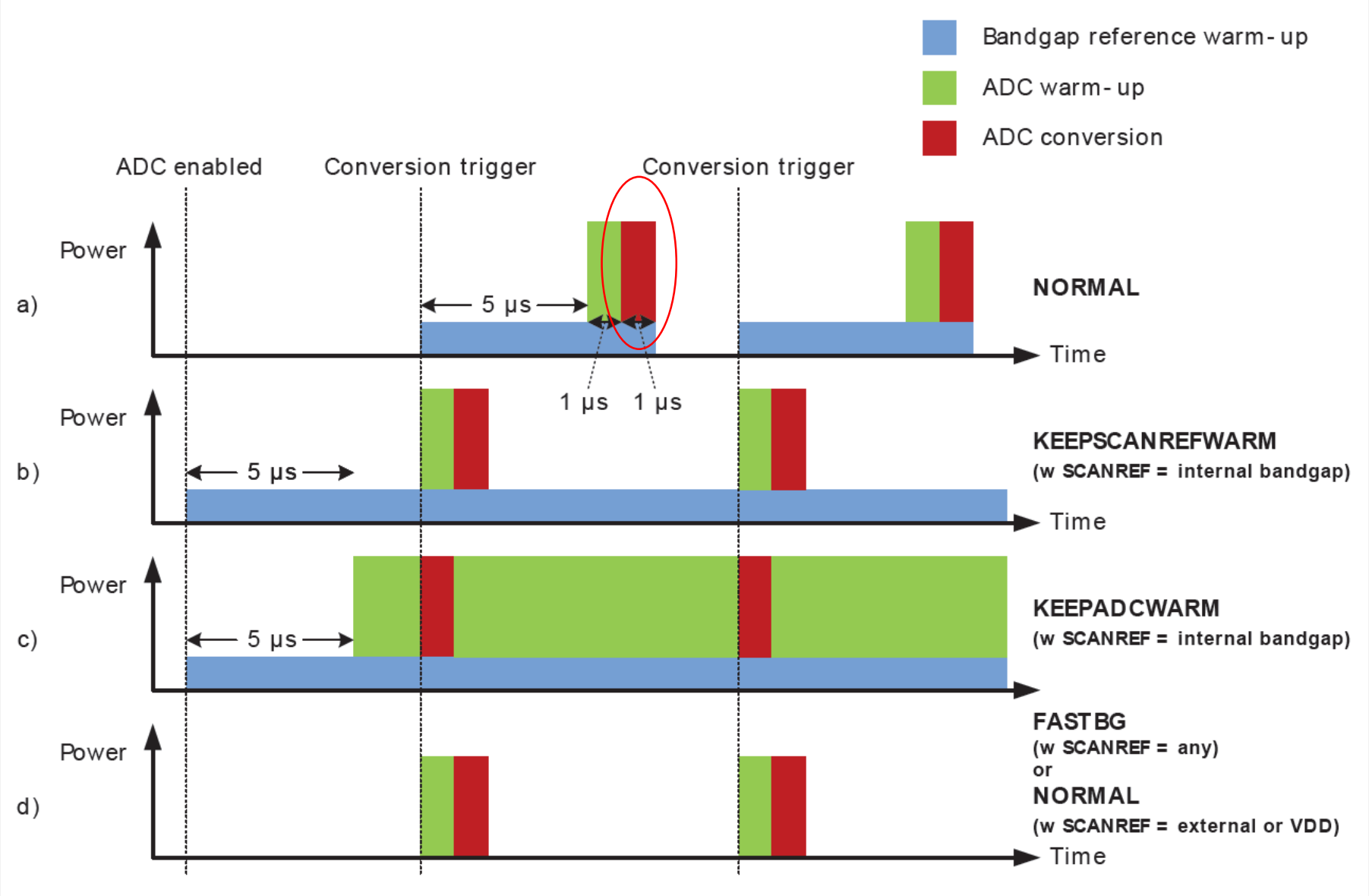
Warm Up


Bandgap reference warm-up
ADC warm-up
ADC conversion
ADC enabled
Conversion trigger
Conversion trigger
Power
a)
5 µs
1 µs
1 µs
NORMAL
Time
Power
b)
5 µs
KEEPSCANREFWARM
(w SCANREF = internal bandgap)
Time
Power
c)
5 µs
KEEPADCWARM
(w SCANREF = internal bandgap)

Warm Up

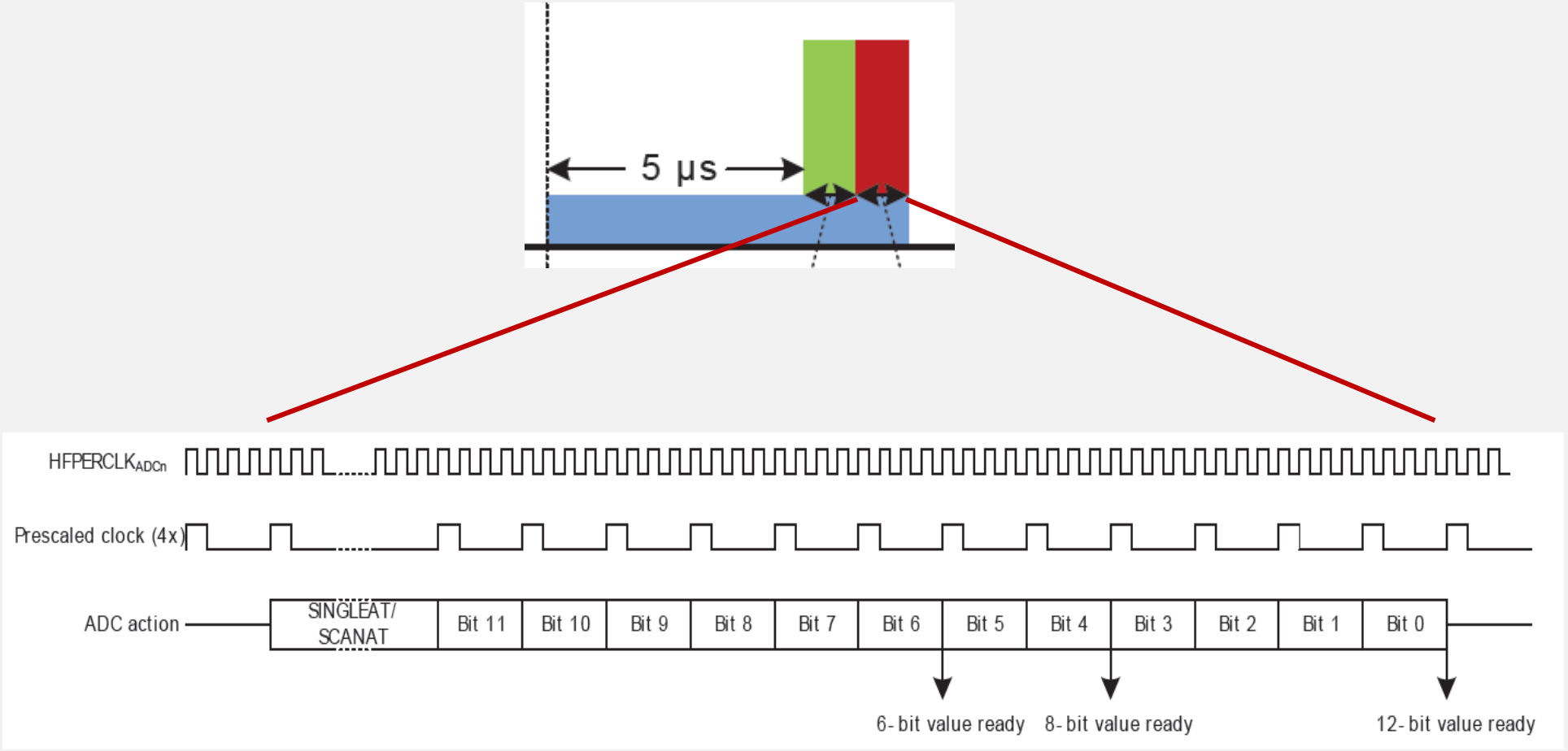

Bandgap reference warm-up
ADC warm-up
ADC conversion
ADC enabled
Conversion trigger
Conversion trigger
Power
a)
5 µs
1 µs
1 µs
Time
NORMAL
Power
b)
5 µs
Time
KEEPSCANREFWARM
(w SCANREF = internal bandgap)
Power
c)
5 µs
KEEPADCWARM
(w SCANREF = internal bandgap)
Power
d)
FASTBG
(w SCANREF = any)
or
NORMAL
(w SCANREF = external or VDD)
Time

Warm Up


Bandgap reference warm-up
ADC warm-up
ADC conversion
ADC enabled
Conversion trigger
Conversion trigger
Power
a)
5 µs
1 µs
1 µs
Time
NORMAL
Power
b)
5 µs
Time
KEEPSCANREFWARM
(w SCANREF = internal bandgap)
Power
c)
5 µs
KEEPADCWARM
(w SCANREF = internal bandgap)
Power
d)
FASTBG
(w SCANREF = any)
or
NORMAL
(w SCANREF = external or VDD)
Time

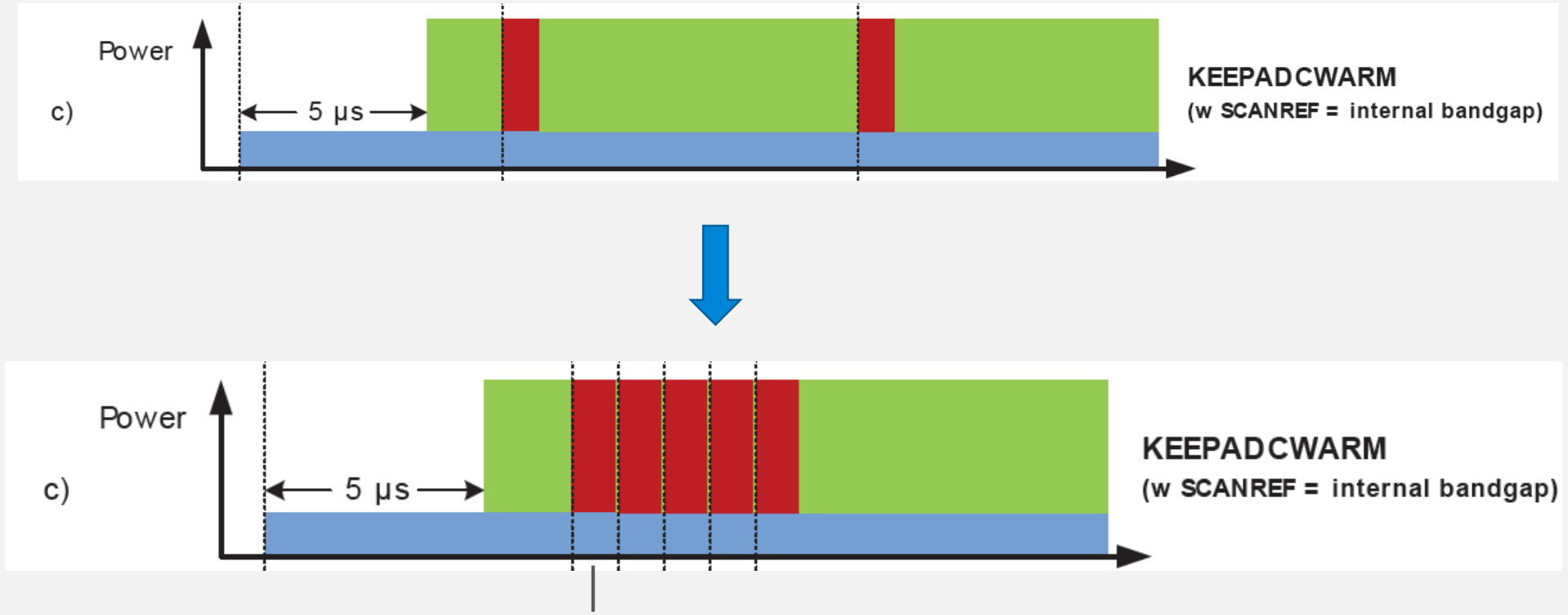
Conversion Timing

- Timing: One conversion, 13 cycles



Tuning for Speed

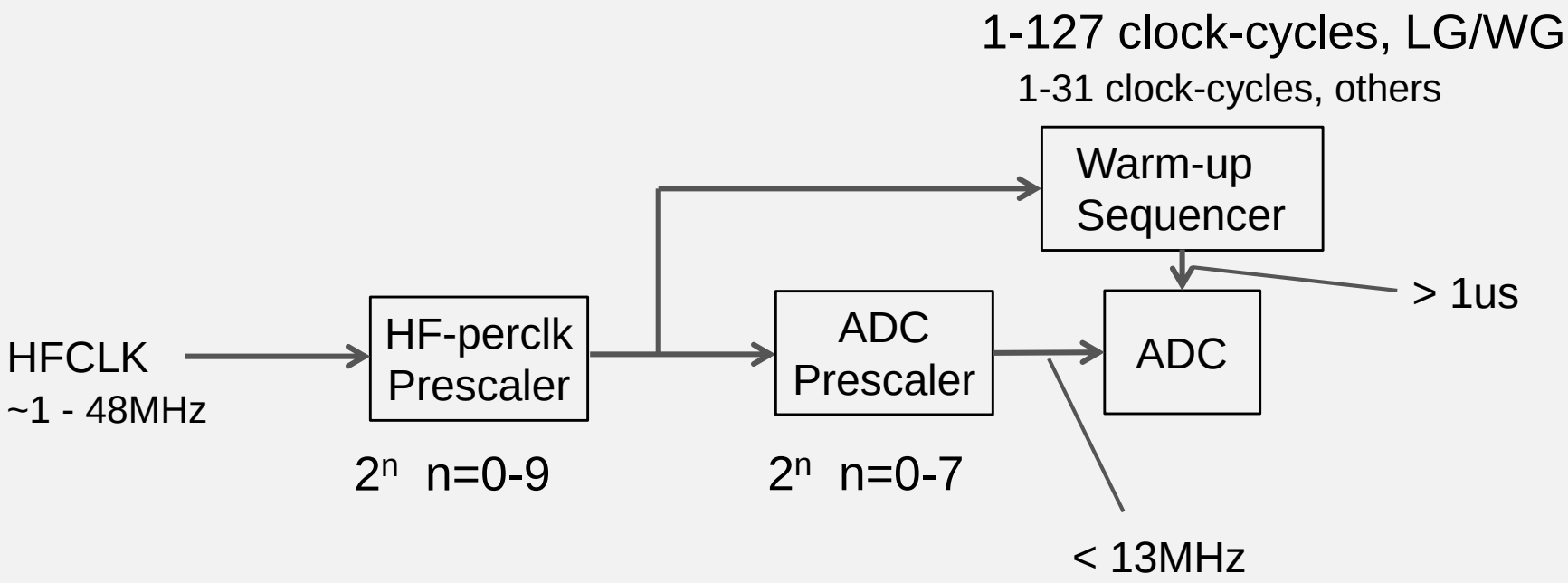
- Only way to achieve 12bit, 1MSPS:



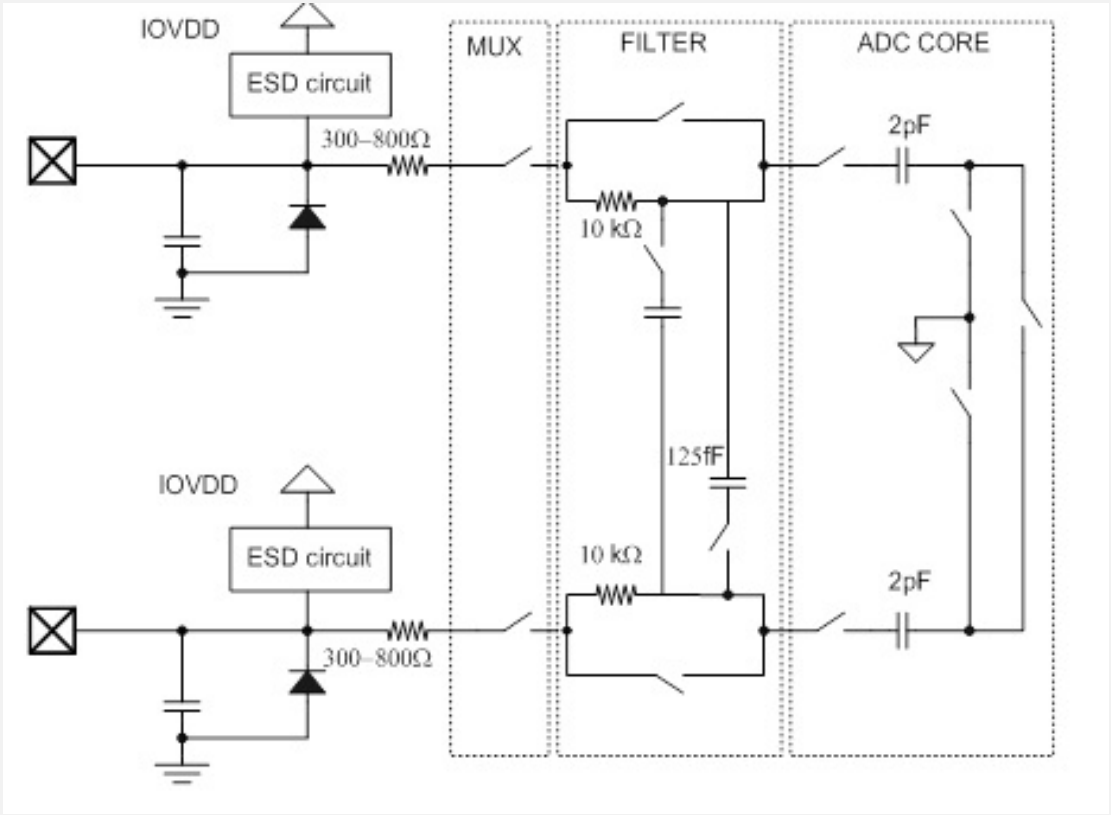
1us, 13 cycles at 13 MHz

13 MHz -> must have HFXO/HFRCO at 13, 26 or 39 MHz

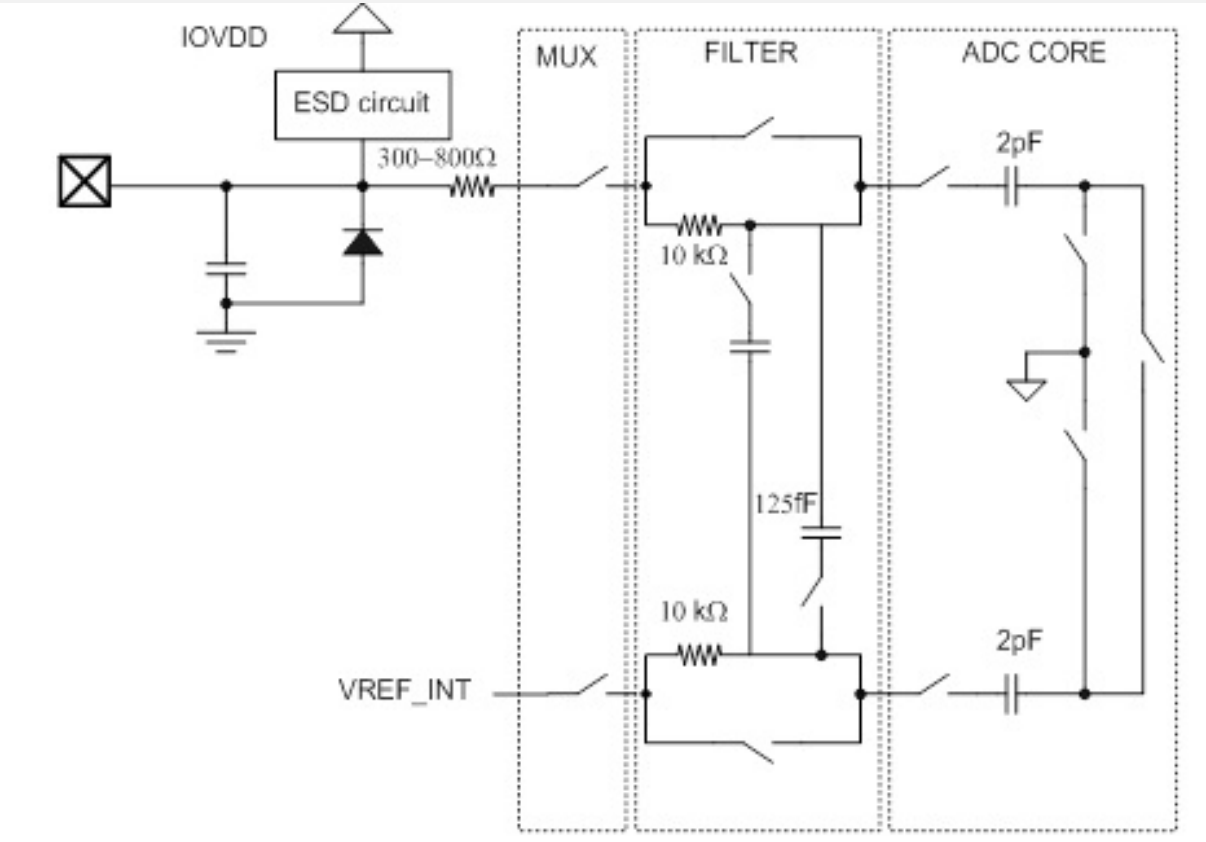
Prescalers


1-127 clock-cycles, LG/WG
1-31 clock-cycles, others
Warm-up Sequencer
> 1us
HFCLK
~1 - 48MHz
HF-perclk Prescaler
ADC Prescaler
ADC
2n n=0-9
2n n=0-7
< 13MHz

ADC Input Circuitry – Differential



ADC Input Circuitry – Single Ended



Hardware Oversampling

- **Up to 4096x oversampling in hardware**
- **Result is accumulated and right-shifted**
- **Not neccessarily true 16-bit result!**

OVS	Right shifts	Result # bits
2x	0	13
4x	0	14
8x	0	15
16x	0	16
32x	1	16
64x	2	16
128x	3	16
256x	4	16
512x	5	16
1024x	6	16
2048x	7	16
4096x	8	16

Calibration

- Internal references calibrated in production test
- Calibration values stored in DI page
- Values for 1.25V BG loaded at reset
- Emlib functions automatically load corresponding calibration values

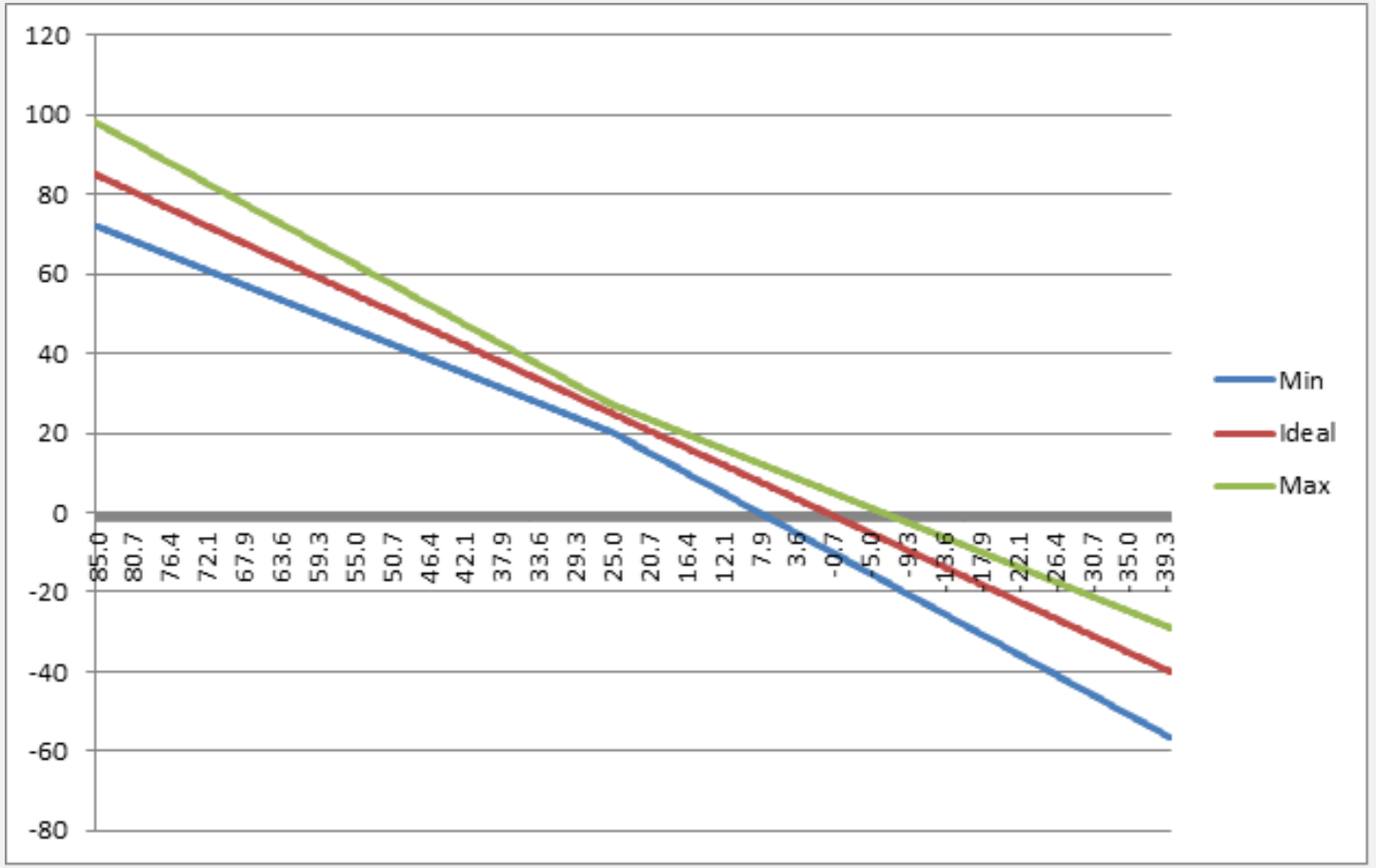
```
00340 /***************************************************************************
00360 void ADC_InitSingle(ADC_TypeDef *adc, const ADC_InitSingle_TypeDef *init)
00361 {
00362   uint32_t tmp;
00363
00364   EFM_ASSERT(ADC_REF_VALID(adc));
00365
00366   /* Make sure single conversion is not in progress */
00367   adc->CMD = ADC_CMD_SINGLESTOP;
00368
00369   /* Load proper calibration data depending on selected reference */
00370   ADC_CalibrateLoadSingle(adc, init->reference);
00371
00372   tmp = ((uint32_t)(init->prsSel) << _ADC_SINGLECTRL_PRSSEL_SHIFT) |
00373         ((uint32_t)(init->acqTime) << _ADC_SINGLECTRL_AT_SHIFT) |
00374         ((uint32_t)(init->reference) << _ADC_SINGLECTRL_REF_SHIFT) |
00375         ((uint32_t)(init->input) << _ADC_SINGLECTRL_INPUTSEL_SHIFT) |
00376         ((uint32_t)(init->resolution) << _ADC_SINGLECTRL_RES_SHIFT);
00377
00378   if (init->prsEnable)
00379   {
00380     tmp |= ADC_SINGLECTRL_PRSEN;
00381   }
00382
```

DI Page

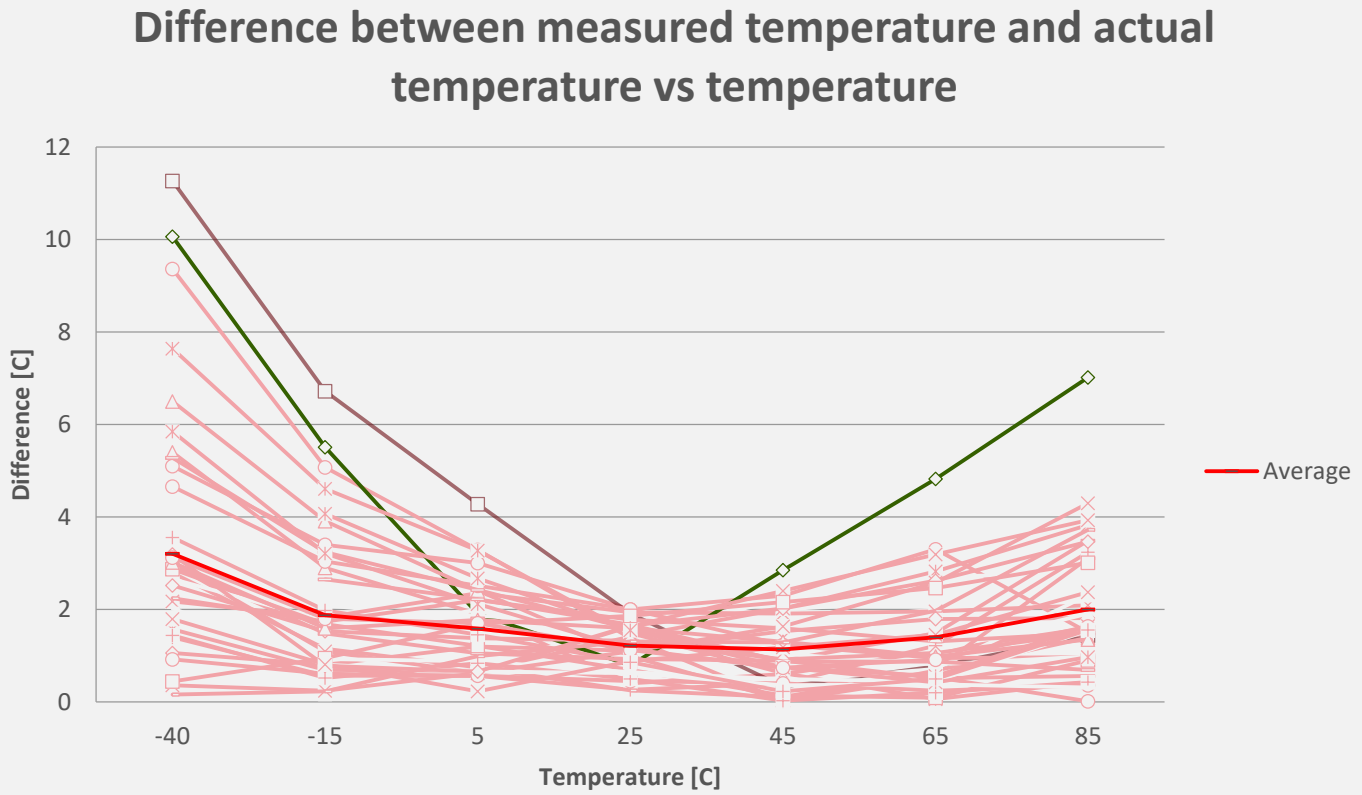
0x0FE081B4	ADC0_CAL_1V25	[14:8]: Gain for 1V25 reference, [6:0]: Offset for 1V25 reference.
0x0FE081B6	ADC0_CAL_2V5	[14:8]: Gain for 2V5 reference, [6:0]: Offset for 2V5 reference.
0x0FE081B8	ADC0_CAL_VDD	[14:8]: Gain for VDD reference, [6:0]: Offset for VDD reference.

Integrated Temperature Sensor

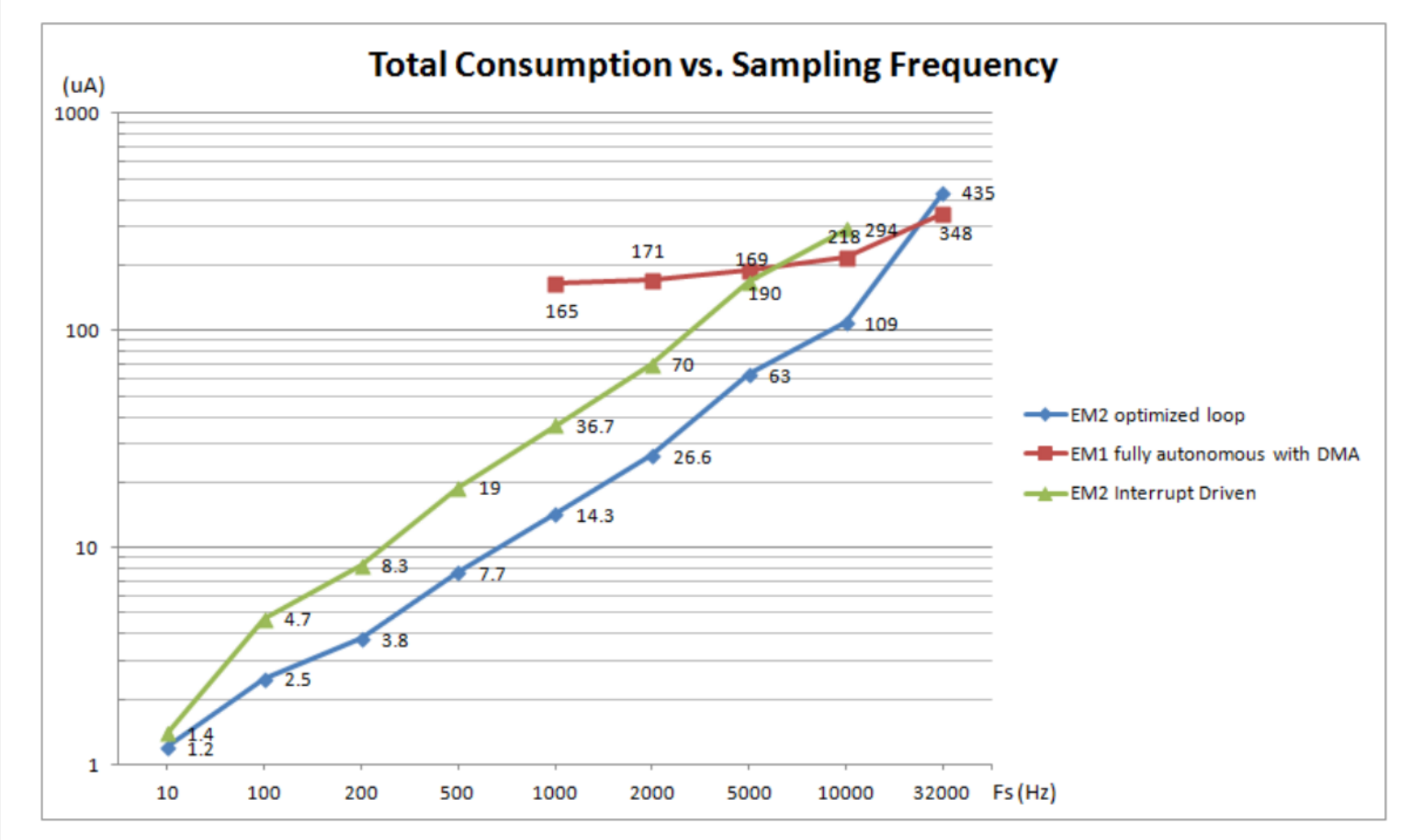
- Calibrated at 25 C



Temperature Sensor Error

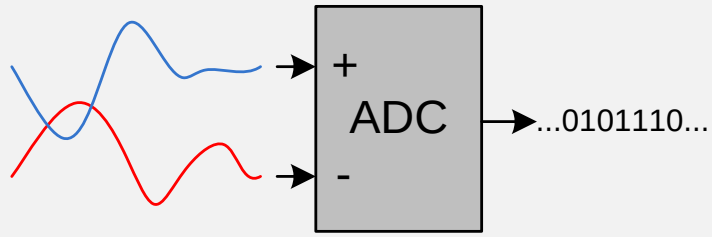


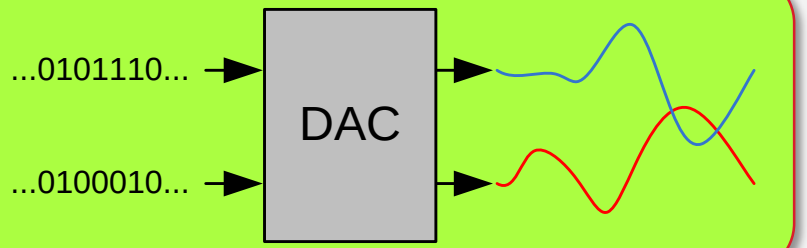
Energy Efficient ADC Sampling

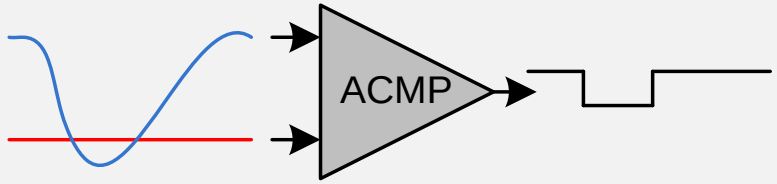


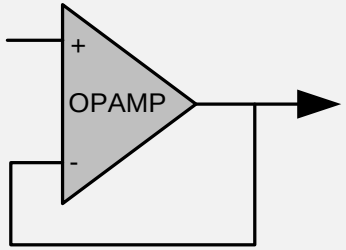
Digital-to-Analog Converter

- **12-bit resolution**
- **200 µA @ 500 kSamples/s**
- **2 independent channels**
- **Internal references**
- **Sine generation mode**
- **PRS/DMA Trigger**

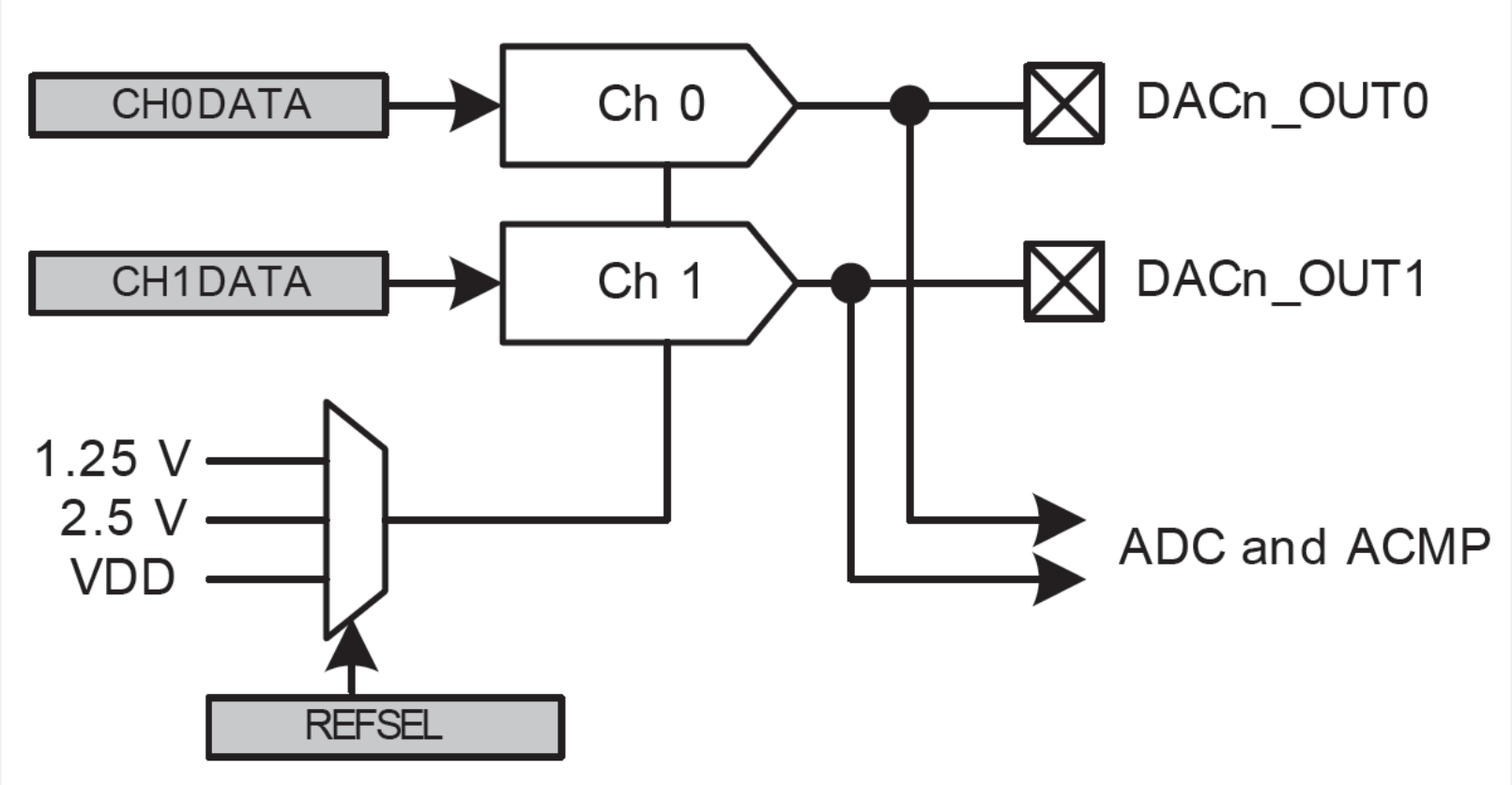








DAC



- Two independent channels,
- Internal references
- Output to internal peripherals

Output Modes

Single Ended

$$V_{OUT} = V_{DACn\_OUTx} - V_{SS} = V_{ref} \times CHxDATA/4095$$

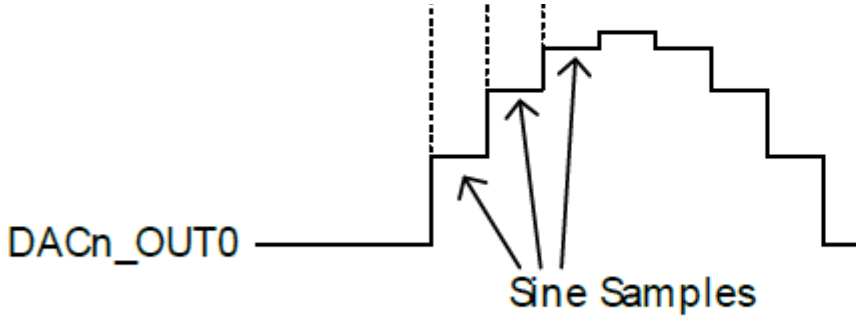
Differential

$$V_{OUT} = V_{DACn\_OUT1} - V_{DACn\_OUT0} = V_{ref} \times CH0DATA/2047$$

Common mode = VDD / 2

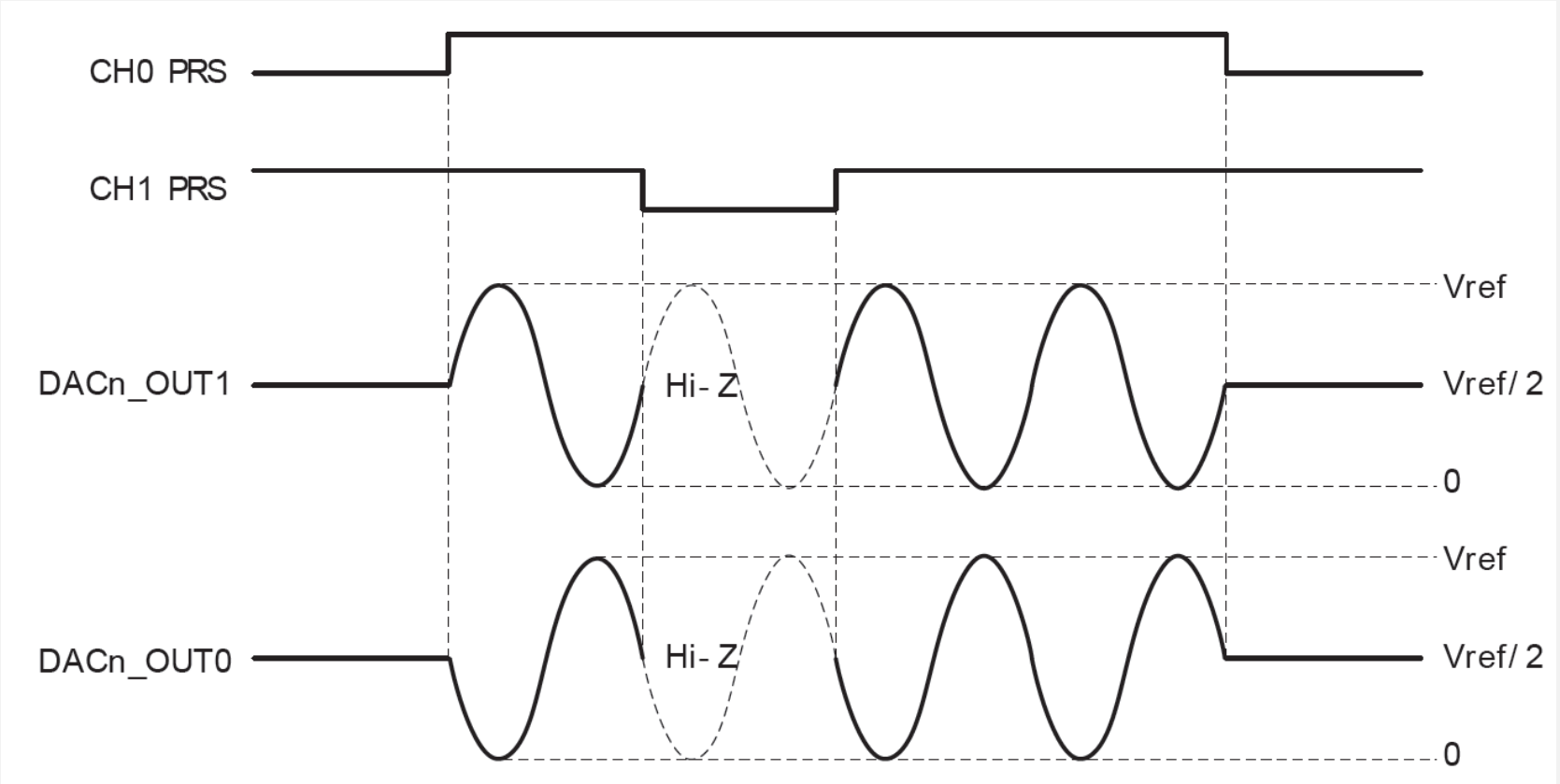
Signed integer

DAC built in Sine Wave Generator



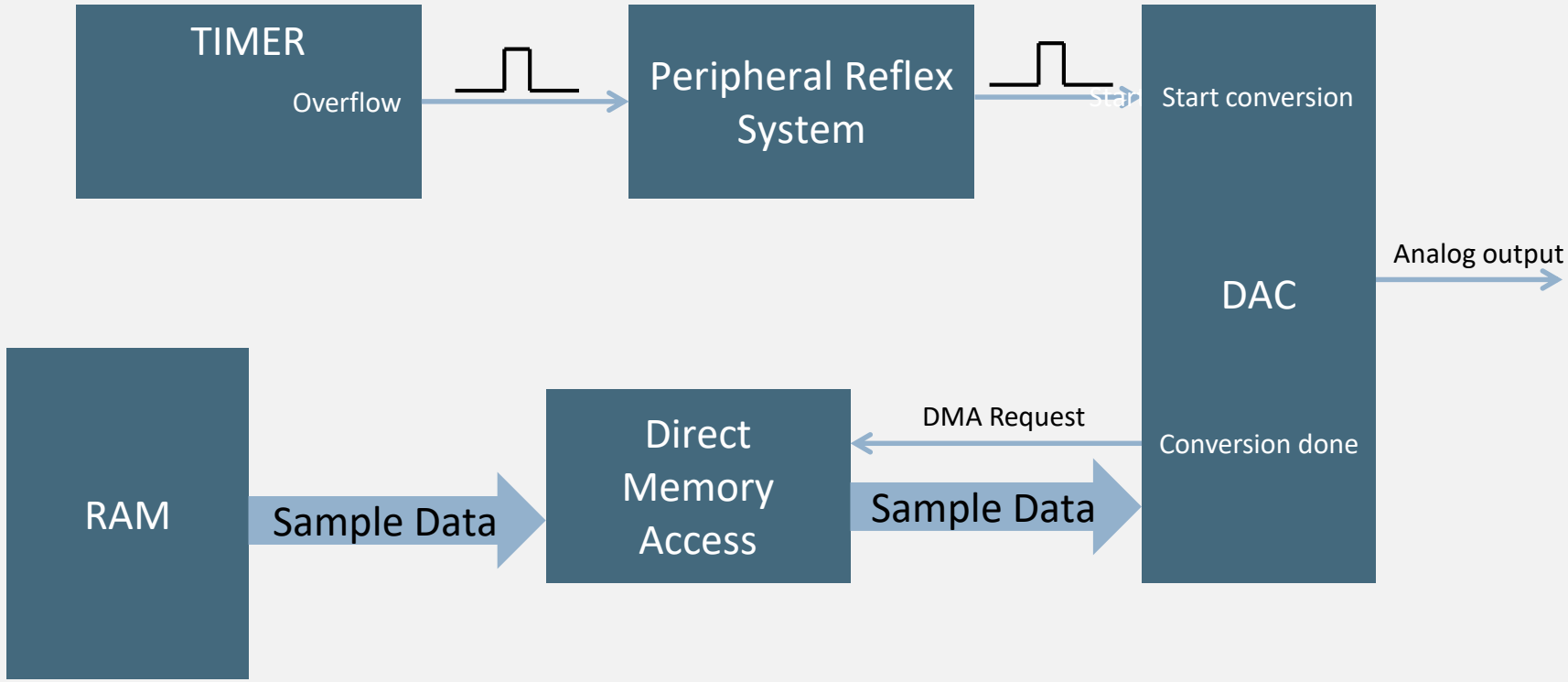
- 16 sample sine wave HW look-up table in DAC
- Frequency set by HFPERCLK supplied to DAC

DAC built in Sine Wave Generator

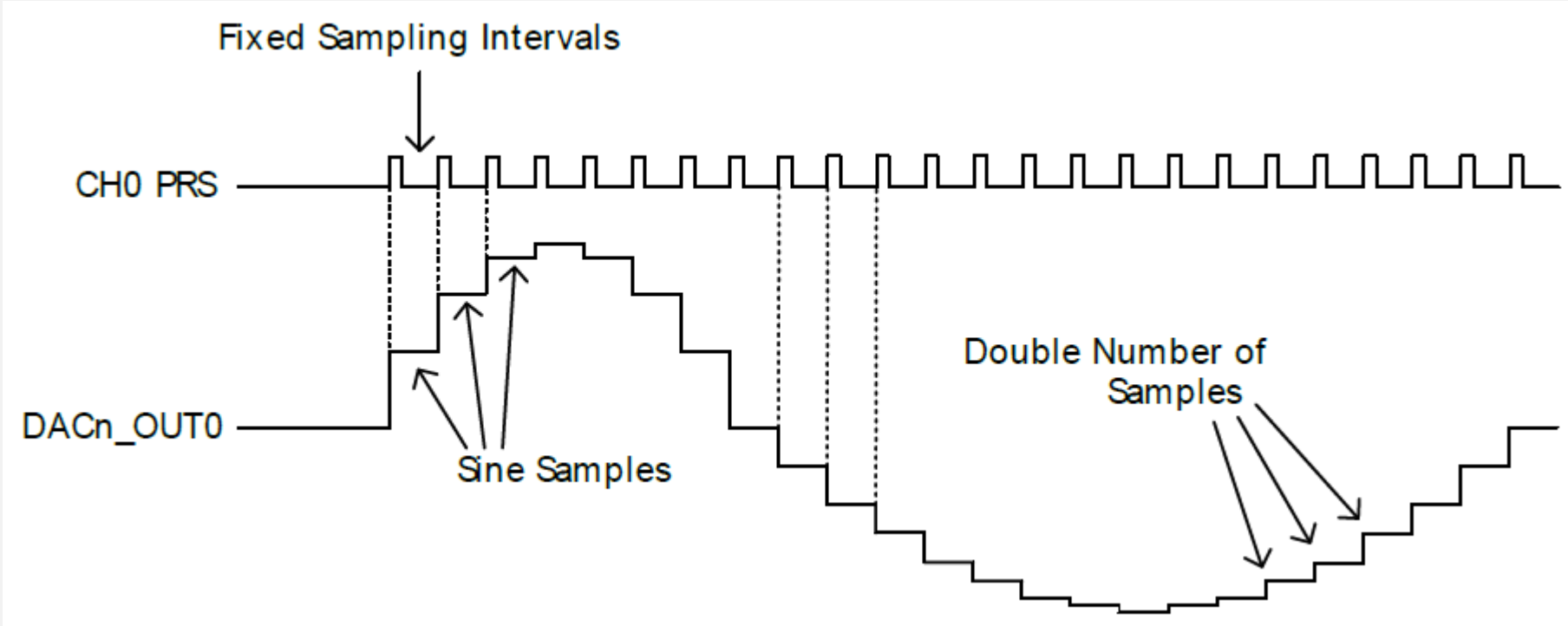


- Signal generation is controlled by PRS
- Output can be tri-stated or driven to Vdd/2
- Can be used for simple power line communication

DAC Signal with DMA and PRS



DAC Sine Wave Generator with PRS



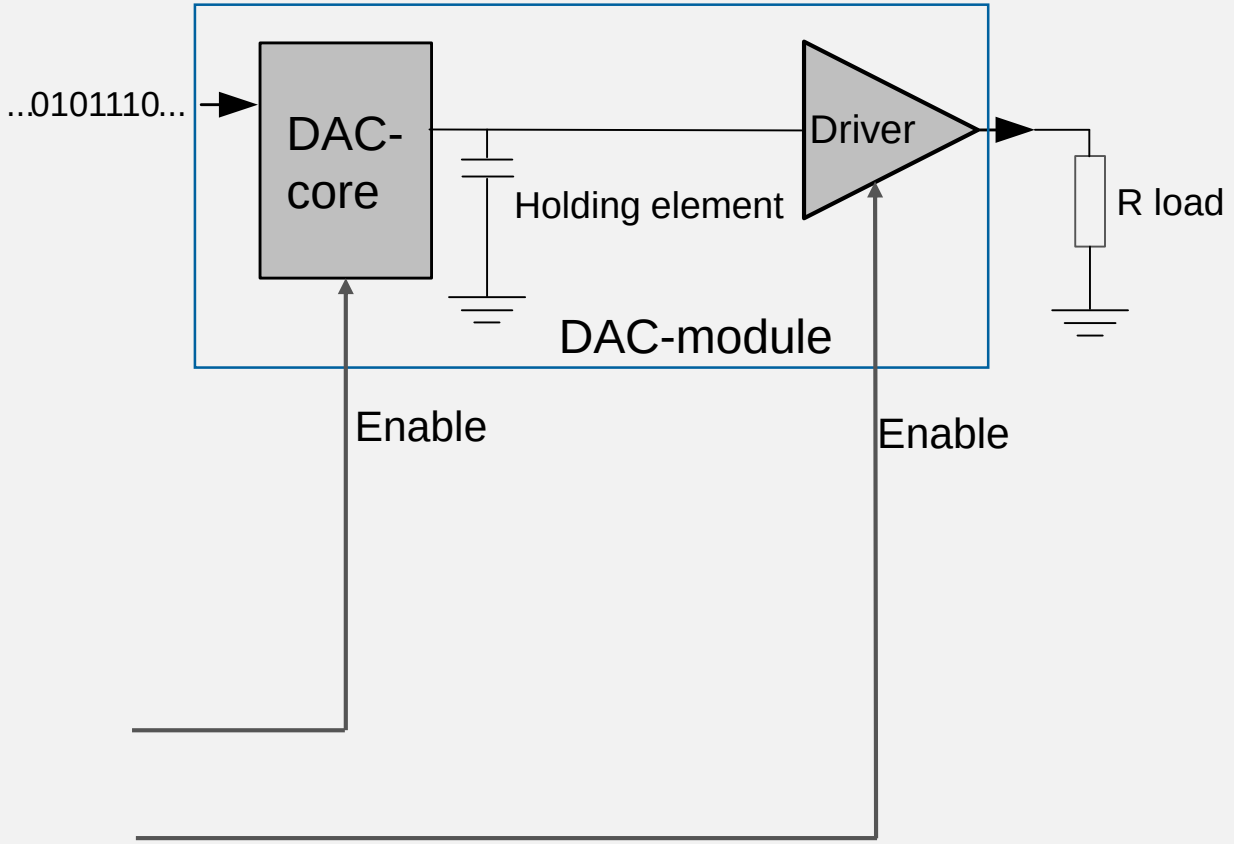
- DMA fetches samples from RAM, can produce any waveform, even music.
- Sample-frequency timed by PRS from TIMER

DAC Details

- What is continuous, sample-hold and sample-off modes?

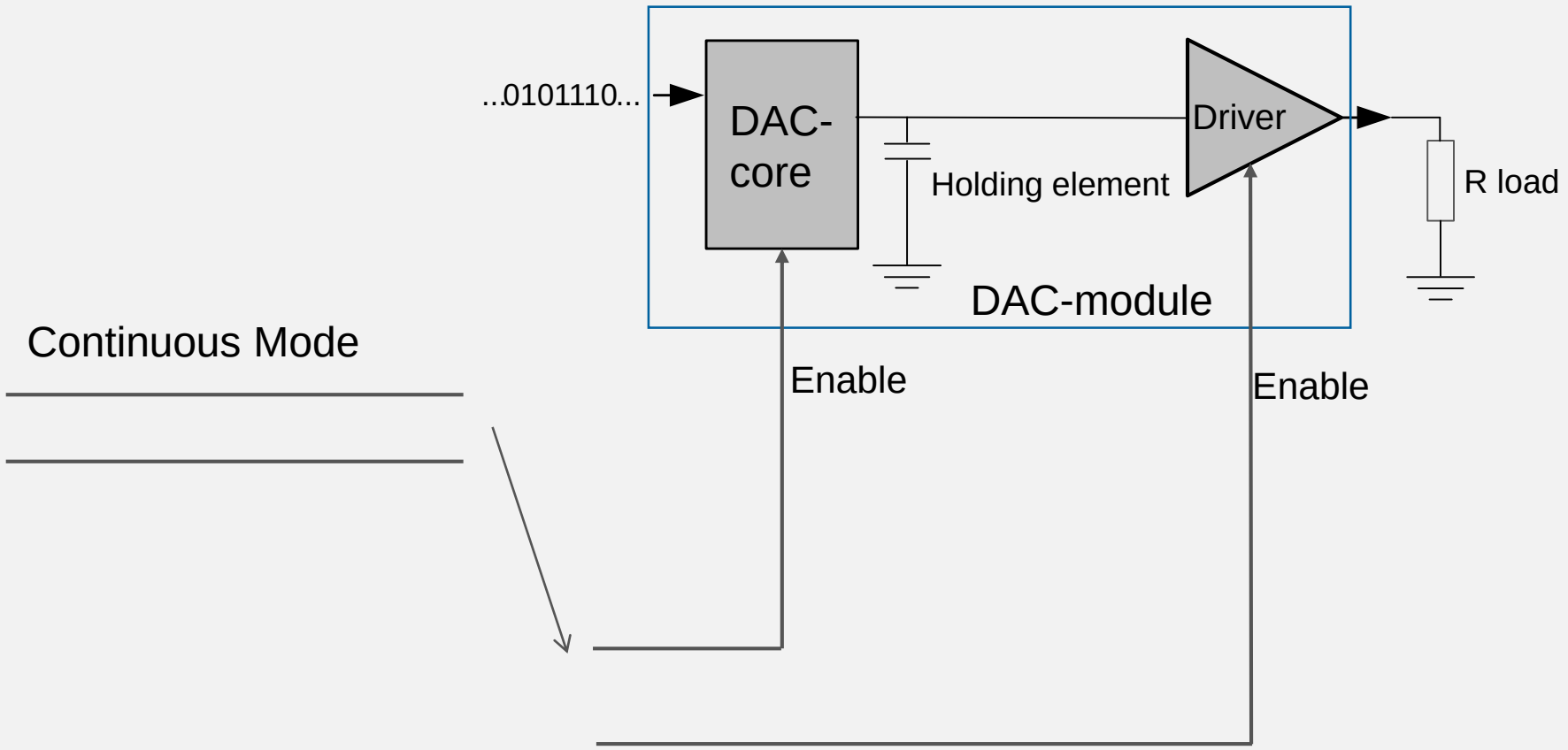
DAC Details

- What is continuous, sample-hold and sample-off modes?



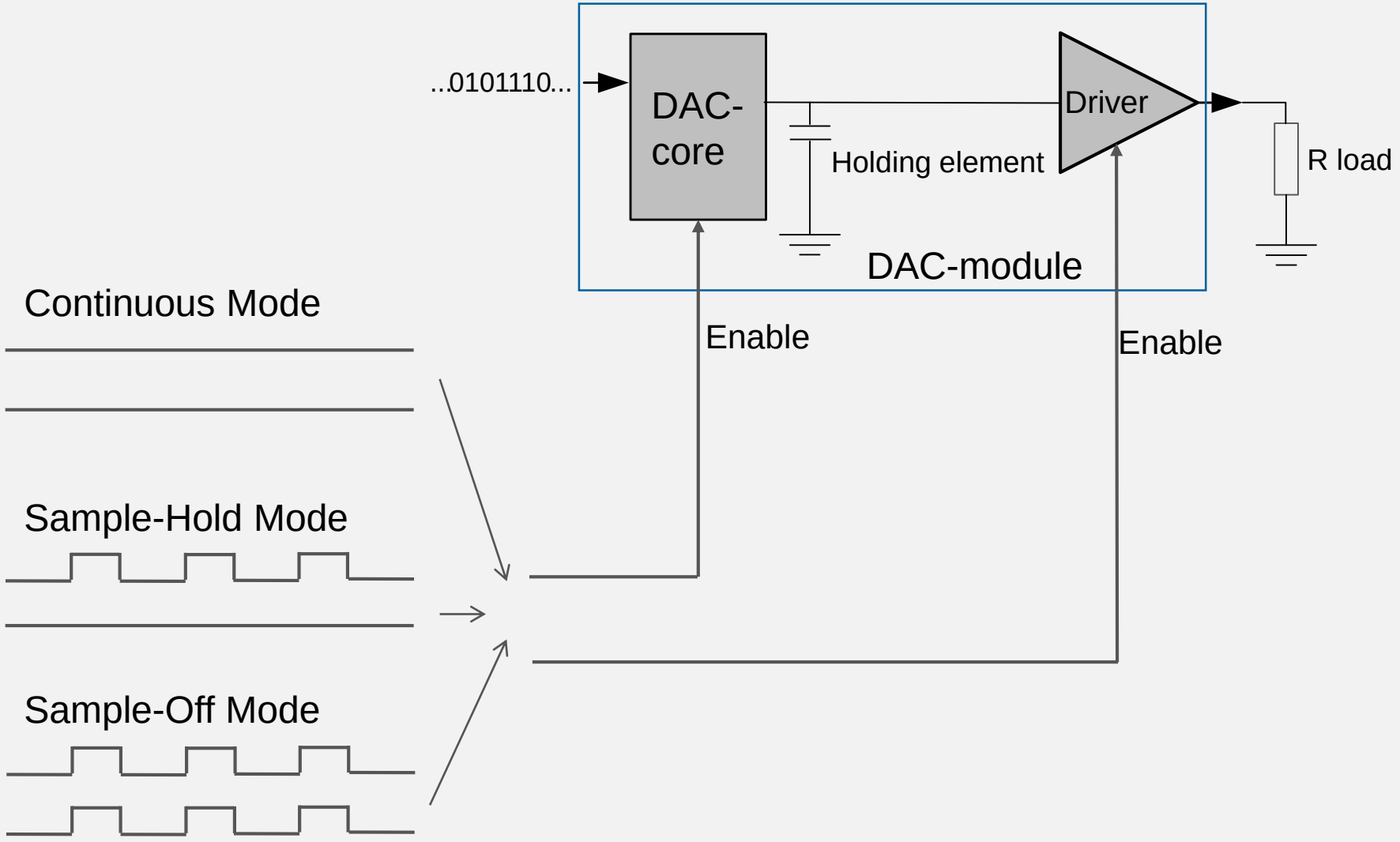
DAC Details

- What is continuous, sample-hold and sample-off modes?



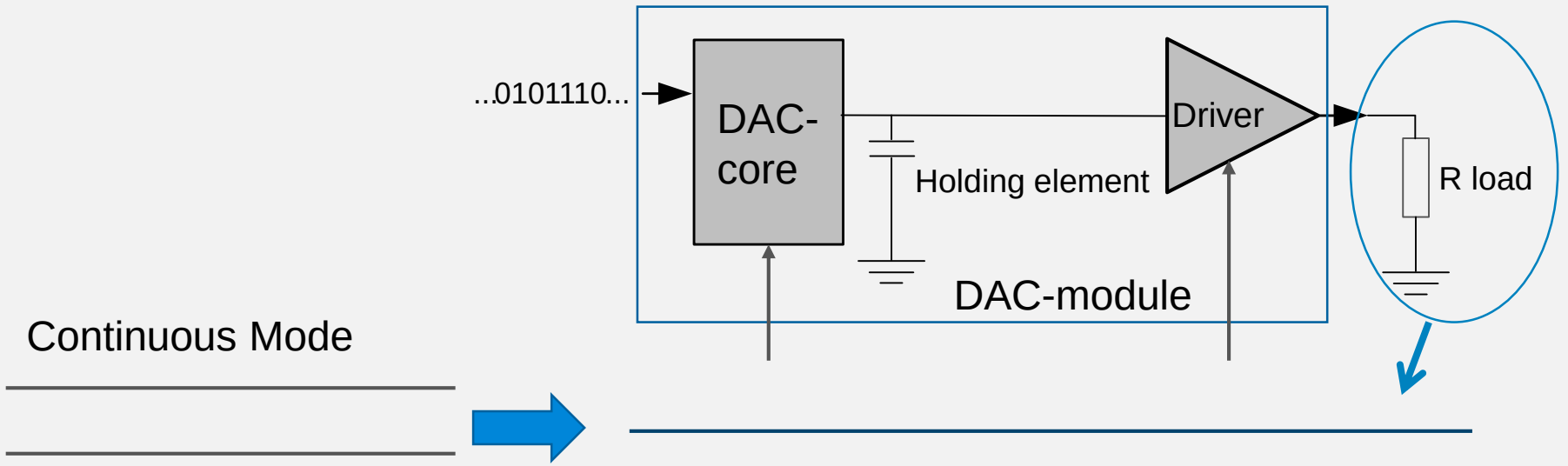
DAC Details

- What is continuous, sample-hold and sample-off modes?



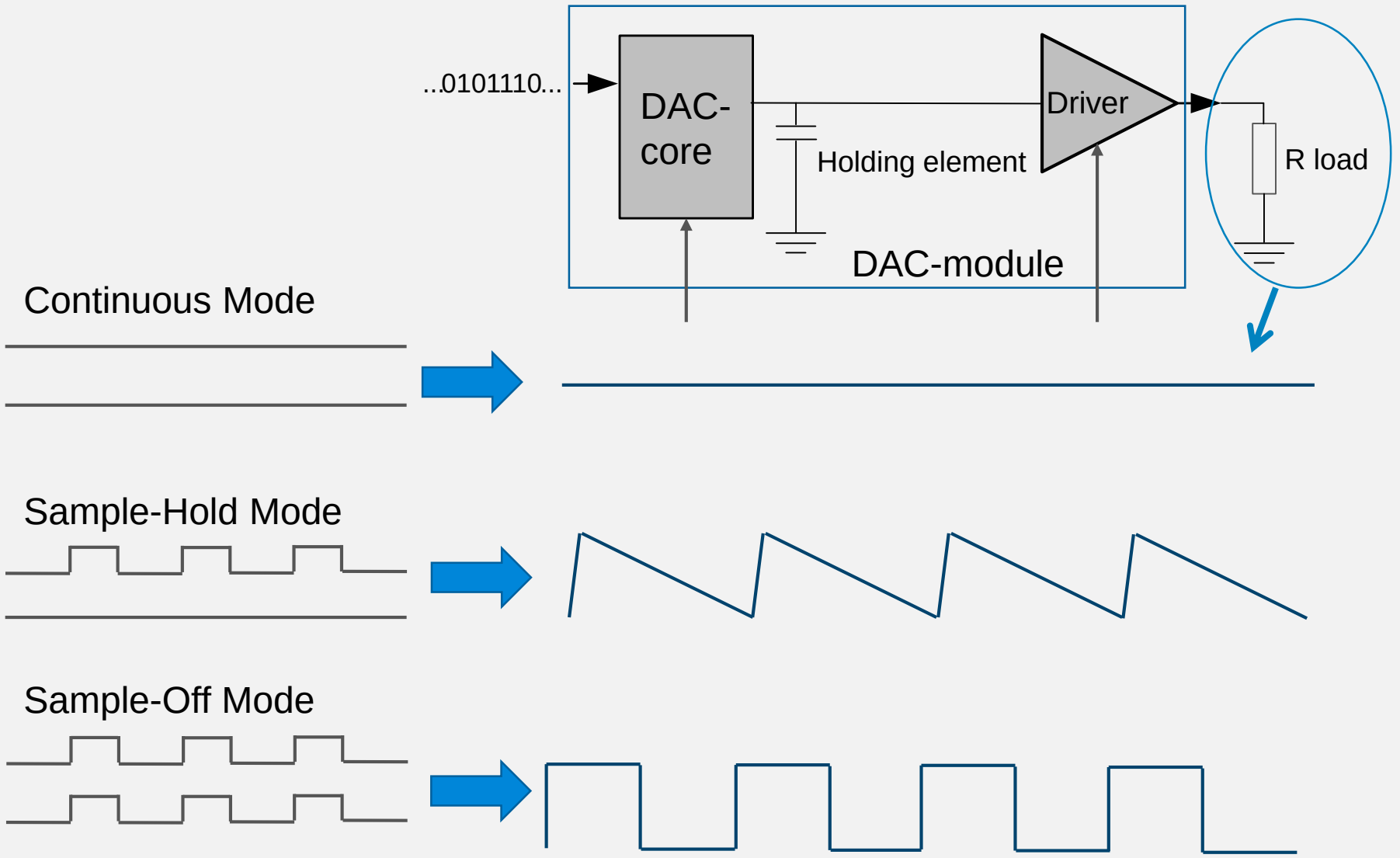
DAC Details

- What is continuous, sample-hold and sample-off modes?



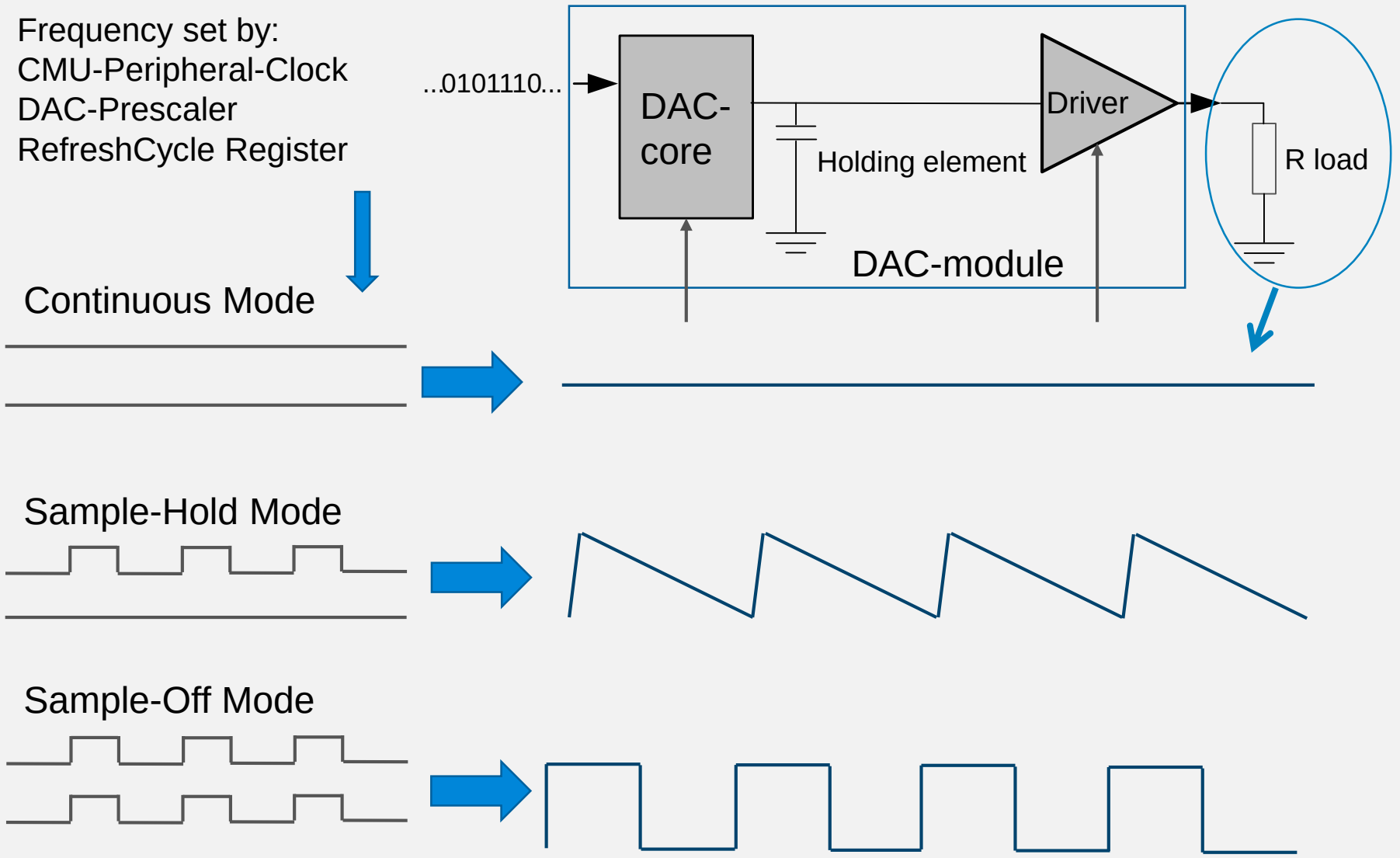
DAC Details

- What is continuous, sample-hold and sample-off modes?



DAC Details

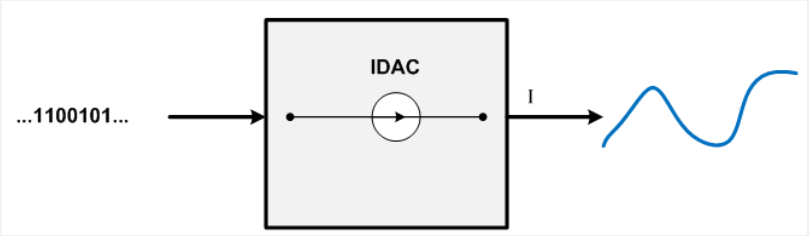
- What is continuous, sample-hold and sample-off modes?



Current DAC

IDAC Highlights

- Zero Gecko only
- Configurable 0.05-64 µA output
  - 4x32 steps
- Only 10 nA current overhead
- PRS triggered operation
- Biasing of external components
  - E.g. amplifiers



IDAC Ranges

- 4 Ranges
- 32 Steps in each range

Table 23.1. Range Selection

Range Select	Range Value [µA]	Step Size [nA]	Step Counts
0	0.05 - 1.6	50	32
1	1.6 - 4.7	100	32
2	0.5 - 16	500	32
3	2 - 64	2000	32

Calibration

- Middle of each range is calibrated in production test
- Calibration values stored in DI page
- Tuning values automatically loaded by emlib

```
00159 /*****************************************
00175 void IDAC_RangeSet(IDAC_TypeDef *idac, const II
00176 {
00177   uint32_t tmp;
00178
00179   EFM_ASSERT(IDAC_REF_VALID(idac));
00180   EFM_ASSERT((range >> _IDAC_CURPROG_RANGESEL_S
00181
00182   /* Load proper calibration data depending on
00183   switch ((IDAC_Range_TypeDef) range)
00184   {
00185   case idacCurrentRange0:
00186     idac->CAL = (DEVINFO->IDAC0CAL0 & _DEVINFO_
00187     break;
00188   case idacCurrentRange1:
00189     idac->CAL = (DEVINFO->IDAC0CAL0 & _DEVINFO_
00190     break;
00191   case idacCurrentRange2:
00192     idac->CAL = (DEVINFO->IDAC0CAL0 & _DEVINFO_
00193     break;
00194   case idacCurrentRange3:
00195     idac->CAL = (DEVINFO->IDAC0CAL0 & _DEVINFO_
00196     break;
00197   }
00198
```

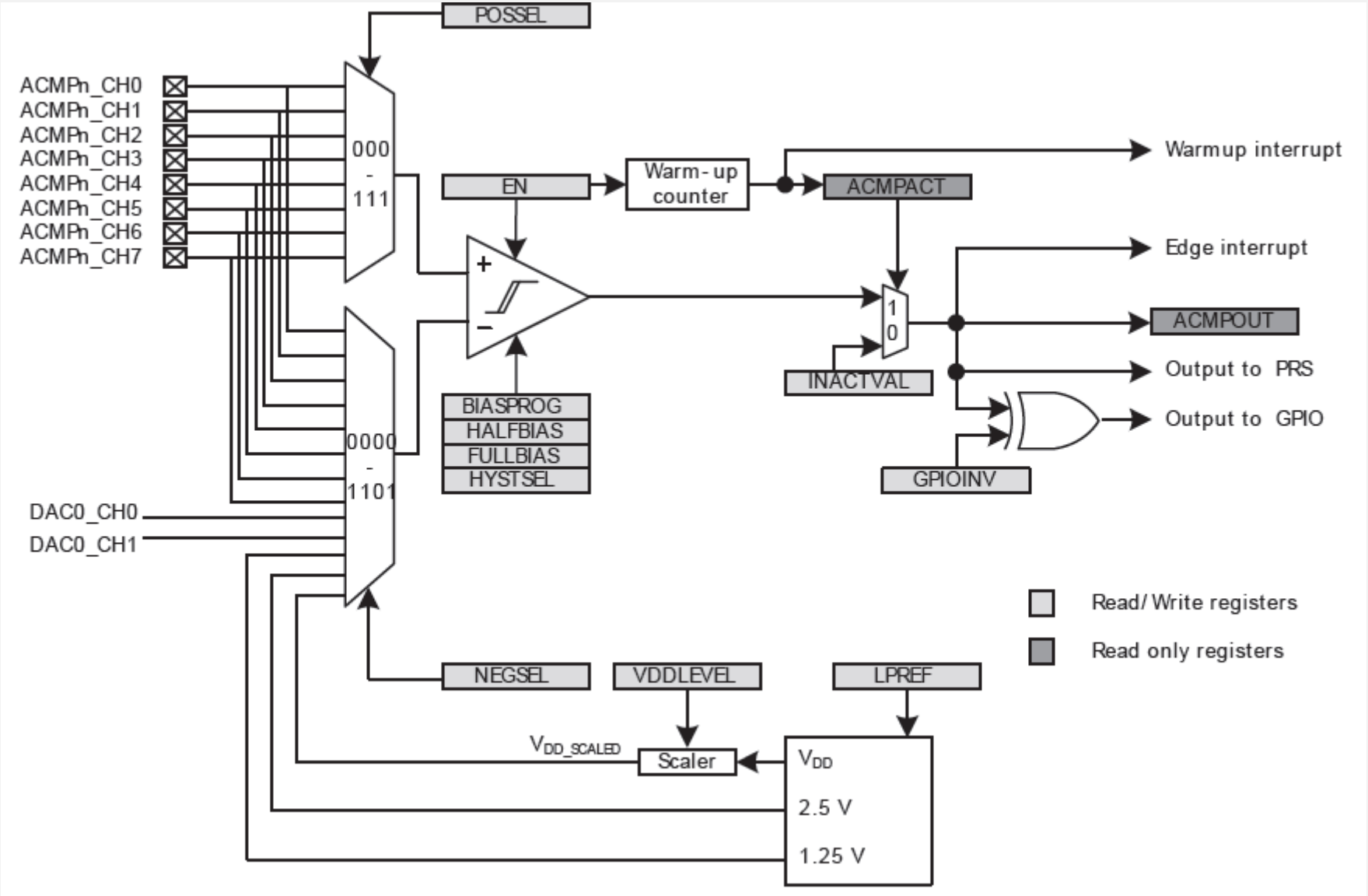
0x0FE081C8	IDAC0_CAL_RANGE0	[7:0]: Current range 0 tuning.
0x0FE081C9	IDAC0_CAL_RANGE1	[7:0]: Current range 1 tuning.
0x0FE081CA	IDAC0_CAL_RANGE2	[7:0]: Current range 2 tuning.
0x0FE081CB	IDAC0_CAL_RANGE3	[7:0]: Current range 3 tuning.

Duty Cycle

- IDAC can be duty-cycled at 4 Hz
- Sources current at low overhead
- Default enabled in EM2/EM3
- Default disabled in EM0/EM1
- Cannot sink in duty-cycle mode

Bit	Name	Reset	Access	Description
31:2	*Reserved*	*To ensure compatibility with future devices, always write bits to 0. M*		
1	EM2DUTYCYCLEDIS	0	RW	**EM2/EM3 Duty Cycle Disable.**
	Set to disable duty cycling in EM2 and EM3.			
0	DUTYCYCLEEN	0	RW	**Duty Cycle Enable.**
	Set to always enable duty cycling. Will override EM2DUTYCYCLEDIS.			

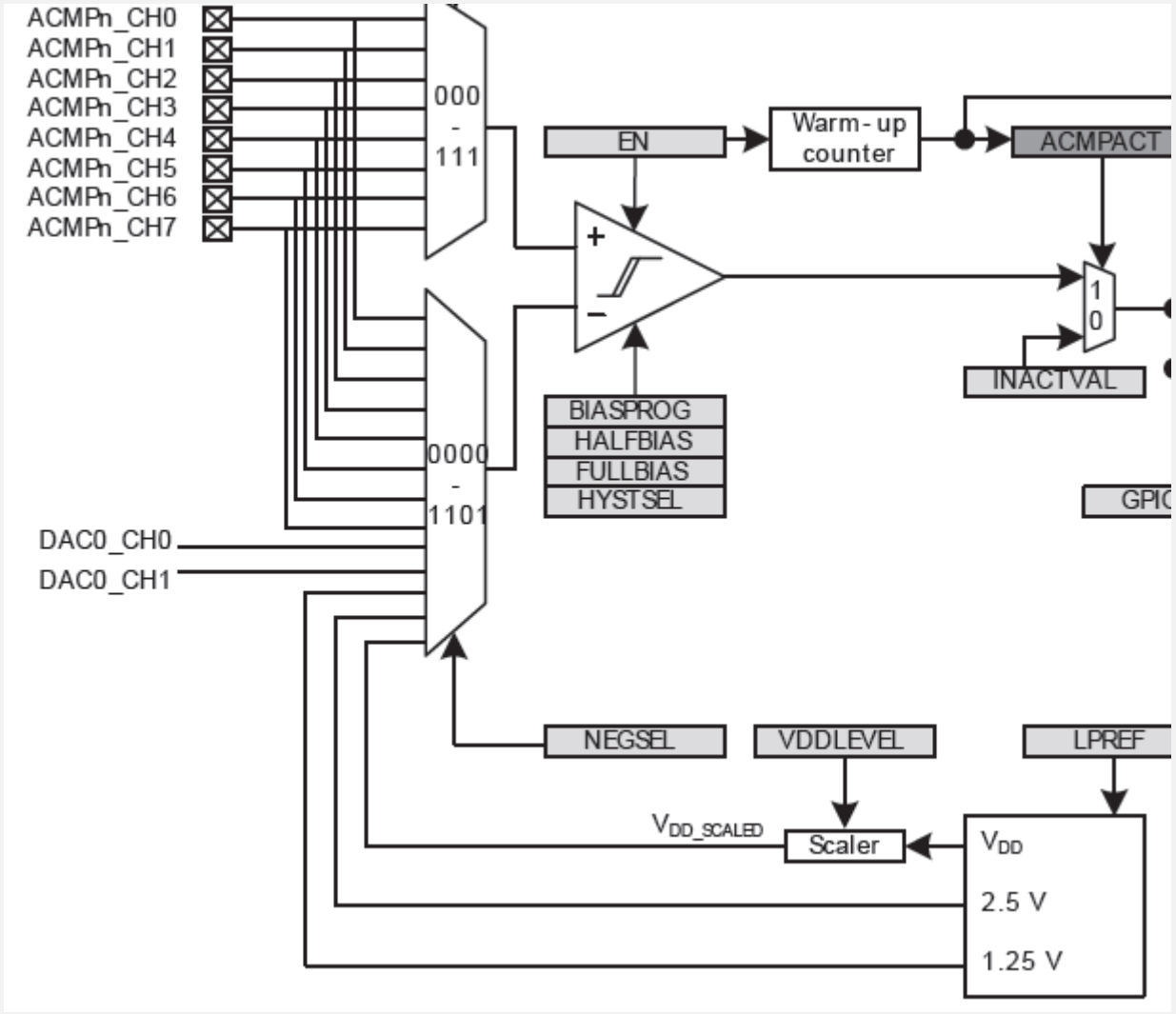
ACMP Overview



ACMP Input Selection

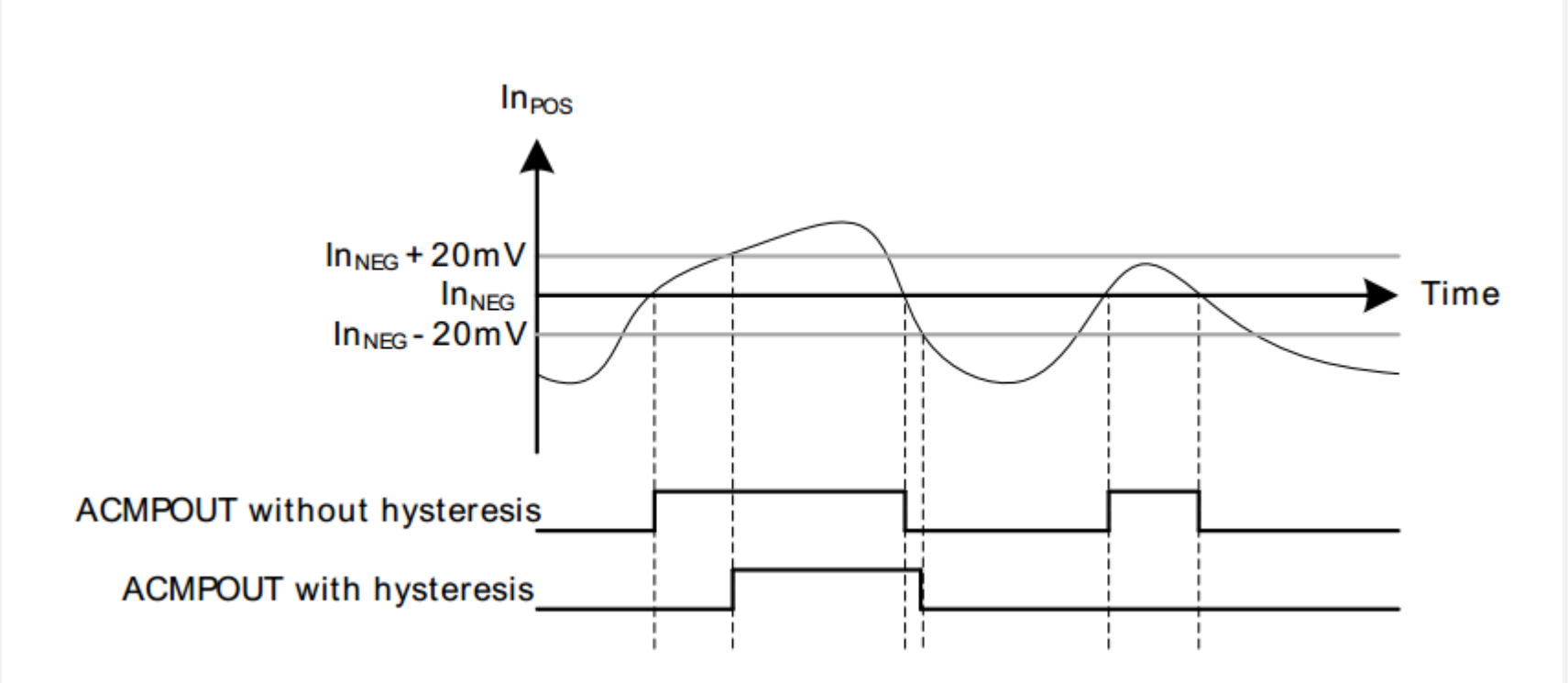
- Positive Channel
  - Input pins 0-7

Negative Channel
- Input pins 0-7
- 1.25V BG
- 2.5V BG
- Scaled VDD
- DAC channels

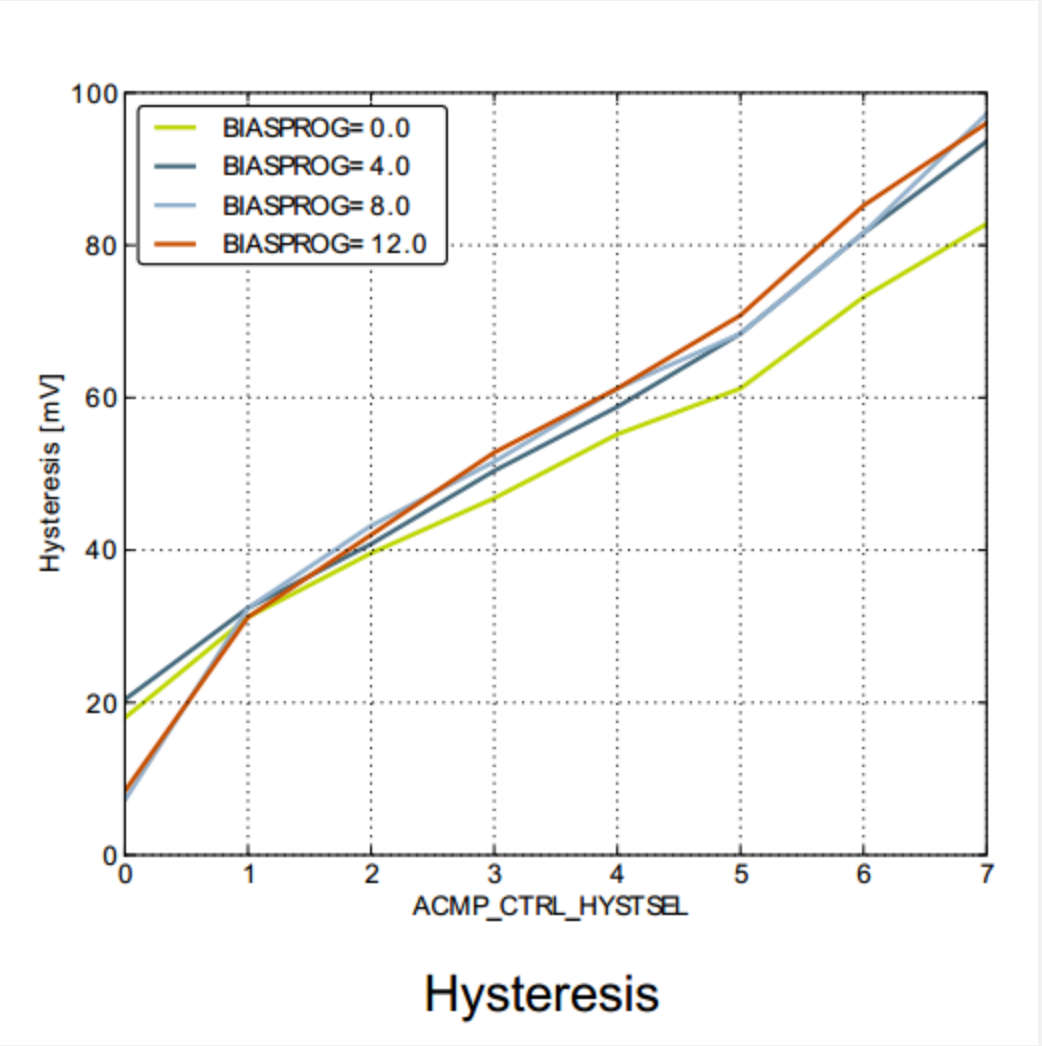


$$V_{DD\_SCALED} = V_{DD} \times VDDLEVEL/63$$

Hysteresis

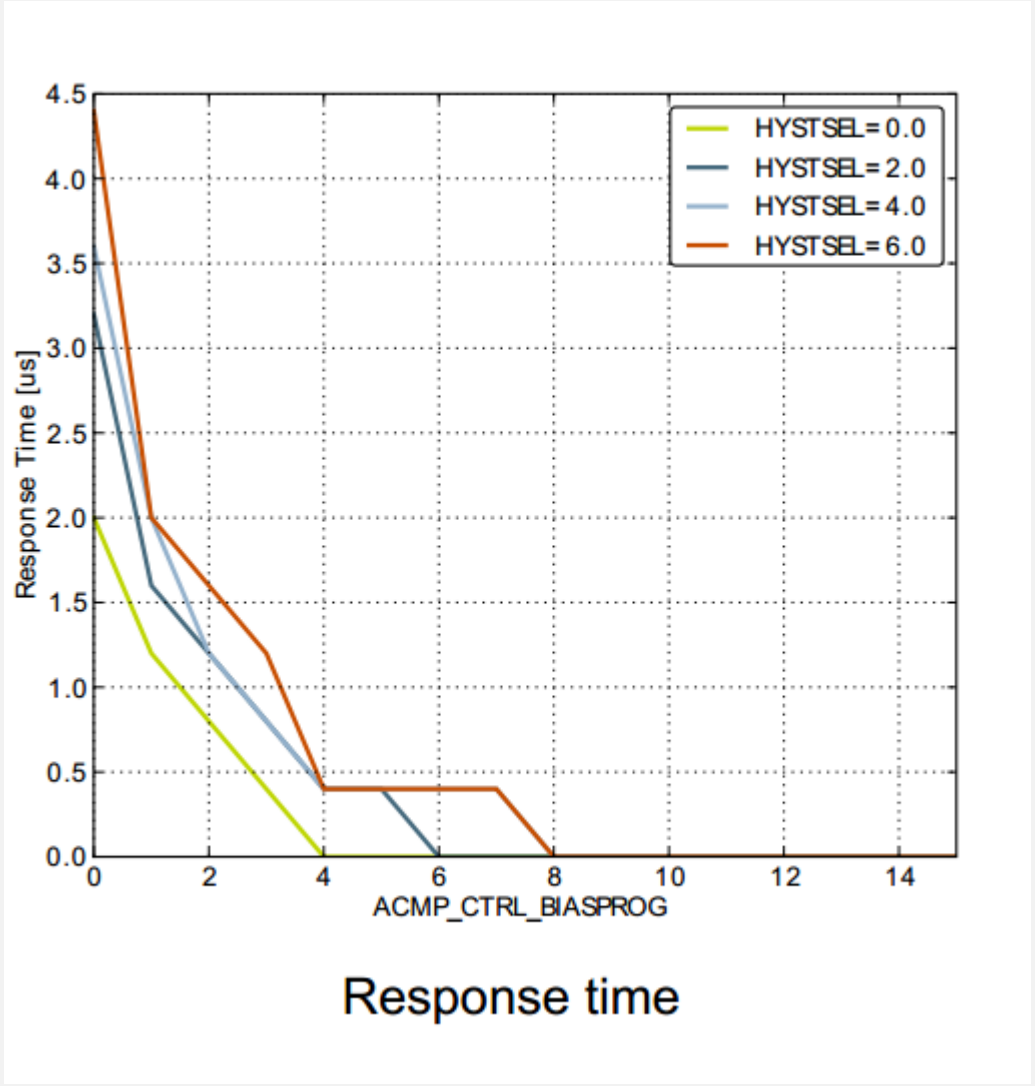

In_{POS}
In_{NEG} + 20mV
In_{NEG}
In_{NEG} - 20mV
Time
ACMPOUT without hysteresis
ACMPOUT with hysteresis

Hysteresis



Hysteresis

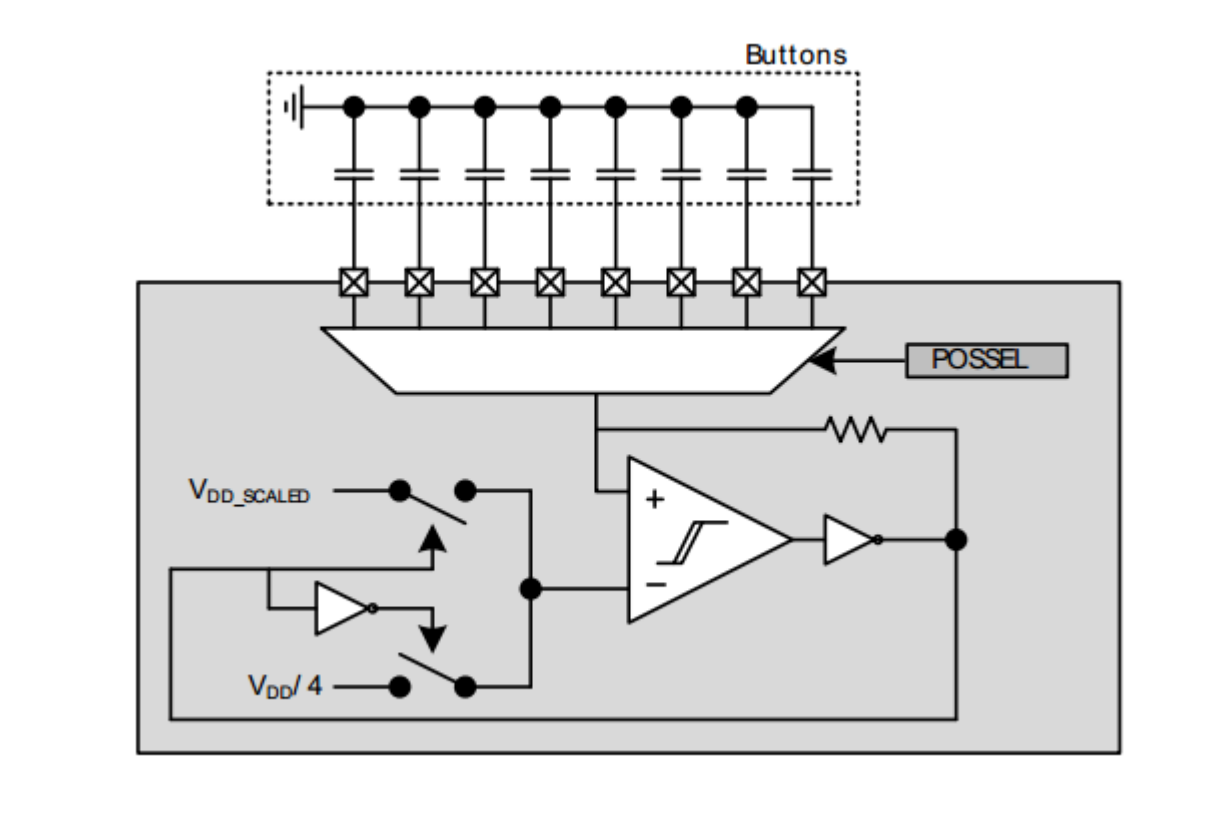
Response Time



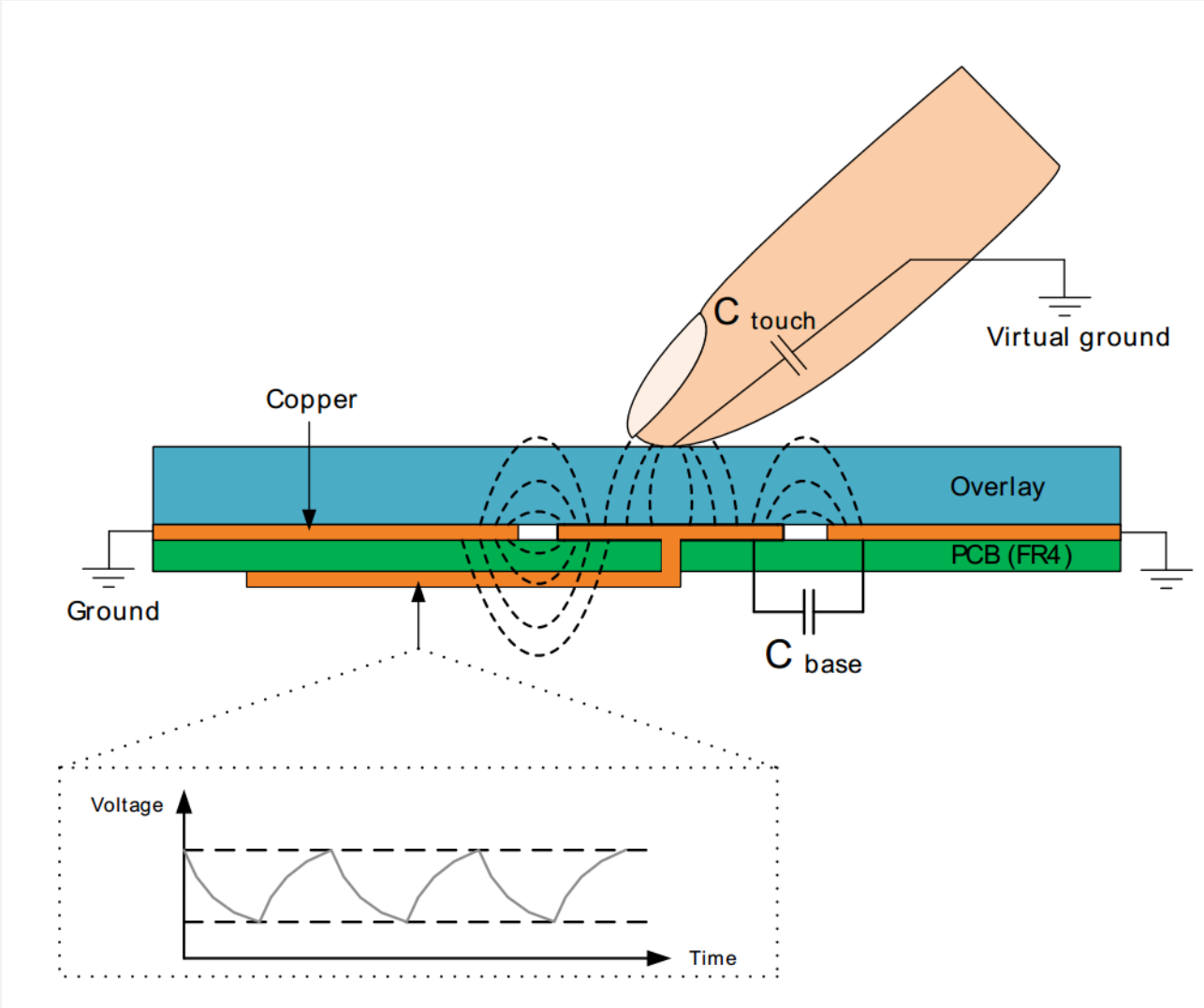
Response time

FULLBIAS = 0, HALFBIAS = 1

Capacitive Sense Mode

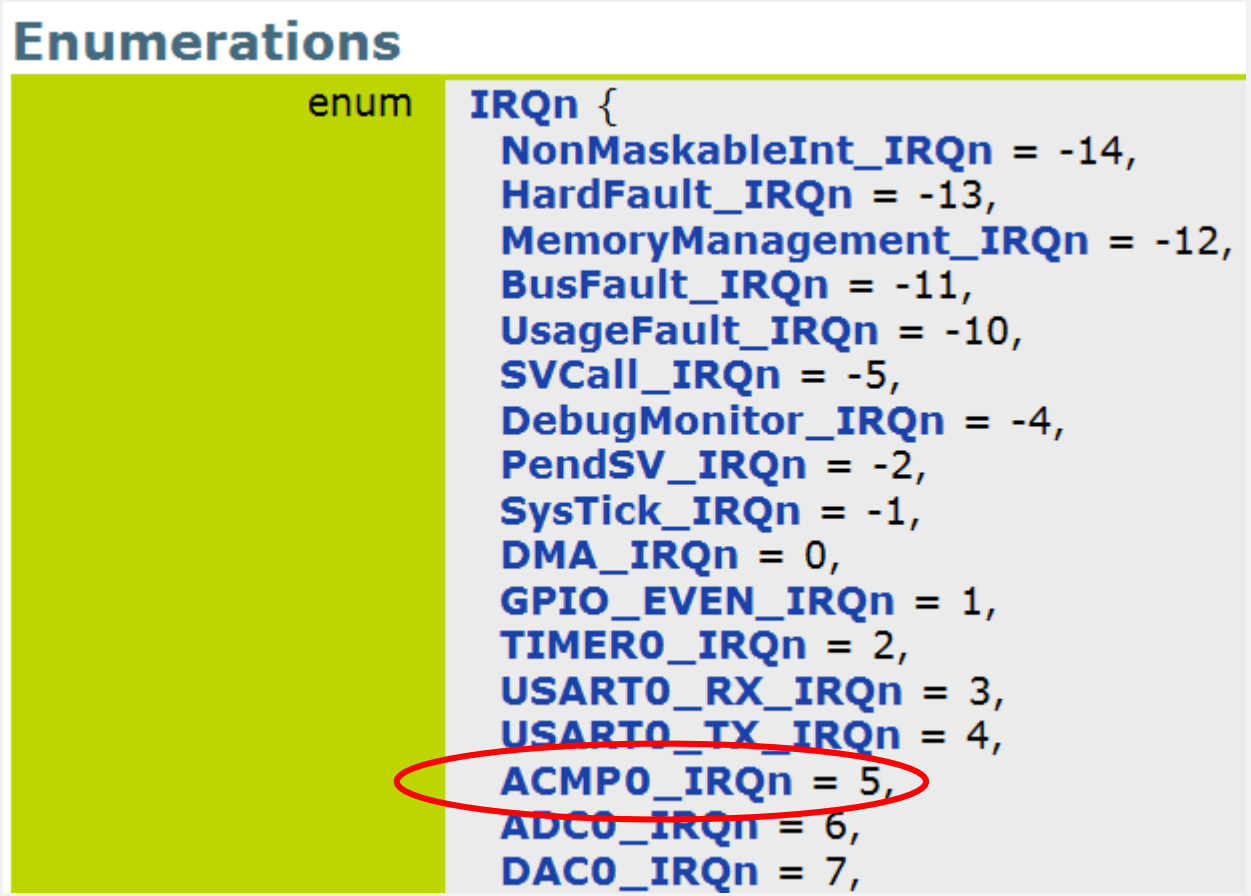


Capacitive Sense Mode

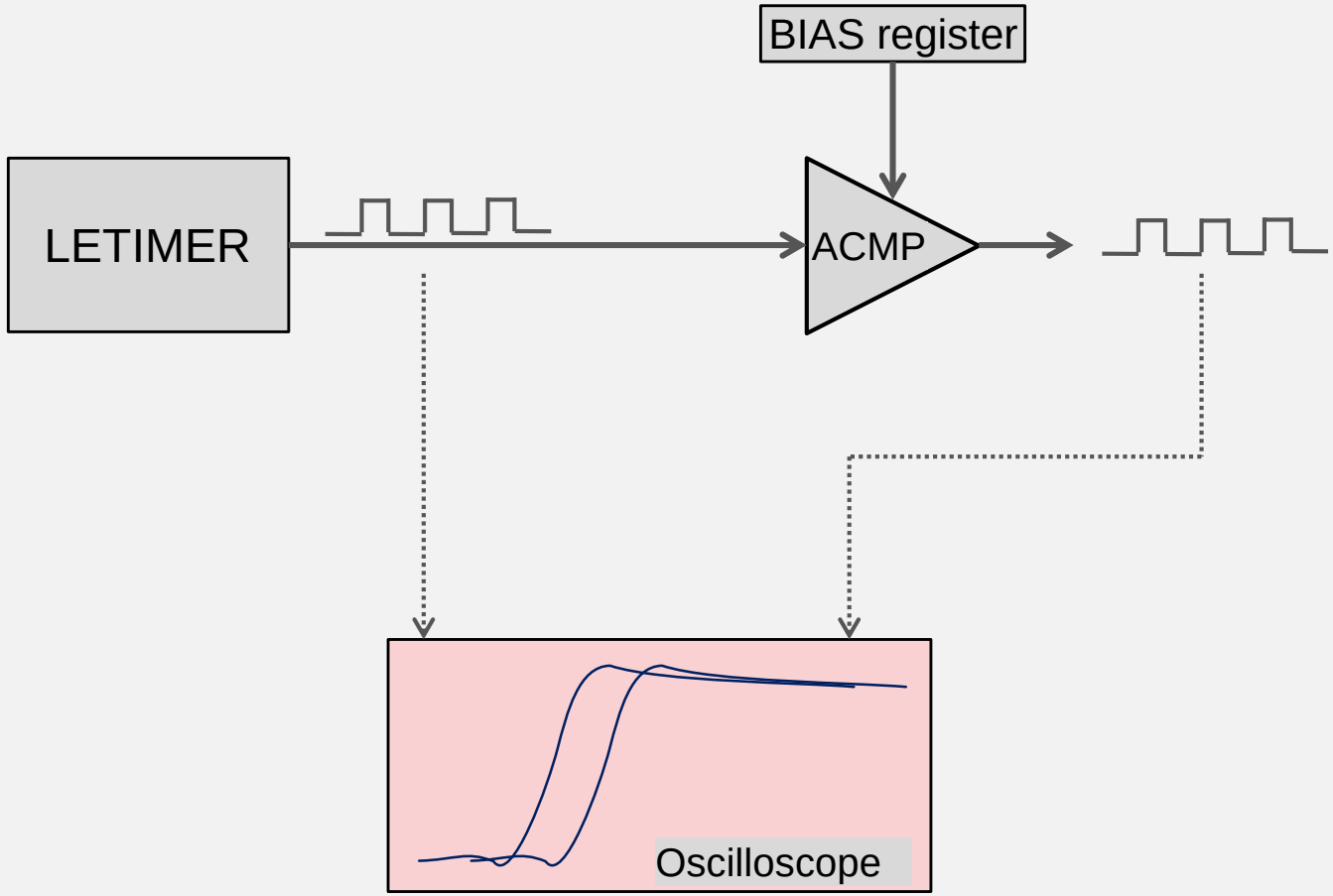


ACMP IRQ

- Both ACMPs share one IRQ, called ACMP0_IRQ

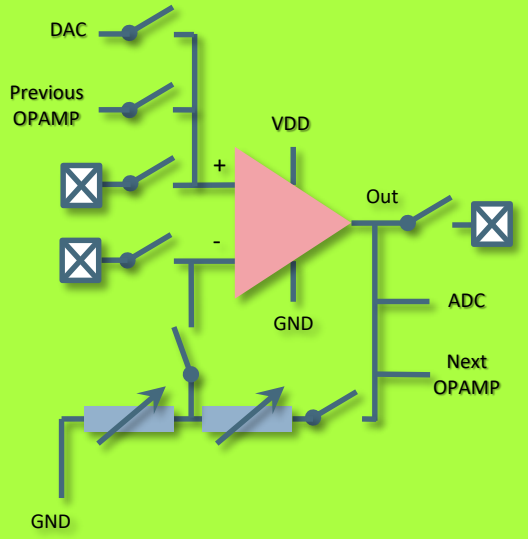


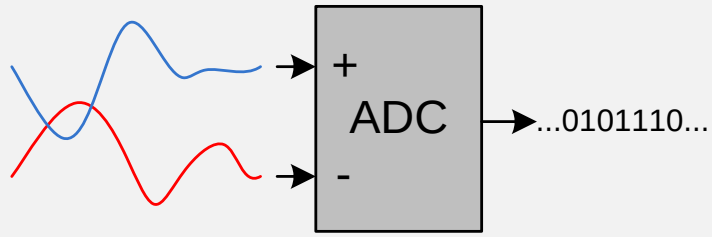
ACMP Bias

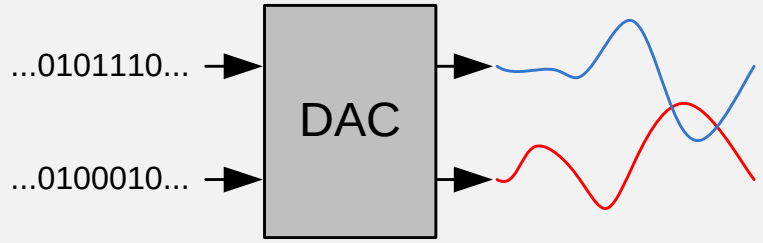


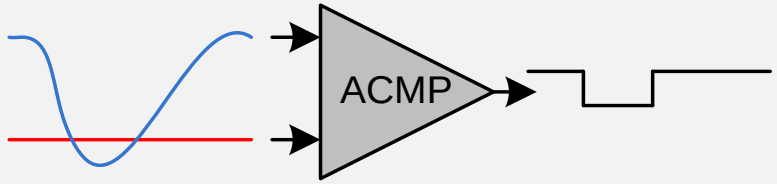
Operational Amplifiers

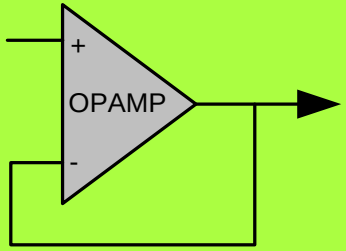
- **3 rail-to-rail OPAMPs integrated**
- **Configurable connections to ADC, DAC and pins**
- **Various configuration modes**
  - **Programmable gain**
  - **Inverting / non-inverting**
  - **Cascading**



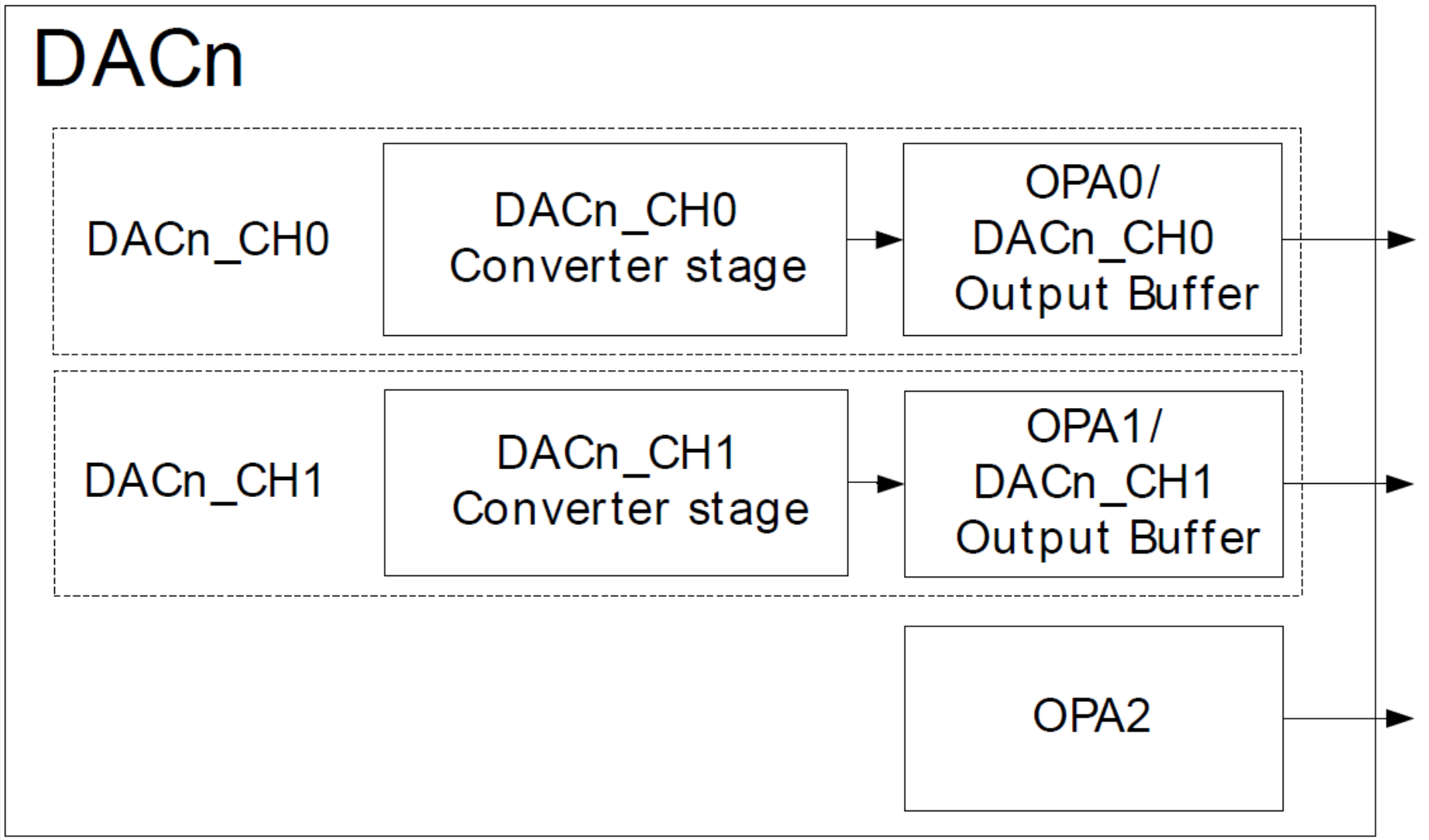




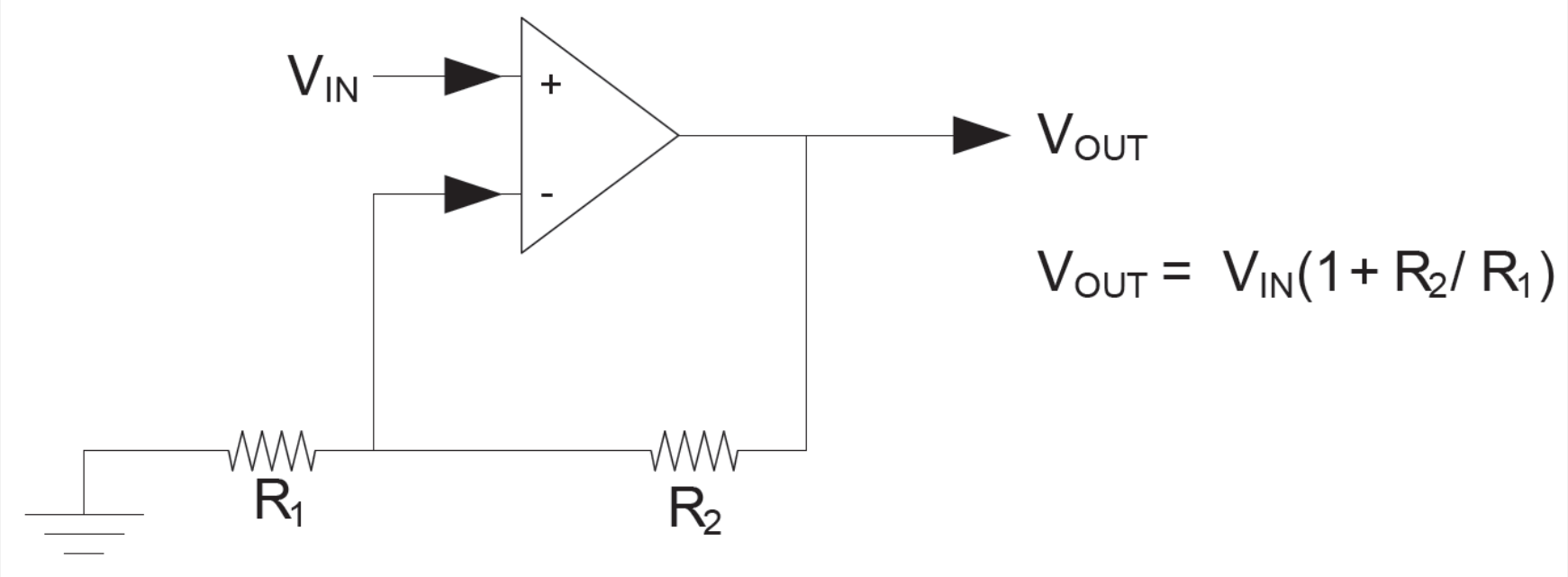




Operational Amplifiers in DAC



Non-Inverting Amplifier



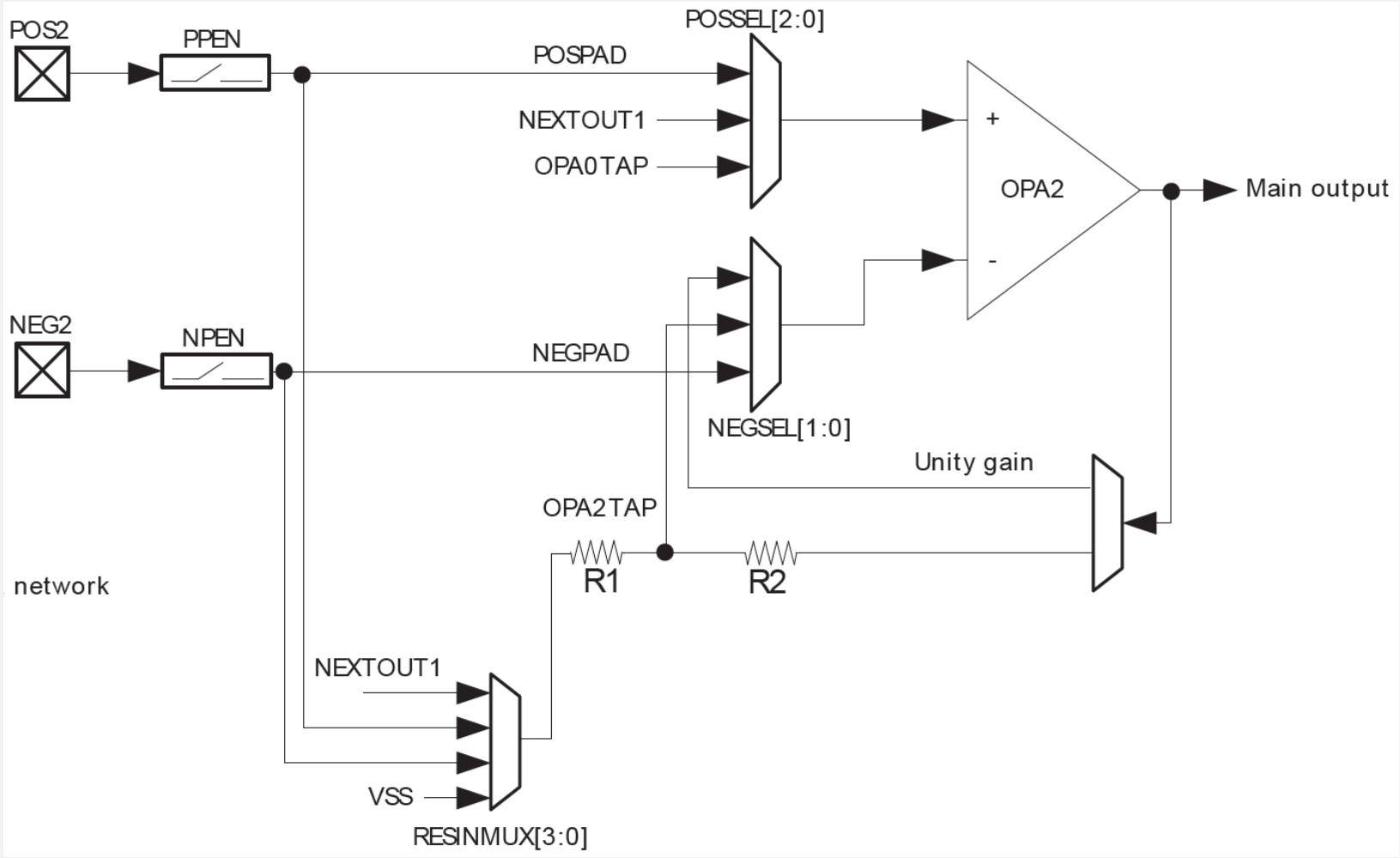
Non-Inverting Amplifier

- **Preset common configurations in emlib**

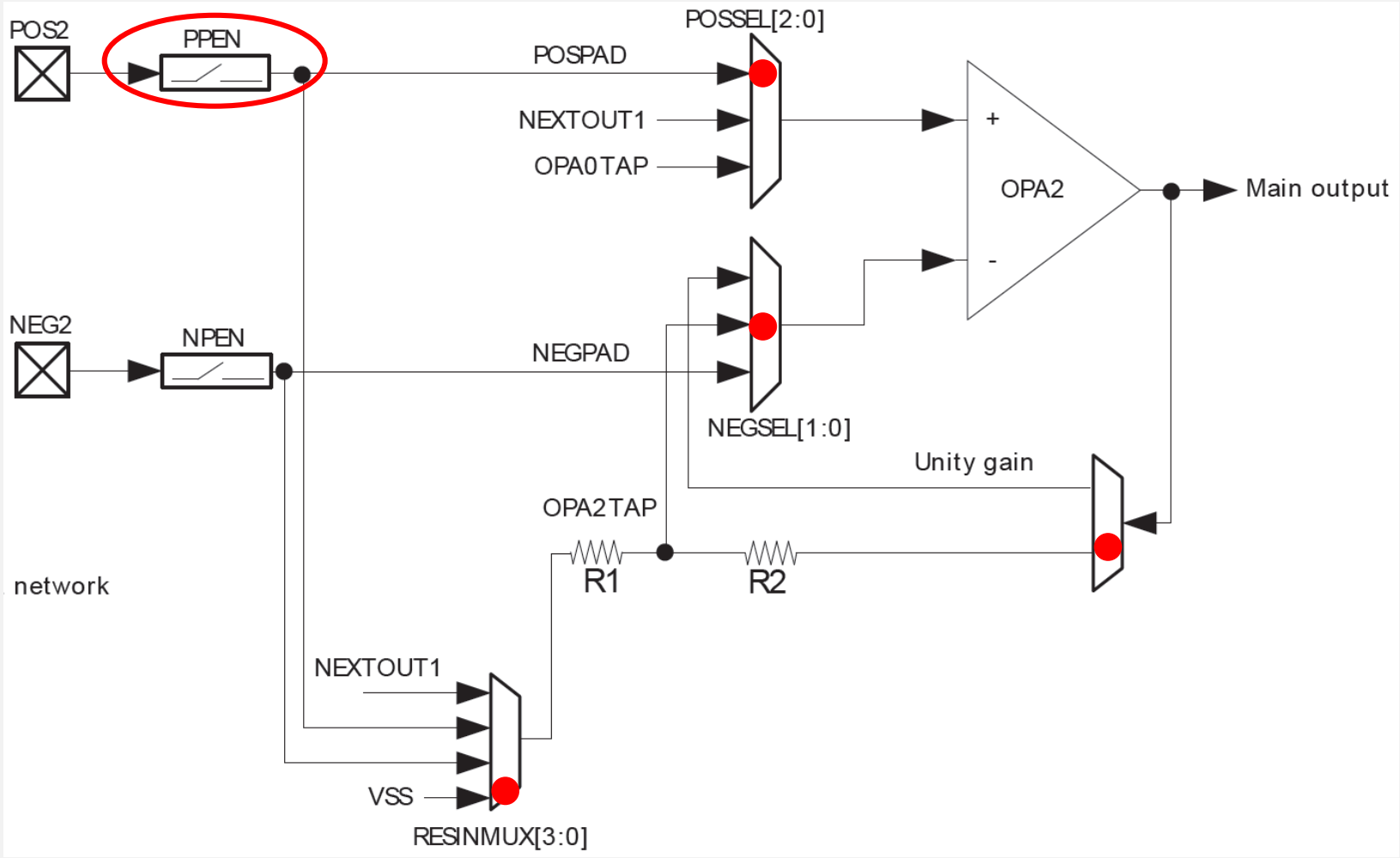
```
/*Turn on the DAC clock*/
CMU_ClockEnable(cmuClock_DAC0, true);

/*Define the configuration for OPA2*/
OPAMP_Init_TypeDef configuration = OPA_INIT_NON_INVERTING_OPA2;

/*Enable OPA2*/
OPAMP_Enable(DAC0, OPA2, &configuration);
```

- **Only some options available in emlib**
- **To fully utilize the opamps, one must go touch the registers directly**

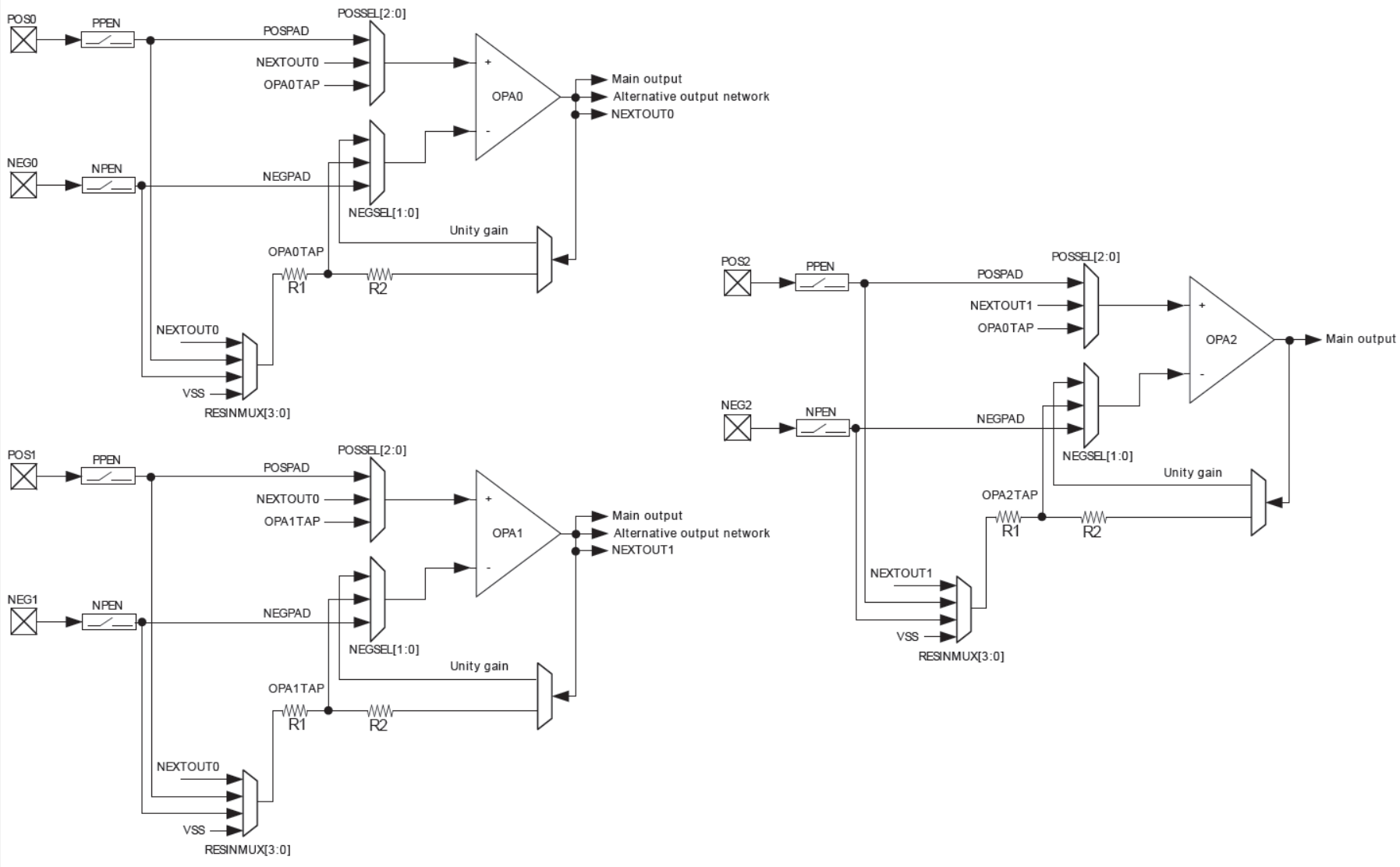
OPAMP Connections



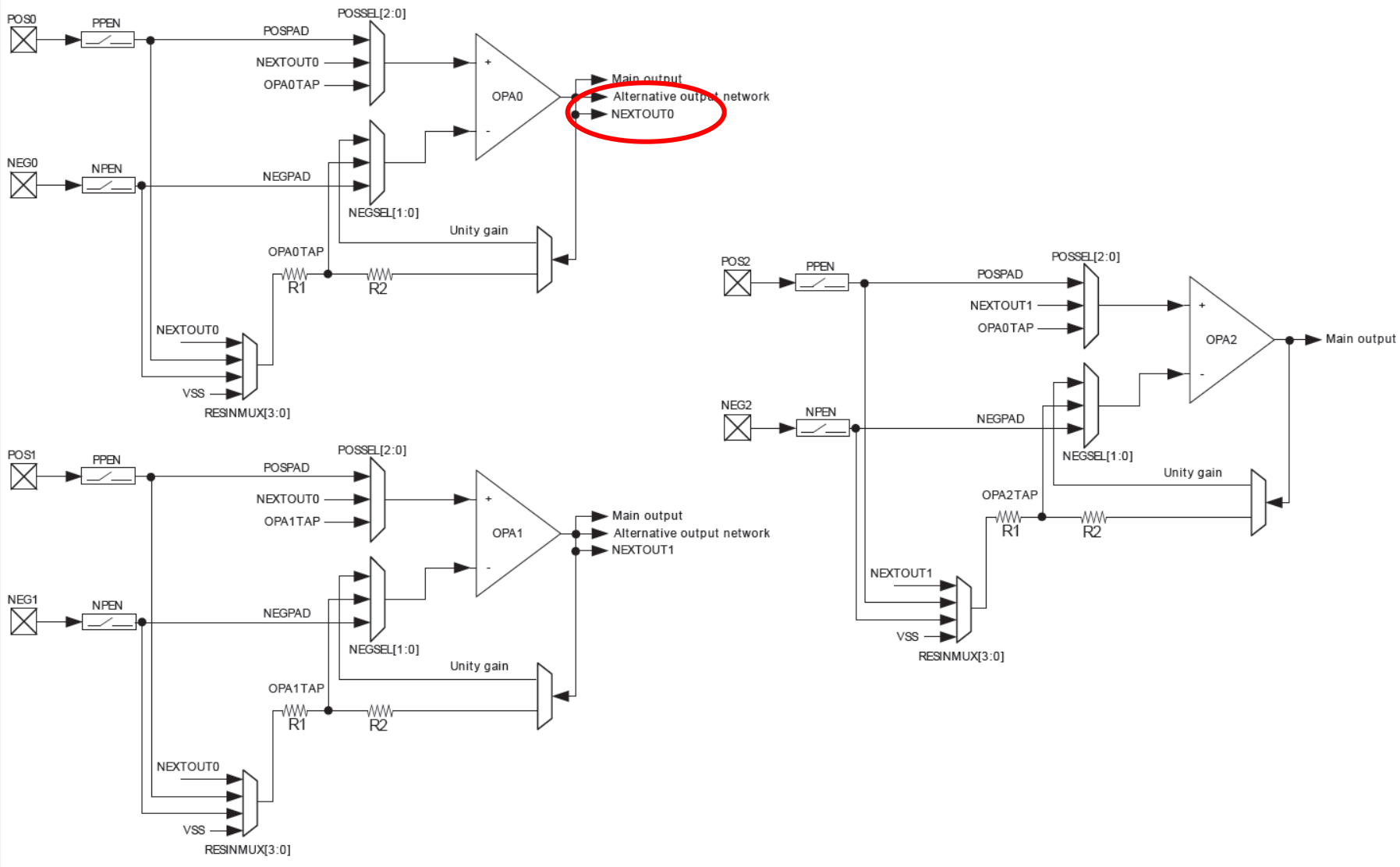
OPAMP Connections



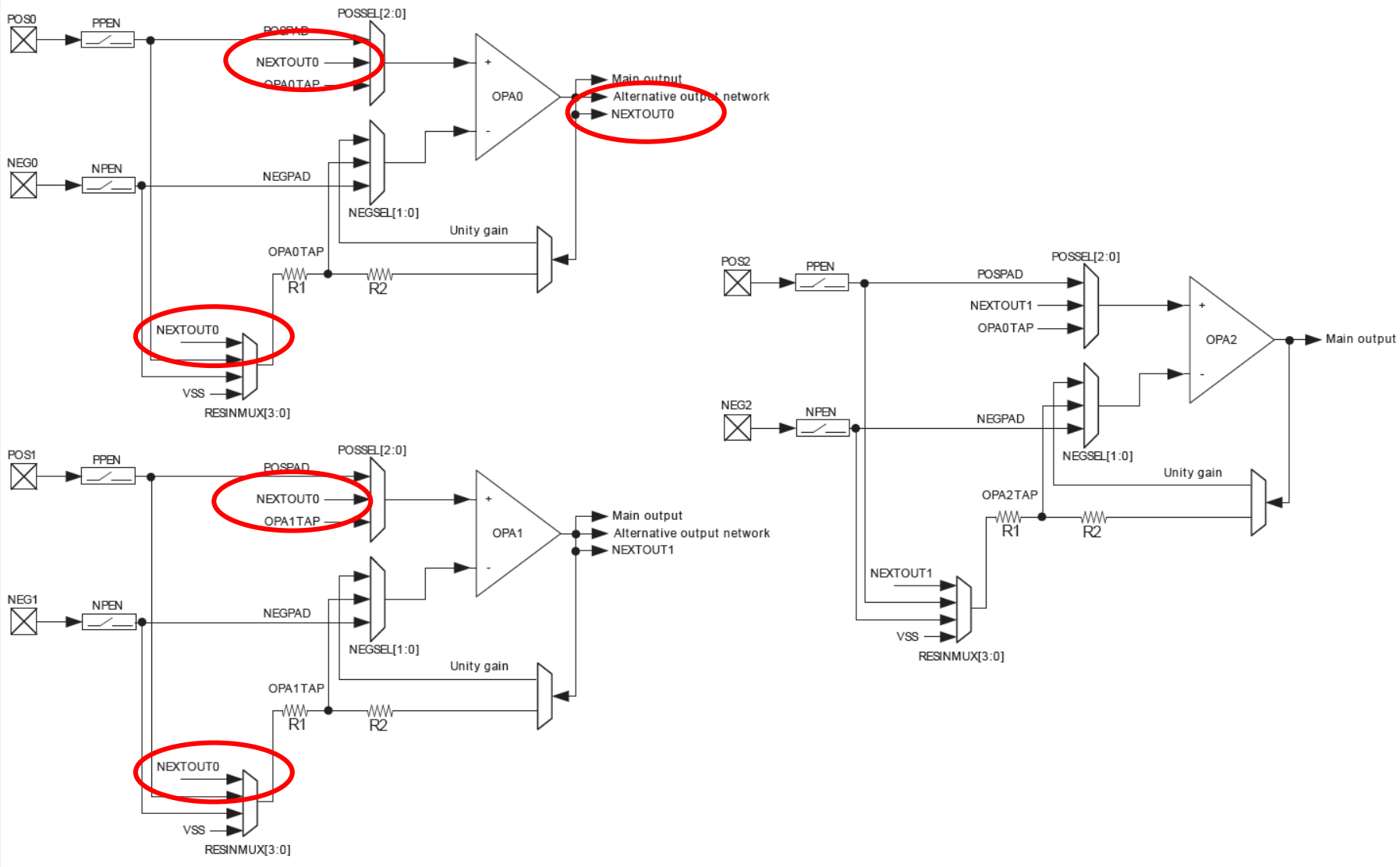
OPAMP Connections



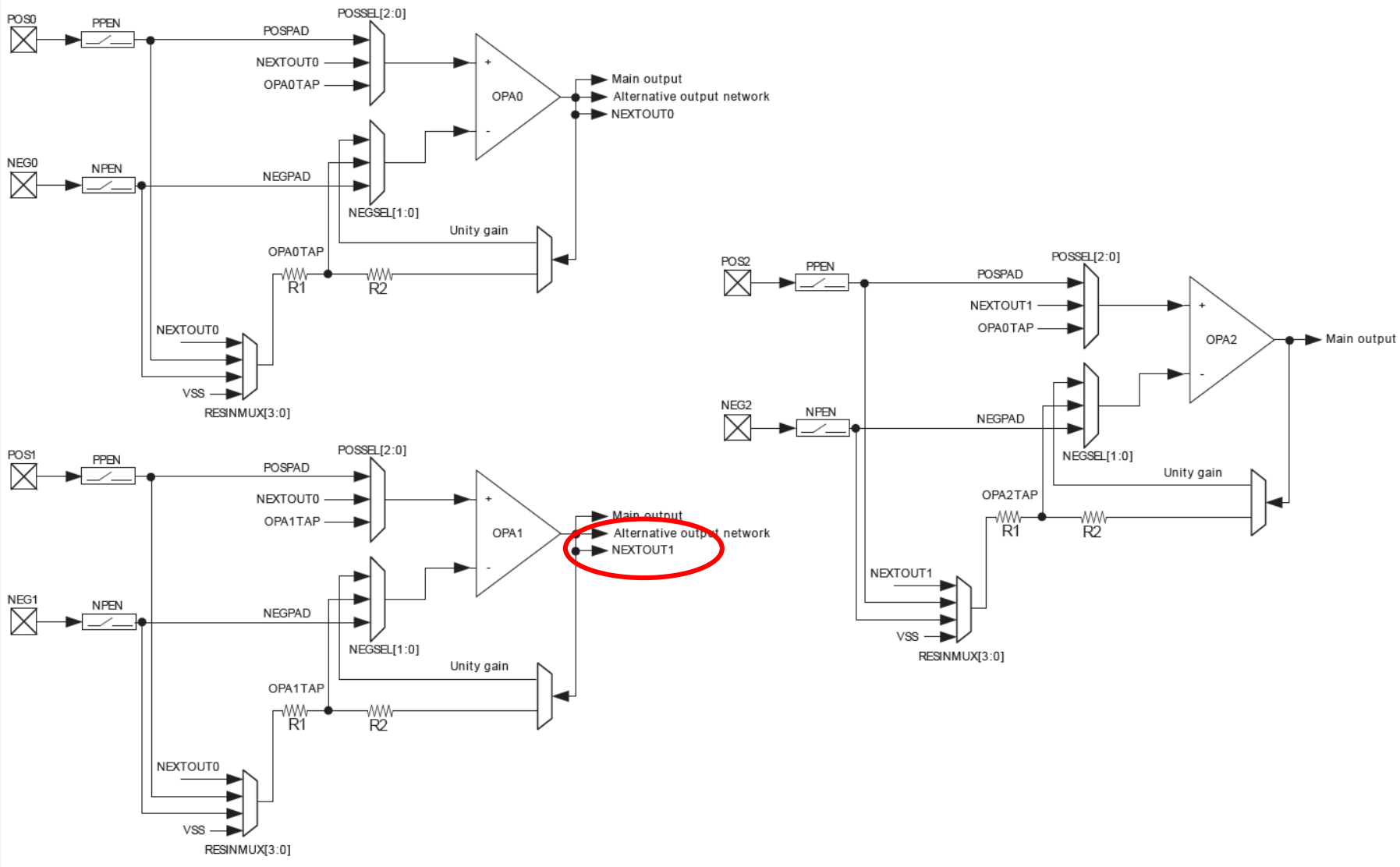
OPAMP Connections



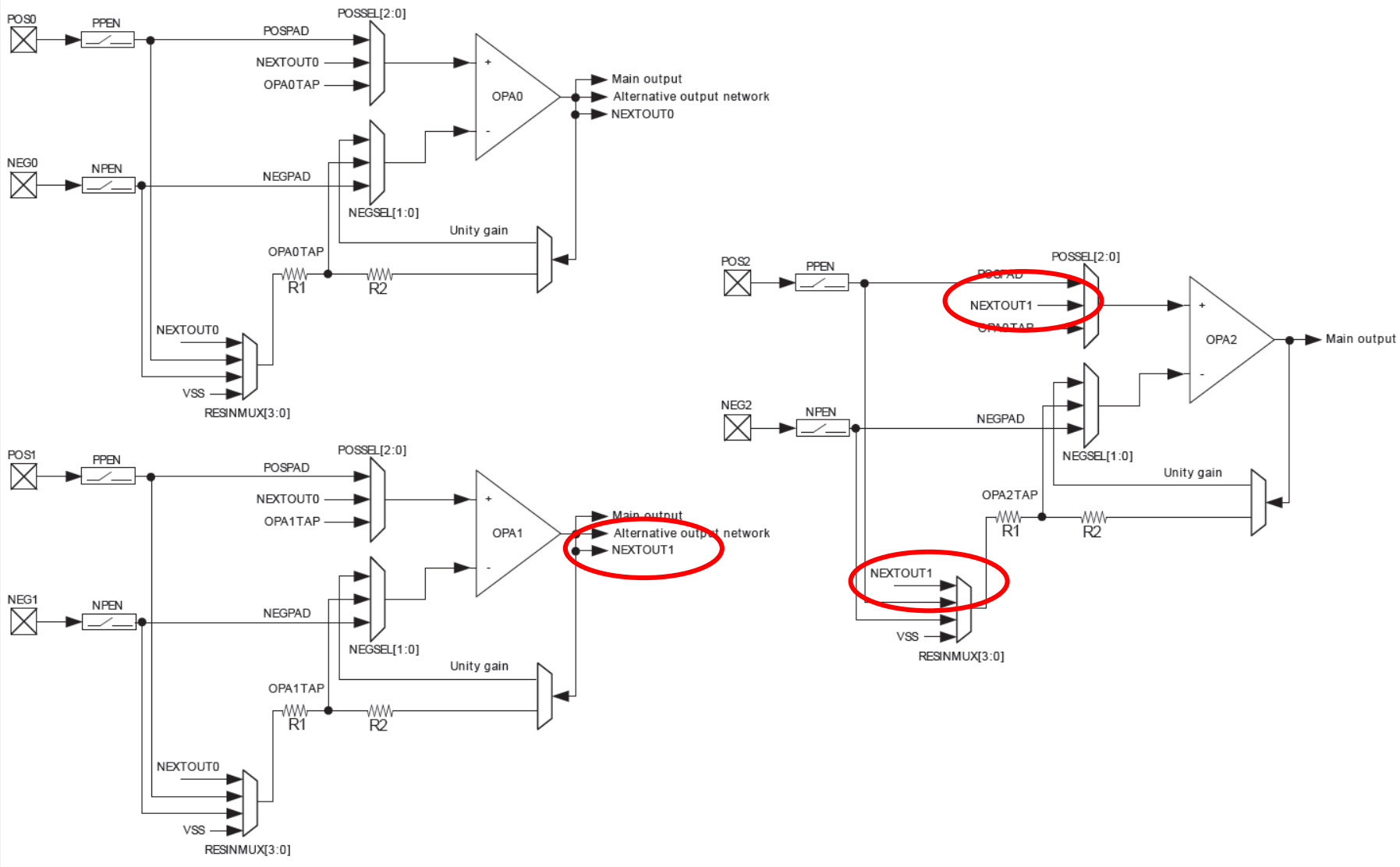
OPAMP Connections



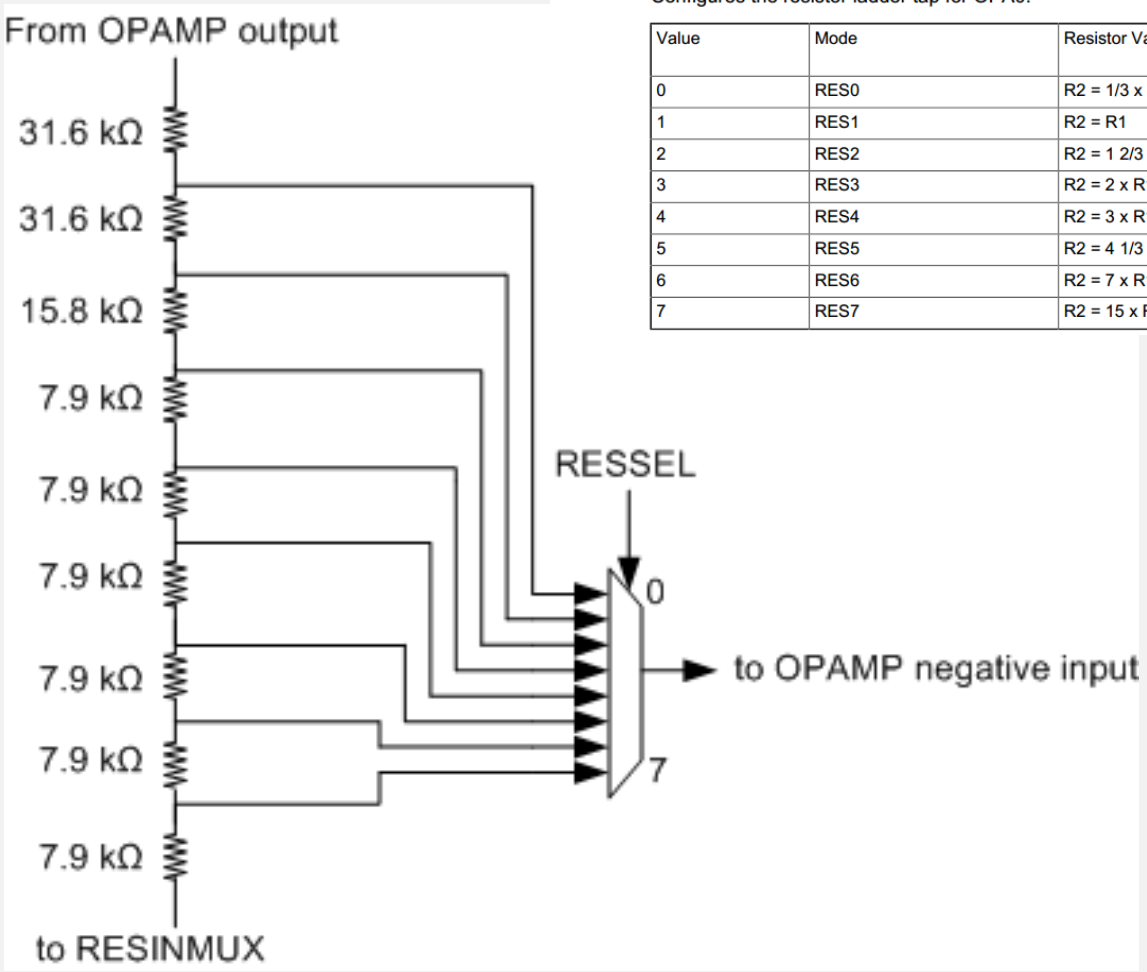
OPAMP Connections



OPAMP Connections



Resistor Ladder

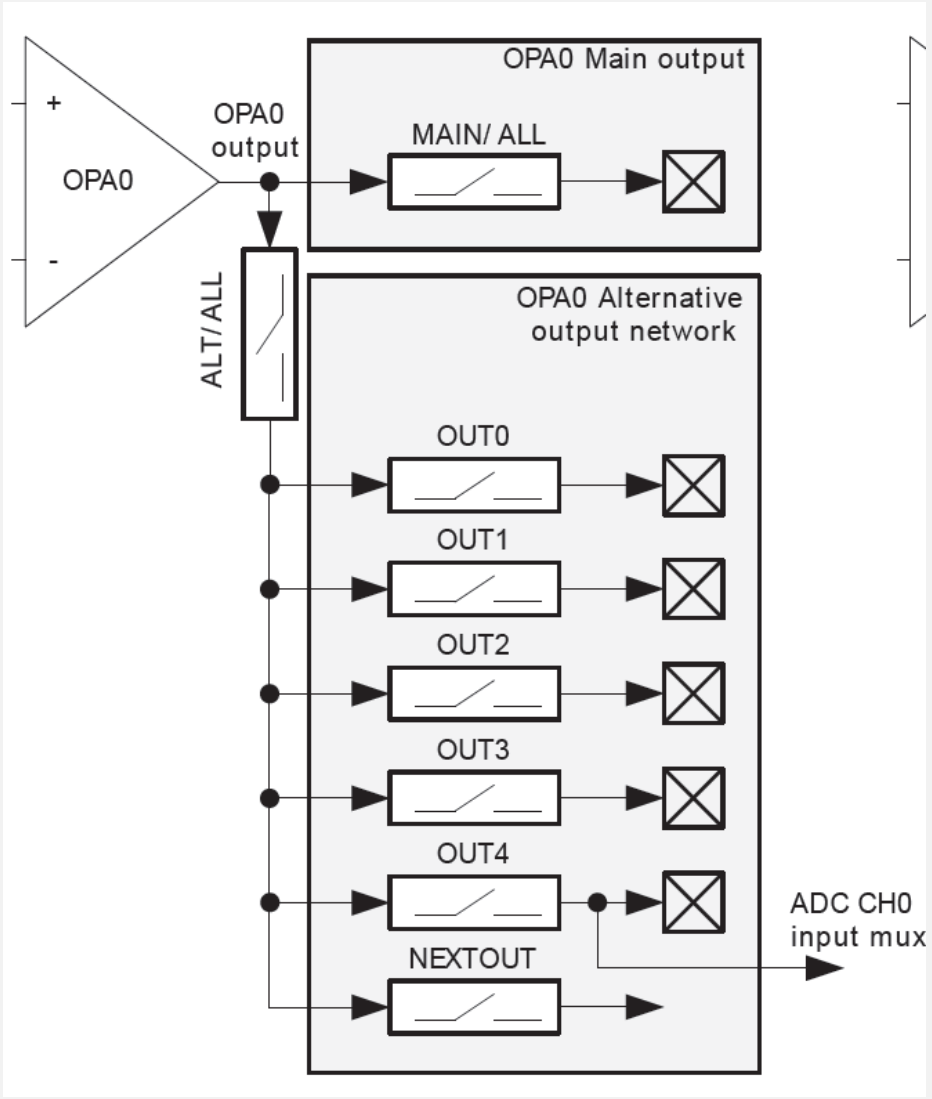


30:28	RESSEL	0x0	RW	**OPA0 Resistor Ladder Select**

Configures the resistor ladder tap for OPA0.

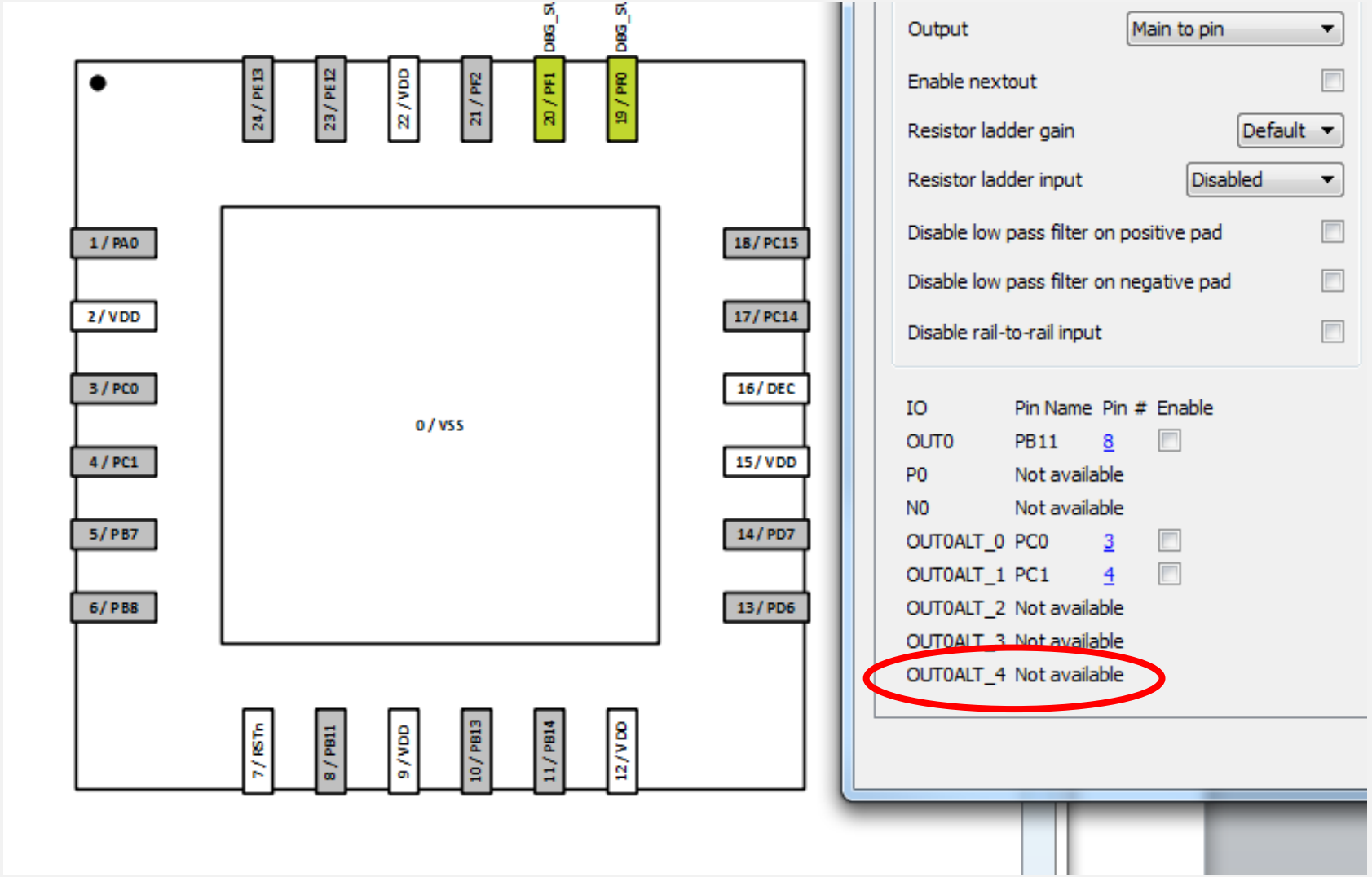
Value	Mode	Resistor Value	Inverting Mode Gain (-R2/R1)	Non-inverting Mode Gain (1+(R2/R1)
0	RES0	R2 = 1/3 x R1	-1/3	1 1/3
1	RES1	R2 = R1	-1	2
2	RES2	R2 = 1 2/3 x R1	-1 2/3	2 2/3
3	RES3	R2 = 2 x R1	-2 1/5	3 1/5
4	RES4	R2 = 3 x R1	-3	4
5	RES5	R2 = 4 1/3 x R1	-4 1/3	5 1/3
6	RES6	R2 = 7 x R1	-7	8
7	RES7	R2 = 15 x R1	-15	16

Output to ADC



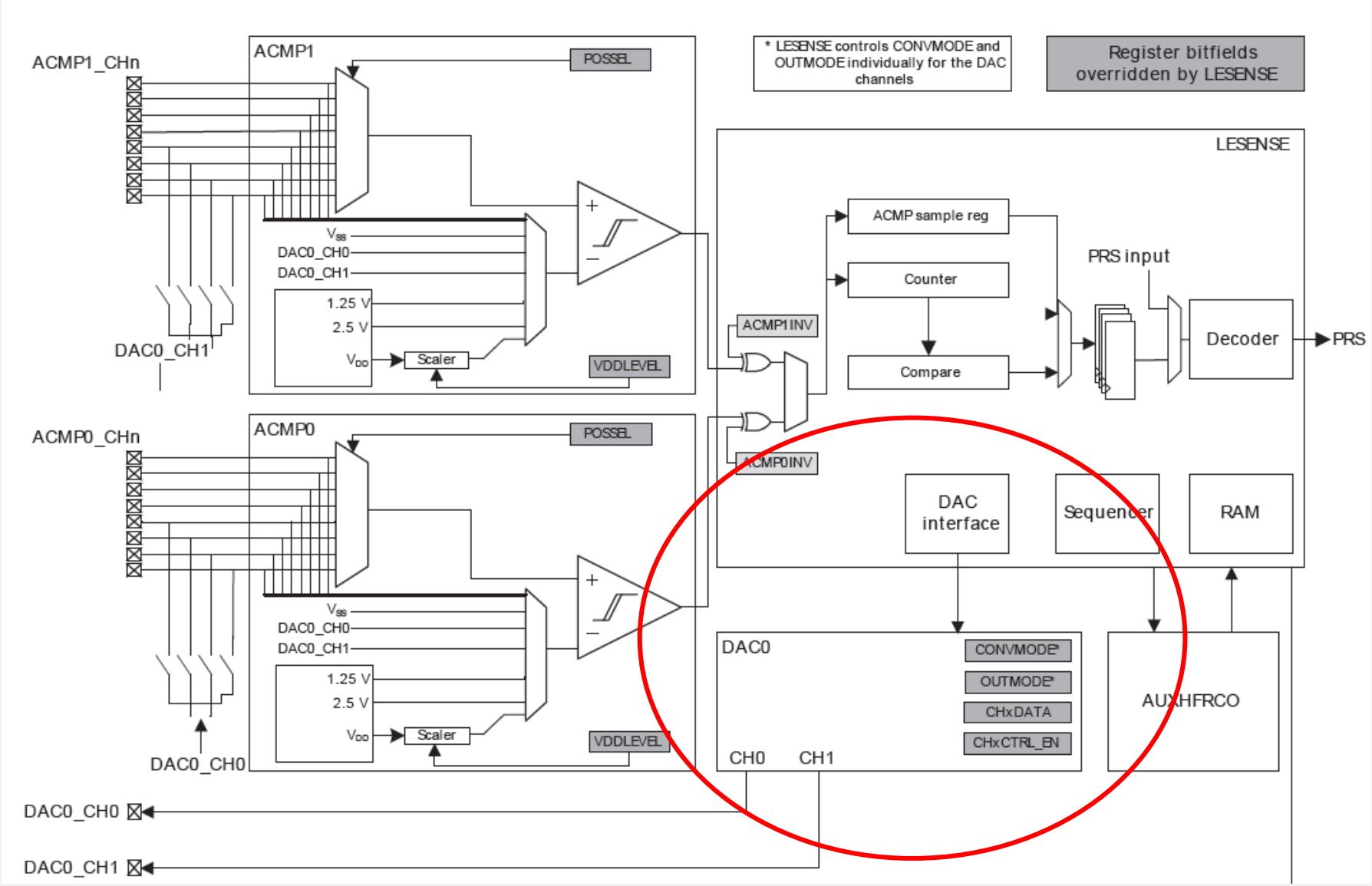
OPAMP Connections

- Output to ADC, what if there is no pin on that package?
  - Still OK to use that pad for internal routing



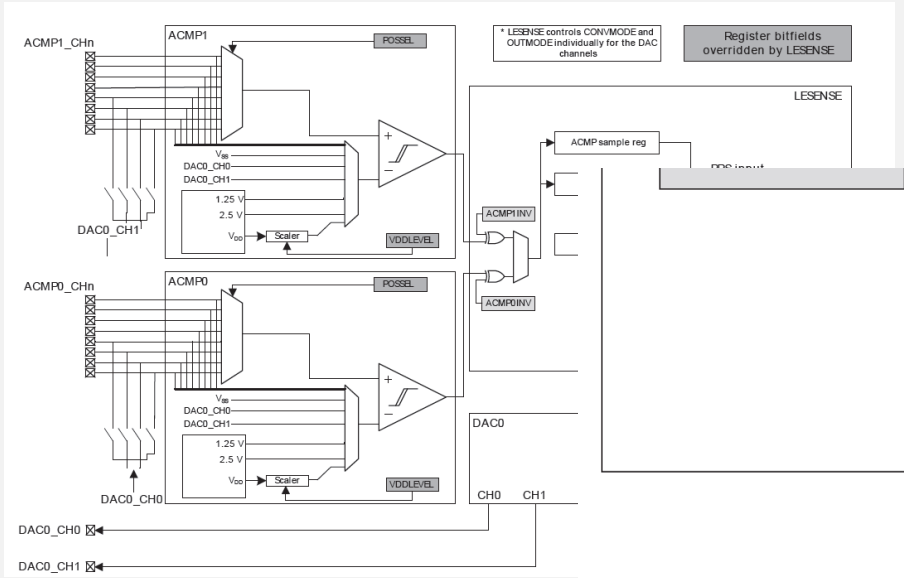
OPAMP + LESENSE

- OPAMP + LESENSE



OPAMP Details

- OPAMP + LESENSE



LESENSE can dutycycle OPAMP 0 and 1 the way it dutycycles the DAC