CALOROC1B datasheet

Documents history

VERSION	DATE	MODIFICATION
1.0	5 Sept, 2025	First release
1.1	3 Oct, 2025	Update with LLR comments + BGA/C4 mapping
1.2	12 Mar, 2026	Update operating modes, fast commands, I2C, CRC, 1 ADC mode
1.3	14 Apr, 2026	SC parameters update

Table of contents

1 CALOROC1B: architectural overview

The CALOROC1B ASIC is made of 36 independent and autonomous channels to provide a measurement of the arrival time (TOA) and of the charge for the 36 input signals.
The overall block diagram is shown below (Figure 1) and described after.

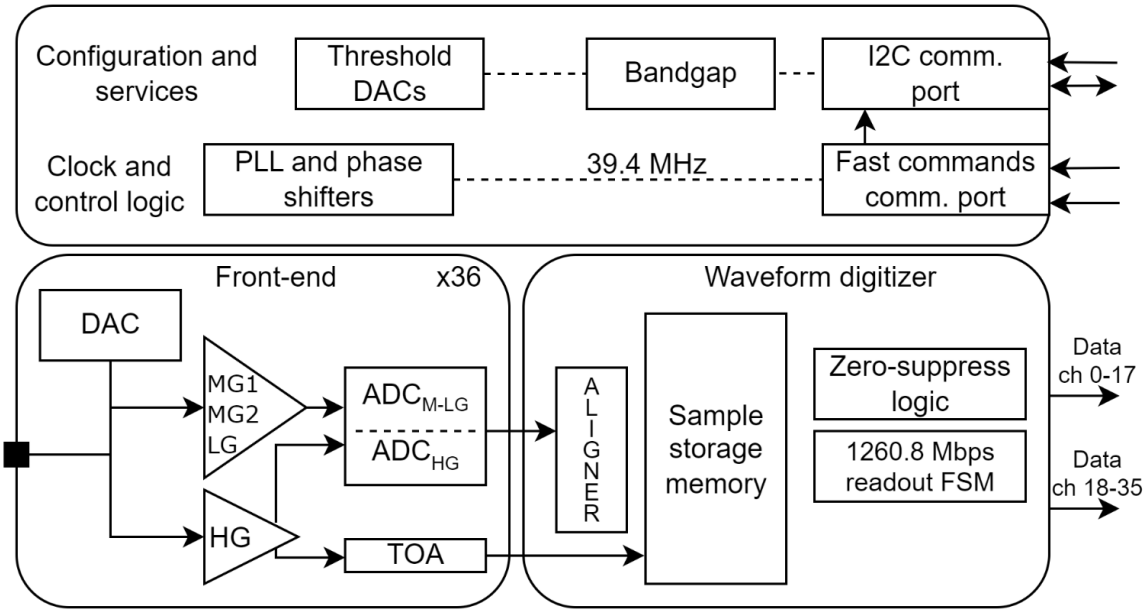


Figure 1 The overall block diagram

The analog front-end part (section 2) amplifies and filters the input signals to allow the charge measurement thanks to a 10-bit SAR ADC (designed by AGH in Krakow) and the arrival time measurement thanks to a 10-bit TDC (designed by the CEA IRFU group in Saclay).

CALOROC1B was developed for the Electron-Proton/Ion Collider (ePIC) at EIC, using a large number of silicon photomultipliers (SiPM). To achieve the large charge dynamic range of the ePIC detector each channel counts on a dynamic gain switching mechanism that allows it to reach a dynamic range up to 140k measured as maximum measurable signal divided by the RMS noise.

Two 1.26 Gbps links (18 channels per link) are dedicated to transmitting digitized event information (charge and time measurements). An internal 32-depth circular DRAM buffer stores this data (encoded with Hamming code for charge and time) until the serial link becomes available. An almost full indicator (Flag_AF) is provided via an output pin to inform the user of the buffer's status.

The Zero Suppression logic outputs data only from hit channels, transmitting it over dedicated links. It is enabled by status bits configured via slow control.

The I2C protocol is used to configure and monitor the chip parameters. This section is triplicated to provide robustness against Single Event Effects (SEE). The chip is controlled by the Fast Command block (see section 3.2), which receives a 315 MHz clock and a 315 MHz command link. This block enables the transmission of commands for configuring specific operating modes, such as link synchronization, reset, calibration, etc. It also generates the 39.4 MHz clock from the 315 MHz command link.

The 39.4 MHz clock is used by the digital part of the ASIC (digital processing, I2C interface) and by the PLL, which generates the additional clocks required by the ASIC. The PLL provides:

- a 630 MHz clock for the 1.26 Gbps links,
- a 39.4 MHz clock with adjustable phase for the ADCs,
- a 157.6 MHz clock with adjustable phase for the TDCs.

Phase adjustment is necessary to align the conversion blocks (ADC and TDC) with the phase of the physical signals, which depend on the rapidity angle.

To simplify the redaction of this document, the 39.4MHz will be approximated to 40MHz. The same for the other clocks which will be approximated to multiples of 40MHz. All blocks (ADC, TDC, PLL) have proven to work with 40MHz and its multiples before.

2 Analog and mixed-signal processing

2.1 Analog front-end

The analog front-end can be divided in six main sub-parts (Figure 2):

- The preamplifiers which provide the first amplification the SiPM signal with the best noise performance.
  The high gain preamplifier has a variable feedback capacitor to tune the gain and a variable feedback resistor to shape the output signal which is sent to the shaper block.
  The Low power preamplifier have a lower power consumption in exchange for a higher noise contribution and lower shaping. This low power preamplifier has a variable input capacitor to tune the gain.
- A high pass active filter is used to reduce the timewalk of the signal in exchange for a lower SNR, a variable feedback resistor can be tuned to adjust this tradeoff. Then a discriminator produces the TOA (Time of Arrival) pulse which is sent to a dedicated TDC.
- Two other discrimitors are used to activate the gain switching as soon as one of the preamplifiers saturates.
- Each preamplifier has a shaper with a variable feedback resistor and feedback capacitance to tune the shape of the signal.
- An analog multiplexer is used to adjust what shaper output will be read by the ADC1 and ADC2. This analog multiplexer is driven by a state machine.
- Two buffers for each ADC are used to produce the differential signal and drive the ADCs.
- An input amplifier which is used channel wise to set the DC voltage at the channel input.

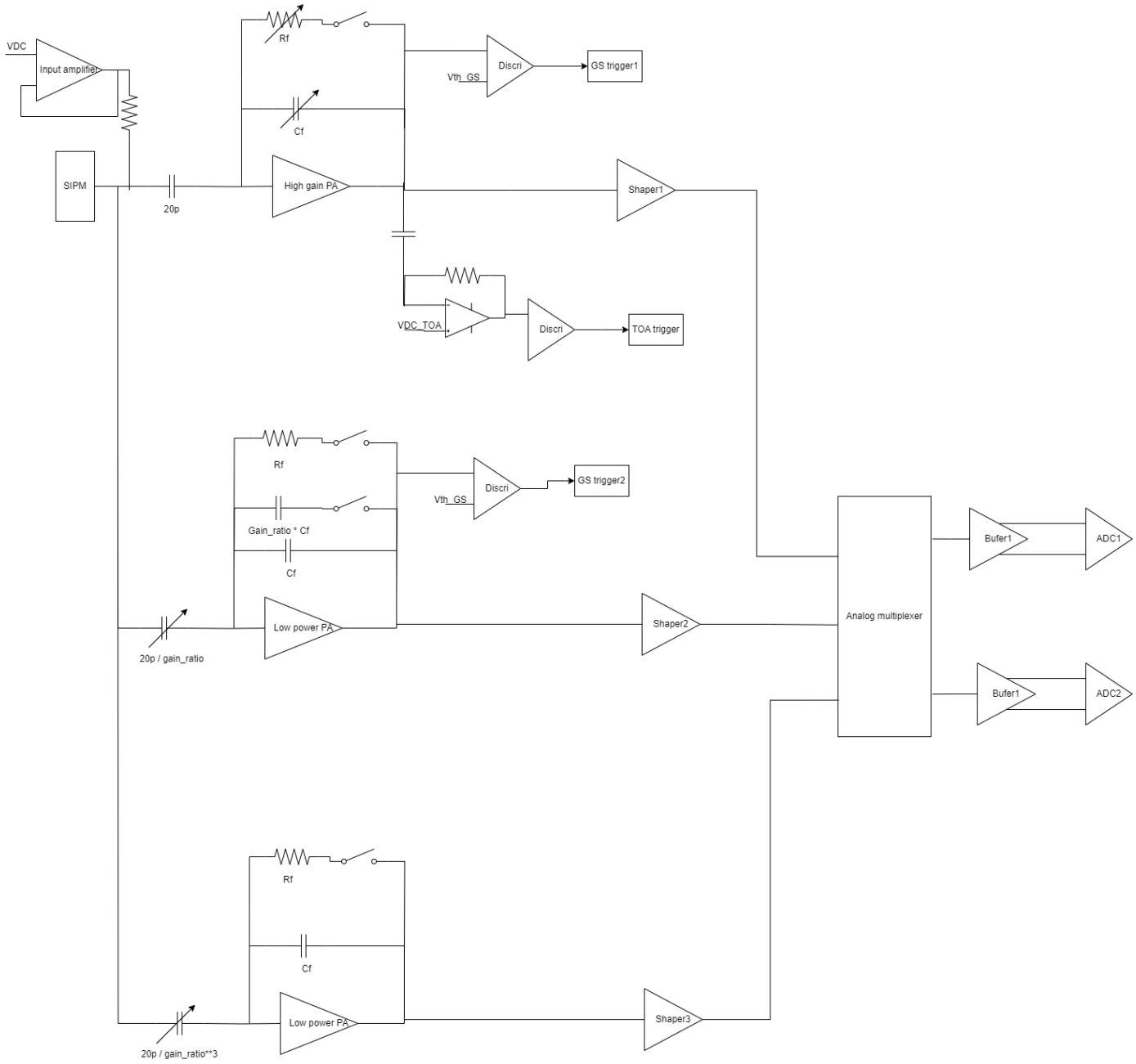


Figure 2 Analog front-end part

As seen on the full schematic, there is a high gain path, and two low gain paths, with the middle one implementing a gain switching mechanism. The output of those three gain paths is connected to an ADC through an analog multiplexor. By using these three gain paths with one gain switching at the medium gain path, we obtain four gains represented by two bits. These four gains are called:

- The high gain (HG): corresponding to the high gain path, also called Gain 0 (Gain 2b'00)
- The medium gain 1 (MG1): corresponding to the medium gain path without switching , also called Gain 1 (Gain 2b'01)
- The medium gain 2 (MG2): corresponding to the medium gain path after switching , also called Gain 2 (Gain 2b'10)

- The low gain (LG): corresponding to the low gain path, which can also be called Gain 3 (Gain 2b'11)

The analog outputs signals are shown in picture Figure 3. The input signal used to simulate a photomultiplier signal is a positive current pulse of 432fC for 15 p.e over a period of 3ns in parallel to a 10nF capacitor.

Figure 3. Analog outputs signals

2.1.1 Preamplifier

The high gain preamplifier is AC coupled to the input with an internal 20pF capacitor. It provides two outputs:

- outpa connected to the shaper path. Its DC operating point is the same than the preamplifier input (480 mV).
- outCf_pa connected to the high pass filter. Its DC operating point is around outpa + Vgs (~ 750 mV).

The purpose of the preamplifier is to amplify the input signal with the best signal-to-noise ratio and a gain adapted to the MIP signal. It provides a variable gain by a feedback capacitor on 3 bits (Cf) (common for the 36 channels). A variable feedback resistor (Rf) allows to adjust the shaping of the PA output signal.

In the table below, the PA feedback parameters are described (these parameters are global, not channel-wise, default values in **bold**):

Rf (Ω)	15K, 60K, **250K**, 1M	In parallel, these resistors provide 15 values to be adjusted with the Cf and Cf_comp values to adapt the decay time constant at the physics requirements.
Cf (fF)	**70**, 140, 280	Can be combined to provide 7 values which provide the gain of the preamplifier.

The following plot (Figure 4) shows the preamplifier response at 15pe input charge for different PA gains (Cf from 70 fF up to 210 fF, more values are possible with the 3 bits) with Rf=250k

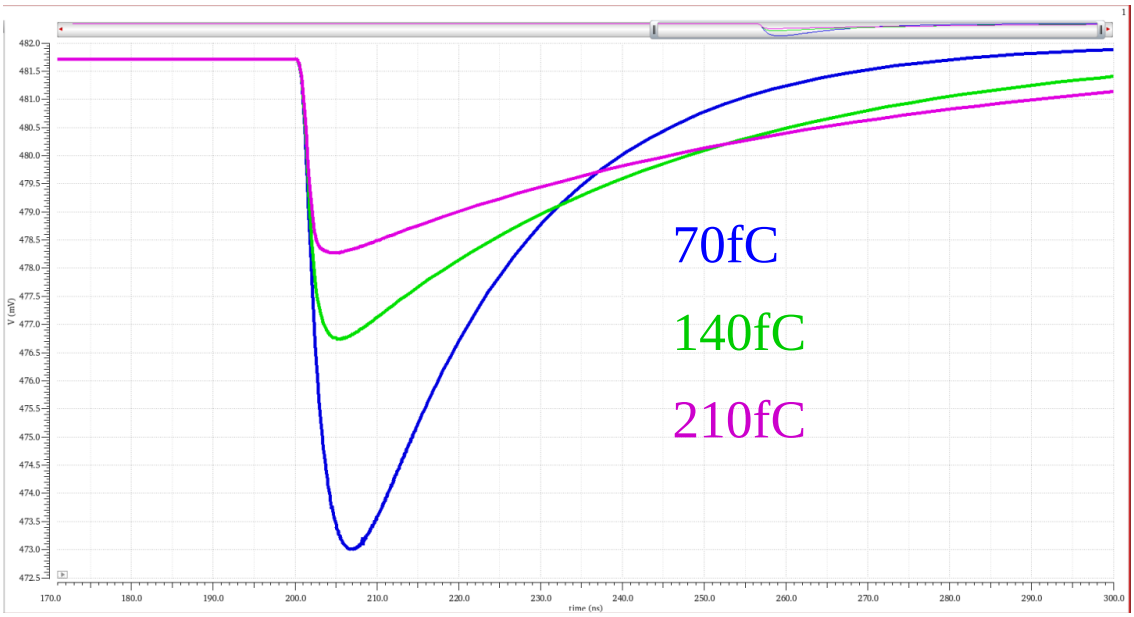


Figure 4. Preamplifier output signals vs Cf (fC)

The low power preamplifiers have a global adjustable Cin to adjust the gain with a constant 70fF Cf and 250k feedback resistor for the medium gains and 100k for the low gain preamplifier.

When the gain switching occurs the feedback resistor is disconnected giving a virtually infinite feedback resistor. This is done to improve the linearity of the measurements.

Also, when the gain switching between medium gain1 and medium gain2 occurs the Cf is increased from 70fF to 560fF.

sw_cin1(fF)	**500**, **500**, **1000**	In parallel, these resistors provide 4 values to adjust the medium gain.
Sw_cin2(fF)	25, **50**, 100	Can be combined to provide 7 values which provide the gain of the preamplifier.
cstab(fF)	**500**	When the cin2 is too low, a 500fF capacitor can be connected between the input and ground to improve the stability of the low power preamplifier.

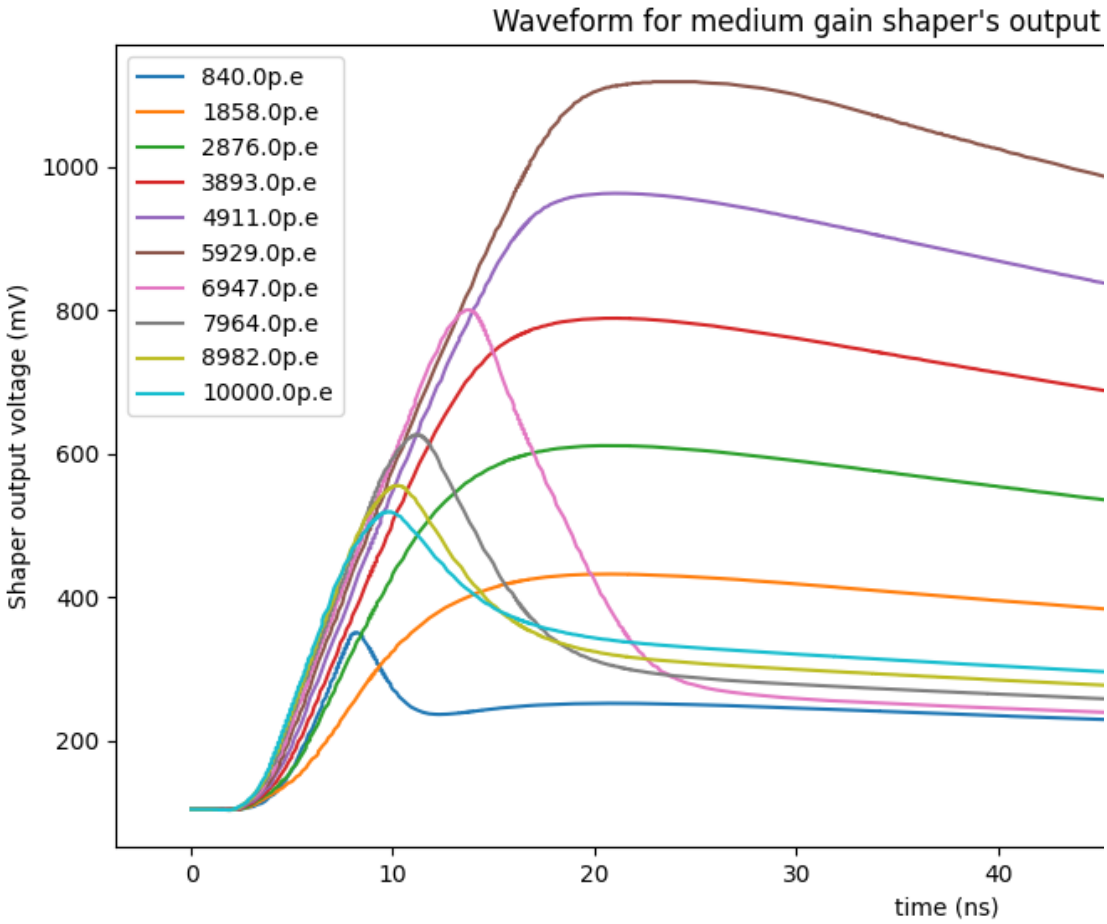


Figure 6. Gain switching shaper output

As it can be seen, there is a transition period between both gains, therefore the measurements done during this transition period should be ignored (first ADC point on the waveform).

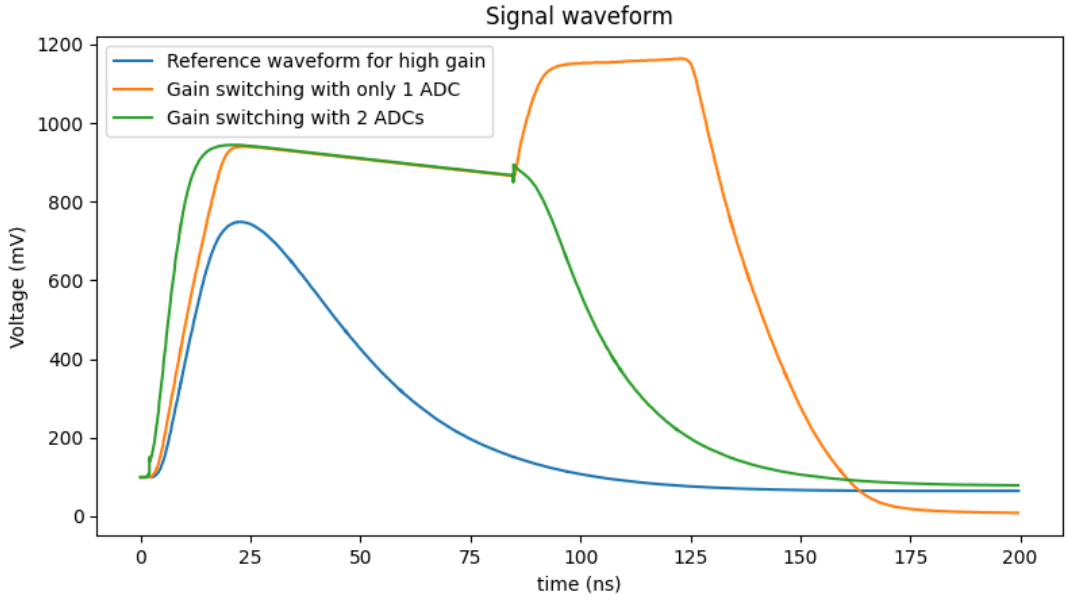


Figure 7. Gain switching reset

In figure 7 we observe the two different waveforms we can have after the reset. In the ADC input will be switched back to the high gain path giving a large (saturated) signal at first and a large undershoot after. In the case of using 2 ADCs the second ADC will switch back to the medium gain path, giving a smaller signal and undershoot. For larger signals we observe the same phenomena (saturation and then a large undershoot) for the medium gain. In all cases the reset will happen 75ns to 100ns after the arrival of the signal depending on the phase of the signal relative to the 40MHz clock.

2.1.2 High pass filter and discriminators

An active high pass filter is connected to "outpa" and the TOA discriminator. This high pass filter has an adjustable feedback resistance which can be used to decrease the time walk in exchange for a reduced SNR at the TOA.

sw_rf_OTA_TOA(Ω)	20k, 40k, **80k**, 160k	In parallel, these resistors provide 7 values to adjust the peak time.

Three other discriminators are connected to the high gain preamplifier and the medium gain preamplifier to generate the gain switching bits and the switching of the resistor.

A global 10b-DAC allows the user to adjust the thresholds of the discriminators and a local triming 6b-DAC (ref_toa<0:5>) allow to reduce the dispersion per channel, the 10b-DAC also has a polarity bit which allows the user to choose if the 10b-DAC will increase or decrease the base value. The 10b-DAC has a 2b cmd_refi to adjust all the 10b-DACs LSB while the trimming 6b-DAC has a 3b dac_GS to adjust its own LSB. The LSB is adjusted globally and not channel wise.

Toa_Threshold = Toa_vref<9:0> - Trim_dac_toa<4:0>

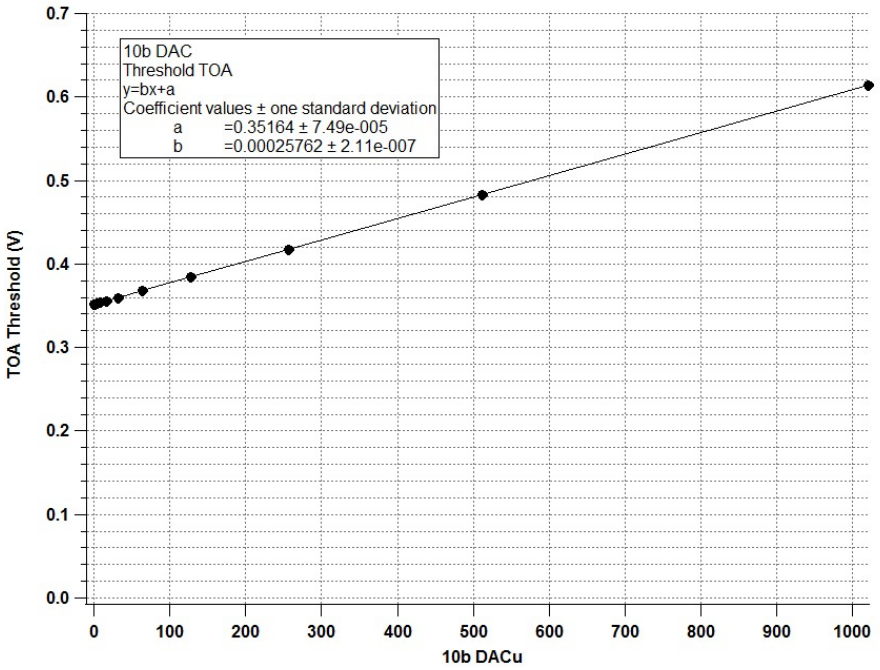


Figure 7. 10 bits DAC linearity. Common TOA threshold.

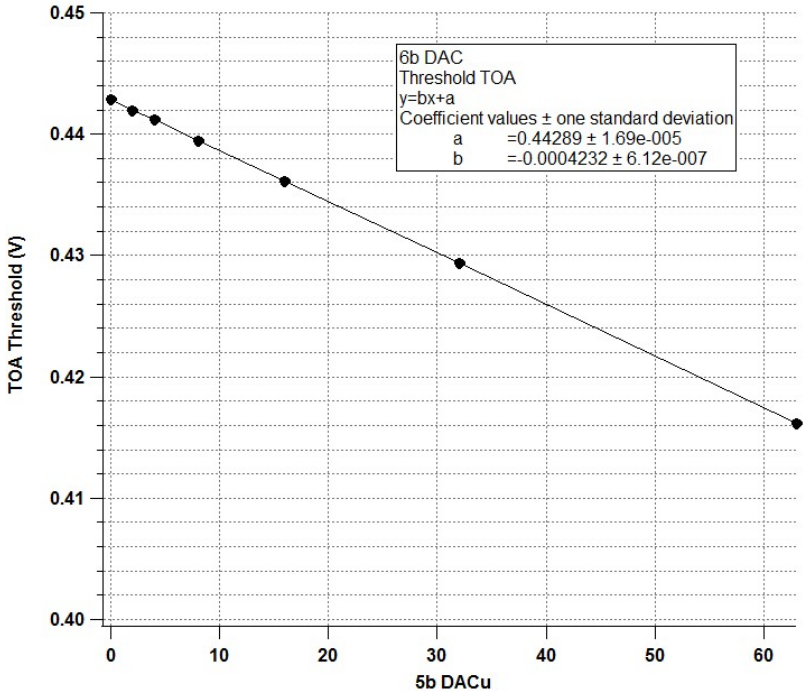


Figure 8. 6 bits DAC linearity. Ch by ch TOA threshold adjustment.

Since this circuit is used to read sensors with a high capacitance the amplification of the signal is considerably large, therefore the noise is also considerable with an RMS noise of 14.3mV after the high pass active filter using the default settings. Considering the peak of the signal, this would correspond to and equivalent noise of 97fC using 16 3x3mm SiPM in parallel (8.96nF). Therefore the minimum measurable signal would be at 6σ (580fC) for such input capacitance. Naturally, a lower input capacitance such as 4 3x3 SiPM in parallel would allow for a smaller threshold (145fC).

With a threshold at ½ of MIP. the time walk simulated gives 2.8ns for an input signal from 15p.e. to 100k p.e. As mentioned before, adjusting the feedback resistor of the high pass active filter can decrease this time walk further but in exchange the minimum threshold (in fC) will increase. (Although the Jitter stays relatively constant)

Figure 9. High pass active filter output voltage noise.

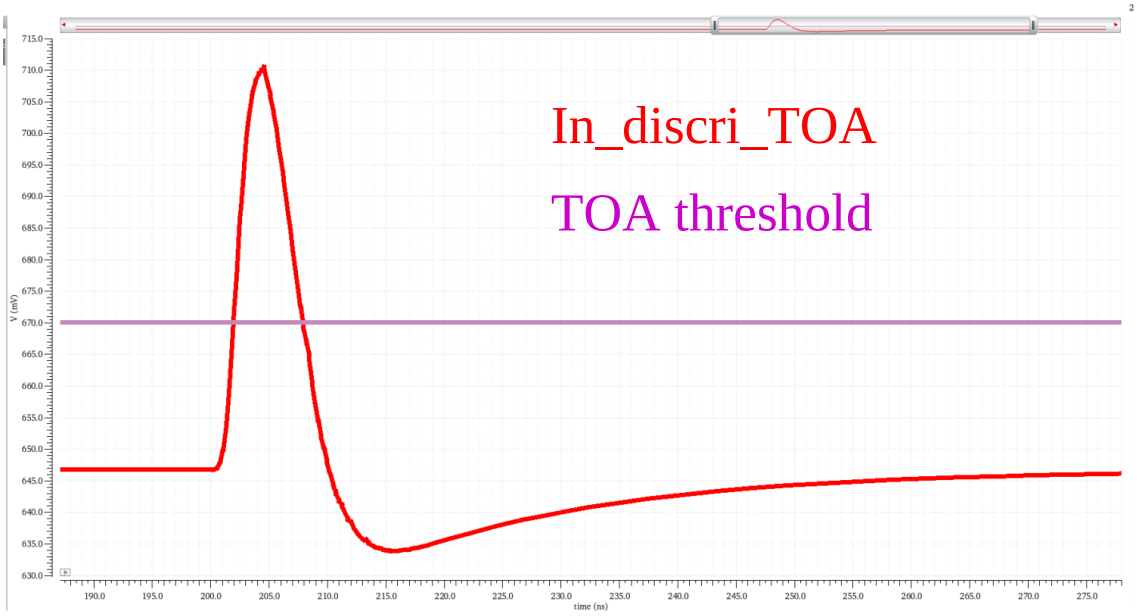


Figure 10 High pass filter output=TOA input signal and the threshold at ½ of MIP.

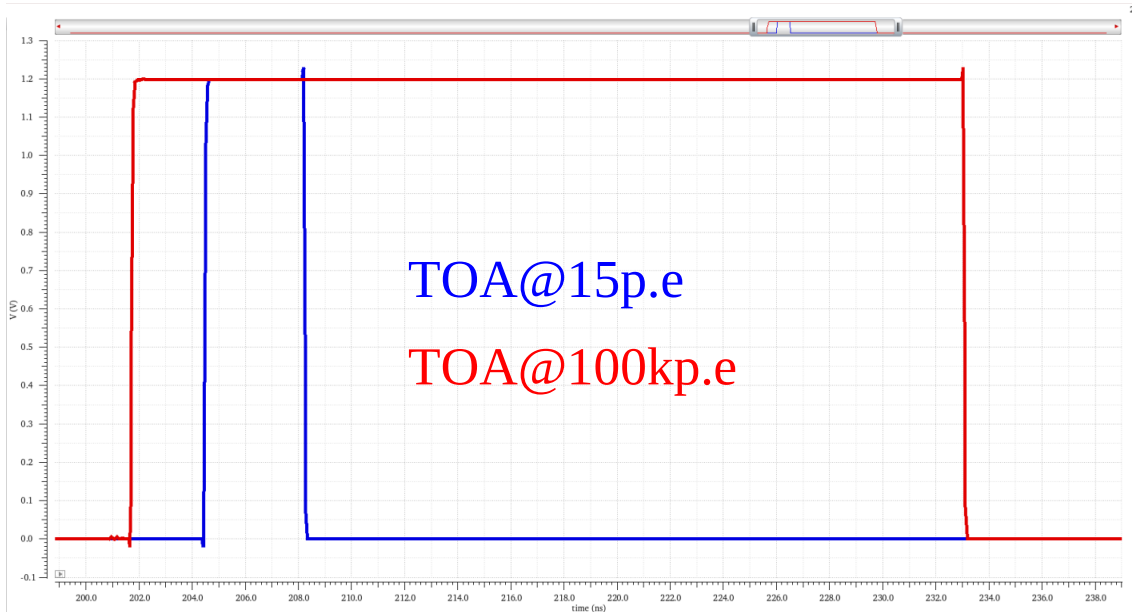


Figure 11 TOA trigger with input for 15 p.e. and 100k p.e. and threshold at ½ of MIP (15p.e). Time walk 2.75ns.

Two external triggers inputs are available (pins: Trig1_ext and Trig2_ext); typically, in the case when the user wants to calibrate the TOA, he can send a trigger for the TOA discriminator for a channel, and the other for a neighboring channel.

The two discriminators outputs can be masked per channel as well.

The following figure show the principle of the external trigger usage. The Trig1/2 are without effect when they are tied to 0: pull-down resistors have been placed to disable them by default. To send a trigger to a chosen channel, all the other channel which won't be triggered either by trig1 nor trig2 must be masked.

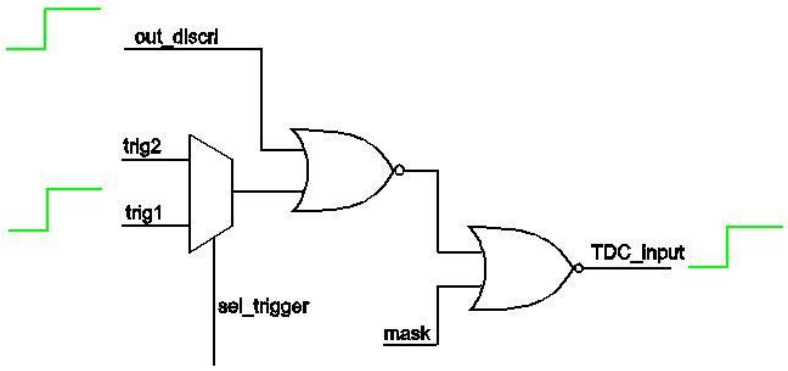


Figure 12 External trigger and mask system.

The two discriminators outputs can be probed and looked at on a scope.

2.1.3 Shapers

There are three shapers, one for each preamplifier:

It is a 1st order RC shaper with the peaking time typically set around 25ns.

The purpose is to optimize the signal-to-noise ratio and use the full available dynamic range (~ 1 V).

The user has to set the Vref_sk, Vref_sk_LP1 and Vref_sk_LP2 using the 10b-DACs to globally set the DC levels of the respective shapers, and so set the ADC pedestal. These 10b-DACs have typically 1mV LSB. In order to reduce the dispersion per channel, the user can play with a channel-wise trimming 8b-DAC (0.8mV/DACu).

Gain_sh<3:0>	40k, **50k**, 100k, 200k	High gain RC variable shaping time
Cf_sk<3:0>	**50fC**, 100fC, **200fC**, 400fC	High gain feedback capacitor for time shaping
Gain_sh_LP<2:0>	**30k**, 45k, 60k	Medium gain RC variable shaping time
Cf_sk_LP<2:0>	**50fC**, 102fC, 189fC	Medium gain feedback capacitor for time shaping
Gain_sh_LP2<2:0>	**30k**, 45k, 60k	Low gain RC variable shaping time
Cf_sk_LP2<2:0>	25fC, **50fC**, 100fC	Low gain feedback capacitor for time shaping

2.1.4 Analog multiplexer

The analog multiplexer is controlled by a state machine which choses what shaper should be connected to each ADC. This state machine has many slow control parameters that allows the analog multiplexer to work with only 1 ADC to decrease the digital noise and reduce the power consumption or to keep the connections static (no dynamic switching) which can be useful for calibration. This parameter will be described in the I2C parameters section to avoid redundancies.

2.1.5 ADC buffers

As mentioned there are two ADC buffers for each ADC, one is a rail to rail shaper connected as a buffer and the other is the same kind of shaper connected as a unitary gain inverter amplifier. This allows us to create the differential signal needed for the ADC without introducing many mismatch variations.

The shaper, which is connected as an inverter amplifier, is connected to Vref_inv, which is set by default to be the common mode of the ADC. An 8b-DAC can be used to increase this value globally and a 6b-DAC can be used for trimming channel wise. This dacs are set to 0 by default but can be used to compensate the mismatch between channels.

The following plots show the non-inverted buffer output for a charge from 15 p.e to 400 p.e., the inverted signals. The pedestal are respectively at ~ 100 mV for the non-inverted, 1.1 V for the inverted signal (obtained with the default values of the slow control bits).

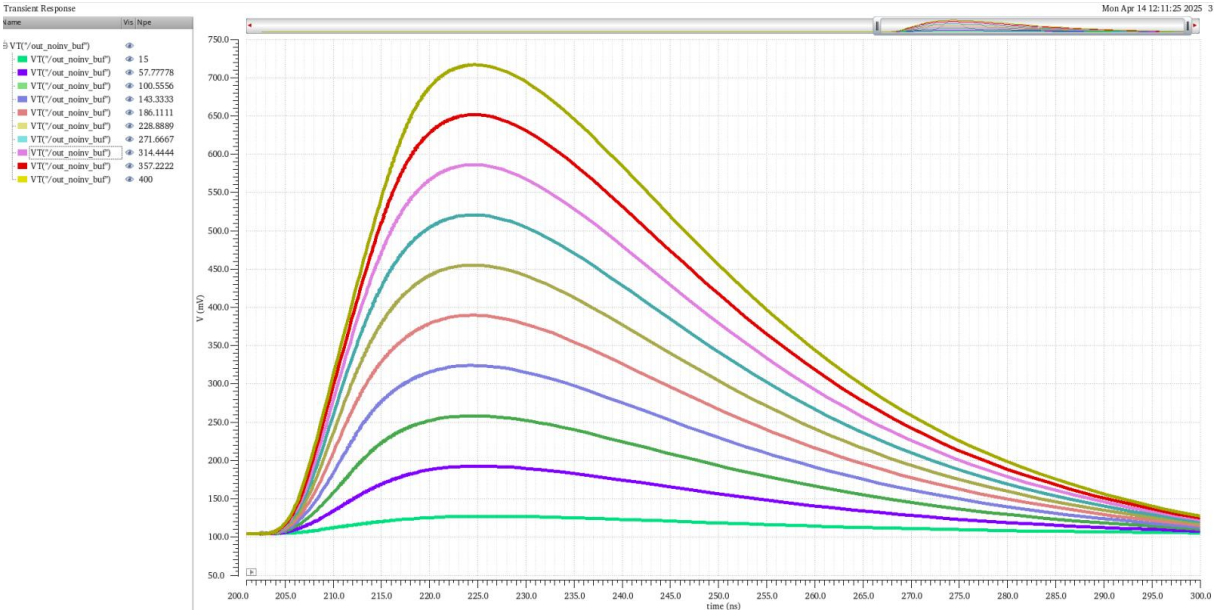


Figure 13 Non inverted buffer output.

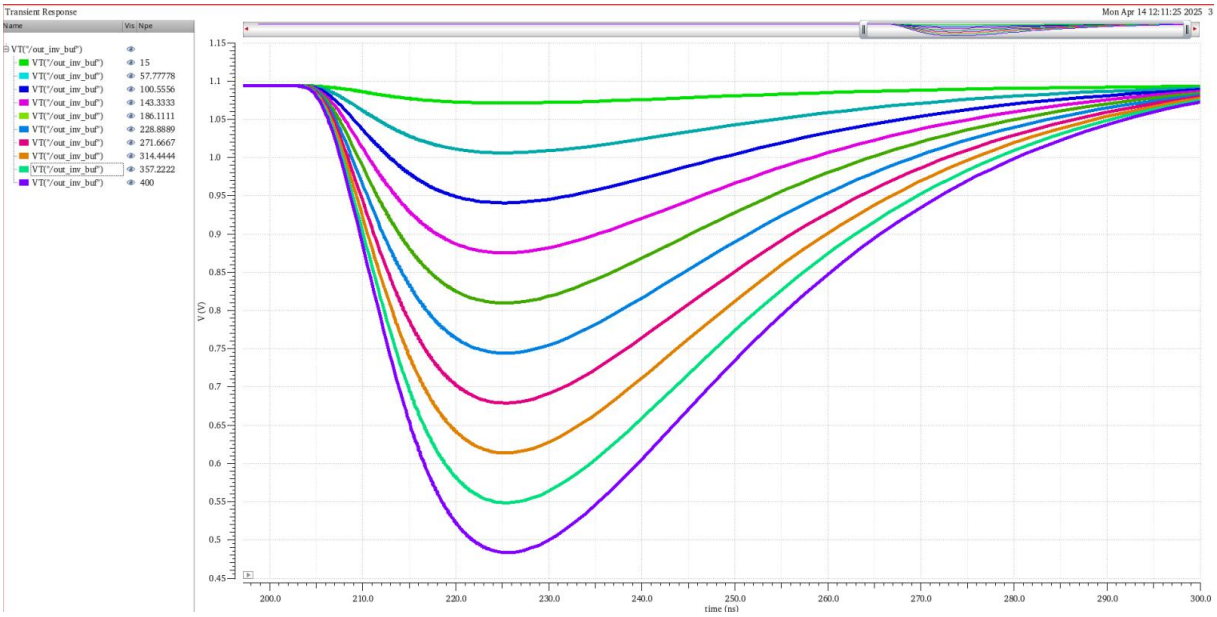


Figure 14 inverted buffer outputs.

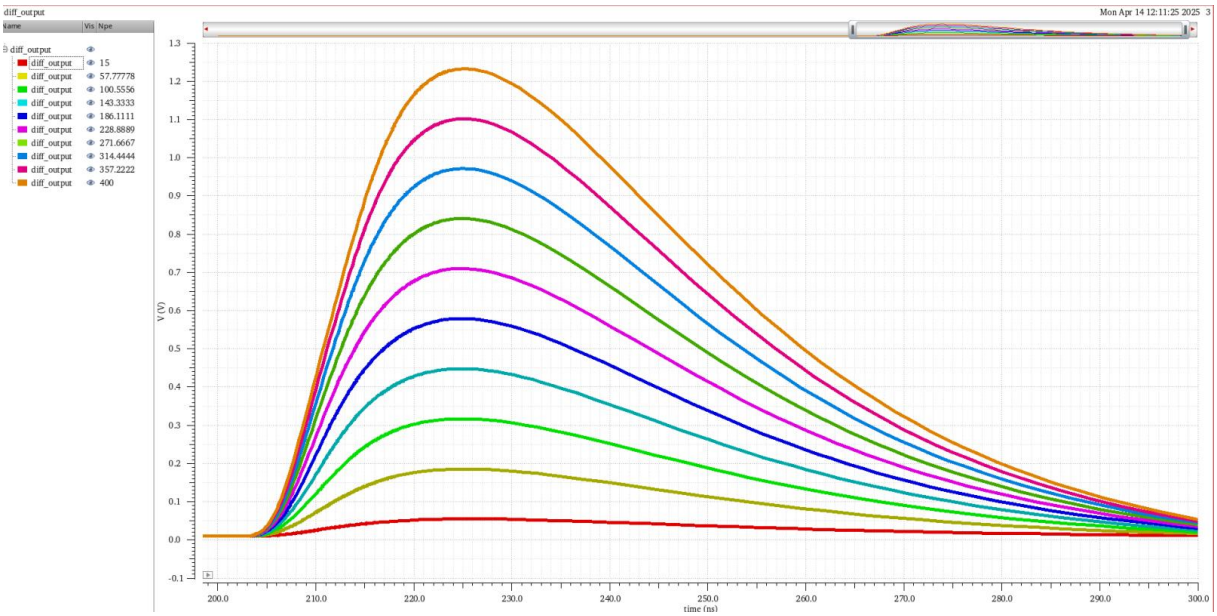


Figure 15 Differential buffer outputs signals from 15 p.e. to 400 p.e.

The linearity of the circuit using the gain switching was simulated taking into account the first point of the signal measured by the ADC (25ns) for the high gain and the second point for the medium and low gains (50ns) to avoid the non-linearity that comes from the transition between gains.

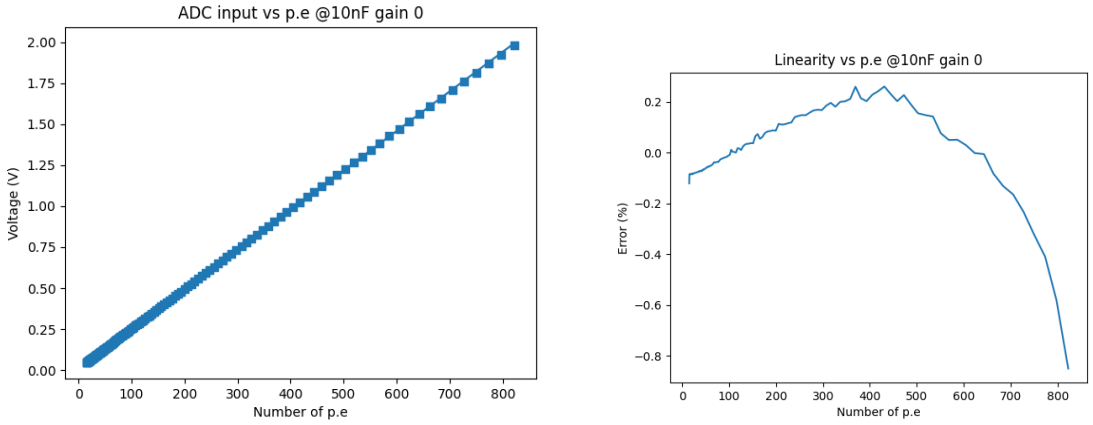


Figure 16 High gain linearity

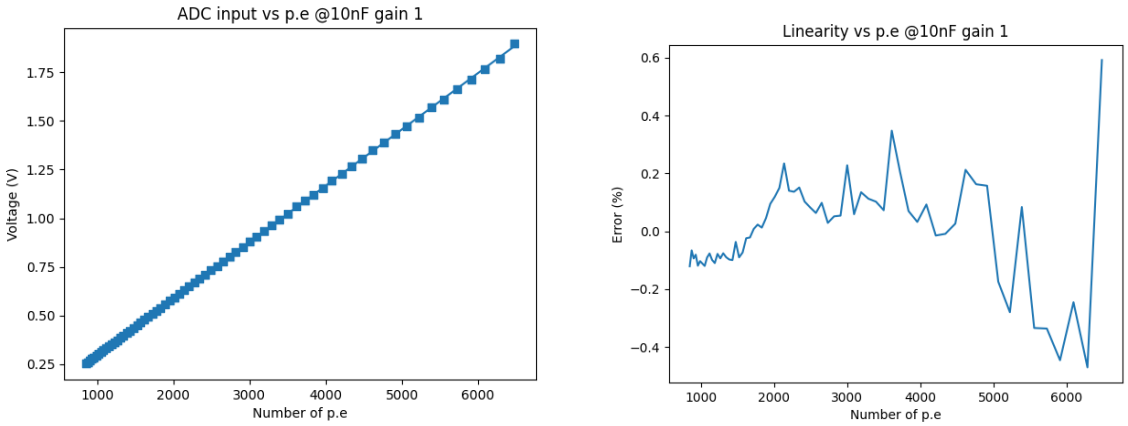


Figure 17 Medium gain 1 linearity

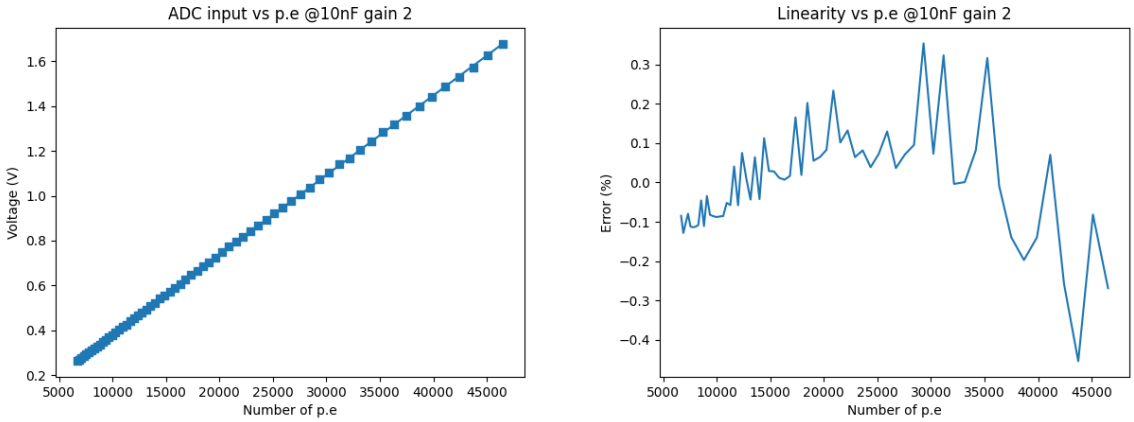


Figure 18 Medium gain 2 linearity

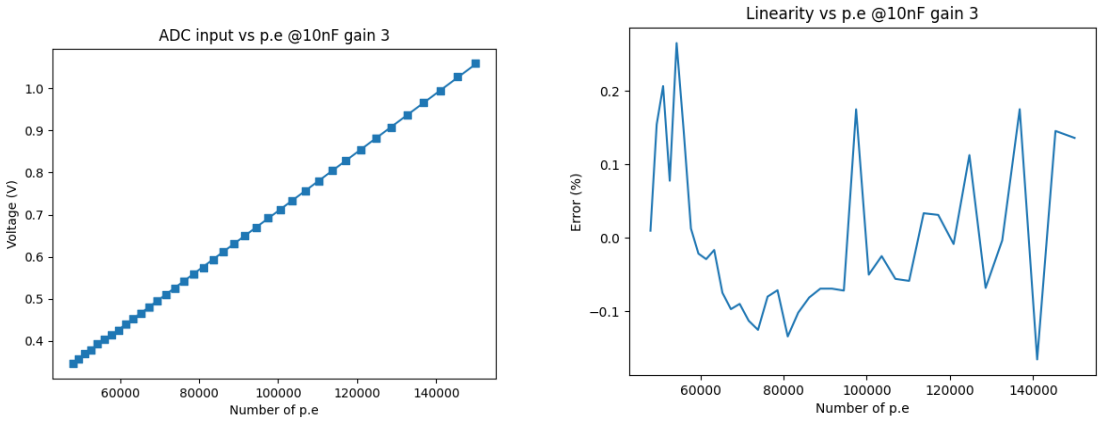


Figure 19 Low gain linearity

2.1.6 Bias

All the bias are generated from the bandgap voltage so that the power consumption and DC values are less sensitive to temperature and power supply variations.

2.1.7 Input amplifier

The channel input DC voltage is set by an amplifier connected as a buffer in series with a resistor. This allows us to decrease the electronic noise while keeping control over the channel input DC voltage.

The DC input voltage is determined by a channel wise 8b dac which will decrease the input voltage from 1.1 volt down to 300mv giving a range of 800mv with a LSB of 3mv.

The resistor can be chosen to be 50Ω or 200 Ω. And in case it is not needed, the input amplifier can be turned off and the resistance can be connected directly to ground. Doing this will reduce the power consumption and noise in exchange for losing the functionality.

When using this functionality it is advised to set the DC voltage as low as possible in order to keep the highest dynamic range possible as this readout circuit has been developed to read positive signals.

2.2 Mixed-signal blocks

2.2.1 10 bits ADC

The CALOROC1B contains two 10b SAR ADC, designed by AGH in Krakow. The ADC's vrefm reference voltage is tied to ground.

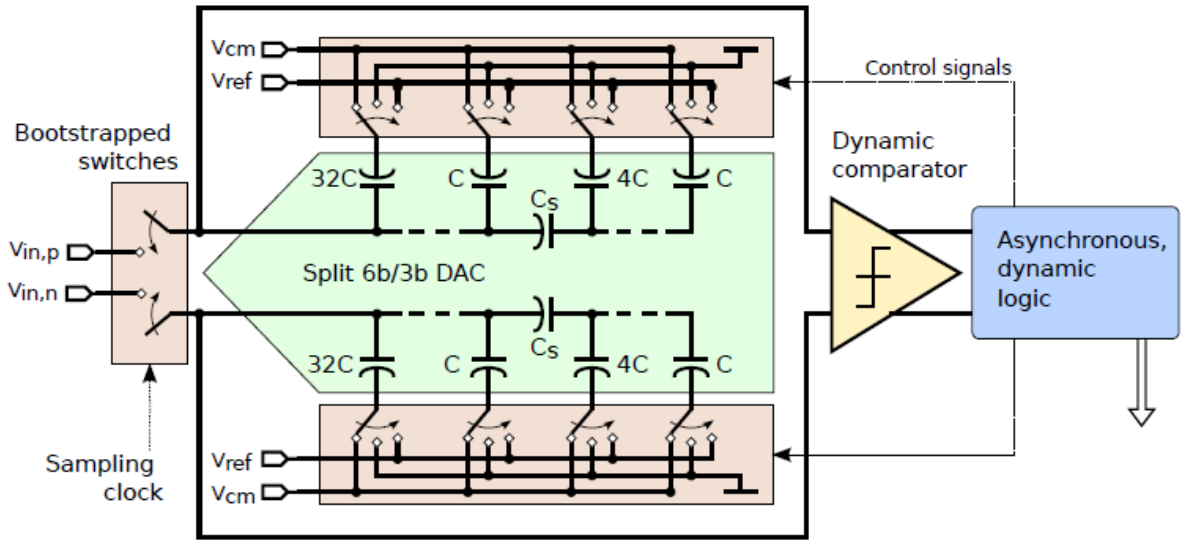


Figure 5 ADC block schematic

The 10 bits data provided by both ADCs are sent to an Align Buffer to align them to the TOT and TOA data.

A mask can be applied channel-wise to each ADC which stops the clock. Half-wise the asynchronous conversion delays can be adjusted over 3 bits.

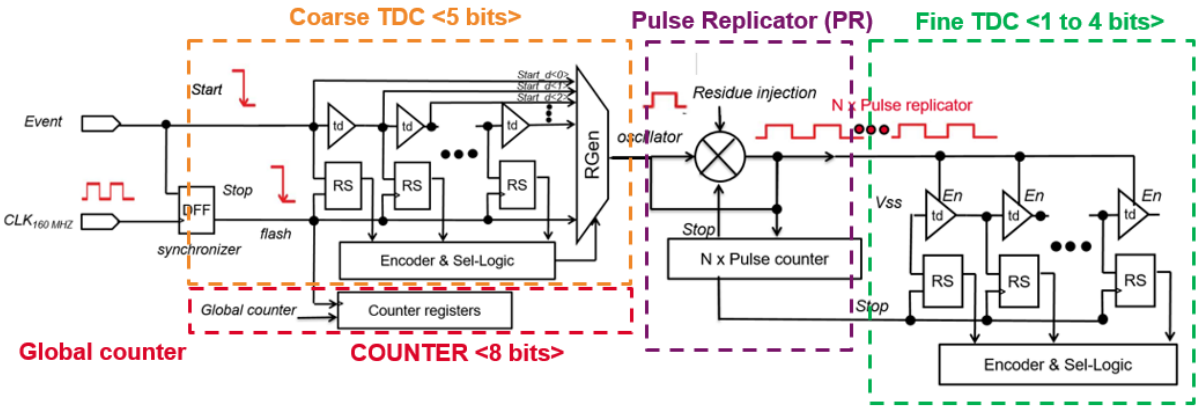
2.2.2 TOA TDCs

One TDC block handles the TOA measurement. It was designed by the CEA IRFU group in Saclay. The following table give the specifications of the TOA.

TDC ToA specifications	
Resolution	about 25 ps RMS
Range	10 bits over 25 ns
Conversion rate	> 40 MHz (bunch clock)
Power consumption	< 2 mW / channel
Area	Pitch 120 µm
Technology	TSMC 130 nm
Temperature	-30 °C

The architecture of the TDC is based on the time residue amplification method. It makes use of three main conversion blocks. A common **Gray counter** is used for the MSBs of all the channels. It is followed by a **Coarse TDC** (CTDC) based on an interpolator built using tapped delay line providing intermediate bits. The CTDC includes a pulse residue extractor. The residue is multiplied into a train of pulses by a **Pulse Replicator (PR)** and the resulting residual pulse train is sent to a residue integrator called **Fine TDC (FTDC)** that makes use of a DLL line to obtain the LSBs.

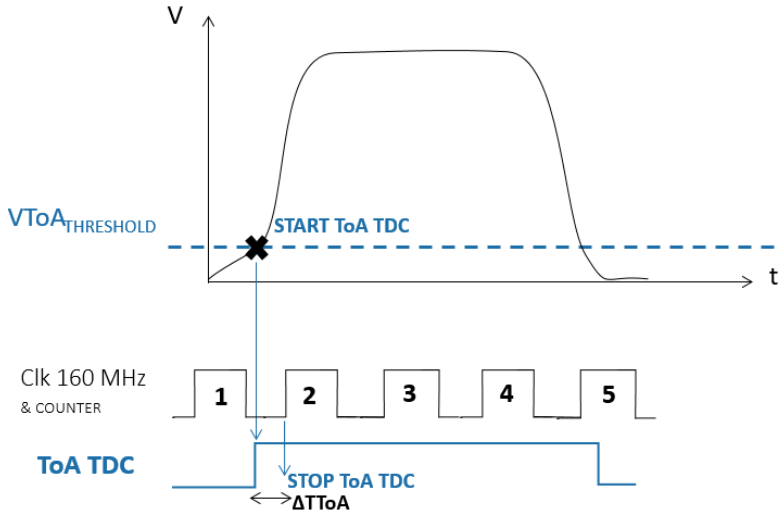
The schematic below gives an overview of the TDC circuitry.



2.2.2.1 The Time of Arrival (TOA) TDC

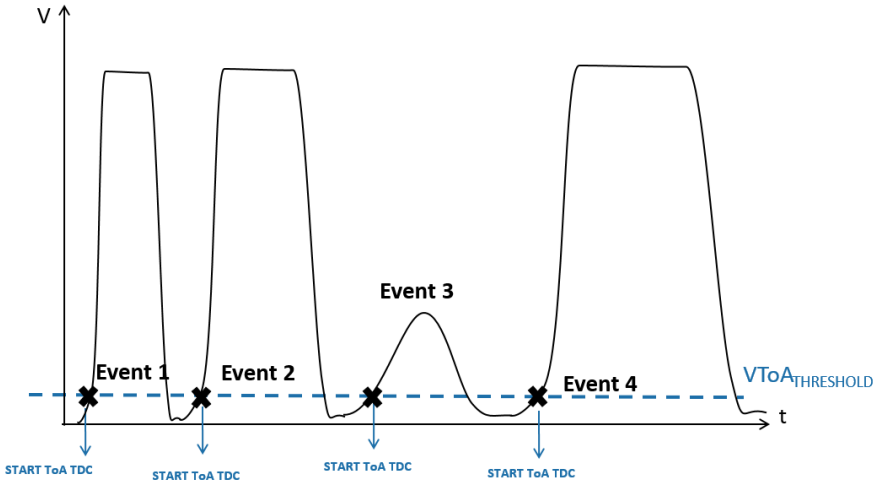
For the TOA measurements, the start signal for the delay line of the CTDC is a digital signal generated by the discriminator of the analog front-end electronics (FEE) of the CALOROC1B chip. The propagation of the signal in the DLL is stopped with the next rising edge of the 160 MHz clock from the on-chip PLL. The thermometer code output obtained from the CTDC and the FTDC delay lines is then encoded and converted to the binary code.

To measure TOA with a TDC, the analog input signal is compared with a programmable threshold (higher than the noise floor by the on-chip FEE). The following figure illustrates the TOA pulses for an input charge.

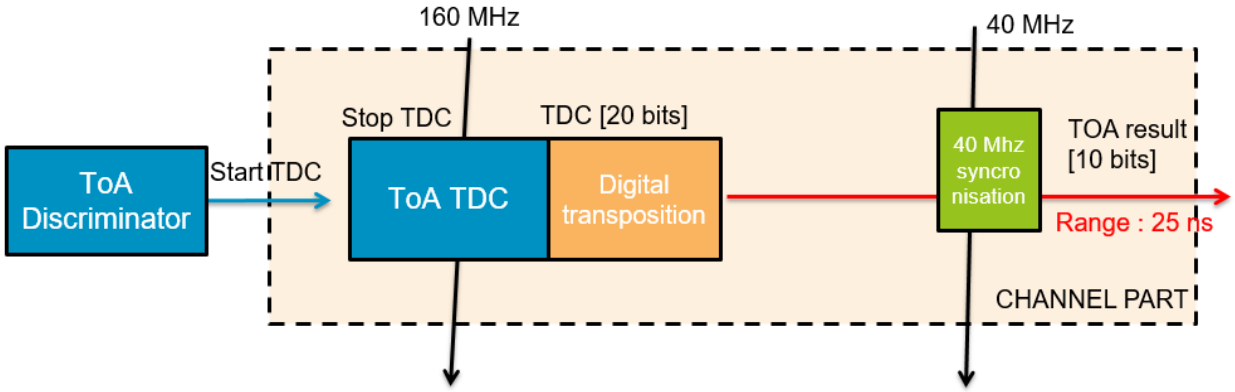


The rising edge of the TOA discriminator initiates TOA measurement that is available immediately after that the conversion is finished. However, the proposed architecture required also a TOT measurement. This TOT measurement is not useful for the CALOROC1B ASIC so to simplify the overall design and to be conservative respect to the original design of this TDC (designed for the ASIC HGCROC) , the TOA is stored in the on-chip DRAM memory when the TOT measurement is over.

The proposed solution for the TOA measurement is to tag the time when the high pass filter output signal cross the threshold. The following figure shows the general principle of the TOA measurement:



This architecture can be used only for input analog signals for which the time difference between the crossing points of the TOA threshold can be detected.



The TOA value is obtained directly from the TOA TDC measurement after a digital transposition in 10 bits. The TOA result is valid during one 40 MHz clock period and is written to the on-chip DRAM. For non-TOA events, the DRAM is filled with the all-zero code. This type of architecture is able to convert at a frequency of 40 MHz

2.2.2.2 Synchronization

The ADC values are available immediately after conversion but the conversion and resynchronization of the TOA requires processing time. However, as described above, the TOA interval may vary from 0 to 25 ns. The on-chip Trigger unit requires the ADC and the TOA measurements to be present simultaneously at the same rising edge of the on-chip PLL 40 MHz clock. To achieve the synchronization, the TOA logic guaranties that the TOA measurement will be available after a fixed number of clock periods starting from the TOA discriminator rising edge. The following TOA logic is used:

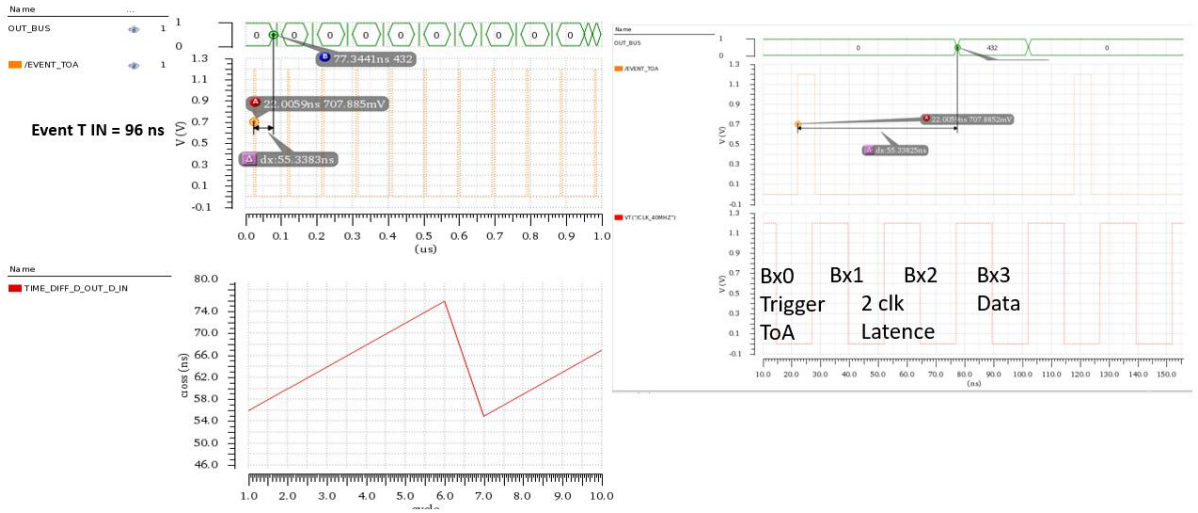
Bx0: the TOA trigger appears

Bx1 and Bx2 => latencies

Bx3 the data is present

- rising edge at start of Bx3 set the TOA value

- rising edge at the end of Bx3 is the one used by the next stage to sample this TOA value



According to the proposed design, after a TOA hit, the TOA is measured and stored into a TOA FIFO and a counter is started to count 2 x 40 MHz clock cycles. This creates a so-called TOA latency window.

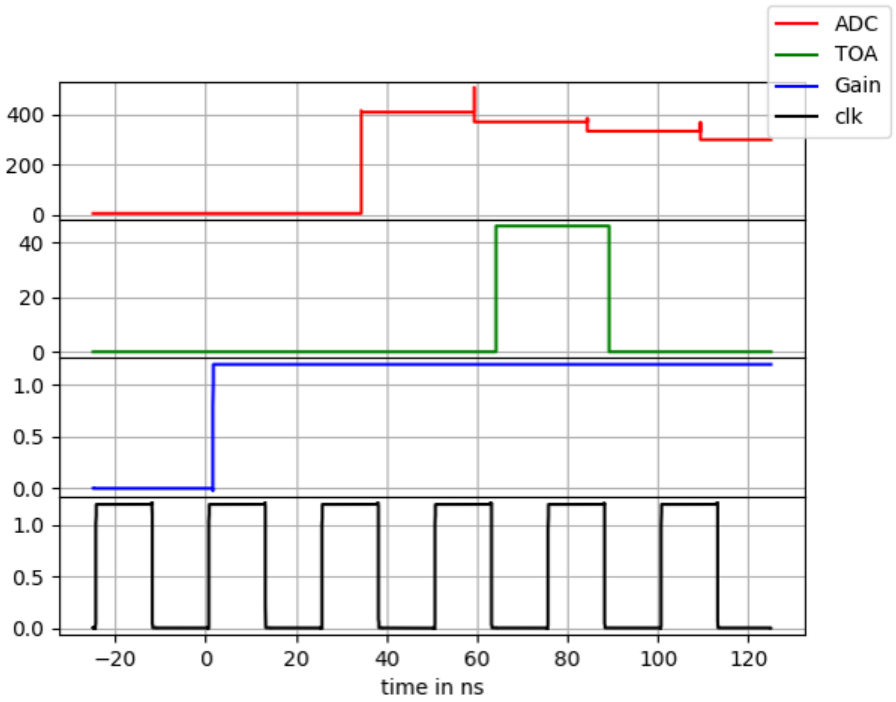
If there is no TOA hit, the TOA data is accompanied by a “0” code to indicate the absence of the TOA information taking into account the latency window.

If a new TOA event occurs during an active pending TOA[n-1] latency window, the new TOA[n] measurement is appended to the FIFOs, a new latency counter is started and the TOA[n] latency window is initiated.

2.2.2.3 Align buffer

The TOA, ADC and gain switching signals are not aligned by default, the align buffer in the waveform digitizer adds a latency to the ADC and gain switching signals to make sure they are aligned with the TOA. Additionally, a smaller second align buffer adds an extra latency to the gain switching signal to make sure it is well aligned with the ADC.

Since the TOA is used to start saving the measured point in the memory, an ADC latency of 3 is used to ensure that for all phases we will measure at least one point of the pedestal. By default, we save 4 samples (N = 4) of the ADC.



For the phase 0 (event aligned with the 40MHz clock), we can observe that with an ADC latency of 3 and 4 samples, we measure two points of the pedestal and two points over the waveform. As mentioned previously, the first point of the ADC signal has nonlinearities when using the gain switching, therefore we obtain one linear sample of the ADC in this case. By adding an additional delay of 1 to the gain switching we obtain:

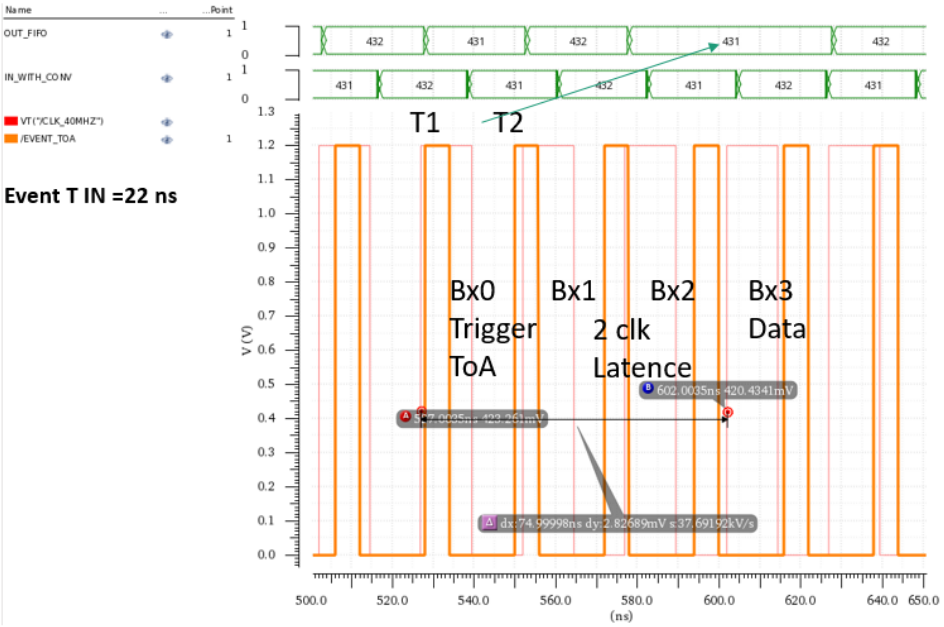
LAT_ADC	0	1	2	3
Gain	3	3	0	0
ADC3g/4g	390	400	60	60
TOA	45	45	45	45

The data acquisition starts with the TOA signal, therefore this aligns the ADC and gain values correctly with the TOA values. A lower latency is equivalent to reading a later sample, for example, the second sample with a latency of 3 will be equivalent to the first sample with a latency of 2, this way we can use this table to see all 4 samples when using a latency of 3.

Depending on the phase, the two middle points may measure the pedestal or the waveform. As we can see in the appendix 6.1), for all phases we have at least one point for the pedestal and at least one linear point for the ADC.

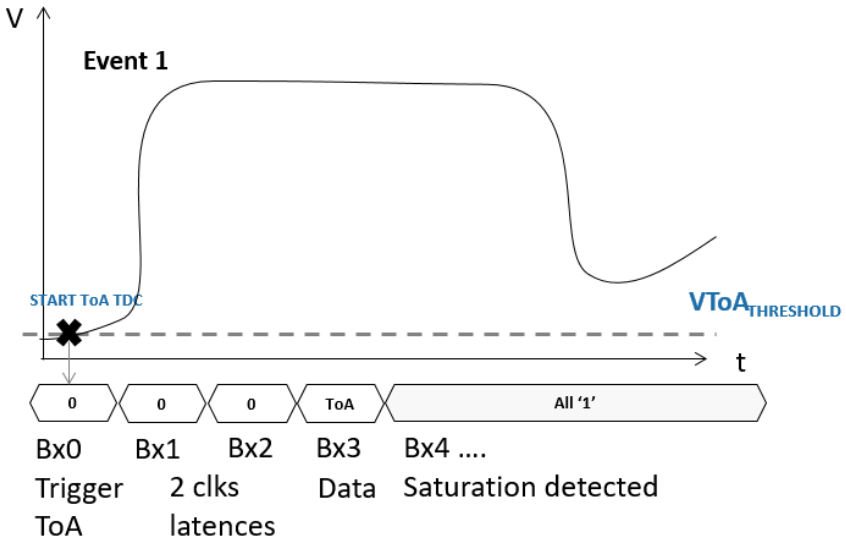
2.2.2.4 Pileup effect

Taking into account the discharge time of the analog part, the minimal time between two consecutive TOA events can be as low as 25 ns. During one TOA Bx, several other charge deposits may occur within the same channel so we can have 2 triggers in the same Bx. The design guarantees that the TOA will be measured the 1st and delete a 2nd trigger in the same Bx:



2.2.2.5 Discriminator saturation

In case of high rate pileup, the preamplifier output signal may not return to its baseline after a TOA event when a new charge arrives. In this case, a second TOA may appear without the TOA threshold being crossed and the discriminator is in saturation region as shown on the next figure.



If the output of the discri is at '1' for more than 4 x Bx I force the outputs of the TOA to '1' until the trigger is re-decent with a deactivation. If the channel discriminator is saturated at the end of 4 BX and after the TOA data, we set all the 10 bits data bits to 1023 (all '1') indicating a saturation of the channel.

As soon as the discriminator drops below the Toa theshold, the TOA value will return to a '0' in the next Bx indicating that the channel is ready to trigger again.

This mode can be interesting for the calibration of discriminators because we could know when the discri is under the threshold in static!

2.3 Ancillary blocks

2.3.1 PLL and clocks distribution

The main specifications of the PLL are described in the following table:

PLL specifications	
Input frequency	40 MHz (LHC bunch clock)
Output frequencies	1.28 GHz and 640, 320, 160 MHz
Jitter cleaner	Low jitter < 15 ps RMS (for an input jitter of 30 ps RMS)
Power consumption	< 2 mW
Area	Pitch 200 µm
Technology	TSMC 130 nm
Temperature	-30 °C

The jitter cleaner and clock synthesizer PLL provides a set of on-chip clocks with frequencies varying from 40 MHz to 1.28 GHz, all phase aligned to its 40 MHz input clock. It consists of a low noise digital phase frequency detector (PFD), a precision charge pump, a loop filter, a voltage-controlled oscillator (VCO) and prescale dividers. The PLL must have its own power supplies and reserved pads (VDD_PLL and GND_PLL).

The chip only receives the Fast-Command link made of the 320 MHz fast command and the 320 MHz clock. The FastCommand block decodes the fast command and outputs two sets of 40 MHz clock. One set (Cki2c_*) for the I2C block and the other set (Ck40fc_*) for all the FSMs, the counters, the two RAMs, the wr/rd pointers and the PLL. Hard and soft resets (hard_resetb and soft-rstb) can reset the FastCommand block.

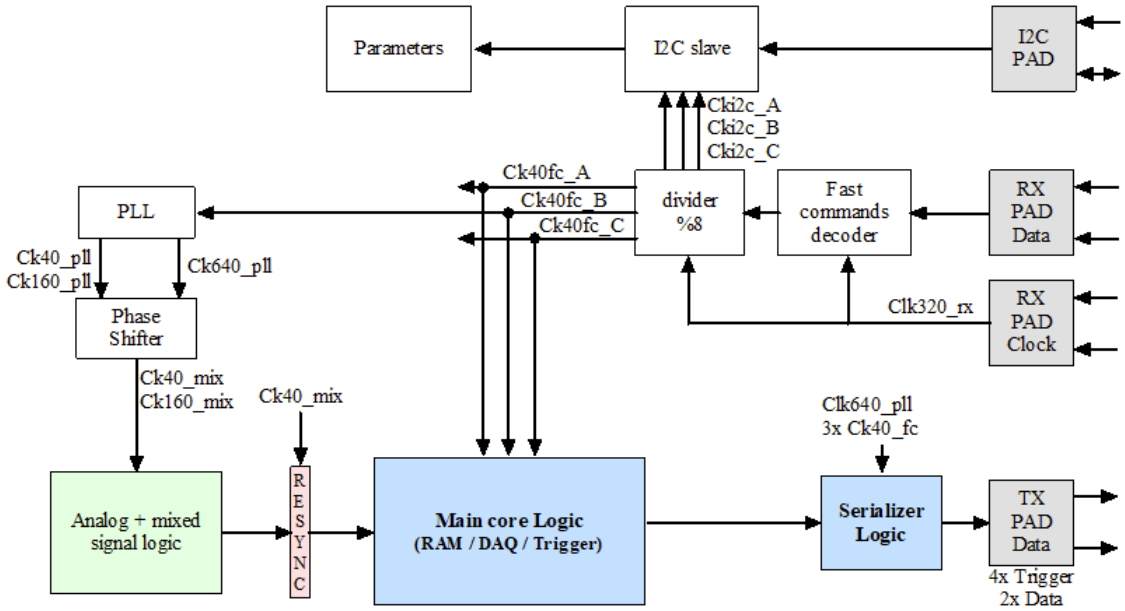


Figure 6

The FastCommand block decodes the 40MHz clocks from the 320MHz links. The 40MHz clock for the I2C is directly divided from the 320MHz clock input, so that the chip can be configurable even if it does not receive any fast commands. The 40MHz clock for all the others parts (digital, PLL, mixed) are decode from both 320MHz and FastCommand links, so that the chip needs correct fast command to provide this clock.

The PLL takes as reference the 40 MHz clock (Ck40fc_B) provided by the FastCommand block. The PLL can only be reset by the hard/soft resets (hard_resetb and soft-rstb). The PLL generates three clocks: clk_40M_pll, clk_160M_pll, clk_640M_pll.

- Clk_40M_pll: this clock comes in a Phase Shift block to achieve the ADCs clock
- Clk_160M_pll: this clock comes in a Phase Shift block to achieve the TDCs clock.
- Clk_640M_pll: this clock comes in the SerDes of the E-links.

2.3.2 Bandgap and voltage references

Typically, the bandgap provides an output voltage of around 330 mV (vbg). This is multiplied to get a usable voltage reference around 1 V (Vbg_1V). This bandgap block provide a Vbg-based current (ref-i).

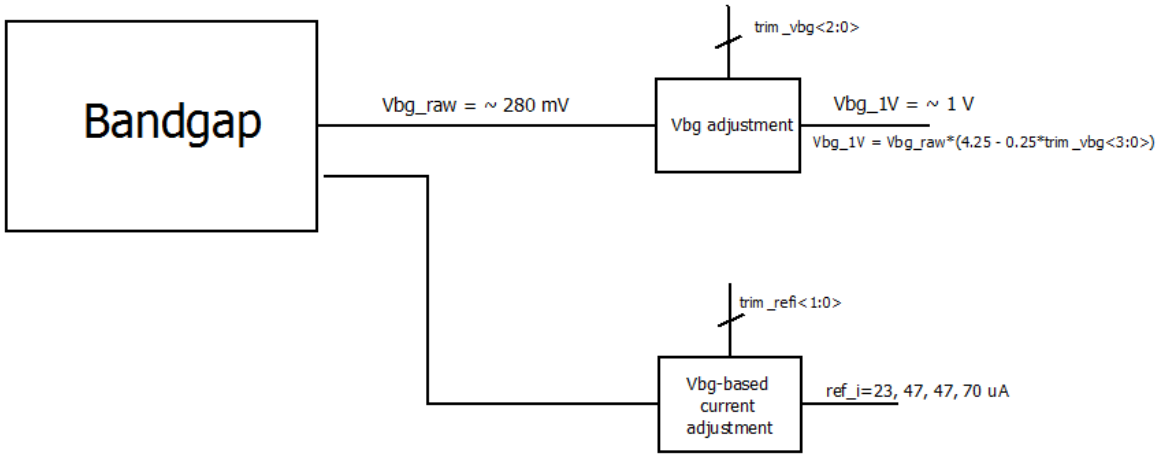


Figure 7 Bandgap schematic block

The Vbg_1V is used to generate:

- The upper reference voltage of the Calibration 12b-DAC,
- The reference voltage of the ADCs,
- Ref. voltage for the TDCs, bias.

The ref_i reference current is used to generate the offset and the steps of the 10b-DACs.

This ref_i allows to fabricate the voltage references for the analog part thanks to 4 10-bit ADCs:

- the two discriminator thresholds,
- the inverted and non-inverted references for the shaper

The outputs voltage values of a preamplifier, similar to the PA of the classical channel (extra PA added in the Bias block), is used to give these voltage references. This is done to make the chip not sensitive to the temperature. So all the references follow the preamplifier input DC value.

2.3.3 Calibration circuit

The following scheme shows the circuitry of the calibration 12-bits DAC. The 12b-DAC ("Reference Voltage I2C" "Calib_dac") provides a voltage up to around 1.2 V (the supply value) and the injection capacitance if of 67.48pF. Have to set Inctest at 1 ("Reference Voltage I2C" "Lowrange"). STROB= Fast command CalPulseInt: send STROBE pulse to the internal calibration DAC

The ctest node comes from an external pad and allows the user both to use an external pulser rather than the calibration DAC and also to calibrate the calibration 12b-DAC.

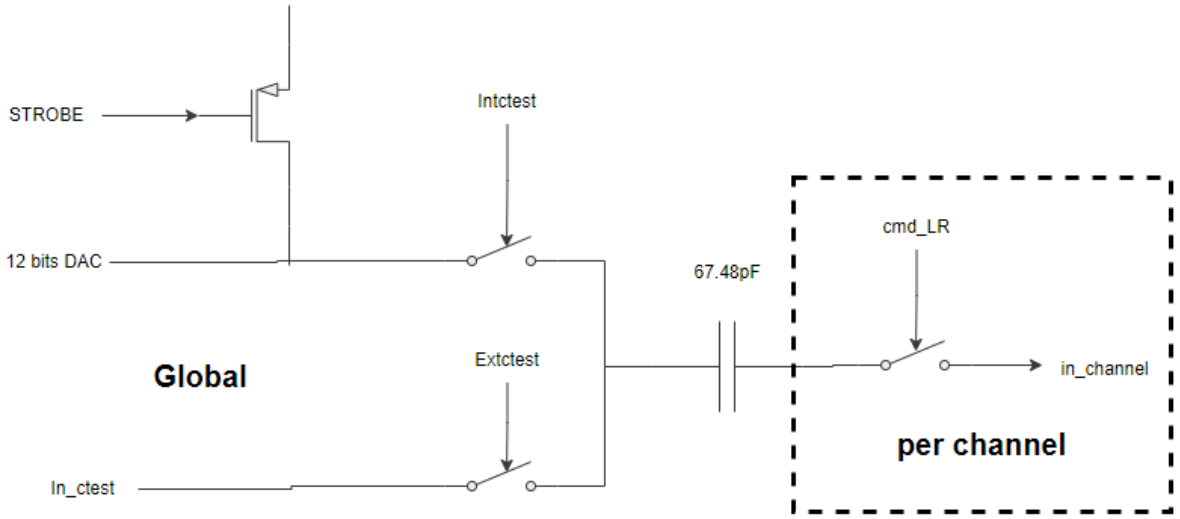


Figure 8.1 Calibration schematic

CalPulseExt and CalPulseInt are internal signals which come from the fast command circuit (integrated in the main digital circuit) while Strobe_ext is a signal which comes from outside the circuit.

Strobe_ph is the signal that drives the calibration circuit shown above, while Sipm_calib is a chip output

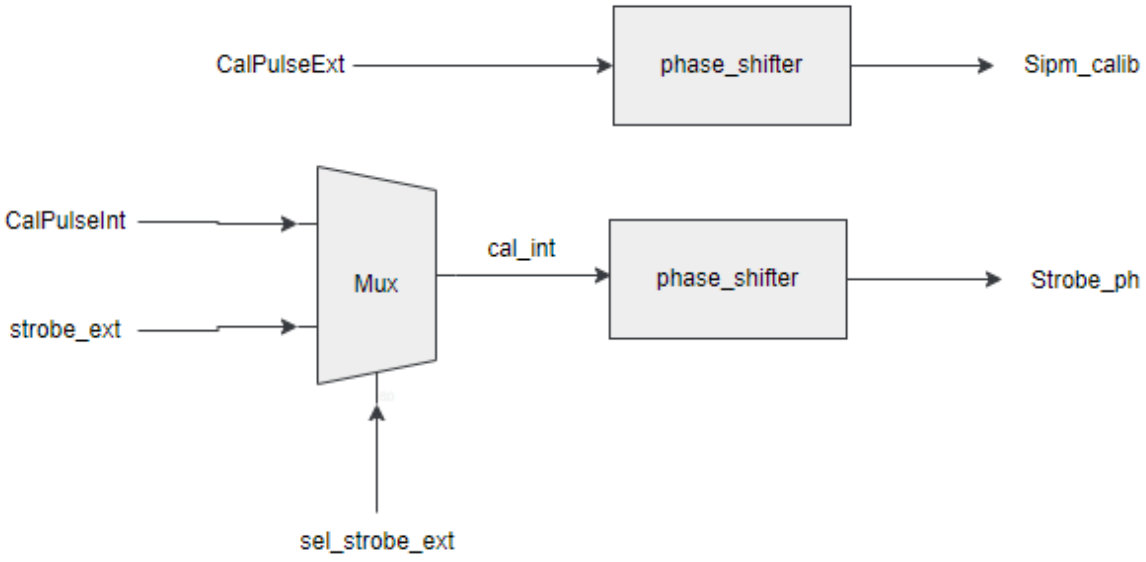


Figure 9.2 Calibration signals

2.3.4 Monitoring

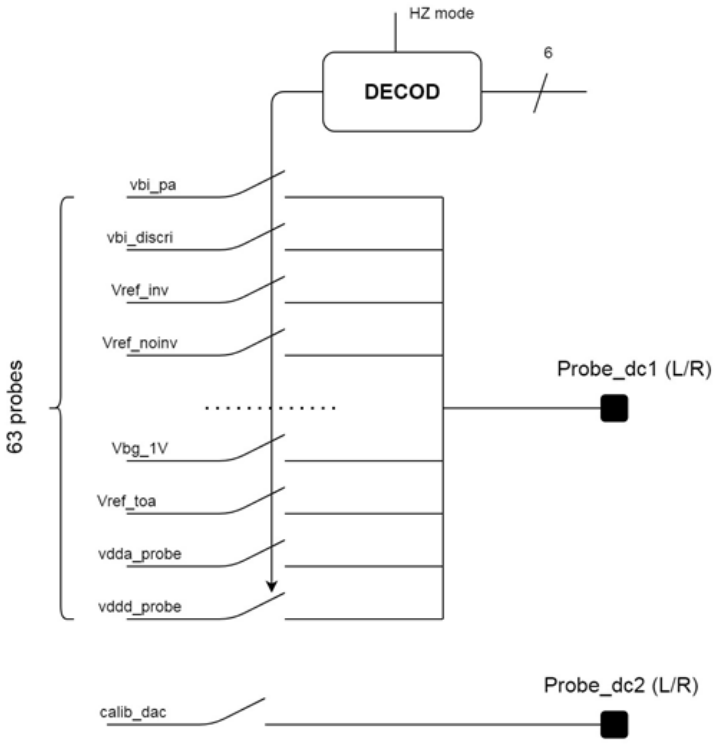


Figure 10 Probes DC

The internal DACs outputs (calibration, reference voltages and Current DAC) are multiplexed as described in the figure above. The signals will be sent to the ADC of the SCA chip. All the monitored points are in the 1V2 range.

On probe_dc1, there are 80 available monitoring points. A command of 3 bits : cmd_probe_dc1 allows to select the probe points from 0 to 31, from 32 to 64 or from 65 to 80.

If cmd_probe_dc1<0> = 1 the next 32 voltages are accessible:

ADDRESS	PROBE POINT	Nominal value (mV)	Comment
0	Vbm1_discri_toa	858	
1	Vbm2_discri_toa	848	
2	Vbo_discri_toa	512	
3	Vcasc_discri_toa	1008	
4	Vb_hyst_toa	475	Tunable over 6b (dac_hyst_toa)
5	Vbi_discri_pmos	848	
6	Vbm_discri_pmos	848	
7	Vbo_discri_pmos	245	
8	Vb_hyst_pmos	450	
9	Vbc_pmos_ampli	1028	
10	Vb2_pmos_ampli	344	Tunable over 6b (dac_pmos_opamp)
11	Vb1_pmos_ampli	959	Tunable over 2b (ibib_pmos_opamp)
12	vbiP_inv_buf	859	Tunable_over 2b (ibib_inv_buf)
13	vbiN_inv_buf	299	Tunable_over 2b (ibib_inv_buf)
14	vbFCP_inv_buf	845	Tunable over 6b (dac_inv_buf)
15	vbFCN_inv_buf	315	Tunable over 6b (dac_inv_buf)

16	vbiP_noinv_buf	859	Tunable_over 2b (ibib_noinv_buf)
17	vbiN_noinv_buf	299	Tunable_over 2b (ibib_noinv_buf)
18	vbFCP_noinv_buf	845	Tunable over 6b (dac_noinv_buf)
19	VbFCN_noinv_buf	315	Tunable over 6b (dac_noinv_buf)
20	Vcp_buf	589	
21	Vcn_buf	589	
22	vbiP_inv_buf2	859	Tunable_over 2b (ibib_inv_buf2)
23	vbiN_inv_buf2	299	Tunable_over 2b (ibib_inv_buf2)
24	vbFCP_inv_buf2	845	Tunable over 6b (dac_inv_buf2)
25	vbFCN_inv_buf2	315	Tunable over 6b (dac_inv_buf2)
26	vbiP_noinv_buf2	859	Tunable_over 2b (ibib_noinv_buf2)
27	vbiN_noinv_buf2	299	Tunable_over 2b (ibib_noinv_buf2)
28	vbFCP_noinv_buf2	845	Tunable over 6b (dac_noinv_buf2)
29	vbFCN_noinv_buf2	315	Tunable over 6b (dac_noinv_buf2)
30	Vcp_buf2	589	
31	Vcn_buf2	589	

If cmd_probe_dc1<1> = 2, the next 32 voltages are accessible:

ADDRESS	PROBE POINT	Nominal value (mV)	Comment
0	Vpa_Cf_ref	735	
1	Vb1_in_ota	436	Tunable over 2b (ibib_input_ampli)
2	Vb2_in_ota	1022	Tunable over 6b (dac_input_ampli)
3	NC_ota_in	NC	
4	Vcm_0p6_inv	600	
5	Vcm_0p6_noinv	600	
6	Vref_adc	1144	
7	Vcm_adc	507.6	
8	VD_FTDC_N_EXT	603	
9	VD_FTDC_P_EXT	77	
10	VC_CTDC_N_EXT	616	
11	VC_CTDC_P_EXT	77	
12	EXT_REF_TDC	467	
13	Vref_inv	600	Tunable over 8b (dac_ota_vref_inv)
14	Vb_5bdac_toa	781	Tunable over 3b (dac_GS)
15	Vb_5bdac_out_inv	681	Tunable over 3b (dac_inv)
16	Vb_5bdac_GS	781	Tunable over 3b (dac_GS)
17	Vb_8bdac_sk	278	Tunable over 3b (dac_vref)
18	Probe_vref_LP	443	Equal to vref_LP used to set Vref_RS
19	Probe_vref_Cf_LP	735	Equal to vref_Cf_LP used to set Vref_GS1
20	Probe_vref_Cf	776	Equal to vref_Cf used to set Vref_GS0
21	Probe_vref_ota_toa	634	Equal to vref_ota_toa used to set Vref_toa
22	Probe_vref_LP2_Rdiv_dac	333	Equal to vref_LP2_Rdiv_dac used to set Vref_sk_LP2
23	Probe_vref_pa	333	Equal to vref_pa used to set Vref_sk
24	Probe_vref_LP_Rdiv_dac	402	Equal to vref_LP_Rdiv_dac used to set Vref_sk_LP
25	Vref_RS	431	Tunable over 10b (bit_Vref_RS)
26	Vref_GS1	377	Tunable over 10b (bit_Vref_GS1)

27	Vref_GS0	378	Tunable over 10b (bit_Vref_GS0)
28	Vref_toa	670	Tunable over 10b (bit_Vref_toa)
29	Vref_sk_LP2	356	Tunable over 10b (bit_Vref_sk_LP2)
30	Vref_sk	418	Tunable over 10b (bit_Vref_sk)
31	Vref_sk_LP	356	Tunable over 10b (bit_Vref_sk_LP)

If cmd_probe_dc1<2> = 1, the next 16 voltages are accessible:

ADDRESS	PROBE POINT	Nominal value (mV)	Comment
0	Probe_vbg_1V	540	Half Vbg_1V Tunable over 3b (cmd_vbg_1v)
1	Probe_vdd_pad	300	A fourth of vdd_pad
2	Probe_vdd_pa	600	Half of vdd_pa
3	Probe_vddd	600	Half of vddd
4	Ibo_ref_adc	821	
5	Probe_center		
6	Vbm_pa	669	Tunable with 1b (half_vbm)
7	Vbi_pa	241	Tunable over 6b (dacb_vbi)
8	Vbm2_pa	396	
9	Vbm3_pa	204	Tunable with 1b (half_vbm)
10	Vbo_pa	373	Tunable over 6b (dacb_vbo)
11	Vbi_pa_LP	568	Tunable over 6b (dacb_vbi_LP)
12	Vb_ota_toa	323	Tunable over 2b (ibib_ota_toa)
13	VDC_ota_toa	634	
14	VC_ota_toa	999	
15	Vbi_discri_toa	333	

If cmd_probe_dc1<2:0> are not set, the probe_dc1 is in high impedance so that several probes can be tied together on the hexaboard.

If cmd_probe_dc2=1 the calibration dac output is sent to probe_dc2.

2.3.5 Efuse

Serial number over 31 bits can be written inside each chip by burning e-fuses. Actually there are 32 e-fuses inside the chip, but the two last MSB, bit<31> and bit<30>, are hard-wired to 0 and to 1 respectively, furthermore the MSB, bit<31>, cannot be read. So in fact the range of the serial number goes from 0x4000 0000 to 0x7FFF FFFF (1 073 741 823 combinations). The serial number has to be first written into registers R9[7..0], R11[7..0], R13[7..0] and R15[5..0] of Top sub-block before being burned bit per bit following the procedure described below.

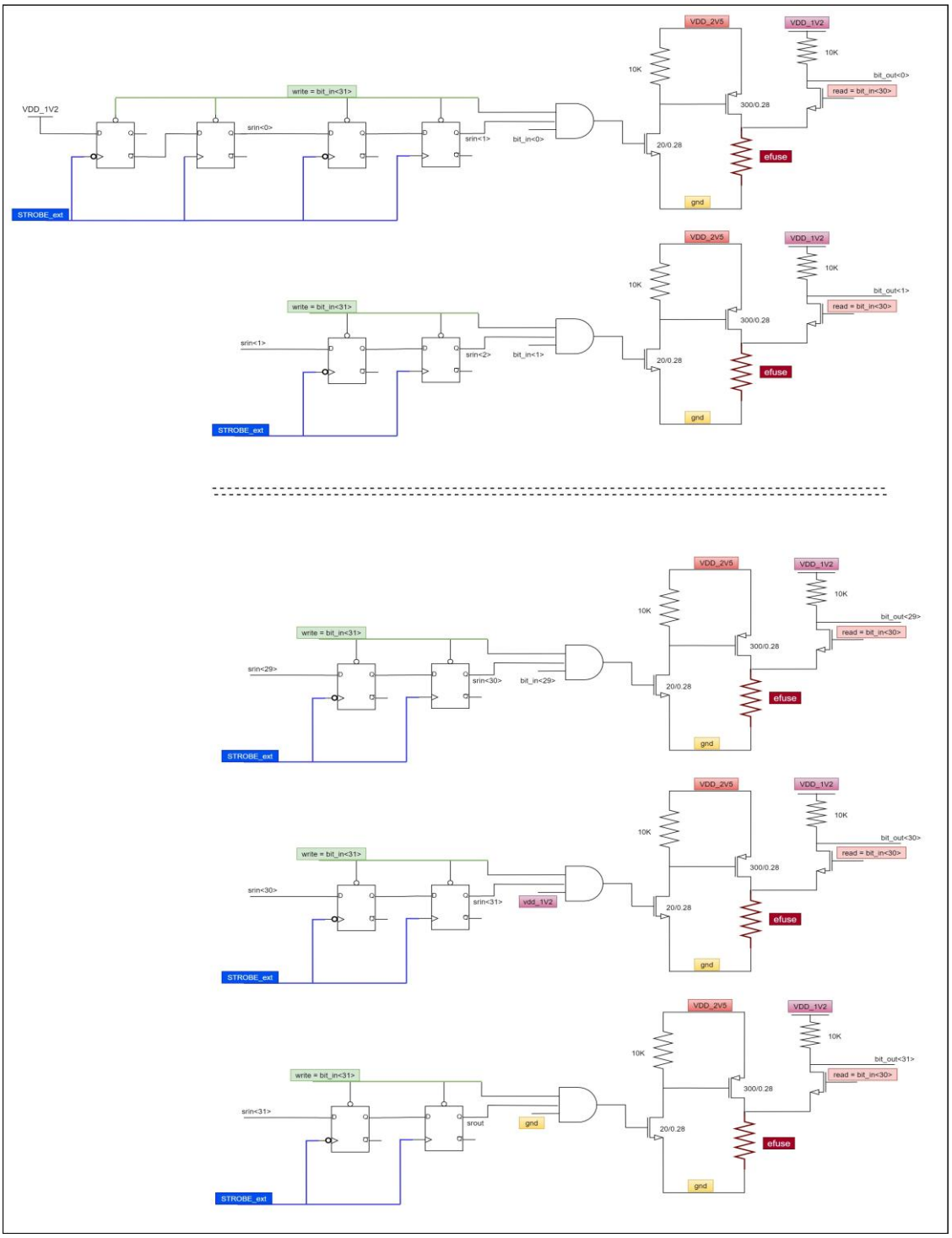


Figure 11 Efuse schematic

Once the serial number is written inside registers R15[5..0], R13[7..0], R11[7..0], R9[7..0] of Top sub-block, R15[7] has to be set to 1 so that the *write* permission is enabled, 2.5 V has to be provided from the *efuse* pin. And then a 33-pulses train has to be provided on the pin *STROBE_ext* which must start and finish to 0 as showed in the chronogram below.

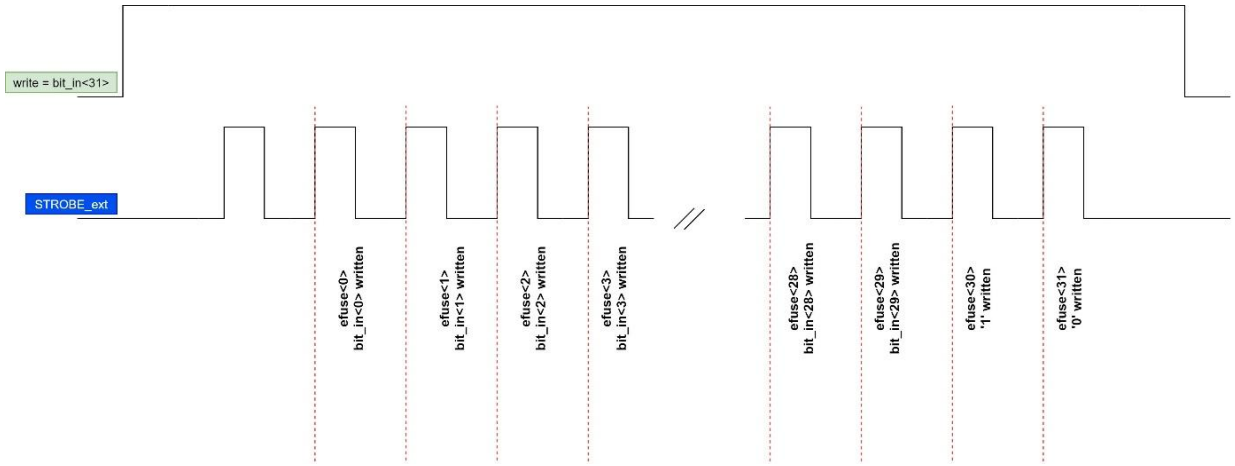


Figure 12 Efuse chronogram

In order to read the serial number back, R15[6] of Top sub-block has to be set to 1 to enable the *read* permission, and then each bit is connected, LSB first, into R8[0..7], R10[0..7], R12[0..7] and R14[0..6]. No matter the value on the *efuse* pin: 0V, 2.5V or floating. The bit R14[7] gives the state of the *srout* net of the figure above, only for debug purpose.

2.3.6 Random Clock Generator [RCG]

A Random Clock Generator has been implemented in order to calibrate the TDCs. This block generates a clock whose frequency is unsynchronized with any clocks inside the chip so that the code distribution can be produced. From this code distribution, the INL and DNL can be extracted offline.

By default, the RCG is disabled and off.

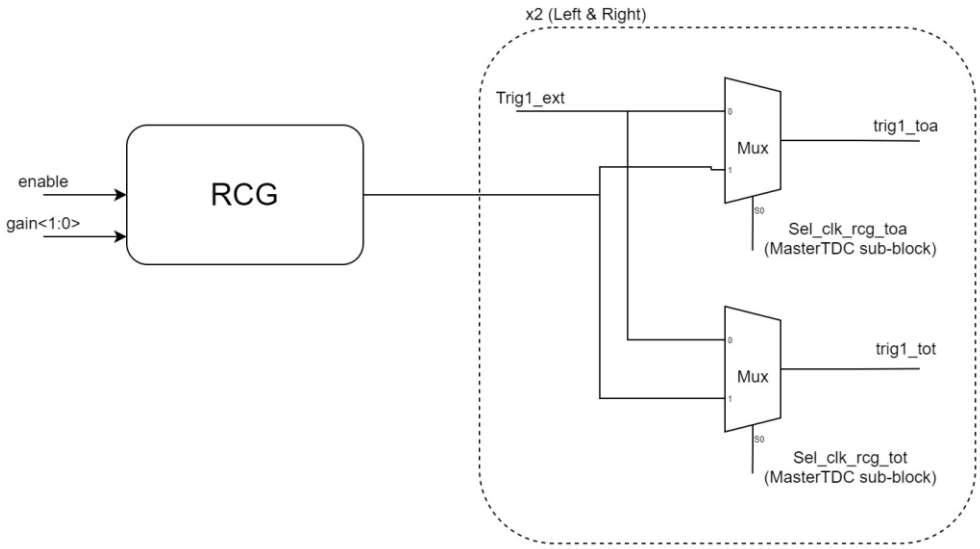


Figure 13 RGC

For triggering a TDC, the user has to select *sel_trigger_toa/tot* (channel_wise) of the channel, enable the RCG (Top sub-block) and select RCG as external trigger with *Sel_clk_rcg_toa/tot* parameter (MasterTDC sub-block).

3 Digital processing

3.1 Overview

CALOROC integrates 36 channels with charge and time measurements. Digitized data are readout through 2 high speed serial links at 1.28 Gbps (each link readout 18 channels respectively).

As shown in the figure below, the ASIC has 3 main digital interfaces:

- 1x I2C interface to change the ASIC parameters
- 1x Fast command interface which generate internal commands and clocks
- 2x 1.28 Gbps serial data transmitter (ASIC outputs)

These interfaces are described in the next section.

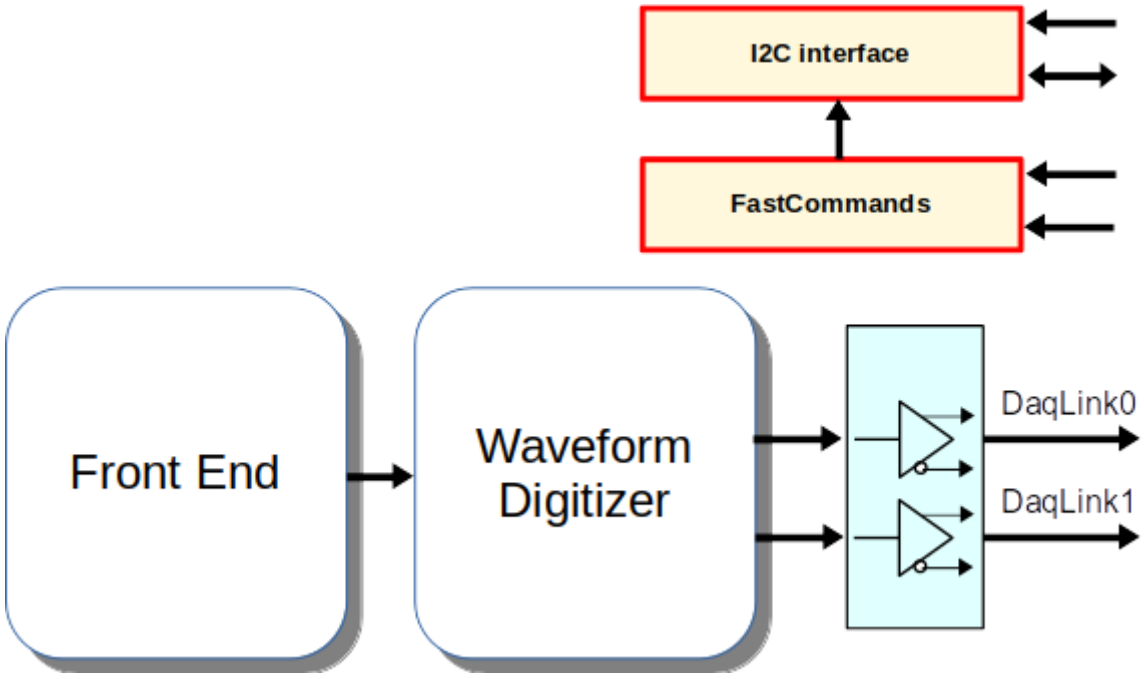


Figure 14

3.2 Control interface blocks

3.2.1 Fast Command

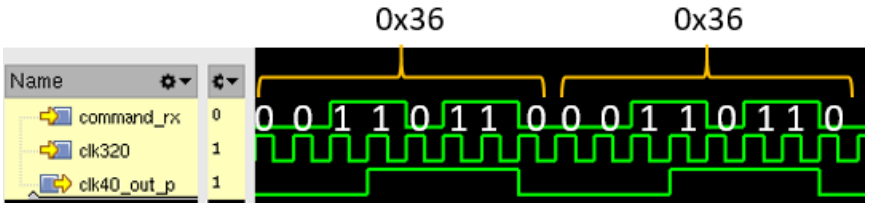
The Fast Command Unit (FCU) is responsible for generating internal fast commands. The table below details each command along with its associated latency. Commands are encoded and transmitted using the MSB first convention. Latency is defined as the delay between the reception of the last bit in the 320 Mbps bitstream and the issuance of the command to configure a specific operating mode of the system. Latency is expressed in clock cycles of the synchronized 40 MHz clock, which is generated internally by the FCU.

The FCU requires 3 clock cycles (at 40 MHz) to decode a command; therefore, the minimum achievable latency is 3 clock cycles.

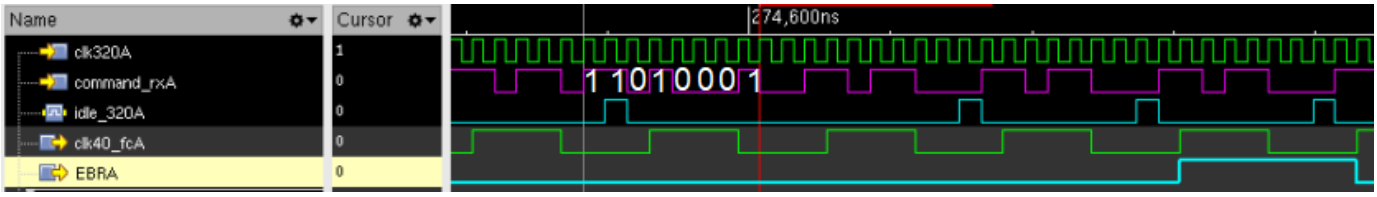
Command	Code	Description	Total latency with FC
CMD_IDLE	00110110 (0x36)	Allow to generate the main internal 40MHz clock. It's derived from the 320MHz and aligned with bit 4 of incoming command.	
CMD_L1A	01001011 (0x4B)	Acts as an internal triggering for all the channels: forcing timestamp and all ADCs-TOAs to be stored in the internal memories.	3+6 clock cycle at 40 MHz.
CMD_CALPULSEINT	00101101 (0x2D)	Send a Calibration Pulse of 32 x 40 MHz clock period.	3+1 clock cycle at 40 MHz.
CMD_CALPULSEEXT	01111000 (0x78)	Send a Calibration Pulse of 4 x 40 MHz clock period.	3+1 clock cycle at 40 MHz.
CMD_CHIPSYNC	11010010 (0xD2)	Reset all Digital FSM. Does not reset analog part (PLL) and the FCU.	3+1 clock cycles at 40 MHz to reset Startup FSM to idle state.
CMD_EBR	11010001 (0xD1)	Reset the buffer pointer of FIFO and the event counter.	3+1 clock cycle at 40 MHz.
CMD_BCR	00011101 (0x1D)	Reset the Timestamp to its default value (parameter TimeOffset).	3+1 clock cycle at 40 MHz.
CMD_PING	10011001 (0x99)	Replace an IDLE by an IDLE_ping.	Fixed
CMD_LINKRESETROCD	10011010 (0x9A)	Link Sync: Upon completion of the current packet, 400 IDLEs are forced on the links.	The link is reset immediately if the readout is finished.
ROC_SERIALIZER_RESET	10011100 (0x9C)	Link Reset: Reset all Serializers, it does a bit alignment for all of them. The chip loses links during 14 clock cycles at 40MHz.	1 clock cycle at 40 MHz to reset and 14 clock cycles at 40 MHz to recover links.
SlowControl_bit0 SC0	01011010	Send 0 for the I2C over fast command module.	NA
SlowControl_bit1 SC1	01011100	Send 1 for the I2C over fast command module.	NA
SlowControl_valid_rst SCV	10001011	Send a validation for the I2C over fast command (reset if 2 consecutive).	NA

- Internal clock phase relationship with external clock:

The fast command bit stream (*command_rx* below), clocked at 320 MHz, defines the phase relationship between internal 40 MHz lock and the 320 MHz clock as shown below.

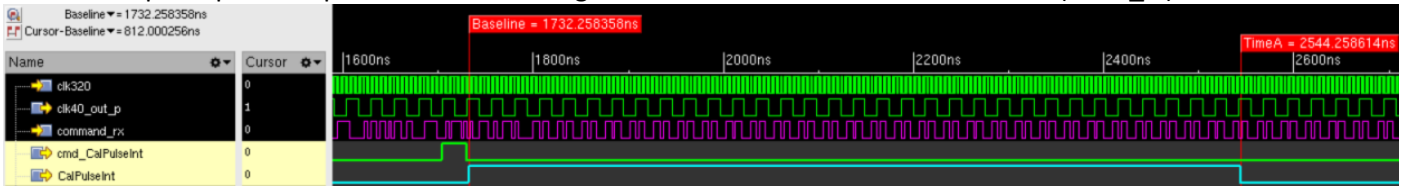


- Example of an internal fast commands decoding (CMD_EBR):

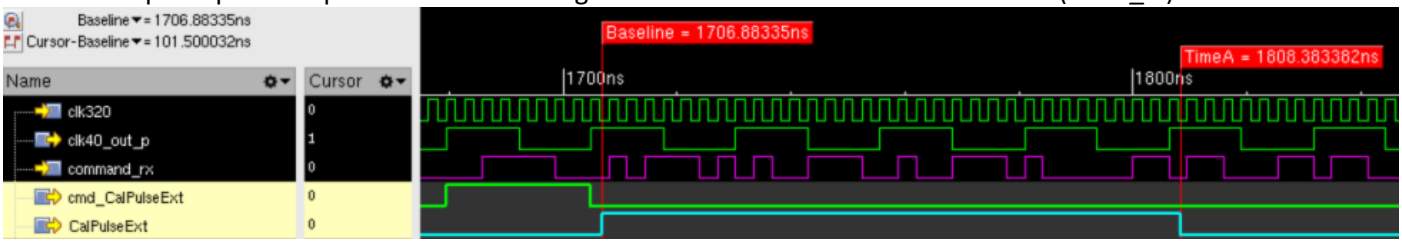


- Detail of the Calibration pulses fast commands:

  - CMD_CALPULSEINT: send a STROBE pulse of 800 ns to the internal calibration DAC. The STROBE pulse phase depends on the main digital clock from Fast Commands module (clk40_fc).



  - CMD_CALPULSEEXT: send a STROBE pulse of 100 ns to the external pin SiPM_calibration. The STROBE pulse phase depends on the main digital clock from Fast Commands module (clk40_fc).



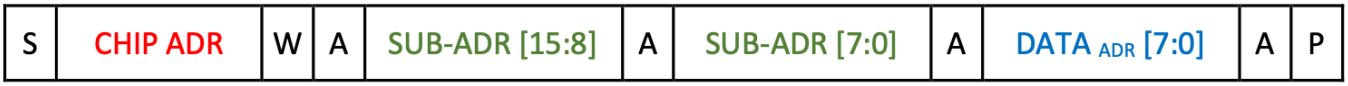
The fast command CMD_L1A which acts as an internal triggering for all the channels, 1 L1A implies writing N_value charge samples in memory.

3.2.2 ASIC parameter access with I2C

I2C protocol is used to access ASIC parameters: SDA and SCL pins are dedicated to this. A standard I2C protocol is used and the default frame to access parameters are given below. This block was originally designed by CERN and modified by OMEGA to bypass wishbone clock (internal stuff not seen outside).

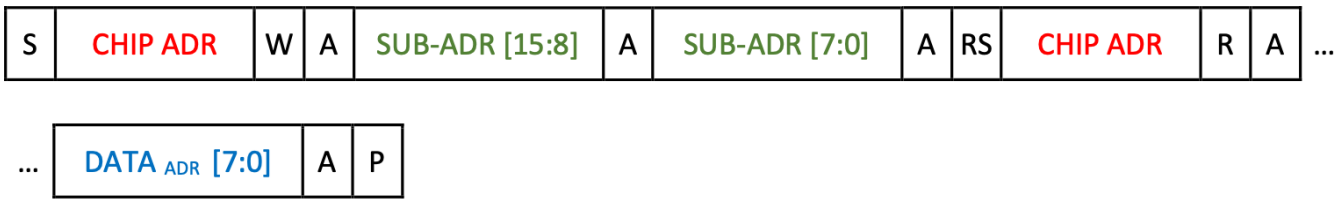
A parameter can be accessed through its sub-block (cluster) number and its register number. The mapping is given in the section “ASIC parameters”.

The default frame to write a single parameter is given below:



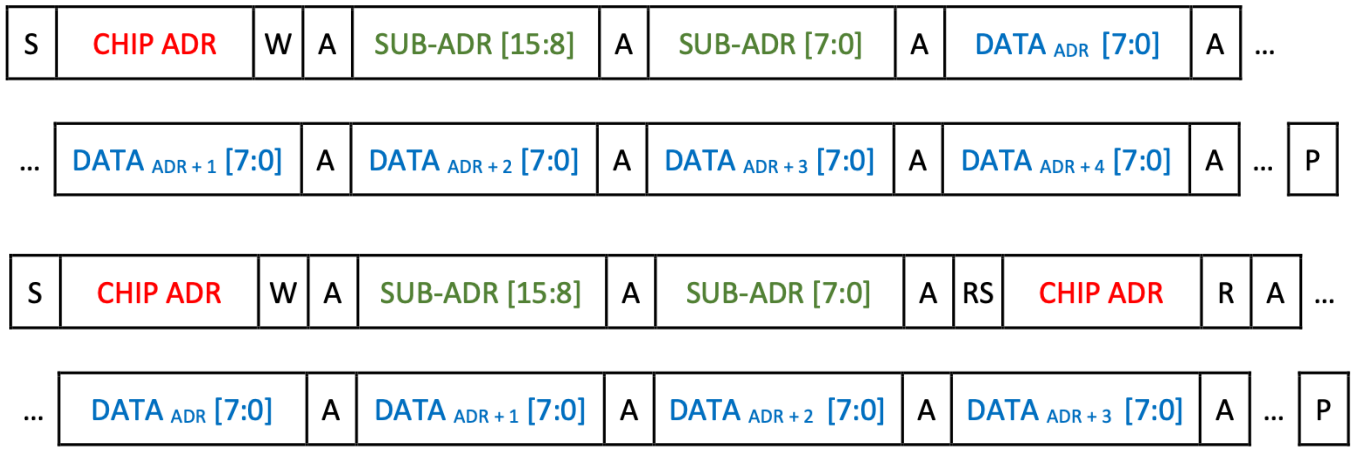
S / P: START and STOP condition of a standard I2C protocol
CHIP ADR: 7-bit chip address (MSB must be set to 1)
SUB-ADR [15:8]: sub-block (cluster) number (from 0 to 255)
SUB-ADR [7:8]: register number (from 0 to 31) – not used bits must be set to 0
DATA[7:0]: data to be written in the parameter
A: Acknowledge bit

The frame for a read operation is:



RS: RESTART condition (can be a STOP/START condition)
A: Acknowledge bit (except for the last one that need to be inverted)

The multibyte write and read can be handled with the following write (top) and read (bottom frames). This operation is only allowed inside the same sub-block (cluster).



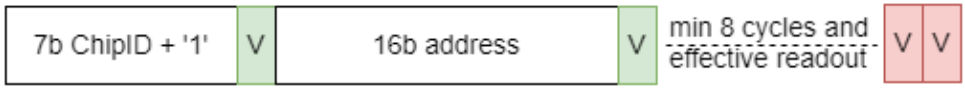
A broadcast address (hardcoded to 100 0000) is also available to write multiple chips at the same time.

3.2.3 ASIC parameter access with fast commands

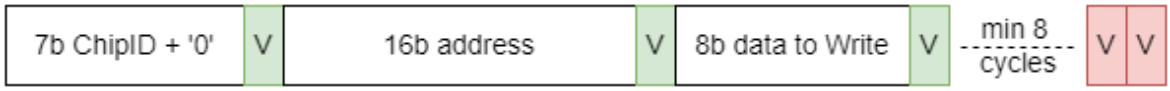
As an alternative, parameters can be accessed with 3 dedicated fast commands (SC0, SC1 and SCV). It mimics simple or multibyte read and write actions (like I2C ones).

SC0 and SC1 are used to send respectively a 0 and 1. SCV is used to validate a transaction.

If there is a readout in progress I2C read transaction is not possible. A read transaction can be done by making the following frame. The effective read of the parameter is done through the first fast serial link.

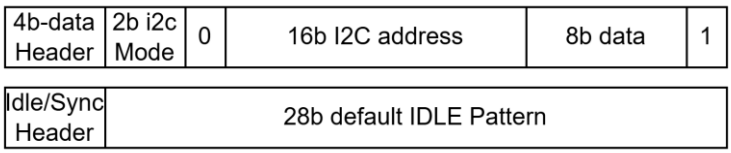


A write transaction can be done by making the following frame.



Sending 2 consecutive SCV cancels the current transaction. When this method is used, the I2C port must remain in idle mode.

The read operation is done through the first high-speed data link (lowest number). This I2S readout has a lower priority than an event data readout and will be postponed until the ASIC has no event to readout



3.2.4 High speed serial links

The output differential links (aka elinks) are composed of a serializer and a driver compatible with the LpGBT protocol (CLPS). The serializer converts parallel 32 bits words at 40 MHz to a serial train of bits send out at 1.28 Gbps. In the table below, the electrical specifications of the driver are given.

The serial output links send the data in MSB first format.

Specification description	Value
Vcm (common voltage)	0,6 V
Vdiff (differential voltage)	100 to 200 mV
Pre-emphasis current	0,5 to 4 mA
Termination load	100 Ω

The termination load resistor must be placed outside the chip.

3.3 Operating mode

3.3.1 Resets

There are three ways to reset the chip.

- The hard reset pin hard_resetb (CMOS input, active low) comes from an external pin and resets everything (analog + digital).
- The soft reset pin soft-rstb (CMOS input, active low) is same as hard_resetb but leaves the SC parameters programmed.
- The fast command ChipSync is equivalent to an external soft reset (but only affects the digital part), it does not reset the Fast Command unit or analog part (like PLL). A BCR is needed after a ChipSync to resynchronize the time tagging counters with the whole system.

A Power-On-Reset has been implemented which performs a hard reset (hard_resetb) while the chip is powered on. The Power-On-Reset is a very critical block of the chip. It has been simulated with the special gate-leakage models and furthermore with additional 1 uA leakage current on the sensitive transistors. Moreover, its output is accessible on a spare pad (Rstb_I2C).

Some fast commands allow to reset specific parts of the chip, as described below:

- Bunch Crossing counter reset (BCR): issued via the fast command receiver, this command resets the internal timestamp counter to its default value (as defined by the corresponding parameter)
- Event Buffer Reset (EBR): issued via the fast command receiver, this command resets only the pointers of the Event Buffer (internal FIFO memory). After an EBR the ASIC can accept a new manual trigger from a fast command (L1A) on the next clock cycle. If an event readout is already in progress, it will complete normally.

3.3.2 Startup procedure

CALOROC is able to read 1 SiPM per channel. The chip has 36 input-channels which are mapped to two 1.28 Gbps serial links (18 channels per link).

On start-up, the user resets the chip, this will force all chip parameters to default values. Once the reset is done, the user can configure the chip by using slow control parameters. The acquisition starts when the parameter named “R*un*” is equal to 1.

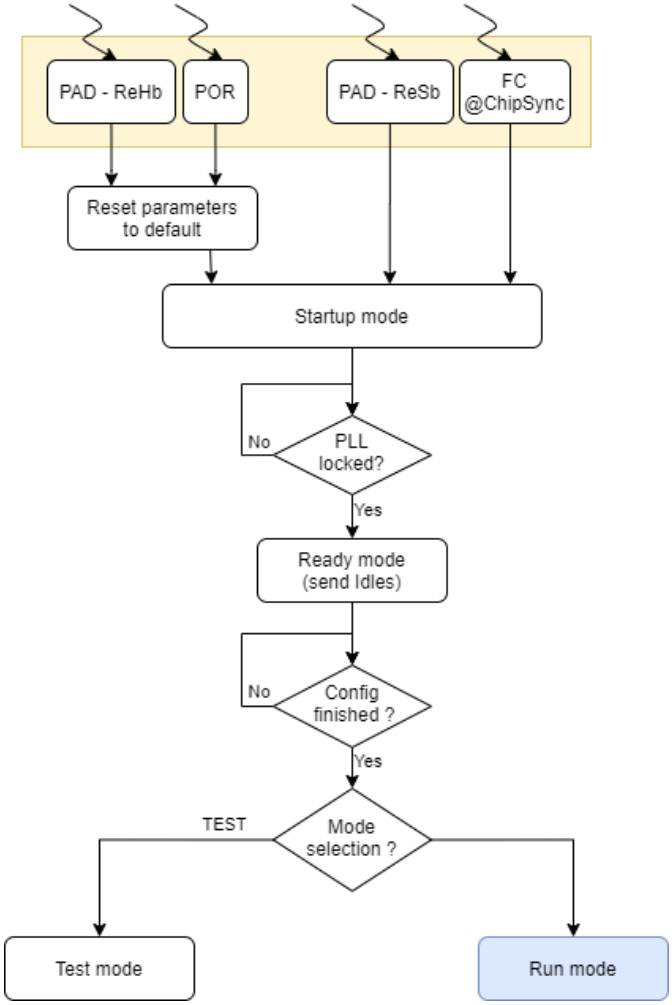


Figure 15

When the chip is powering on, the Power-On-Reset (POR) block applies a reset (like the hard_resetb).
These describe the state machine or sequence the ASIC goes through after power-up and during operation. They are dynamic, meaning the ASIC transitions between them based on internal conditions or user actions:

1. **Startup Mode**

- Occurs immediately after reset or power-on.
- ASIC uses default configuration parameters.
- It stays here until the PLL locks.

2. **Ready Mode**

- The PLL is now locked, and the ASIC is ready for configuration.
- The user have to load operating parameters via I2C.
- Once configuration is done, a signal “Run” have to be set.
- ASIC starts sending a forced 32-bit IDLE pattern on the trigger links.

3. **Run Mode**

- The ASIC is fully configured and begins data acquisition.
- This is the default acquisition mode during system operation.
-

4. **Test Mode**

- Used to test the internal DRAM memory.
- Not part of normal acquisition, meant for diagnostics.

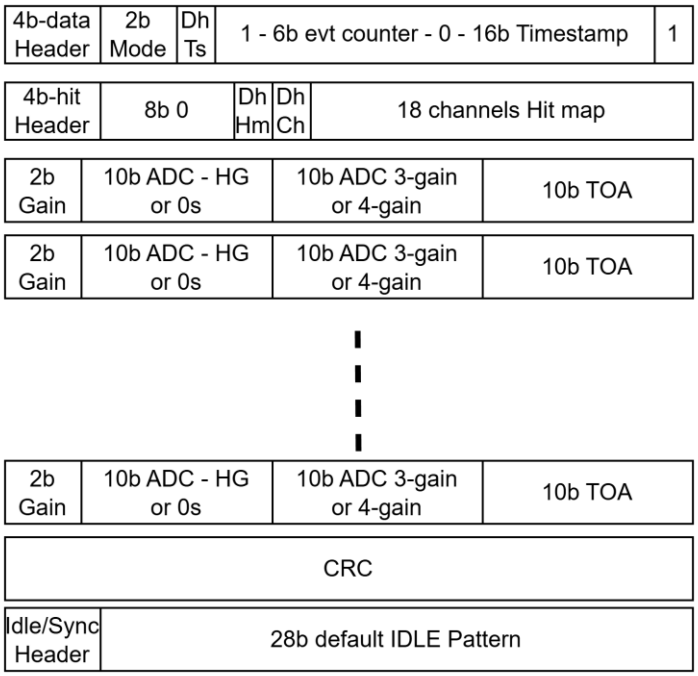
Low Power mode disables the chip's analog section, including the preamplifier, shaper, and discriminators. However, since the PLL remains active, the ADC, TDC, and AlignBuffer also remain operational.

3.3.3 RUN mode

3.3.3.1 Readout

During normal operation, when all channels are active, the readout consists of 22 words of 32 bits each.
When operating in zero-suppressed mode, the readout may contain fewer words than the nominal 3-20 word range. (see figure and explanations below):

- **Words 1-2:** Contain the timestamp and hit channel information
- **Words 3-20:** Depending on whether Zero Supress is enabled, each word includes:
  - 2 bits indicating the ADC gain used
  - The remaining bits containing the Charge and Time of Arrival (TOA**)** data for each channel. Based on the selected AsicMode, Charge and TOA are measured for each PMT channel across the high, medium, and low gain paths.
- **Word 21:** Contains the CRC for error checking
- **Word 22:** Idle word



2b Gain (2 bits): Tells which is the gain being read by the ADC.

- 00 → High gain
- 01 → Medium gain 1
- 10 → Medium gain 2
- 11 → Low gain

Data Header (4 bits):
Identify the type of frame with dedicated header: 0x5

Hit Header (4 bits):
Identify the type of frame with dedicated header: 0xF

Mode (2 bits):
Indicates the ASIC mode:

- 00 → Same as H2GCROC
- 01 → 2 ADCs mode with gain switching
- 10 → 1 ADC mode with gain switching – not working in CALOROC1. It's still possible to mimic this mode by using the 2 ADCs mode and deactivating the unused ADC. In this case, the HG manual mode should be switched off. The following parameters must be set to 1: channel wise mask_ADC2 and manual_modeb .
- 11 → Only appear in data readout to tag an I2C read through fast commands

Dh (1 bit):
Double error flag — active high if a double error is detected during internal Hamming decoding.

- Dh Ts: double error for Hamming decoding Timestamp bits
- Dh Ch: double error for Hamming decoding 18 CH data words
- Dh Hm: double error for Hamming decoding HitMap bits

Timestamp (16 bits):
Provides the timestamp corresponding to the detected event.

Event counter (6 bits):
Provides the sample event. First one is tagged 1.

HitMap (18 bits):
Regardless of whether Zero suppress mode is enabled or not, this bus indicates which channels were hit during the event.
Data words for each channel are transmitted in ascending channel index order.

- Link 0 -> DATA(i) if HitMap(i) = 1, for i = 0 to 17
- Link 1 -> DATA(i) if HitMap(i) = 1, for i = 18 to 35

However, when the L1A fast command is enabled, all bits of HitMap are set to '1'.

3.3.3.2 Data to DRAM

As seen previously ASIC measurement data are formatted into a 32-bit word for DRAM storage (Gain, ADC, TOA). Two slow control signals allow selecting and defining a specific test pattern instead of the charge and TOA data. The details are provided below:

- **SC_testRAM_cmd:** 1 bit signal forces the module to bypassing normal data formatting.
- **SC_testRAM_data:** 32-bit test pattern word for DRAM storage (test or initialization).

Register	Value	Description
SC_testRAM_cmd	0	Default value. Enables normal readout operation in RUN mode as described in the previous section.
SC_testRAM_cmd	1	Enables test mode for SC_testRAM.

When **SC_testRAM_cmd = 1** :

Word	Content
Word 1	The 24-bit word, originally formatted as {'1', 6b event counter, 16b Timestamp, '0'}, is remapped to {SC_testRAM_data[2:0], SC_testRAM_data[20:0]} in test mode.
Words 3–20	Fixed 32 bits pattern: 0x55555555

If the DRAM is full while events are still being detected, partial events may be written to the DRAM.

3.3.3.3 CRC

CRC (32 bits):

Cyclic Redundancy Check for data integrity verification.

CRC function used is polynomial: 0x04C11DB7

x^32 + x^26 + x^23 + x^22 + x^16 + x^12 + x^11 + x^10 + x^8 + x^7 + x^5 + x^4 + x^2 + x^1 + 1

data width: 32

CRC includes all data from word 1 to word 20

3.3.4 TEST mode

3.3.4.1 DRAM Self-Test via Memory BIST Module

The Test Mode is used to verify the integrity of the internal DRAM arrays using the integrated Memory BIST (Built-In Self-Test) module.
In the CALOROC ASIC, there are six DRAM instances, organized into two groups:

- Group 0: DRAM0, DRAM1, DRAM2
- Group 1: DRAM3, DRAM4, DRAM5

Each group is managed by a dedicated Memory BIST module.

3.3.4.2 Memory BIST Functionality

Each Memory BIST module performs automated self-tests by:

- Writing a predefined set of test patterns to each DRAM instance.
- Applying a retention time (delay before readback).
- Reading back the stored data.
- Comparing the results to detect memory faults.

This module is clock-gated to minimize power consumption and operates exclusively in Test Mode.

3.3.4.3 Activation Conditions

Memory BIST tests are triggered automatically during startup if the following slow control parameters are set after reset:

- `TestMode = 1`
- `Run = 0`

In this case, the startup state machine enters DRAM test mode and executes the BIST sequence.

3.3.4.4 Interface Signals (Slow Control Registers)

Signal	Direction	Width	Description
`Ret_time`	Input	12	Retention time (wait cycles before readback)
`BistRamErrors`	Output	24	Error flags (4 bits per DRAM)
`Bist_dones`	Output	2	BIST completion flags (1 per DRAM group)

3.3.4.5 Monitoring Test Progress

The `Bist_dones` signal provides status on BIST execution:

Signal Bit	DRAM Group	Description
`Bist_dones[1]`	DRAM3, DRAM4, DRAM5	Indicates BIST completion for upper group
`Bist_dones[0]`	DRAM0, DRAM1, DRAM2	Indicates BIST completion for lower group

- While `Bist_dones = 0`, the BIST is in progress.
- When `Bist_dones = 1`, the test has completed for that group.

Each bit is set to '1' once the test finishes for the associated group.

3.3.4.6 Analyzing Test Results

The `BistRamErrors` signal provides a 24-bit error flag, organized as 4 bits per DRAM instance.

A bit set to '1' indicates that a fault has been detected for a specific test pattern.

Bits are grouped as follows:

- Bits [3:0] → DRAM0
- Bits [7:4] → DRAM1
- Bits [11:8] → DRAM2
- Bits [15:12] → DRAM3
- Bits [19:16] → DRAM4
- Bits [23:20] → DRAM5

Example:

If `BistRamErrors = 24'b0000_0100_0000_0000_1000_0001`, then:

- Bit 0 = '1' → Fault in DRAM0
- Bit 7 = '1' → Fault in DRAM1
- Bit 18 = '1' → Fault in DRAM4

In this example 3 faults detected on DRAM0, DRAM1, and DRAM4.

3.3.5 *Asic modes (functional configuration)*

These are user-selectable configurations that determine how the ASIC internally processes analog signals. Set via a slow control register called `AsicMode`. They are static until changed by the user:

Two ADCs Mode

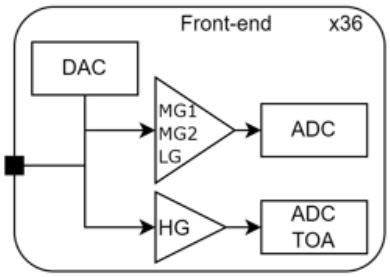
- The ASIC uses two separate ADCs to digitize input channels.
- Likely for higher throughput or parallel processing.

Single ADC with 4 Gain Stages Mode

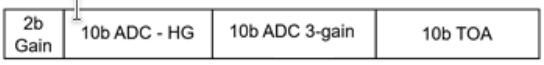
- Uses one ADC, but with four selectable gain settings.
- Optimized for dynamic range adaptation and good when signals vary in amplitude.

- AsicMode = "01" → Two ADCs mode
- AsicMode = "10" → Single ADC mode – not working in CALOROC1 (possible to mimic with 2 ADC mode)

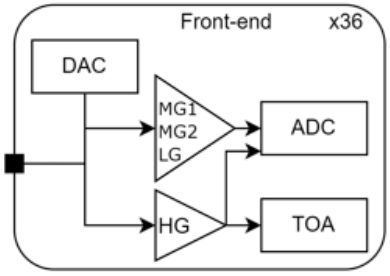
CALOROCB (2 ADCs)



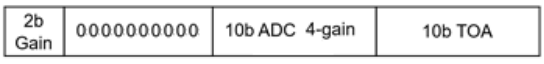
Corresponding readout



CALOROCB (1 ADCs)



Corresponding readout



3.3.6 *Idle word composition*

The default word sent over the high-speed links is defined as the idle word. It's a 32-bit word composed by 2 fields: a 4-bit header (defined in the next sections) and a 28-bit pattern (defined as a slow-control parameter with its LSB forced to 0).

Idle/Sync Header	28b default IDLE parameter

This idle word is automatically modified to carry the almost-full (AF) information. The bit 1 is set to 1 when the flag is internally active as it's shown below.

Idle/Sync Header	26b default IDLE parameter (MSBs)	AF	0

In response to a ping fast command (CMD_PING), the chip will change this idle word to a ping-idle word. With a fixed latency the chip will output the following word containing the timestamp. The LSB is set to 1 to identify this ping-idle word.

Idle/Sync Header	4b default idle	16b timestamp	6b default idle	AF	1

3.3.7 Number of samples (N_value)

N_value is a 3 bits parameter. It specifies the desired number of charge samples per event. Its value must be greater than 0 (from 1 to 7). The next figure illustrates the number of samples in normal mode. Red dots represent the acquired and transmitted samples, while the blue dot indicates the Time of Arrival (TOA).

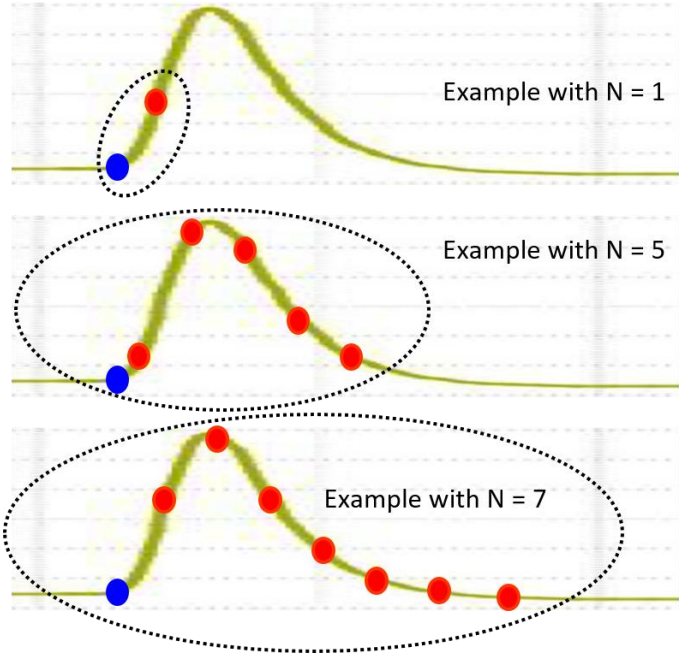


Figure 16

3.3.8 Synchronization header (Time_trigger)

When the internal counter (16b Timestamps) of the chip crosses a defined value (parameter Time_trigger), the 4-bit header of the next readout 32-bit word will change according to table below.

4-bit header value	Signification
0x5	Data: first readout word (timestamp or I2C read)
0xF	Data: readout word is the event Hit Map
0xA	IDLE: default IDLE word
0x9	IDLE: idle word with synchronization header

3.3.9 Clocks

The only external clock needed to operate the ASIC is the 320MHz. All other needed clocks are generated internally. The state machines are clocked at 40MHz (which is generated by the fast command block). The chip embeds a PLL to provide the internal clocks.

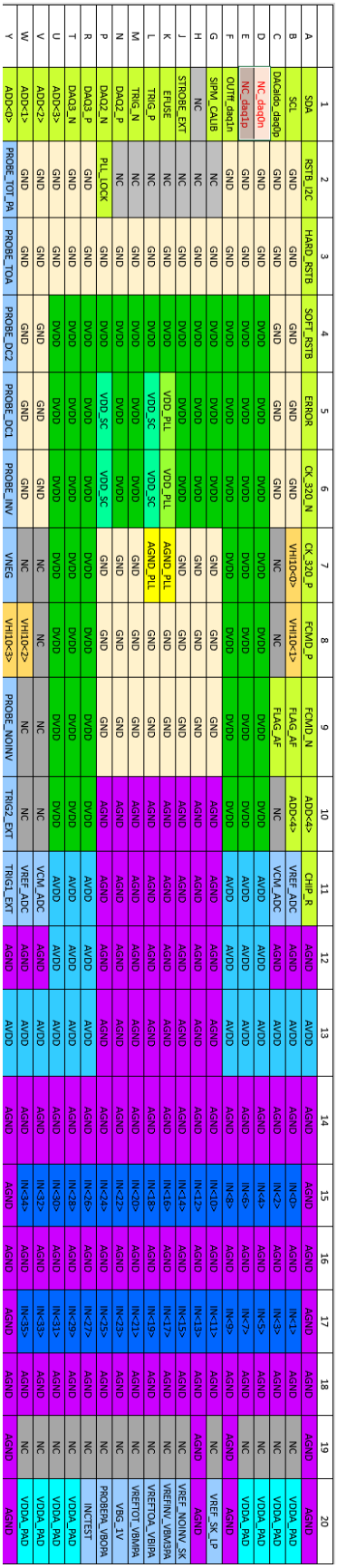
3.3.10 Almost full flag (Flag_AF)

Flag_AF is an output pin of the chip that notifies the user when the internal memory of the chip is almost full. Val_AF (value almost full) is a 6 bits parameter that contains the almost-full threshold (value from 1 to 32). If Val_AF is set to 32, Flag_AF will be active when the memory is full.

This information can be inferred from bit 1 of the idle word sent through the high-speed links.

4 Packaging, I/Os and powering scheme

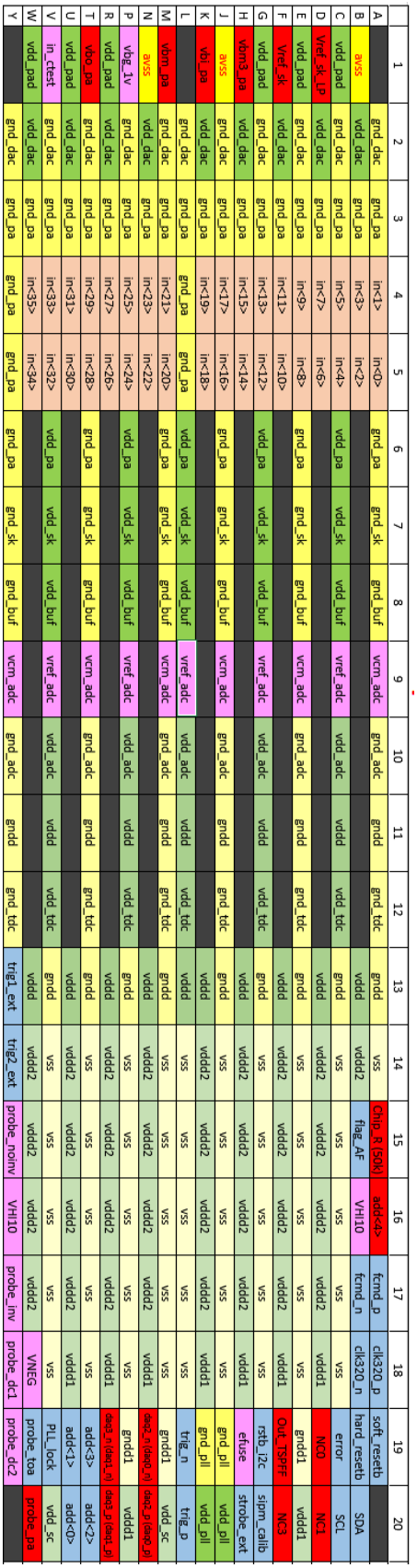
The chip will be packaged in BGA. The BGA ballmap is given below (valid for all the CALOROC versions).



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
A	SDA	RSTB_I2C	HARD_RSTB	SOFT_RSTB	ERROR	CK_320_N	CK_320_P	FCMD_P	FCMD_N	ADD<4>	ChIP_R	AGND	AVDD	AGND	AGND	AGND	AGND	AGND	AGND	AGND
B	SCL	GND	GND	GND	GND	GND	VHI10<0>	VHI10<1>	FLAG_AF	ADD<4>	VREF_ADC	AGND	AVDD	AGND	IN<0>	AGND	IN<1>	AGND	NC	VDDA_PAD
C	DACaldo_daq0p	GND	GND	GND	GND	GND	NC	NC	FLAG_AF	NC	VCM_ADC	AGND	AVDD	AGND	IN<2>	AGND	IN<3>	AGND	NC	VDDA_PAD
D	NC_daq0n	GND	GND	DVDD	DVDD	DVDD	DVDD	DVDD	DVDD	DVDD	AVDD	AVDD	AVDD	AGND	IN<4>	AGND	IN<5>	AGND	NC	VDDA_PAD
E	NC_daq1p	GND	GND	DVDD	DVDD	DVDD	DVDD	DVDD	DVDD	DVDD	AVDD	AVDD	AVDD	AGND	IN<6>	AGND	IN<7>	AGND	NC	VDDA_PAD
F	OUTff_daq1n	GND	GND	DVDD	DVDD	DVDD	DVDD	DVDD	DVDD	DVDD	AVDD	AVDD	AVDD	AGND	IN<8>	AGND	IN<9>	AGND	AGND	AGND
G	SIPM_CALIB	NC	GND	DVDD	DVDD	DVDD	GND	GND	GND	DVDD	AGND	AGND	AGND	AGND	IN<10>	AGND	IN<11>	AGND	NC	VREF_SK_LP
H	NC	NC	GND	DVDD	DVDD	DVDD	GND	GND	GND	DVDD	AGND	AGND	AGND	AGND	IN<12>	AGND	IN<13>	AGND	AGND	AGND
J	STROBE_EXT	NC	GND	DVDD	DVDD	DVDD	GND	GND	GND	DVDD	AGND	AGND	AGND	AGND	IN<14>	AGND	IN<15>	AGND	NC	VREF_MONV_SK
K	EFUSE	NC	GND	DVDD	VDD_PLL	VDD_PLL	AGND_PLL	GND	GND	DVDD	AGND	AGND	AGND	AGND	IN<16>	AGND	IN<17>	AGND	NC	VREFINV_VBM3PA
L	TRIG_P	NC	GND	DVDD	VDD_SC	VDD_SC	AGND_PLL	GND	GND	DVDD	AGND	AGND	AGND	AGND	IN<18>	AGND	IN<19>	AGND	NC	VREFTOA_VBIPA
M	TRIG_N	NC	GND	DVDD	DVDD	DVDD	GND	GND	GND	DVDD	AGND	AGND	AGND	AGND	IN<20>	AGND	IN<21>	AGND	NC	VREFTOT_VBMPA
N	DAQ2_P	NC	GND	DVDD	DVDD	DVDD	GND	GND	GND	DVDD	AGND	AGND	AGND	AGND	IN<22>	AGND	IN<23>	AGND	NC	VBG_1V
P	DAQ2_N	PLL_LOCK	GND	DVDD	VDD_SC	VDD_SC	GND	GND	GND	DVDD	AGND	AGND	AGND	AGND	IN<24>	AGND	IN<25>	AGND	NC	PROBEPA_VBOPA
R	DAQ3_P	GND	GND	DVDD	DVDD	DVDD	DVDD	DVDD	DVDD	DVDD	AVDD	AVDD	AVDD	AGND	IN<26>	AGND	IN<27>	AGND	NC	INCTEST
T	DAQ3_N	GND	GND	DVDD	DVDD	DVDD	DVDD	DVDD	DVDD	DVDD	AVDD	AVDD	AVDD	AGND	IN<28>	AGND	IN<29>	AGND	NC	VDDA_PAD
U	ADD<3>	GND	GND	DVDD	DVDD	DVDD	DVDD	DVDD	DVDD	DVDD	AVDD	AVDD	AVDD	AGND	IN<30>	AGND	IN<31>	AGND	NC	VDDA_PAD
V	ADD<2>	GND	GND	DVDD	DVDD	DVDD	NC	NC	NC	NC	VCM_ADC	AGND	AVDD	AGND	IN<32>	AGND	IN<33>	AGND	NC	VDDA_PAD
W	ADD<1>	GND	GND	GND	GND	GND	NC	NC	NC	NC	VREF_ADC	AGND	AVDD	AGND	IN<34>	AGND	IN<35>	AGND	NC	VDDA_PAD
Y	ADD<0>	PROBE_TOT_PA	PROBE_TOA	PROBE_DC2	PROBE_DC1	PROBE_INV	VNEG	VHI10<3>	PROBE_NONINV	TRIG2_EXT	TRIG1_EXT	AGND	AVDD	AGND	AGND	AGND	AGND	AGND	AGND	AGND

4.1 Pin maps

The figure below shows the pinout of the die bumps of CALOROC (to be used as a flip-chip). The view is from the TOP view of ASIC design tools. The black spots show location where bumps were removed.



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
A		gnd_dac	gnd_pa	inn<1>	inn<0>	gnd_pa	gnd_sk	gnd_buf	vcm_adc	gnd_adc	gndd	gnd_tdc	gndd	vssi	Chnr_R (SDA)	add<4>	fcmd_p	clk320_p	soft_resetb	
B	avss	vdd_dac	gnd_pa	inn<3>	inn<2>								vddd	vddd2	flag_AF	VH10	fcmd_n	clk320_n	hard_resetb	SDA
C	vdd_pad	gnd_dac	gnd_pa	inn<5>	inn<4>	vdd_pa	vdd_sk	vdd_buf	vref_adc	vdd_adc	vddd	vdd_tdc	gndd	vss	vss	vss	vss	vss	error	SCL
D	Vref_sk_LP	vdd_dac	gnd_pa	inn<7>	inn<6>								vddd	vddd2	vddd2	vddd2	vddd2	vddd1	NC0	NC1
E	vdd_pad	gnd_dac	gnd_pa	inn<9>	inn<8>	gnd_pa	gnd_sk	gnd_buf	vcm_adc	gnd_adc	gndd	gnd_tdc	gndd	vss	vss	vss	vss	vss	gndd1	vddd1
F	Vref_sk	vdd_dac	gnd_pa	inn<11>	inn<10>								vddd	vddd2	vddd2	vddd2	vddd2	vddd1	Out_TSPFF	NC3
G	vdd_pad	gnd_dac	gnd_pa	inn<13>	inn<12>	vdd_pa	vdd_sk	vdd_buf	vref_adc	vdd_adc	vddd	vdd_tdc	gndd	vss	vss	vss	vss	vss	rstb_i2c	sipm_calib
H	vbm3_pa	vdd_dac	gnd_pa	inn<15>	inn<14>								vddd	vddd2	vddd2	vddd2	vddd2	vddd1	efuse	strobe_ext
J	avss	gnd_dac	gnd_pa	inn<17>	inn<16>	gnd_pa	gnd_sk	gnd_buf	vcm_adc	gnd_adc	gndd	gnd_tdc	gndd	vss	vss	vss	vss	vss	gnd_pll	vdd_pll
K	vbi_pa	vdd_dac	gnd_pa	inn<19>	inn<18>								vddd	vddd2	vddd2	vddd2	vddd2	vddd1	gnd_pll	vdd_pll
L		gnd_dac	gnd_pa	gnd_pa	gnd_pa	vdd_pa	vdd_sk	vdd_buf	vref_adc	vdd_adc	vddd	vdd_tdc	vddd	vss	vss	vss	vss	vss	trig_n	trig_p
M	vbm_pa	gnd_dac	gnd_pa	inn<21>	inn<20>								vddd	vddd2	vddd2	vddd2	vddd2	vddd1	gnddi1	vdd_sc
N	avss	vdd_dac	gnd_pa	inn<23>	inn<22>	gnd_pa	gnd_sk	gnd_buf	vcm_adc	gnd_adc	gndd	gnd_tdc	vddd	vddd2	vddd2	vddd2	vddd2	vddd1	spa3_ptdap_n	spa2_ptdap_n
P	vbg_1v	gnd_dac	gnd_pa	inn<25>	inn<24>	vdd_pa	vdd_sk	vdd_buf	vref_adc	vdd_adc	vddd	vdd_tdc	gndd	vss	vss	vss	vss	vss	gnddi1	vddd1
R	vdd_pad	vdd_dac	gnd_pa	inn<27>	inn<26>								vddd	vddd2	vddd2	vddd2	vddd2	vddd1	spa3_ptdap_p	spa2_ptdap_p
T	vbo_pa	gnd_dac	gnd_pa	inn<29>	inn<28>	gnd_pa	gnd_sk	gnd_buf	vcm_adc	gnd_adc	gndd	gnd_tdc	vddd	vss	vss	vss	vss	vss	add<3>	add<2>
U	vdd_pad	vdd_dac	gnd_pa	inn<31>	inn<30>								vddd	vddd2	vddd2	vddd2	vddd2	vddd1	add<1>	add<0>
V	in_ctest	gnd_dac	gnd_pa	inn<33>	inn<32>	vdd_pa	vdd_sk	vdd_buf	vref_adc	vdd_adc	vddd	vdd_tdc	gndd	vss	vss	vss	vss	vss	PLL_lock	vdd_sc
W	vdd_pad	vdd_dac	gnd_pa	inn<35>	inn<34>								vddd	vddd2	vddd2	vddd2	vddd2	vddd1	probe_toa	probe_pa
Y		gnd_dac	gnd_pa	gnd_pa	gnd_pa	gnd_pa	gnd_sk	gnd_buf	vcm_adc	gnd_adc	gndd	gnd_tdc	trig1_ext	trig2_ext	probe_noinv	VH10	probe_inv	probe_dc1	probe_dc2	

4.2 Pin list

	Pin Name	Pin type		Comments	Default value
POWERS/GROUNDS	VDDD	POWER		FIFO, ADC power supply	1.2V
	VDD_PAD	POWER		PAD power supply	1.2V
	VDD_TDC	POWER		TDC power supply	1.2V
	VDD_ADC	POWER		ADC power supply	1.2V
	VDD_BUF	POWER		BUFFER power supply	1.2V
	VDD_SK	POWER		Shaper power supply	1.2V
	VDD_PA	POWER		Preamplifier power supply	1.2V
	VDD_DAC	POWER		DAC power supply	1.2V
	VDD_PLL	POWER		PLL power supply	1,2V
	VDD_SC	POWER		Slow Control power supply	1,2V
	VHI10	POWER		DRAM positive power supply	1V
	VDD2	POWER		Digital power supply	1.2V
	VDD1	POWER		Digital power supply	1.2V
	VSS	GROUND		digital ground	0V
	GNDD	GROUND		FIFO, ADC ground	0V
	GND_TDC	GROUND		TDC ground	0V
	GND_ADC	GROUND		ADC ground	0V
	GND_BUFF	GROUND		BUFF ground	0V
	GND_SK	GROUND		Shaper ground	0V
	GND_PA	GROUND		Preamplifier ground	0V
	GND_DAC	GROUND		DAC ground	0V
	GND_PLL	GROUND		PLL ground	0V
	GNDD1	GROUND		Digital ground	0V
Common pins	Error	Open drain		Open collector; external resistor must be added. OR of all the slow-control cells' error signals. Errorb	
	SCL	CMOS	input	I2C clock	
	SDA	Open drain	bidir	I2C data	
	Daq0_p	CLPS	Output	Data 0 link	
	Daq0_n	CLPS	Output		
	Daq1_p	CLPS	Output	Data 1 link	
	Daq1_n	CLPS	Output		
				TSPFF test signal	
	Out_TSPFF	CLPS	Output		
				NC	
	Trig_p	CLPS	Output	Analog Probe: Trigger_OR or PLL Probe (Clock) If SC EN_Probe_PLL=0 → Trigger_OR EN_trig=1	
	Trig_n	CLPS	Output		
	SiPM_Calib	CMOS	Output	Calibration signal for SiPM	
	Rstb_ I2C	CMOS	Input/Out	Output: Probe Power-On reset Input: Bypass Power-On reset	0V

			Chip reset ACTIVE LOW	
Hard_resetb	CMOS	Input	Global reset (PLL, I2C, FSM,SLOW CONTROL); ACTIVE LOW	0V
Soft-rstb	CMOS	Input	Global reset (except SLOW CONTROL); ACTIVE LOW	0V
Clk320_p	CLPS	Input	Fast command 320 MHz clock	
Clk320_n	CLPS	Input		
Fcmd_p	CLPS	Input	Fast command data	
Fcmd_n	CLPS	Input		
Chip_R		Input	50kohm resistor to gnd	
Strobe_ext	CMOS	Input	External calibration signal selectable by slow control	
efuse	ANALOG		Efuse connection	
ADD<4:0>	I2C chip adress		I2C chip address	
PLL_lock	CMOS	Output		
VNEG	Power		DRAM negative power supply	-100mV
Flag_AF	Open drain	Output	Indicates internal almost full memory (active low)	

Analog and mixed part (ch. 0 to 35)

In<0:35>	ANALOG	Input	Analog inputs	
Trig1_ext	CMOS	Input	External trigger 1	
Trig2_ext	CMOS	Input	External trigger 2	
Vcm_ADC	ANALOG	Input/Output	Common mode reference voltage of the ADC.	150mV
Vref_ADC	ANALOG	Input/Output	Reference voltage of the ADC.	1.07V
Vref_SK	ANALOG	Input/Output	Reference voltage of the Shaper amplifier, value close to the preamp input voltage. By default must be kept free. Can be monitored/checked.	418mV
Vref_SK_LP	ANALOG	Input/Output	Reference voltage of the non-inverted shaper. Value given by an internal 10b_DAC. Can be checked/monitored externally. By default must be kept free.	356mV
Vbm3_pa	ANALOG	Input/Output	Preamplifier bias, a 100nF external capacitance can be connected to decrease the crosstalk	204mv
Vbi_pa	ANALOG	Input/Output	Preamplifier bias, a 100nF capacitance must be connected to decrease the noise and crosstalk	241mv
Vbm_pa	ANALOG	Input/Output	Preamplifier bias, a 100nF external capacitance can be connected to decrease the crosstalk	669mv
VBG_1V	ANALOG	Input/Output	BANDGAP voltage. Value can be tuned by slow-control. Default value = 1V	1V

	Vbo_pa	ANALOG	Input/Output	Preamplifier bias, a 100nF external capacitance can be connected to decrease the crosstalk	373mv
	In_ctest	ANALOG	Input	By default, external pin test, but possibility to connect the calibration DAC output to this pin by slow-control.	
	Probe_noinv	CMOS	Output	Buffer Noinv output probe. Channel-wise	
	Probe_inv	CMOS	Output	Buffer inv output probe. Channel-wise	
	Probe_Toa	CMOS	Output	TOA discri output probe. Channel-wise	
	Probe_pa	CMOS	Output	preamplifier analog probe. Channel-wise	
	Probe_DC1	ANALOG	Input/Output	DC analog probe (bias, ref.)	
	Probe_DC2	ANALOG	Input/Output	DC analog probe (bias, ref.)	

5 ASIC parameters

The I2C slave has 4 internal registers (aka "Slim internal registers"): 2 read/write registers and 2 read-only status registers. These 4 registers are in the cluster 255 (register 0 to 3) and are described below.

Cluster 255 – register 0 - read/write			
Bit	**Name**	**Default**	**Description**
0	NA	0	
1	NA	0	
2	NA	0	
3	NA	0	
4	NA	0	
5	NA	0	
6	NA	0	
7	NA	0	

Cluster 255 – register 1 - read/write			
Bit	**Name**	**Default**	**Description**
0	AutoReload	1	"1" = automatically correct SEU
1	NA	1	
2	NA	1	
3	NA	1	
4	NA	1	
5	NA	1	
6	NA	1	
7	NA	1	

Cluster 255 – register 2 – read only (status)			
Bit	**Name**	**Default**	**Description**
0	ErrorOut_global		SEU error in slow control registers
1	ParityOut_global		Parity error in slow control registers
2	LOCK_PLL		
3	NA		
4	NA		
5	NA		
6	NA		
7	NA		

Cluster 255 – register 3 – read only (status)			
Bit	**Name**	**Default**	**Description**
0	NA		
1	NA		
2	NA		
3	NA		
4	NA		
5	NA		
6	NA		
7	NA		

5.1 I2C Addressing

The chip is divided in sub-blocks (aka clusters) containing a maximum of 32 registers. The 16-bit I2C address was divided in 2 parts:

- The 8 MSB of this address to encode the sub-block (cluster) number (from 0 to 255)
- The 8 LSB to encore the register number (from 0 to 31) of this sub-block (cluster)

The unused bits of the register number must be set to 0. The table below gives the address, the name and a short description of all the sub-blocks (clusters).

Sub-block name	Sub-block address	Description
Channel_0	0	Registers described in "**channel wise**" I2C parameters table
Channel_1	1	
Channel_2	2	
Channel_3	3	
Channel_4	4	
Channel_5	5	
Channel_6	6	
Channel_7	7	
Channel_8	8	
Channel_9	9	
Channel_10	10	
Channel_11	11	
Channel_12	12	
Channel_13	13	
Channel_14	14	
Channel_15	15	
Channel_16	16	
Channel_17	17	
Channel_18	18	
Channel_19	19	
Channel_20	20	
Channel_21	21	
Channel_22	22	
Channel_23	23	
Channel_24	24	
Channel_25	25	
Channel_26	26	
Channel_27	27	
Channel_28	28	
Channel_29	29	
Channel_30	30	
Channel_31	31	
Channel_32	32	
Channel_33	33	
Channel_34	34	
Channel_35	35	
No sub-block		
UNUSED	36	
Reference_Voltage	37	Registers described in "**Reference Voltage**" table
Global_Analog	38	Registers described in "**Global analog**" table
Master_TDC	39	Registers described in "**Master TDC**" table
Digital	40	Registers described in "**Digital**" table
ChannelWise	41	Registers described in "**channel wise**" I2C table
Top	42	Registers described in "**Top sub-block**" table

5.2 "Channel-wise" I2C parameters

Register # 0			
Bit	**Name**	**Default**	**Description**
0	ref_sk_lp2<0>	"0"	Local 8b-DAC for low gain pedestal tuning
1	ref_sk_lp2<1>	"0"	
2	ref_sk_lp2<2>	"0"	
3	ref_sk_lp2<3>	"0"	
4	ref_sk_lp2<4>	"0"	
5	ref_sk_lp2<5>	"0"	
6	ref_sk_lp2<6>	"0"	
7	ref_sk_lp2<7>	"0"	

Register # 1			
Bit	**Name**	**Default**	**Description**
0	ref_sk_lp<0>	"0"	Local 8b-DAC for medium gain pedestal tuning
1	ref_sk_lp<1>	"0"	
2	ref_sk_lp<2>	"0"	
3	ref_sk_lp<3>	"0"	
4	ref_sk_lp<4>	"0"	
5	ref_sk_lp<5>	"0"	
6	ref_sk_lp<6>	"0"	
7	ref_sk_lp<7>	"0"	

Register # 2			
Bit	**Name**	**Default**	**Description**
0	ref_sk<0>	"0"	Local 8b-DAC for high gain pedestal tuning
1	ref_sk<1>	"0"	
2	ref_sk<2>	"0"	
3	ref_sk<3>	"0"	
4	ref_sk<4>	"0"	
5	ref_sk<5>	"0"	
6	ref_sk<6>	"0"	
7	ref_sk<7>	"0"	

Register # 3			
Bit	**Name**	**Default**	**Description**
0	ref_inv<0>	"0"	Local 6b-DAC for pedestal tuning of ADC0 buffers
1	ref _inv<1>	"0"	
2	ref _inv<2>	"0"	
3	ref _inv<3>	"0"	
4	ref _inv<4>	"0"	
5	ref _inv<5>	"0"	
6	channel_off	"0"	Turn off the channel
7	extended_dr_off	"0"	Turn off the low gain preamplifier

Register # 4			
Bit	**Name**	**Default**	**Description**
0	ref_gs1<0>	"0"	Local 6b-DAC for second gain switching threshold tuning
1	ref_gs1<1>	"0"	
2	ref_gs1<2>	"0"	
3	ref_gs1<3>	"0"	

4	ref _gs1<4>	"0"	
5	ref _gs1<5>	"0"	
6	mask_pmos1	"0"	"1" = gain switching 0 off
7	mask_pmos2	"0"	"1" = gain switching 1 off

Register # 5			
Bit	**Name**	**Default**	**Description**
0	ref_inv_lp<0>	"0"	Local 6b-DAC for pedestal tuning of ADC1 buffers
1	ref _inv_lp <1>	"0"	
2	ref _inv_lp <2>	"0"	
3	ref _inv_lp <3>	"0"	
4	ref _inv_lp <4>	"0"	
5	ref _inv_lp <5>	"0"	
6	mask_toa	"0"	"1" = TOA discriminator off
7	mask_pmos3	"0"	"1" = resistor switching off

Register # 6			
Bit	**Name**	**Default**	**Description**
0	ref_gs0<0>	"0"	Local 6b-DAC for first gain switching threshold tuning
1	ref_gs0<1>	"0"	
2	ref_gs0<2>	"0"	
3	ref_gs0<3>	"0"	
4	ref _gs0<4>	"0"	
5	ref _gs0<5>	"0"	
6	cmd_probe_pa	"0"	Preamplifier output probe
7	cmd_probe_toa	"0"	Preamplifier output toa

Register # 7			
Bit	**Name**	**Default**	**Description**
0	ref_rs<0>	"0"	Local 6b-DAC for resistor switching threshold tuning
1	ref_rs<1>	"0"	
2	ref_rs<2>	"0"	
3	ref_rs<3>	"0"	
4	ref _rs<4>	"0"	
5	ref _rs<5>	"0"	
6	en_or_toa	"0"	To enable the toa channel to the global NOR
7	cmd_lr	"0"	Enable external charge injection

Register # 8			
Bit	**Name**	**Default**	**Description**
0	ref_toa<0>	"0"	Local 6b-DAC for TOA threshold tuning
1	ref_toa<1>	"0"	
2	ref_toa<2>	"0"	
3	ref_toa<3>	"0"	
4	ref _toa<4>	"0"	
5	ref _toa<5>	"0"	
6	cmd_probe_inv	"0"	Inv buffer output probe
7	cmd_probe_noinv	"0"	Noinv buffer output probe

Register # 9			
Bit	**Name**	**Default**	**Description**

0	ref_vin<0>	“0”	Local 8b-DAC for setting the input DC voltage
1	ref_vin<1>	“0”	
2	ref_vin<2>	“0”	
3	ref_vin<3>	“0”	
4	ref_vin<4>	“0”	
5	ref_vin<5>	“0”	
6	ref_vin<6>	“0”	
7	ref_vin<7>	“0”	

Register # 10			
Bit	**Name**	**Default**	**Description**
0	mask_adc	“0”	“0” = ADC0 off
1	mask_adc2	“0”	“0” = ADC1 off
2	hz_inv	“0”	Inv input of ADC0 connected to 600mv
3	hz_noinv	“0”	noinv input of ADC0 connected to 600mv
4	hz_inv_lp	“0”	Inv input of ADC1 connected to 600mv
5	hz_noinv_lp	“0”	noinv input of ADC1 connected to 600mv
6	manual_modeb	“0”	“1” = Gain switching active for ADC0
7	manual_mode_lp	“0”	“0” = Gain switching active for ADC1

Register # 11			
Bit	**Name**	**Default**	**Description**
0	DAC_CAL_CTDC_TOA<0>	“0”	Tune the fine gain of the TOA FTDC: 5 bits DAC <0:4> (1kΩ x 32) by the BIAS_CAL_DAC_P and the sign 1 bit <5> VD_P CAL = VD_P + (sign <5> x <0:4> x 1kΩ x BIAS_CAL_DAC_P)
1	DAC_CAL_CTDC_TOA <1>	“0”	
2	DAC_CAL_CTDC_TOA <2>	“0”	
3	DAC_CAL_CTDC_TOA <3>	“0”	
4	DAC_CAL_CTDC_TOA <4>	“0”	
5	DAC_CAL_CTDC_TOA <5>	“0”	
6	NC	“0”	
7	NC	“0”	

Register # 12			
Bit	**Name**	**Default**	**Description**
0	DAC_CAL_FTDC_TOA<0>	“0”	Tune the fine gain of the TOA FTDC: 5 bits DAC <0:4> (1kΩ x 32) by the BIAS_CAL_DAC_P and the sign 1 bit <5> VD_P CAL = VD_P + (sign <5> x <0:4> x 1kΩ x BIAS_CAL_DAC_P)
1	DAC_CAL_FTDC_ TOA <1>	“0”	
2	DAC_CAL_FTDC_ TOA <2>	“0”	
3	DAC_CAL_FTDC_ TOA <3>	“0”	
4	DAC_CAL_FTDC_ TOA <4>	“0”	
5	DAC_CAL_FTDC_ TOA <5>	“0”	
6	NC	“0”	
7	EN_TDC	“0”	

Register # 13			
Bit	**Name**	**Default**	**Description**
0	IN_FTDC_ENCODER_TOA<0>	“0”	Adjust the TOA FTDC offset by 5 bits <0:4> and the sign 1 bit <5> OFFSET = FTDC_TDC<0:5> + (sign <5> x DATA<0:4>)
1	IN_FTDC_ENCODER_TOA<1>	“0”	
2	IN_FTDC_ENCODER_TOA<2>	“0”	
3	IN_FTDC_ENCODER_TOA<3>	“0”	
4	IN_FTDC_ENCODER_TOA<4>	“0”	
5	IN_FTDC_ENCODER_TOA<5>	“0”	
6	NA		

7	NA		

5.3 "Global analog" I2C parameters

Register # 0			
Bit	**Name**	**Default**	**Description**
0	dacb_vbo_pa<0>	"0"	6b-DAC for preamp output stage current tuning
1	dacb _vbo_pa<1>	"0"	
2	dacb _vbo_pa<2>	"0"	
3	dacb_vbo_pa<3>	"0"	
4	dacb_vbo_pa<4>	"0"	
5	dacb_vbo_pa<5>	"1"	
6	on_pa_sc	"1"	"1" = enable preamplifier bias
7	half_vbm	"0"	Cut in half the current set by vbm

Register # 1			
Bit	**Name**	**Default**	**Description**
0	dacb_hyst_toa<0>	"0"	6b-DAC for TOA discri hysteresis tuning
1	dacb_hyst_toa<1>	"0"	
2	dacb_hyst_toa<2>	"0"	
3	dacb_hyst_toa<3>	"0"	
4	dacb_hyst_toa<4>	"0"	
5	dacb_hyst_toa<5>	"1"	
6	on_pmos_discri_sc	"1"	"1" = enable all gain switching discris
7	on_rtr_sc	"1"	"1" = enable ADC0 buffers

Register # 2			
Bit	**Name**	**Default**	**Description**
0	dac_inv_buf2<0>	"0"	Inverter buffer of ADC1 output stage current
1	dac_inv_buf2<1>	"0"	
2	dac_inv_buf2<2>	"1"	
3	dac_inv_buf2<3>	"1"	
4	dac_inv_buf2<4>	"0"	
5	dac_inv_buf2<5>	"0"	
6	ibib_inv_buf2<0>	"1"	Inverter buffer of ADC1 input stage current
7	ibib_inv_buf2<1>	"1"	

Register # 3			
Bit	**Name**	**Default**	**Description**
0	dac_noinv_buf2<0>	"0"	Non inverter buffer of ADC1 output stage current
1	dac_noinv_buf2<1>	"0"	
2	dac_noinv_buf2<2>	"1"	
3	dac_noinv_buf2<3>	"1"	
4	dac_noinv_buf2<4>	"0"	
5	dac_noinv_buf2<5>	"0"	
6	ibib_noinv_buf2<0>	"1"	Non inverter buffer of ADC1 input stage current
7	ibib_noinv_buf2<1>	"1"	

Register # 4

Bit	Name	Default	Description
0	dacb_vbi_pa<0>	"0"	6b-DAC for high gain preamp input stage current tuning
1	dacb_vbi_pa <1>	"0"	
2	dacb_vbi_pa <2>	"0"	
3	dacb_vbi_pa <3>	"0"	
4	dacb_vbi_pa <4>	"0"	
5	dacb_vbi_pa <5>	"0"	
6	on_pmos_opamp_sc	"1"	"1" = enable all shapers
7	on_toa_sc	"1"	"1" = enable TOA discri

Register # 5			
Bit	**Name**	**Default**	**Description**
0	dacb_vbi_pa_LP<0>	"0"	6b-DAC for medium and low gain preamp input stage current tuning
1	dacb_vbi_pa_LP <1>	"0"	
2	dacb_vbi_pa_LP <2>	"0"	
3	dacb_vbi_pa_LP <3>	"0"	
4	dacb_vbi_pa_LP <4>	"0"	
5	dacb_vbi_pa_LP <5>	"1"	
6	on_ota_toa_sc	"1"	"1" = enable high pass active filter
7	on_rtr2_sc	"1"	"1" = enable ADC1 buffers

Register # 6			
Bit	**Name**	**Default**	**Description**
0	dac_inv_buf<0>	"0"	Inverter buffer of ADC0 output stage current tuning
1	dac_inv_buf<1>	"0"	
2	dac_inv_buf<2>	"1"	
3	dac_inv_buf<3>	"1"	
4	dac_inv_buf<4>	"0"	
5	dac_inv_buf<5>	"0"	
6	ibib_inv_buf<0>	"1"	Inverter buffer of ADC0 input stage current tuning
7	ibib_inv_buf<1>	"1"	

Register # 7			
Bit	**Name**	**Default**	**Description**
0	dac_noinv_buf<0>	"0"	Non inverter buffer of ADC0 output stage current tuning
1	dac_noinv_buf<1>	"0"	
2	dac_noinv_buf<2>	"1"	
3	dac_noinv_buf<3>	"1"	
4	dac_noinv_buf<4>	"0"	
5	dac_noinv_buf<5>	"0"	
6	ibib_noinv_buf<0>	"1"	Non inverter buffer of ADC0 input stage current tuning
7	ibib_noinv_buf<1>	"1"	

Register # 8			
Bit	**Name**	**Default**	**Description**
0	cf_sk<0>	"1"	High gain shaper feedback capacitor: "0" = 50fF, "1" = 100fF, "2" = 200fF, "3" = 400fF
1	cf_sk<1>	"0"	
2	cf_sk<2>	"1"	
3	cf_sk<3>	"0"	
4	gain_sh<0>	"0"	High gain shaper feedback resistor: "0" = 40kΩ, "1" = 50kΩ, "2" = 100kΩ,
5	gain_sh<1>	"1"	

6	gain_sh<2>	“0”	“3” = 200kΩ
7	gain_sh<3>	“0”	

Register # 9			
Bit	**Name**	**Default**	**Description**
0	cf_sk_LP<0>	“1”	Medium gain shaper feedback capacitor: “0” = 50fF, “1” = 102fF, “2” = 189fF
1	cf_sk_LP<1>	“0”	
2	cf_sk_LP<2>	“0”	
3	gain_shLP<0>	“1”	Medium gain shaper feedback resistor: “0” = 30kΩ, “1” = 45kΩ, “2” = 60kΩ
4	gain_shLP<1>	“0”	
5	gain_shLP<2>	“0”	
6	en_deglitcher	“0”	“1” = enable the TOA deglitcher
7	en_hyst_toa	“1”	“1” = enable the TOA hysteresis

Register # 10			
Bit	**Name**	**Default**	**Description**
0	cf_sk_LP2<0>	“0”	Medium gain shaper feedback capacitor: “0” = 25fF, “1” = 50fF, “2” = 100fF
1	cf_sk_LP2<1>	“1”	
2	cf_sk_LP2<2>	“0”	
3	gain_shLP2<0>	“1”	Medium gain shaper feedback resistor: “0” = 30kΩ, “1” = 45kΩ, “2” = 60kΩ
4	gain_shLP2<1>	“0”	
5	gain_shLP2<2>	“0”	
6	sel_trig_toa	“0”	“1” = select external trigger for TOA discri
7	sel_trig_pmos	“0”	“1” = select external trigger for GS discri

Register # 11			
Bit	**Name**	**Default**	**Description**
0	dac_pmos_opamp<0>	“0”	6b-DAC for shapers output stage current tuning
1	dac_pmos_opamp<1>	“0”	
2	dac_pmos_opamp<2>	“0”	
3	dac_pmos_opamp<3>	“0”	
4	dac_pmos_opamp<4>	“0”	
5	dac_pmos_opamp<5>	“1”	
6	ibib_pmos_opamp<0>	“0”	2b for shapers input stage current tuning
7	ibib_pmos_opamp<1>	“0”	

Register # 12			
Bit	**Name**	**Default**	**Description**
0	s_noinv_buf_LP<0>	“1”	ADC1 noinv buffer miller capacitance: “0” = 50fF, “1” = 100fF, “2” = 200fF
1	s_noinv_buf_LP<1>	“1”	
2	s_noinv_buf_LP<2>	“0”	
3	s_inv_buf_LP<0>	“0”	ADC1 inv buffer miller capacitance: “0” = 50fF, “1” = 100fF, “2” = 200fF
4	s_inv_buf_LP<1>	“1”	
5	s_inv_buf_LP<2>	“0”	
6	pol_trig_toa	“1”	Polarity of TOA
7	NC	“0”	

Register # 13			
Bit	**Name**	**Default**	**Description**
0	s_sk<0>	“0”	Shapers miller capacitance: “0” = 50fF, “1” = 100fF, “2” = 200fF
1	s_sk<1>	“0”	

2	s_sk<2>	"1"	
3	read_gain_lp	"0"	Read gain with ADC1: "0" = medium gain, "1" = low gain. Only active when manual_mode_LP = "1"
4	gs2_delay<0>	"0"	Tune the threshold needed to activate the third gain switching.
5	gs2_delay<1>	"1"	
6	read_gain<0>	"0"	Read gain with ADC0: "0" = high gain path, "1" = medium gain path. "2" = low gain path. Only active when manual_modeb = "0"
7	read_gain<1>	"0"	

Register # 14			
Bit	**Name**	**Default**	**Description**
0	on_adc2	"1"	"0" = disconnects the analog mux form the ADC1 buffers
1	NC	"0"	
2	NC	"0"	
3	NC	"0"	
4	sw_rf_ota_toa<0>	"0"	High pass filter feedback resistance: "0" = 20kΩ, "1" = 40kΩ, "2" = 80kΩ, "3" = 160kΩ
5	sw_rf_ota_toa<1>	"0"	
6	sw_rf_ota_toa<2>	"1"	
7	sw_rf_ota_toa<3>	"0"	

Register # 15			
Bit	**Name**	**Default**	**Description**
0	s_noinv_buf<0>	"1"	ADC0 noinv buffer miller capacitance: "0" = 50fF, "1" = 100fF, "2" = 200fF
1	s_noinv_buf<1>	"1"	
2	s_noinv_buf<2>	"0"	
3	s_inv_buf<0>	"0"	ADC0 inv buffer miller capacitance: "0" = 50fF, "1" = 100fF, "2" = 200fF
4	s_inv_buf<1>	"1"	
5	s_inv_buf<2>	"0"	
6	NC	"0"	
7	NC	"0"	

Register # 16			
Bit	**Name**	**Default**	**Description**
0	ibib_ota_toa<0>	"0"	High pass filter current tuning
1	ibib_ota_toa<1>	"0"	
2	negp1	"0"	"1" = negative input polarity for GS0
3	en_hyst_pmos1	"0"	"1" = enable hysteresis for GS0
4	negp2	"0"	"1" = negative input polarity for GS1
5	en_hyst_pmos2	"0"	"1" = enable hysteresis for GS1
6	negp3	"0"	"1" = negative input polarity for Resistance switching
7	en_hyst_pmos3	"0"	"1" = enable hysteresis for Resistance switching

Register # 17			
Bit	**Name**	**Default**	**Description**
0	sw_rf<0>	"0"	Feedback resistor of high gain preamplifier: "0" = 1.05MΩ, "1" = 250kΩ, "2" = 60kΩ, "3" = 25kΩ
1	sw_rf<1>	"1"	
2	sw_rf<2>	"0"	
3	sw_rf<3>	"0"	

4	sw_cf<0>	"1"	Feedback capacitance of high gain preamplifier: "0" = 70fF, "1" = 140fF, "2" = 280fF
5	sw_cf<1>	"0"	
6	sw_cf<2>	"0"	
7	NC	"0"	

Register # 18			
Bit	**Name**	**Default**	**Description**
0	dac_vref<0>	"0"	Vref_sk, Vref_sk_LP and Vref_sk_LP2 channel wise DACs current tuning
1	dac_vref<1>	"1"	
2	dac_vref<2>	"0"	
3	dac_GS<0>	"0"	TOA, GS and RS channel wise DACs current tuning
4	dac_GS<1>	"1"	
5	dac_GS<2>	"0"	
6	on_vref_inv_sc	"1"	"1" = enable channel wise Vref tuning DACs
7	on_vpa_cf_ref_sc	"1"	"1" = enable second gain switching DC reference

Register # 19			
Bit	**Name**	**Default**	**Description**
0	dac_ota_vref_inv<0>	"0"	Global common mode voltage tuning (positive polarity)
1	dac_ota_vref_inv<1>	"0"	
2	dac_ota_vref_inv<2>	"0"	
3	dac_ota_vref_inv<3>	"0"	
4	dac_ota_vref_inv<4>	"0"	
5	dac_ota_vref_inv<5>	"0"	
6	dac_ota_vref_inv<6>	"0"	
7	dac_ota_vref_inv<7>	"0"	

Register # 20			
Bit	**Name**	**Default**	**Description**
0	dac_input_ampli<0>	"1"	6b-DAC for input amplifier output stage current tuning
1	dac_input_ampli<1>	"1"	
2	dac_input_ampli<2>	"1"	
3	dac_input_ampli<3>	"1"	
4	dac_input_ampli<4>	"1"	
5	dac_input_ampli<5>	"0"	
6	ibib_input_ampli<0>	"0"	2b for input amplifier input stage current tuning
7	ibib_input_ampli<1>	"1"	

Register # 21			
Bit	**Name**	**Default**	**Description**
0	sw_cin1<0>	"1"	Medium gain input capacitance tuning: "0" = 500fF, "1" = 500fF, "2" = 1pF
1	sw_cin1<1>	"1"	
2	sw_cin1<2>	"1"	
3	cstab	"1"	"1" = connect the stabilizing 100fF capcitance to the low power preamplifier
4	sw_cin2<0>	"0"	Low gain input capacitance tuning: "0" = 25fF, "1" = 50fF, "2" = 100fF
5	sw_cin2<1>	"1"	
6	sw_cin2<2>	"0"	
7	on_input_ampli_sc	"1"	"1" = enable input amplifier

Register # 22			
Bit	**Name**	**Default**	**Description**
0	dac_inv<0>	"1"	Common mode voltage channel wise DACs current tuning
1	dac_inv<1>	"1"	
2	dac_inv<2>	"1"	
3	rstb_gs	"1"	"1" = Enable gain switching state machine
4	connect2gnd	"0"	"1" = connect channel input to ground
5	r50	"1"	Channel input resistance set to: "0" = 200Ω, "1" = 50Ω
6	NC	"0"	
7	NC	"0"	

Register # 23			
Bit	**Name**	**Default**	**Description**
0	delay9<0>	"0"	Delay tuning for bit <9> "000" = faster conversion
1	delay9<1>	"0"	
2	delay9<2>	"0"	
3	delay87<0>	"0"	Delay tuning for bits <8:7> "000" = faster conversion
4	delay87<1>	"0"	
5	delay87<2>	"0"	
6	b_ref_adc<0>	"0"	Input stage current of the Ref ADC OTA
7	b_ref_adc<1>	"0"	

Register # 24			
Bit	**Name**	**Default**	**Description**
0	delay65<0>	"0"	Delay tuning for bits <6:5> "000" = faster conversion
1	delay65<1>	"0"	
2	delay65<2>	"0"	
3	delay40<0>	"0"	Delay tuning for bits <4:0> "000" = faster conversion
4	delay40<1>	"0"	
5	delay40<2>	"0"	
6	off_ref_adc	"0"	"0" = enable ADC ref OTA
7	on_ref_adc	"1"	"1" = enable ADC ref OTA

5.4 "Reference Voltage" I2C parameters

Register # 0			
Bit	**Name**	**Default**	**Description**
0	calib_dac<0>	"0"	Calibration DAC value
1	calib_dac<1>	"0"	
2	calib_dac<2>	"0"	
3	calib_dac<3>	"0"	
4	calib_dac<4>	"0"	
5	calib_dac<5>	"0"	
6	calib_dac<6>	"0"	
7	calib_dac<7>	"0"	

Register # 1			
Bit	**Name**	**Default**	**Description**
0	calib_dac<8>	"0"	Calibration DAC value

1	calib_dac<9>	"0"	
2	calib_dac<10>	"0"	
3	calib_dac<11>	"0"	
4	NA		
5	NA		
6	intctest	"0"	Selection of the Calibration DAC
7	extctest	"0"	Selection of the external pulse test

Register # 2			
Bit	**Name**	**Default**	**Description**
0	NA		
1	NA		
2	cmd_refi<0>	"0"	Bandgap current tuning
3	cmd_refi<1>	"0"	
4	cmd_vbg_1v<0>	"1"	1V bandgap ref. tuning
5	cmd_vbg_1v<1>	"1"	
6	cmd_vbg_1v<2>	"0"	
7	on_dac	"1"	"1" = enable DACs

Register # 3			
Bit	**Name**	**Default**	**Description**
0	bit_vref_rs<2>	"0"	Bits 2 to 9 used to set the global threshold of the resistor switching
1	bit_vref_rs<3>	"0"	
2	bit_vref_rs<4>	"1"	
3	bit_vref_rs<5>	"1"	
4	bit_vref_rs<6>	"0"	
5	bit_vref_rs<7>	"0"	
6	bit_vref_rs<8>	"0"	
7	bit_vref_rs<9>	"0"	

Register # 4			
Bit	**Name**	**Default**	**Description**
0	polarity_sk	"1"	Polarity of DAC used to set the global threshold for Vref_sk
1	polarity_sk_lp	"1"	Polarity of DAC used to set the global threshold for Vref_sk_LP
2	polarity_sk_lp2	"1"	Polarity of DAC used to set the global threshold for Vref_sk_LP2
3	polarity_toa	"1"	Polarity of DAC used to set the global threshold for Vref_toa
4	polarity_gs0	"0"	Polarity of DAC used to set the global threshold for GS0
5	polarity_gs1	"0"	Polarity of DAC used to set the global threshold for GS1
6	polarity_rs	"0"	Polarity of DAC used to set the global threshold for RS
7	cmd_probe_dc3	"0"	Select the probes 64:79 to probe_dc1 pin

Register # 5			
Bit	**Name**	**Default**	**Description**

0	bit_vref_gs1<2>	“1”	Bits 2 to 9 used to set the global threshold of the GS1
1	bit_vref_gs1<3>	“1”	
2	bit_vref_gs1<4>	“1”	
3	bit_vref_gs1<5>	“1”	
4	bit_vref_gs1<6>	“0”	
5	bit_vref_gs1<7>	“1”	
6	bit_vref_gs1<8>	“0”	
7	bit_vref_gs1<9>	“1”	

Register # 6			
Bit	**Name**	**Default**	**Description**
0	bit_vref_gs0<2>	“1”	Bits 2 to 9 used to set the global threshold of the GS0
1	bit_vref_gs0<3>	“1”	
2	bit_vref_gs0<4>	“1”	
3	bit_vref_gs0<5>	“1”	
4	bit_vref_gs0<6>	“1”	
5	bit_vref_gs0<7>	“0”	
6	bit_vref_gs0<8>	“1”	
7	bit_vref_gs0<9>	“1”	

Register # 7			
Bit	**Name**	**Default**	**Description**
0	bit_vref_gs0<0>	“0”	Bits 0 to 1 used to set the global threshold of the GS0
1	bit_vref_gs0<1>	“1”	
2	bit_vref_gs1<0>	“1”	Bits 0 to 1 used to set the global threshold of the GS1
3	bit_vref_gs1<1>	“1”	
4	bit_vref_rs<0>	“0”	Bits 0 to 1 used to set the global threshold of the RS
5	bit_vref_rs<1>	“0”	
6	NC	“0”	
7	NC	“0”	

Register # 8			
Bit	**Name**	**Default**	**Description**
0	bit_vref_toa<2>	“1”	Bits 2 to 9 used to set the global threshold of Vref_toa
1	bit_vref_toa<3>	“0”	
2	bit_vref_toa<4>	“0”	
3	bit_vref_toa<5>	“0”	
4	bit_vref_toa<6>	“1”	
5	bit_vref_toa<7>	“0”	
6	bit_vref_toa<8>	“1”	
7	bit_vref_toa<9>	“0”	

Register # 9			
Bit	**Name**	**Default**	**Description**
0	bit_vref_sk_lp2<2>	“1”	Bits 2 to 9 used to set the global threshold of Vref_sk_LP2
1	bit_vref_sk_lp2<3>	“0”	
2	bit_vref_sk_lp2<4>	“0”	
3	bit_vref_sk_lp2<5>	“1”	
4	bit_vref_sk_lp2<6>	“0”	
5	bit_vref_sk_lp2<7>	“0”	
6	bit_vref_sk_lp2<8>	“1”	

7	bit_vref_sk_lp2<9>	"0"	

Register # 10			
Bit	**Name**	**Default**	**Description**
0	bit_vref_sk_lp<2>	"1"	Bits 2 to 9 used to set the global threshold of Vref_sk_LP
1	bit_vref_sk_lp<3>	"0"	
2	bit_vref_sk_lp<4>	"0"	
3	bit_vref_sk_lp<5>	"1"	
4	bit_vref_sk_lp<6>	"0"	
5	bit_vref_sk_lp<7>	"0"	
6	bit_vref_sk_lp<8>	"1"	
7	bit_vref_sk_lp<9>	"0"	

Register # 11			
Bit	**Name**	**Default**	**Description**
0	bit_vref_sk<2>	"1"	Bits 2 to 9 used to set the global threshold of Vref_sk
1	bit_vref_sk<3>	"0"	
2	bit_vref_sk<4>	"1"	
3	bit_vref_sk<5>	"1"	
4	bit_vref_sk<6>	"0"	
5	bit_vref_sk<7>	"1"	
6	bit_vref_sk<8>	"0"	
7	bit_vref_sk<9>	"0"	

Register # 12			
Bit	**Name**	**Default**	**Description**
0	bit_vref_sk<0>	"1"	Bits 0 to 1 used to set the global threshold of Vref_sk
1	bit_vref_sk<1>	"1"	
2	bit_vref_sk_lp<0>	"0"	Bits 0 to 1 used to set the global threshold of Vref_sk_LP
3	bit_vref_sk_lp<1>	"1"	
4	bit_vref_sk_lp2<0>	"0"	Bits 0 to 1 used to set the global threshold of Vref_sk_LP2
5	bit_vref_sk_lp2<1>	"1"	
6	bit_vref_toa<0>	"0"	Bits 0 to 1 used to set the global threshold of Vref_toa
7	bit_vref_toa<1>	"0"	

Register # 13			
Bit	**Name**	**Default**	**Description**
0	probe_dc<0>	"0"	Selection among 32 probes to probe_dc1 pin
1	probe_dc<1>	"0"	
2	probe_dc<2>	"0"	
3	probe_dc<3>	"0"	
4	probe_dc<4>	"0"	
5	probe_dc1_0	"0"	Select the 32 first probes to probe_dc1 pin
6	probe_dc1_1	"0"	Select the 32 last probes to probe_dc1 pin
7	probe_dc2	"0"	Select calibration dac output to probe_dc2

5.5 "Master TDC" I2C parameters

Register # 0			
Bit	**Name**	**Default**	**Description**
0	GLOBAL_TA_SELECT_GAIN_TOA<0>	"1"	TOA pulse replicator gain : '1000' : 2 pulses (TDC LSB : 97,6 ps) '1100' : 4 pulses (TDC LSB : 48,8 ps)
1	GLOBAL_TA_SELECT_GAIN_TOA<1>	"1"	
2	GLOBAL_TA_SELECT_GAIN_TOA<2>	"1"	

3	GLOBAL_TA_SELECT_GAIN_TOA<3>	"0"	'1110' : 8 pulses (TDC LSB : 24,4 ps) (Default) '1111' : 16 pulses (TDC LSB : 12,2 ps)
4	NA		
5	COUNTER_PHASE<0>	"0"	TBD
6	COUNTER_PHASE<1>	"0"	
7	FIFO_DELL_40MHZ	"0"	TBD

Register # 1			
Bit	**Name**	**Default**	**Description**
0	GLOBAL_SEU_TIME_OUT	"1"	Activate the SEU time-out in the control logic channel
1	NA		
2	NA		
3	NA		
4	NA		
5	NA		
6	NA		
7	NA		

Register # 2			
Bit	**Name**	**Default**	**Description**
0	BIAS_FOLLOWER_CAL_P_D<0>	"0"	Configure the VTC follower for the CTDC channel calibration follower bias P current: Current = DATA<0:3> Weight addition 'base' : 50 nA <0>: 100 nA <1>: 500 nA <2>: 2,5 µA <3>: 5 µA
1	BIAS_FOLLOWER_CAL_P_D<1>	"0"	
2	BIAS_FOLLOWER_CAL_P_D<2>	"0"	
3	BIAS_FOLLOWER_CAL_P_D<3>	"0"	
4	BIAS_FOLLOWER_CAL_P_CTDC_EN	"0"	Enable the VTC follower for the CTDC channel calibration follower bias P
5	RST_COUNTER	"0"	TBD
6	DISABLE_AUTO_REFRESH_ COUNTER	"1"	TBD
7	CALIB_CHANNEL_DLL	"0"	Activate the automatic calibration of all TDC channel

Register # 3			
Bit	**Name**	**Default**	**Description**
0	VD_CTDC_P_D<0>	"0"	5 bits DAC for the VD_CTDC_P voltage : '00000' = 0,2 V '11111' = 0,75 V
1	VD_CTDC_P_D<1>	"0"	
2	VD_CTDC_P_D<2>	"0"	
3	VD_CTDC_P_D<3>	"0"	
4	VD_CTDC_P_D<4>	"0"	
5	VD_CTDC_P_DAC_EN	"0"	Activate the DAC for the VD_CTDC_P voltage : '0' : external PAD "VD_CTDC_P_EXT" '1' : 5 bits DAC "VD_CTDC_P_D<0:4>"
6	EN_MASTER_CTDC_VOUT_INIT	"0"	Activate the voltage load in "VD_FTDC_P" of the master FTDC DLL pump charge output : '0' : High Z pump charge (Default) '1' : FTDC_P INIT voltage load « VD_FTDC_P_DAC MODE»
7	EN_MASTER_CTDC_DLL	"1"	Activate the CTDC master DLL to provide master CLKs and global channel calibration : '0' : OFF : No CLKs "power off channels" '1' : ON : MASTER TDC activated (Default)

Register # 4			
Bit	**Name**	**Default**	**Description**

0	BIAS_CAL_DAC_CTDC_P_D<0>	"0"	Configure the VTC BIAS_CAL_DAC_CTDC_P bias of the current calibration channel: Current = DATA<0:5> Weight addition <0>: 100 nA <1>: 500 nA <2>: 1 µA <3>: 4 µA
1	BIAS_CAL_DAC_CTDC_P_D<1>	"0"	
2	CTDC_CALIB_FREQUENCY<0>	"0"	CTDC Master DLL calibration period : '000000' : 32 ns '100000' : 50 ns '010000' : 100 ns '001000' : 200 ns '000100' : 400 ns '000010' : 800 ns '000001' : 1600 ns '111111' : 3200 ns
3	CTDC_CALIB_FREQUENCY<1>	"1"	
4	CTDC_CALIB_FREQUENCY<2>	"0"	
5	CTDC_CALIB_FREQUENCY<3>	"0"	
6	CTDC_CALIB_FREQUENCY<4>	"0"	
7	CTDC_CALIB_FREQUENCY<5>	"0"	

Register # 5			
Bit	**Name**	**Default**	**Description**
0	GLOBAL_MODE_TOA_DIRECT_OUTPUT	"0"	The output of the TOA TDC are directly connected to the TOA & TOT output
1	BIAS_I_CTDC_D<0>	"0"	Configure the VTC the bias CTDC BIAS_N_CTDC bias (pump charge injection current) : Current = DATA<0:5> Weight addition (default 30 µA) 'base' : 50 nA <0>: 100 nA <1>: 500 nA <2>: 2,5 µA <3>: 5 µA <4>: 25 µA <5>: 50 µA
2	BIAS_I_CTDC_D<1>	"0"	
3	BIAS_I_CTDC_D<2>	"0"	
4	BIAS_I_CTDC_D<3>	"1"	
5	BIAS_I_CTDC_D<4>	"1"	
6	BIAS_I_CTDC_D<5>	"0"	
7	FOLLOWER_CTDC_EN	"1"	Activate the VTC for the bias MASTER CTDC BIAS_N_CTDC bias (pump charge injection): '0' : OFF '1' : VTC 6 bits "BIAS_I_CTDC_D<0:5>

Register # 6			
Bit	**Name**	**Default**	**Description**
0	GLOBAL_EN_BUFFER_CTDC	"0"	Enable the BUFFER of the calibration FTDC channel '0' : OFF '1' : 'VTC 6 bits "BIAS_CAL_DAC_FTDC_P_D<0:3>'
1	VD_CTDC_N_FORCE_MAX	"1"	**Force the VD_CTDC_N voltage to 1,2 V**
2	NC		
3	VD_CTDC_N_D<0>	"0"	5 bits DAC for the VD_CTDC_N voltage : '00000' = 0,6 V '11111' = 1 V
4	VD_CTDC_N_D<1>	"0"	
5	VD_CTDC_N_D<2>	"0"	
6	VD_CTDC_N_D<3>	"0"	
7	VD_CTDC_N_DAC_EN	"0"	Activate the DAC for the VD_CTDC_N voltage : '0' : external PAD "VD_CTDC_N_EXT" '1' : 5 bits DAC "VD_CTDC_N_D<0:4>"

Register # 7			
Bit	**Name**	**Default**	**Description**
0	EN_MASTER_CTDC_START_UP	"0"	TBD
1	CTRL_IN_ CTDC_P_SIG	"0"	TBD
2	NC	"0"	
3	NC	"0"	
4	NC	"0"	
5	NC	"0"	

6	BIAS_CAL_DAC_CTDC_P_D<2>	"0"	Configure the VTC BIAS_CAL_DAC_CTDC_P bias of the current calibration channel: Current = DATA<0:5> Weight addition <0>: 100 nA <1>: 500 nA <2>: 1 µA <3>: 4 µA
7	BIAS_CAL_DAC_CTDC_P_D<3>	"0"	

Register # 8

Bit	Name	Default	Description
0	CTRL_IN_SIG_CTDC_P_D<0>	"0"	5 bits DAC for the CTRL_SIG_CTDC calibration voltage : '00000' = 0,2 V '11111' = 0,8 V
1	CTRL_IN_SIG_CTDC_P_D<1>	"0"	
2	CTRL_IN_SIG_CTDC_P_D<2>	"0"	
3	CTRL_IN_SIG_CTDC_P_D<3>	"0"	
4	CTRL_IN_SIG_CTDC_P_D<4>	"0"	
5	CTRL_IN_SIG_CTDC_P_EN	"0"	Activate the DAC for CTRL_SIG_CTDC calibration voltage : **'0' : DAC OFF 0 V output** '1' : DAC 5 bits "CTRL_IN_SIG_CTDC_P_D<0:4>"
6	GLOBAL_INIT_DAC_B_CTDC	"0"	Load the channel calibration registers by DAC_CAL_CTDC_TOA <0:5> and DAC_CAL_CTDC_TOT <0:5> **'0' : LOAD by SC registers** '1' : Adjustment in automatic calibration mode
7	BIAS_CAL_DAC_CTDC_P_EN	"0"	Activate the VTC for the bias BIAS_CAL_DAC_CTDC_P bias of the current calibration channel **'0' : OFF** '1' : VTC 6 bits 'BIAS_CAL_DAC_CTDC_P_D<0:3>'

Register # 9

Bit	Name	Default	Description
0	VD_FTDC_P_D<0>	"0"	5 bits DAC for the VD_FTDC_P voltage : '00000' = 0,4 V '11111' = 0,9 V
1	VD_FTDC_P_D<1>	"0"	
2	VD_FTDC_P_D<2>	"0"	
3	VD_FTDC_P_D<3>	"0"	
4	VD_FTDC_P_D<4>	"0"	
5	VD_FTDC_P_DAC_EN	"0"	Activate the DAC for the VD_FTDC_P voltage : '0' : external PAD "VD_FTDC_P_EXT" '1' : 5 bits DAC "VD_FTDC_P_D<0:4>"
6	EN_MASTER_FTDC_VOUT_INIT	"0"	Activate the voltage load in "VD_FTDC_P" of the master FTDC DLL pump charge output : **'0' : High Z pump charge (Default)** '1' : FTDC_P INIT voltage load « VD_FTDC_P_DAC MODE»
7	EN_MASTER_FTDC_DLL	"1"	Activate the FTDC master DLL to provide master CLKs and global channel calibration : '0' : OFF : No CLKs "power off channels" **'1' : ON : MASTER TDC activated (Default)**

Register # 10

Bit	Name	Default	Description
0	NA	0	
1	BIAS_FOLLOWER_CAL_P_FTDC_EN	"0"	Activate the VTC for the bias MASTER FTDC BIAS_N_FTDC bias (pump charge injection): '0' : OFF '1' : VTC 6 bits "BIAS_I_FTDC_D<0:5>
2	FTDC_CALIB_FREQUENCY<0>	"0"	FTDC Master DLL calibration period : '000000' : 32 ns '100000' : 50 ns '010000' : 100 ns '001000' : 200 ns '000100' : 400 ns '000010' : 800 ns
3	FTDC_CALIB_FREQUENCY<1>	"1"	
4	FTDC_CALIB_FREQUENCY<2>	"0"	
5	FTDC_CALIB_FREQUENCY<3>	"0"	
6	FTDC_CALIB_FREQUENCY<4>	"0"	

7	FTDC_CALIB_FREQUENCY<5>	“0”	'000001' : 1600 ns '111111' : 3200 ns

Register # 11			
Bit	**Name**	**Default**	**Description**
0	EN_REF_BG	“1”	Activate the BANDGAP source 1V to the calibration part ‘0’ : External PAD ‘EXT_REF_TDC’ **‘1’ : Internal 1V Bandgap ‘BG_1V’**
1	BIAS_I_FTDC_D<0>	“0”	Configure the VTC the bias PLL BIAS_N_PLL bias (pump charge injection current) : Current = DATA<0:5> Weight addition (default 30 µA) ‘base’ : 50 nA <0>: 100 nA <1>: 500 nA <2>: 2,5 µA <3>: 5 µA <4>: 25 µA <5>: 50 µA
2	BIAS_I_FTDC_D<1>	“0”	
3	BIAS_I_FTDC_D<2>	“0”	
4	BIAS_I_FTDC_D<3>	“1”	
5	BIAS_I_FTDC_D<4>	“1”	
6	BIAS_I_FTDC_D<5>	“0”	
7	FOLLOWER_FTDC_EN	“1”	Activate the VTC for the bias MASTER FTDC BIAS_N_FTDC bias (pump charge injection): ‘0’ : OFF **‘1’ : VTC 6 bits “BIAS_I_FTDC_D<0:5>**

Register # 12			
Bit	**Name**	**Default**	**Description**
0	GLOBAL_EN_BUFFER_FTDC	“0”	Enable the BUFFER of the calibration FTDC channel **‘0’ : OFF** ‘1’ : ‘VTC 6 bits “BIAS_CAL_DAC_FTDC_P_D<0:3>’
1	VD_FTDC_N_FORCE_MAX	“1”	**Force the VD_FTDC_N voltage to 1,2 V**
2	NC		5 bits DAC for the VD_FTDC_N voltage : '00000' = 0,4 V '11111' = 0,9 V
3	VD_FTDC_N_D<0>	“0”	
4	VD_FTDC_N_D<1>	“0”	
5	VD_FTDC_N_D<2>	“0”	
6	VD_FTDC_N_D<3>	“0”	
7	VD_FTDC_N_DAC_EN	“0”	Activate the DAC for the VD_FTDC_N voltage : ‘0’ : external PAD “VD_FTDC_N_EXT” ‘1’ : 5 bits DAC “VD_FTDC_N_D<0:4>”

Register # 13			
Bit	**Name**	**Default**	**Description**
0	CTRL_IN_SIG_FTDC_P_D<0>	“0”	5 bits DAC for the CTRL_SIG_FTDC calibration voltage : '00000' = 0,2 V '11111' = 0,8 V
1	CTRL_IN_SIG_FTDC_P_D<1>	“0”	
2	CTRL_IN_SIG_FTDC_P_D<2>	“0”	
3	CTRL_IN_SIG_FTDC_P_D<3>	“0”	
4	CTRL_IN_SIG_FTDC_P_D<4>	“0”	
5	CTRL_IN_SIG_FTDC_P_EN	“0”	Activate the DAC for CTRL_SIG_FTDC calibration voltage : ‘0’ : DAC OFF 0 V output ‘1’ : DAC 5 bits “CTRL_IN_SIG_FTDC_P_D<0:4>”
6	BIAS_FOLLOWER_CAL_P_FTDC_D<0>	“0”	Configure the VTC follower for the FTDC channel calibration follower bias P current : Current = DATA<0:3> Weight addition **‘base’ : 50 nA** <0>: 100 nA <1>: 500 nA <2>: 2,5 µA <3>: 5 µA
7	BIAS_FOLLOWER_CAL_P_FTDC_D<1>	“0”	

Register # 14

Bit	Name	Default	Description
0	EN_MASTER_FTDC_START_UP	"0"	TBD
1	CTRL_IN_FTDC_P_SIG	"0"	TBD
2	NC		
3	NC		
4	NC		
5	NC		
6	BIAS_FOLLOWER_CAL_P_FTDC_D<2>	"0"	Configure the VTC follower for the FTDC channel calibration follower bias P current : Current = DATA<0:3> Weight addition **'base' : 50 nA** <0>: 100 nA <1>: 500 nA <2>: 2,5 µA <3>: 5 µA
7	BIAS_FOLLOWER_CAL_P_FTDC_D<3>	"0"	

Register # 15			
Bit	**Name**	**Default**	**Description**
0	GLOBAL_SYNC_OUT	"0"	Fixe the output data synchronization with the edge of the 40 MHz '0' : rising edge
1	GLOBAL_FORCE_EN_CLK	"0"	Force the output data transfer clk (always clock gating) '0' : 2 clk cycles dedicated to 1 data transfer (1 for the data & 1 for the 0 after) (low power mode) '1' : clk is always present for the data transfer (40 MHz power consumption)
2	GLOBAL_FORCE_EN_OUTPUT_DATA	"0"	Force the En flag event to '1' (always read TOA)
3	GLOBAL_FORCE_EN_TOA<0>	"0"	TOA pulse replicator gain : S0 & S1 for the digital '00' : 2 pulses (TDC LSB : 97,6 ps) '10' : 4 pulses (TDC LSB : 48,8 ps) '01' : 8 pulses (TDC LSB : 24,4 ps) (Default) '11' : 16 pulses (TDC LSB : 12,2 ps)
4	GLOBAL_FORCE_EN_TOA<1>	"1"	
5	GLOBAL_EN_TUNE_GAIN_DAC	"0"	Load the FTDC channel calibration registers by DAC_CAL_FTDC_TOA <0:5>: '0' : OFF '1' : LOAD by SC registers
6	Sel_clk_rcg_toa	"0"	'1': select clk_rcg to trig1_tot
7	Sel_clk_rcg_tot	"0"	'1': select clk_rcg to trig1_toa

5.6 "Digital part" I2C parameters

Register # 0			
Bit	**Name**	**Default**	**Description**
0	IdleFrame<0>	"0"	Default 28 LSB "1100 --- 1100" data readout on output links when no readout request – or at startup
1	IdleFrame<1>	"0"	
2	IdleFrame<2>	"1"	
3	IdleFrame<3>	"1"	
4	IdleFrame<4>	"0"	
5	IdleFrame<5>	"0"	
6	IdleFrame<6>	"1"	
7	IdleFrame<7>	"1"	

Register # 1			
Bit	**Name**	**Default**	**Description**
0	IdleFrame<8>	"0"	Default 28 LSB "1100 --- 1100" data readout on output links when no readout request – or at startup
1	IdleFrame<9>	"0"	
2	IdleFrame<10>	"1"	
3	IdleFrame<11>	"1"	
4	IdleFrame<12>	"0"	
5	IdleFrame<13>	"0"	
6	IdleFrame<14>	"1"	
7	IdleFrame<15>	"1"	

Register # 2			
Bit	**Name**	**Default**	**Description**
0	IdleFrame<16>	"0"	Default 28 LSB "1100 --- 1100" data readout on output links when no readout request – or at startup
1	IdleFrame<17>	"0"	
2	IdleFrame<18>	"1"	
3	IdleFrame<19>	"1"	
4	IdleFrame<20>	"0"	
5	IdleFrame<21>	"0"	
6	IdleFrame<22>	"1"	
7	IdleFrame<23>	"1"	

Register # 3			
Bit	**Name**	**Default**	**Description**
0	IdleFrame<24>	"0"	Default 28 LSB "1100 --- 1100" data readout on output links when no readout request – or at startup
1	IdleFrame<25>	"0"	
2	IdleFrame<26>	"1"	
3	IdleFrame<27>	"1"	
4	LAT_ADC<0>	"1"	needed ADC Latency from 0 to 3
5	LAT_ADC<1>	"1"	
6	AsicMode<0>	"1"	AsicMode<1:0>: 00 → **H2GEIC (not used in CALOROC1B)** 01 → **CALO3g** (2 ADCs) 10 → **CALO4g** (1 ADC, 4 gains) 11 → **STATUS** (not an input mode; used for I2C-read output only)
7	AsicMode<1>	"0"	

Register # 4			
Bit	**Name**	**Default**	**Description**
0	SC_testRAM_data<0>	"1"	Default test DRAM pattern
1	SC_testRAM_data<1>	"0"	
2	SC_testRAM_data<2>	"1"	
3	SC_testRAM_data<3>	"0"	
4	SC_testRAM_data<4>	"1"	
5	SC_testRAM_data<5>	"0"	
6	SC_testRAM_data<6>	"1"	
7	SC_testRAM_data<7>	"0"	

Register # 5			
Bit	**Name**	**Default**	**Description**

0	SC_testRAM_data<8>	“1”	Default test DRAM pattern
1	SC_testRAM_data<9>	“0”	
2	SC_testRAM_data<10>	“1”	
3	SC_testRAM_data<11>	“0”	
4	SC_testRAM_data<12>	“1”	
5	SC_testRAM_data<13>	“0”	
6	SC_testRAM_data<14>	“1”	
7	SC_testRAM_data<15>	“0”	

Register # 6			
Bit	**Name**	**Default**	**Description**
0	SC_testRAM_data<16>	“1”	Default test DRAM pattern
1	SC_testRAM_data<17>	“0”	
2	SC_testRAM_data<18>	“1”	
3	SC_testRAM_data<19>	“0”	
4	SC_testRAM_data<20>	“1”	
5	SC_testRAM_data<21>	“0”	
6	SC_testRAM_data<22>	“1”	
7	SC_testRAM_data<23>	“0”	

Register # 7			
Bit	**Name**	**Default**	**Description**
0	SC_testRAM_data<24>	“1”	Default test DRAM pattern
1	SC_testRAM_data<25>	“0”	
2	SC_testRAM_data<26>	“1”	
3	SC_testRAM_data<27>	“0”	
4	SC_testRAM_data<28>	“1”	
5	SC_testRAM_data<29>	“0”	
6	SC_testRAM_data<30>	“1”	
7	SC_testRAM_data<31>	“0”	

Register # 8			
Bit	**Name**	**Default**	**Description**
0	OFF_LDO	“0”	TBD
1	NC	“0”	
2	NC	“0”	
3	NC	“0”	
4	NC	“0”	
5	NC	“0”	
6	NC	“0”	
7	NC	“0”	

Register # 9			
Bit	**Name**	**Default**	**Description**
0	NC	"0"	
1	NC	"0"	
2	NC	"0"	
3	NC	"0"	
4	NC	"0"	
5	NC	"0"	
6	NC	"0"	
7	NC	"0"	

Register # 10			
Bit	**Name**	**Default**	**Description**
0	NC	"0"	
1	NC	"0"	
2	NC	"0"	
3	NC	"0"	
4	NC	"0"	
5	NC	"0"	
6	NC	"0"	
7	NC	"0"	

Register # 11			
Bit	**Name**	**Default**	**Description**
0	Ret_time<0>	"0"	Time RETENTION SC parameter for test DRAM RETENTION time
1	Ret_time<1>	"0"	
2	Ret_time<2>	"0"	
3	Ret_time<3>	"0"	
4	Ret_time<4>	"1"	
5	Ret_time<5>	"0"	
6	Ret_time<6>	"0"	
7	Ret_time<7>	"1"	

Register # 12			
Bit	**Name**	**Default**	**Description**
0	Ret_time<8>	"1"	Time RETENTION SC parameter for test DRAM RETENTION time
1	Ret_time<9>	"0"	
2	Ret_time<10>	"0"	
3	Ret_time<11>	"0"	
4	ZeroSuppressEn	"1"	SC: If =1 -> DaqData out with zero suppress, only hit channels will be output, If=0 -> DaqData all CH will be output on link
5	CharacterisationSC	"0"	Not used in CALOROC1B, default value
6	TestMode	"0"	Test mode to enter BIST (test of DRAM)
7	RUN	"0"	Run mode

Register # 13			
Bit	**Name**	**Default**	**Description**
0	TimeOffset<0>	"1"	After reset, the time counter starts from this offset value.
1	TimeOffset<1>	"0"	
2	TimeOffset<2>	"0"	
3	TimeOffset<3>	"0"	
4	TimeOffset<4>	"0"	
5	TimeOffset<5>	"0"	
6	TimeOffset<6>	"0"	
7	TimeOffset<7>	"0"	

Register # 14			
Bit	**Name**	**Default**	**Description**
0	TimeOffset<8>	"0"	After reset, the time counter starts from this offset value.
1	TimeOffset<9>	"0"	
2	TimeOffset<10>	"0"	
3	TimeOffset<11>	"0"	
4	TimeOffset<12>	"0"	
5	TimeOffset<13>	"0"	
6	TimeOffset<14>	"0"	
7	TimeOffset<15>	"0"	

Register # 15			
Bit	**Name**	**Default**	**Description**
0	TimeOffset<16>	"0"	After reset, the time counter starts from this offset value.
1	TimeOffset<17>	"0"	
2	TimeOffset<18>	"0"	
3	TimeOffset<19>	"0"	
4	TimeOffset<20>	"0"	
5	TimeOffset<21>	"0"	
6	TimeOffset<22>	"0"	
7	TimeOffset<23>	"0"	

Register #16			
Bit	**Name**	**Default**	**Description**
0	N_VALUE<0>	"0"	N consecutive snapshot of the digitized data normal op =5
1	N_VALUE<1>	"0"	
2	N_VALUE<2>	"1"	
3	ADCsampleR	"1"	ADC -> 1: Sample on rising edge, 0: Sample on falling edge
4	SC_testRAM_cmd	"0"	Bit set to test/fill DRAM with specific pattern
5	EdgeSel_T1A	"0"	Selection clock edge of the fast command unit
6	In_inv_cmd_rx	"0"	Invert fast command decoding
7	n_counter_rst	"1"	Reset SEU counter of the serializer

Register # 17			
Bit	**Name**	**Default**	**Description**
0	Time_trigger<0>	"1"	Generate Special header on links @ time = Time_trigger
1	Time_trigger<1>	"1"	
2	Time_trigger<2>	"1"	
3	Time_trigger<3>	"1"	
4	Time_trigger<4>	"0"	
5	Time_trigger<5>	"0"	
6	Time_trigger<6>	"0"	
7	Time_trigger<7>	"0"	

Register # 18			
Bit	**Name**	**Default**	**Description**
0	Time_trigger<8>	"0"	Generate Special header on links @ time = Time_trigger
1	Time_trigger<9>	"0"	
2	Time_trigger<10>	"0"	
3	Time_trigger<11>	"0"	
4	Time_trigger<12>	"0"	
5	Time_trigger<13>	"0"	
6	Time_trigger<14>	"0"	
7	Time_trigger<15>	"0"	

Register # 19			
Bit	**Name**	**Default**	**Description**
0	Time_trigger<16>	"0"	Generate Special header on links @ time = Time_trigger
1	Time_trigger<17>	"0"	
2	Time_trigger<18>	"0"	
3	Time_trigger<19>	"0"	
4	Time_trigger<20>	"0"	
5	Time_trigger<21>	"0"	
6	Time_trigger<22>	"0"	
7	Time_trigger<23>	"0"	

Register #20			
Bit	**Name**	**Default**	**Description**
0	VaL_AF<0>	"1"	Fixed Almost full value for DRAM
1	VaL_AF<1>	"1"	
2	VaL_AF<2>	"0"	
3	VaL_AF<3>	"1"	
4	VaL_AF<4>	"1"	
5	VaL_AF<5>	"0"	
6	TOAsampleR	"1"	TOA -> 1: Sample on rising edge, 0: Sample on falling edge
7	TOTsampleR	"1"	TOT -> 1: Sample on rising edge, 0: Sample on falling edge

5.7 "Top sub-block" I2C parameters

Register # 0			
Bit	**Name**	**Default**	**Description**
0	hf_test_tspff	"0"	Frequency for testing TSPFF: "0" = 40MHz, "1" = 640MHz
1	on_test_tspff	"0"	"1" = enable the TSPFF test circuit
2	NC		
3	NC		
4	NC		
5	NC		
6	NC		
7	NC		

Register # 1			
Bit	**Name**	**Default**	**Description**
0	EN_START_UP	"1"	Enable PLL

1	DIV_PLL_A	"1"	Choose the "OUT_CLK_SEL_READ" output frequency <A:B>: '00' : 40 MHz '10' : 160 MHz '01' : 320 MHz **'11' : 1280 MHz**
2	DIV_PLL_B	"1"	
3	FOLLOWER_PLL_EN	"1"	Activate the VTC for the bias PLL bias (pump charge injection): '0' : OFF **'1' : VTC 6 bits "BIAS_I_PLL_D<0:5>**
4	Sel_strobe_ext	"0"	'0': CalPulseInt; '1': Strobe_ext pin
5	NC_set		
6	Sel_error	"0"	'0': I2C error; '1': fast command error
7	Sel_lock	"0"	'0': PLL lock; '1': fast command lock

Register # 2			
Bit	**Name**	**Default**	**Description**
0	VOUT_INIT_EXT_EN	"0"	Activate the DAC for V_PUMP_P voltage : '0' : external PAD "VOUT_INIT_EXT PLL" '1' : 5 bits DAC "VOUT_INIT_EXT_D<0:4>"
1	VOUT_INIT_EXT_D<0>	"0"	5 bits DAC for VOUT_INIT PLL voltage : '00000' = 0 V '11111' = 1.2 V
2	VOUT_INIT_EXT_D<1>	"0"	
3	VOUT_INIT_EXT_D<2>	"0"	
4	VOUT_INIT_EXT_D<3>	"0"	
5	VOUT_INIT_EXT_D<4>	"0"	
6	EN_REF_BG	"1"	Activate the BANDGAP source 1V to the calibration part '0' : External PAD 'EXT_REF_TDC' **'1' : Internal 1V Bandgap 'BG_1V'**
7	VOUT_INIT_EN	"0"	Activate the voltage load in "V_PUMP_P" of the PLL pump charge output : **'0' : High Z pump charge (Default)** '1' : VOUT_INIT voltage load « VOUT_INIT MODE»

Register # 3			
Bit	**Name**	**Default**	**Description**
0	EN_HIGH_CAPA	"1"	Add another large capacitor in the PLL LPF
1	BIAS_I_PLL_D<0>	"0"	Configure the VTC the bias PLL BIAS_N_PLL bias (pump charge injection current) : Current = DATA<0:5> Weight addition (**default 30 µA**) 'base' : 50 nA <0>: 100 nA <1>: 500 nA <2>: 2,5 µA <3>: 5 µA <4>: 25 µA <5>: 50 µA
2	BIAS_I_PLL_D<1>	"0"	
3	BIAS_I_PLL_D<2>	"0"	
4	BIAS_I_PLL_D<3>	"1"	
5	BIAS_I_PLL_D<4>	"1"	
6	BIAS_I_PLL_D<5>	"0"	
7	EN_RCG	"0"	Active the RCG: **'0' : OFF** '1' : ON : RCG activated

Register # 4			
Bit	**Name**	**Default**	**Description**
0	INIT_DAC_EN	"0"	Activate the DAC for the VCO voltage : '0' : external PAD "INIT_EXT" '1' : 5 bits DAC "INIT_D<0:4>"

1	INIT_D<0>	"0"	**Tune VCO frequency of the RCG: 5 bits DAC <0:4> (1kΩ x 32)**
2	INIT_D<1>	"0"	
3	INIT_D<2>	"0"	
4	INIT_D<3>	"0"	
5	INIT_D<4>	"0"	
6	Rcg_gain<0>	"0"	**Choose the "OUT_CLK_ RCG" output frequency <A:B>: '00' : /4096 '10' : /512 '01' : /32 '11' : pure**
7	Rcg_gain<1>	"0"	

Register # 5			
Bit	**Name**	**Default**	**Description**
0	EN<0>	"1"	Current value of the CLPS drivers (Data/Trigger)
1	EN<1>	"1"	
2	EN<2>	"0"	
3	ENpE<0>	"0"	Current value of the CLPS pre-emphasis driver (Data/Trigger)
4	ENpE<1>	"0"	
5	ENpE<2>	"0"	
6	S<0>	"0"	Delay value of the CLPS pre-emphasis driver (Data/Trigger)
7	S<1>	"0"	

Register # 6			
Bit	**Name**	**Default**	**Description**
0	EN_LOCK_CONTROL	"1"	Active the PLL lock detector : '0' : OFF **'1' : ON : PLL lock detector activated**
1	ERROR_LIMIT_SC<0>	"0"	Configure the maximal error limit between EXT 40 MHz & PLL 40 MHz : '000' : 0 difference '100' : 1 difference … '111' : 7 difference
2	ERROR_LIMIT_SC<1>	"1"	
3	ERROR_LIMIT_SC<2>	"0"	
4	EN_trig	"0"	
5	Pll_Lockec_sc	"0"	Slow-control lock pll signal
6	EN_probe_pll	"0"	Enable probe PLL
7	EN_PhaseShift	"1"	Enable phase shifter

Register # 7			
Bit	**Name**	**Default**	**Description**
0	Phase_ck<0>	"0"	Phase shifter value for 40M & 160M clocks
1	Phase_ck<1>	"0"	
2	Phase_ck<2>	"0"	
3	Phase_ck<3>	"0"	
4	Phase_strobe<0>	"0"	Phase shifter value for calibration pulse
5	Phase_strobe<1>	"0"	
6	Phase_strobe<2>	"0"	
7	Phase_strobe<3>	"0"	

Register # 8 Read-Only			
Bit	**Name**	**Default**	**Description**

0	B_out<0>	“0”	Read e-fuse values
1	B_out<1>	“0”	
2	B_out<2>	“0”	
3	B_out<3>	“0”	
4	B_out<4>	“0”	
5	B_out<5>	“0”	
6	B_out<6>	“0”	
7	B_out<7>	“0”	

Register # 9			
Bit	**Name**	**Default**	**Description**
0	B_in<0>	“0”	Write e-fuse values
1	B_in<1>	“0”	
2	B_in<2>	“0”	
3	B_in<3>	“0”	
4	B_in<4>	“0”	
5	B_in<5>	“0”	
6	B_in<6>	“0”	
7	B_in<7>	“0”	

Register # 10 Read-Only			
Bit	**Name**	**Default**	**Description**
0	B_out<8>	“0”	Read e-fuse values
1	B_out<9>	“0”	
2	B_out<10>	“0”	
3	B_out<11>	“0”	
4	B_out<12>	“0”	
5	B_out<13>	“0”	
6	B_out<14>	“0”	
7	B_out<15>	“0”	

Register # 11			
Bit	**Name**	**Default**	**Description**
0	B_in<8>	“0”	Write e-fuse values
1	B_in<9>	“0”	
2	B_in<10>	“0”	
3	B_in<11>	“0”	
4	B_in<12>	“0”	
5	B_in<13>	“0”	
6	B_in<14>	“0”	
7	B_in<15>	“0”	

Register # 12 Read-Only			
Bit	**Name**	**Default**	**Description**
0	B_out<16>	“0”	Read e-fuse values
1	B_out<17>	“0”	
2	B_out<18>	“0”	

3	B_out<19>	“0”	
4	B_out<20>	“0”	
5	B_out<21>	“0”	
6	B_out<22>	“0”	
7	B_out<23>	“0”	

Register # 13			
Bit	**Name**	**Default**	**Description**
0	B_in<16>	“0”	Write e-fuse values
1	B_in<17>	“0”	
2	B_in<18>	“0”	
3	B_in<19>	“0”	
4	B_in<20>	“0”	
5	B_in<21>	“0”	
6	B_in<22>	“0”	
7	B_in<23>	“0”	

Register # 14 Read-Only			
Bit	**Name**	**Default**	**Description**
0	B_out<24>	“0”	Read e-fuse values
1	B_out<25>	“0”	
2	B_out<26>	“0”	
3	B_out<27>	“0”	
4	B_out<28>	“0”	
5	B_out<29>	“0”	
6	B_out<30>	“0”	
7	SROUT	“0”	Srout read

Register # 15			
Bit	**Name**	**Default**	**Description**
0	B_in<24>	“0”	Write e-fuse values
1	B_in<25>	“0”	
2	B_in<26>	“0”	
3	B_in<27>	“0”	
4	B_in<28>	“0”	
5	B_in<29>	“0”	
6	B_in<30>	“0”	E-fuse read signal
7	B_in<31>	“0”	E-fuse write signal

Register # 16 Read-Only			
Bit	**Name**	**Default**	**Description**
0	BistRamErrors<16>	“0”	**Error Count per DRAM (4 bits):** Each bit represents a possible error in one of four test patterns (bit set to ‘1’
1	BistRamErrors<17>	“0”	
2	BistRamErrors<18>	“0”	

3	BistRamErrors<19>	“0”	indicates an error). With 6 DRAM blocks, this yields a total of **24 bits** of error information.
4	BistRamErrors<20>	“0”	
5	BistRamErrors<21>	“0”	
6	BistRamErrors<22>	“0”	
7	BistRamErrors<23>	“0”	

Register # 17 Read-Only			
Bit	**Name**	**Default**	**Description**
0	BistRamErrors<8>	“0”	**Error Count per DRAM (4 bits):** Each bit represents a possible error in one of four test patterns (bit set to ‘1’ indicates an error). With 6 DRAM blocks, this yields a total of **24 bits** of error information.
1	BistRamErrors<9>	“0”	
2	BistRamErrors<10>	“0”	
3	BistRamErrors<11>	“0”	
4	BistRamErrors<12>	“0”	
5	BistRamErrors<13>	“0”	
6	BistRamErrors<14>	“0”	
7	BistRamErrors<15>	“0”	

Register # 18 Read-Only			
Bit	**Name**	**Default**	**Description**
0	BistRamErrors<0>	“0”	**Error Count per DRAM (4 bits):** Each bit represents a possible error in one of four test patterns (bit set to ‘1’ indicates an error). With 6 DRAM blocks, this yields a total of **24 bits** of error information.
1	BistRamErrors<1>	“0”	
2	BistRamErrors<2>	“0”	
3	BistRamErrors<3>	“0”	
4	BistRamErrors<4>	“0”	
5	BistRamErrors<5>	“0”	
6	BistRamErrors<6>	“0”	
7	BistRamErrors<7>	“0”	

Register # 19 Read-Only			
Bit	**Name**	**Default**	**Description**
0	Status<0>	“0”	FSM state from global startup state machine
1	Status<1>	“0”	
2	Status<2>	“0”	
3	Bist_dones<0>	“0”	Flags indicates Bists complete verifications and remain in Finish state until reset
4	Bist_dones<1>	“0”	
5	vss	“0”	Always “0”
6	vss	“0”	
7	vss	“0”	

Register # 20 Read-Only			
Bit	**Name**	**Default**	**Description**
0	Lock_count<0>	“0”	Running count of number of times the phase lock is successful:no-wrap,cleared on reset
1	Lock_count<1>	“0”	
2	Lock_count<2>	“0”	
3	Lock_count<3>	“0”	
4	Lock_count<4>	“0”	

5	Lock_count<5>	“0”	
6	Lock_count<6>	“0”	
7	Lock_count<7>	“0”	

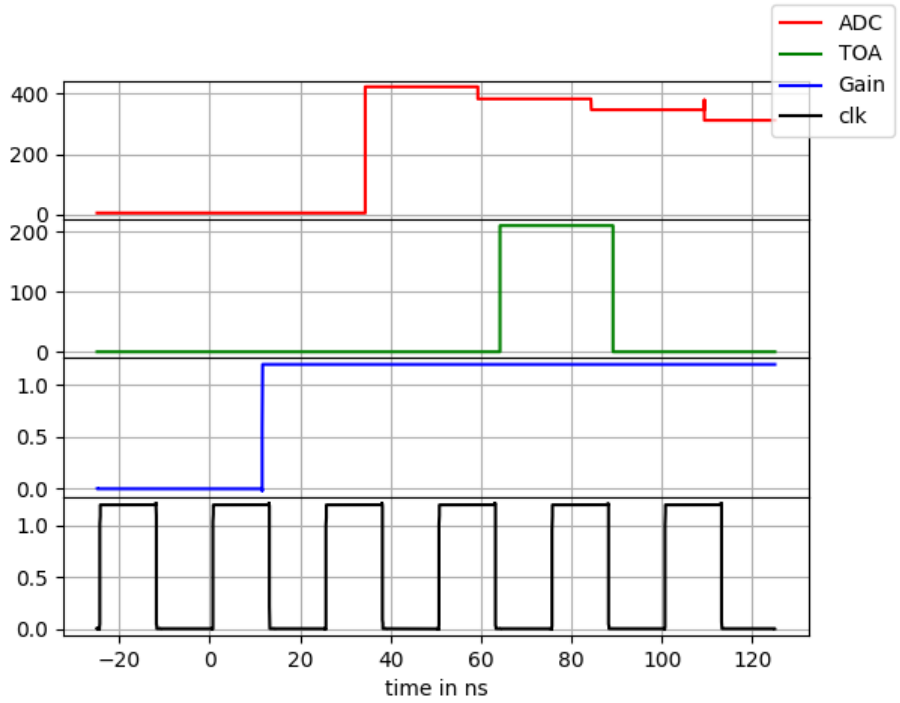
Register # 21 Read-Only			
Bit	**Name**	**Default**	**Description**
0	Fc_error_count<0>	“0”	Running count of number of times the fastcmd is not correct: no-wrap,cleared on reset
1	Fc_error_count<1>	“0”	
2	Fc_error_count<2>	“0”	
3	Fc_error_count<3>	“0”	
4	Fc_error_count<4>	“0”	
5	Fc_error_count<5>	“0”	
6	Fc_error_count<6>	“0”	
7	Fc_error_count<7>	“0”	

Register # 22 Read-Only			
Bit	**Name**	**Default**	**Description**
0	Err_count0<0>	“0”	Number of pll_lock loss seen by the serializer
1	Err_count0<1>	“0”	
2	Err_count0<2>	“0”	
3	Err_count0<3>	“0”	
4	Err_count0<4>	“0”	
5	Err_count0<5>	“0”	
6	Err_count0<6>	“0”	
7	Err_count0<7>	“0”	

6 Annexe A

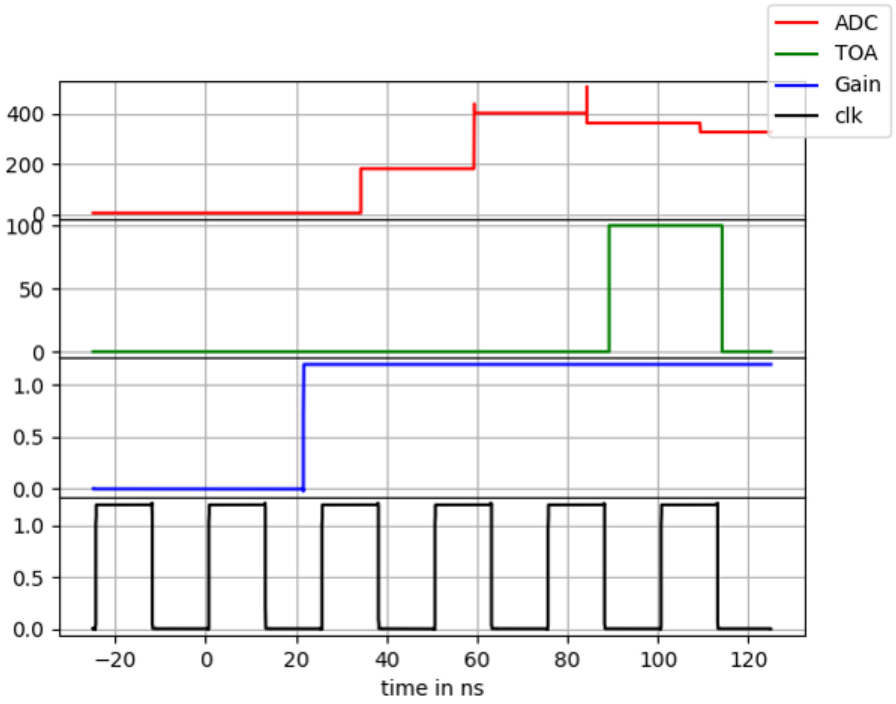
6.1 Align buffer simulation for different phases

6.1.1 Align buffer for the 10ns phase



LAT_ADC	0	1	2	3
Gain	3	3	0	0
ADC3g/4g	390	410	60	60
TOA	210	210	210	210

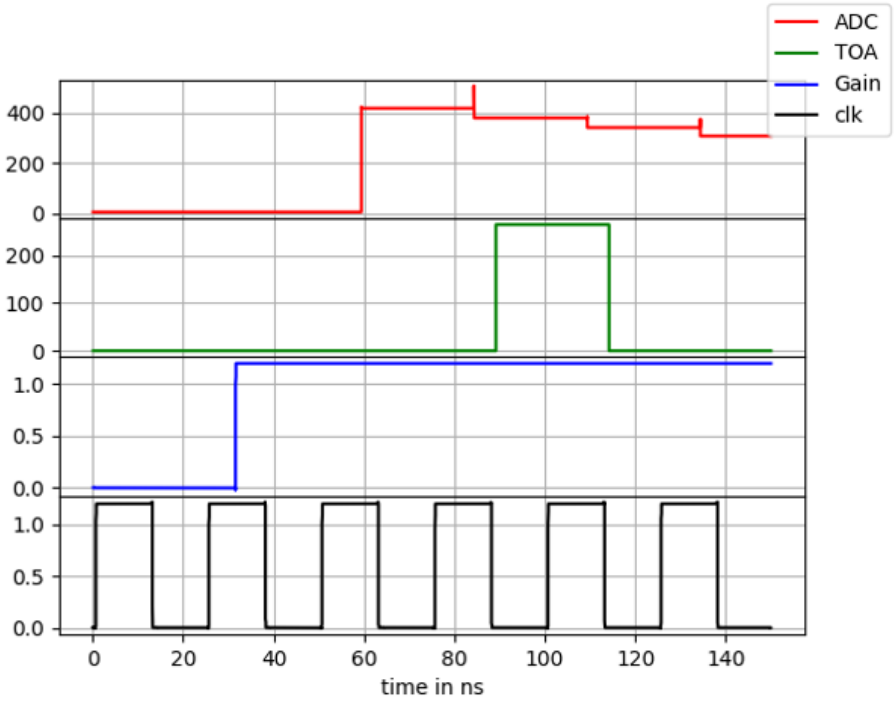
6.1.2 Align buffer for the 20ns phase



LAT_ADC	0	1	2	3
Gain	3	3	0	0
ADC3g/4g	350	400	195	60
TOA	100	100	100	100

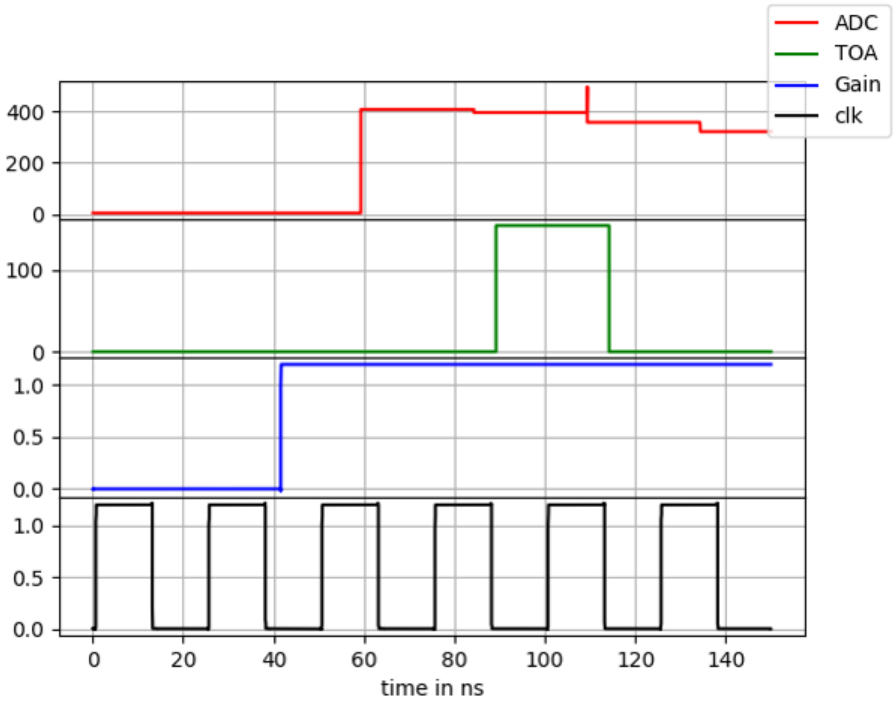
For this phase we only have one pedestal point, thereof the need of having a latency of 3 to sample at least one pedestal point.

6.1.3 Align buffer for the 30ns phase



LAT_ADC	0	1	2	3
Gain	3	3	0	0
ADC3g/4g	390	410	60	60
TOA	280	280	280	280

6.1.4 Align buffer for the 40ns phase



LAT_ADC	0	1	2	3
Gain	3	3	0	0
ADC3g/4g	400	405	60	60
TOA	150	150	150	150