H2GCROC3 Datasheet

Requested additions/clarifications:

-

Modified from H2GCROC2 datasheet and HGCROC3-Si Datasheet				
https://cernbox.cern.ch/index.php/s/PX8mv0zUDdSRdPb#office				
File version	Modified by	Date	Section	Changes
1_2	José Gonzalez	19/11/2021	1	Update diagram for HGCROC3
			1.1.1	Rf and Cf_comp values
				Compensation with common mode channel. Description, schematic and plot
				Dacb compensation dynamic range. Description, schematic and plot
			1.1.2	New Section
			1.1.3 (Before 1.1.2)	Added
			1.3.1.1	New Section
			1.3.1.2	New Section
			1.3.1.3	New Section
			1.3.2.1	New Section
			1.3.2.2	New Section
			1.3.2.3	New Section
			1.3.2.4	New Section
			1.3.2.5	New Section
			1.3.3	New Section
			1.3.3.1	New Section
			1.3.5	Name of Section changed
			1.3.5.1	New Section
			1.3.5.2	New Section
			1.3.5.3	New Section
			1.3.5.4	New Section
			1.3.5.5	New Section
			1.4.1	Updated
			1.4.2	Updated
			1.4.3	Updated
			1.5.1	Updated
			1.5.2	Updated

			1.5.3	New calibdac, descriptions, plots and tables added
			1.5.3.1	New section
			1.5.4	New section
			2.1	New section
			2.2	Previously Section 2.1
			3.2	Added
				Add Register 14
			3.3	Added
			3.4	Added
				Add Registers 9, 10
			3.5	Added
			3.6	Added
				Add Registers 15,16,17,18,19,20,21,22,23,24,25,26
			3.7	Added
				Add Register 8,9,10,11,12,13,14,15,16,17,18,19,20
1_3	José Gonzalez	15/12/2021	1.1.1	CC schematic updated
			1.1.4	Minor changes
			1.7.3	Calibration circuit updated
			all	Information removed from working doc
			all	Information added from HGCROC3-Si Datasheet

Table of contents

1 H2GCROC3: architectural overview

The overall block diagram is described in Figure 1.1. Most of the blocks and functionalities of HGCROC2 remain unchanged for the third and final version. The main changes are regarding the digital processing of the DAQ path like the L1 triggered event FIFO, the hamming coding and encoding. For simplicity, the Circular Buffer will be called RAM1 and the L1 triggered event FIFO will be called RAM2. Every Control and Command's Modules are triplicated with the TMRG tool. The data inside RAM1 and RAM2 are all processed with a Hamming encoding.

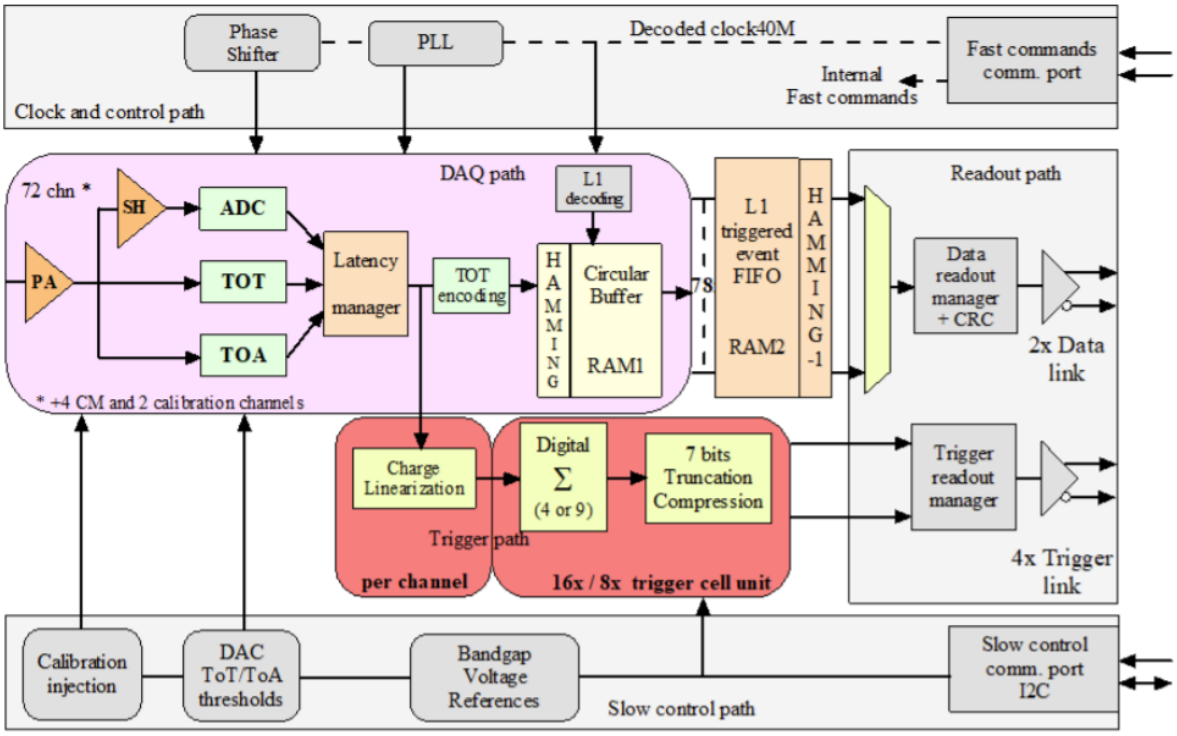


Figure 1.1: The block diagram for the HGCROC3 ASIC

The H2GCROC3 chip is made of: 72 channels of the full analog chain achieving charge and timing information; 4 common mode channels for subtracting coherent noise; 2 calibration channels for the MIP calibration.

Four 1.28 Gbps links are devoted to send out an image of the deposited charge of each bunch crossing event by summing and compressing data over 4 (or 9) channels. These data will contribute to the L1 trigger generation and therefore are processed within the "Trigger Path". The data are sent to the ECON-T concentrator chip.

Two more 1.28 Gbps links are dedicated to sending out the full event information (charge and time) of selected bunch crossings after a L1 trigger request. This path is so referred to as "Data path" or "DAQ path". A 512-deep D-RAM circular memory keeps the entire information (charge and time) for 12.5 us. After a L1A external trigger requesting for the read out of a selected bunch-crossing (BX) event, the data are sent to RAM2 while waiting to be sent to the ECON-D concentrator chip.

The I2C protocol is used to set or read the more than 7900 parameters of the chip. This part is triplicated to resist to the Single Event Effect (SEE). The chip is controlled by the Fast Command block which receives a clock and a command link at 320 MHz. This allows to configure the operating mode of the system: link synchronization, reset, calibration, L1 request, etc. The 40 MHz clock, in phase with the LHC clock, is extracted from the 320 MHz fast command link and provides the clock to the digital part of the ASIC (digital processing, I2C) and to the PLL which generates the others clocks needed to operate the chip: the 640 MHz clock for the 1.28 Gbps links, the phase adjustable 40 MHz clock for the ADCs, the phase adjustable 160 MHz clock for the TDCs. Indeed it is needed to be able to adjust the phase of the conversion blocks (ADC and TDC) to the phase of the physical signals which depend on the rapidity angle.

1.1 Analog front-end

The front-end may be divided in three main sub-parts:

- The very front-end (VFE) part which converts the input charge coming from the SiPM to an output voltage. It must provide the first amplification of the signal with the best noise performance. The VFE stage is made of a current conveyor with variable gain and a charge preamplifier. The conveyor gives the ability to tune the input channel voltage using a DAC channel-wise in order to compensate for SiPM gain variation.
  In the linear part of the preamplifier, the feedback capacitors and feedback resistors provide the gain and the shape of the output signal which is send to the shaper and ADC. From the saturation and above, the feedback discriminator triggers and provides the charge measurement by using the “Time Over Threshold” technique (TOT). Another discriminator allows to give the timing information. This part of the chip is powered at 2.5 V.
- The shaper part is composed of three stages: a Sallen-Key filter, a RC filter and a unity gain amplifier to drive the ADC.
- The two discriminators providing the TOT and TOA (Time of Arrival) pulses, each one sent to a dedicated TDC.

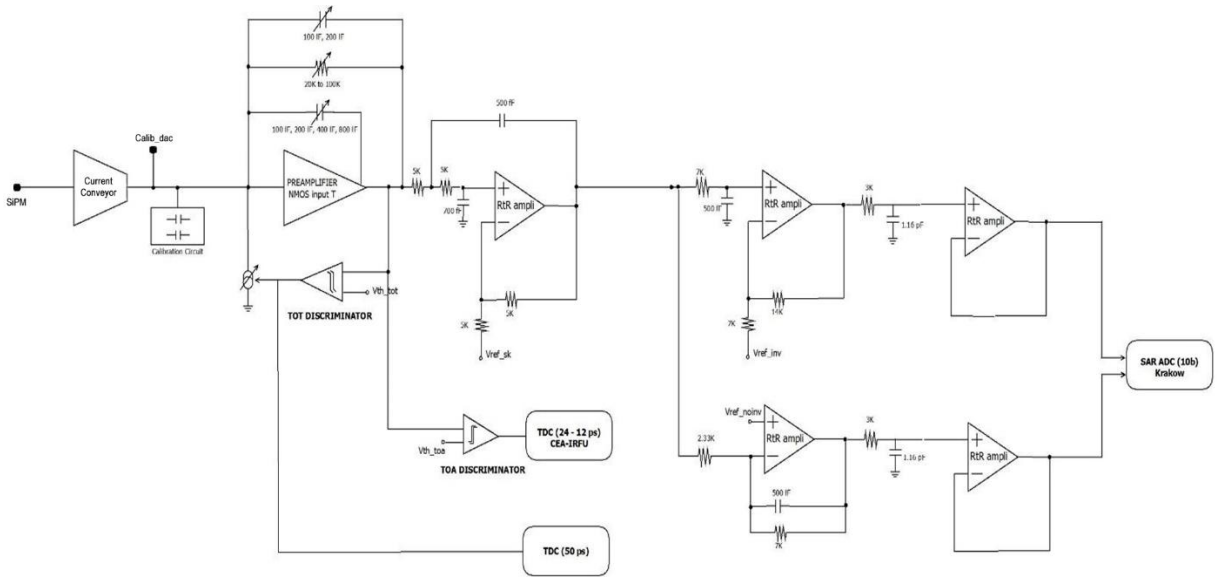


Figure 1.2: Analog channel architecture

The image below shows a layout view of the 4 channels block. The bump pads are placed above either large capacitors or I2C registers so that the area above the sensitive analog parts are kept free of bump pads.

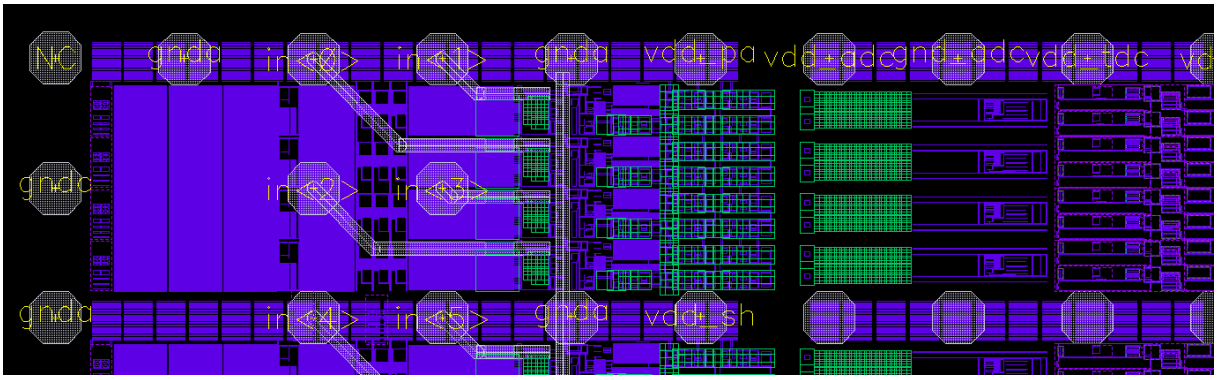


Figure 1.3: Bump bonding pattern

In addition to the 72 readout channels, there are 4 channels for common mode subtraction and 2 channels for MIP calibration. The common mode channels are similar to the regular channels except they stop at the ADC (there are not TDCs). They do not enter the trigger path but are read out in the data path.
In the following sections, more details are given for each block.

1.1.1 Very front-end stage

The current conveyor of the VFE stage attenuates the input signal coming from the SiPM with variable gain and send the signal to the preamplifier input. The current conveyor is divided in four sub-parts (see following image):

- The 'Vdac' input voltage gives the ability to tune the channel input voltage channel-wise in order to adapt the SiPM gain. The input channel voltage is proportional to $V_{gs3} - V_{gs0}$, with V_{gs} values controlled by Ibi and Vdac.
- The 'Ibi' set the current going into the input of the current conveyor half-wise.
- The 'Ibo' compensates the 'Ibi' current considering the current conveyor gain.
- There is one DAC ('dacb') to compensate for the residual current coming from 'Ibi' channel-wise. This DAC can subtract or add current using the 'sign_dacb' signal.

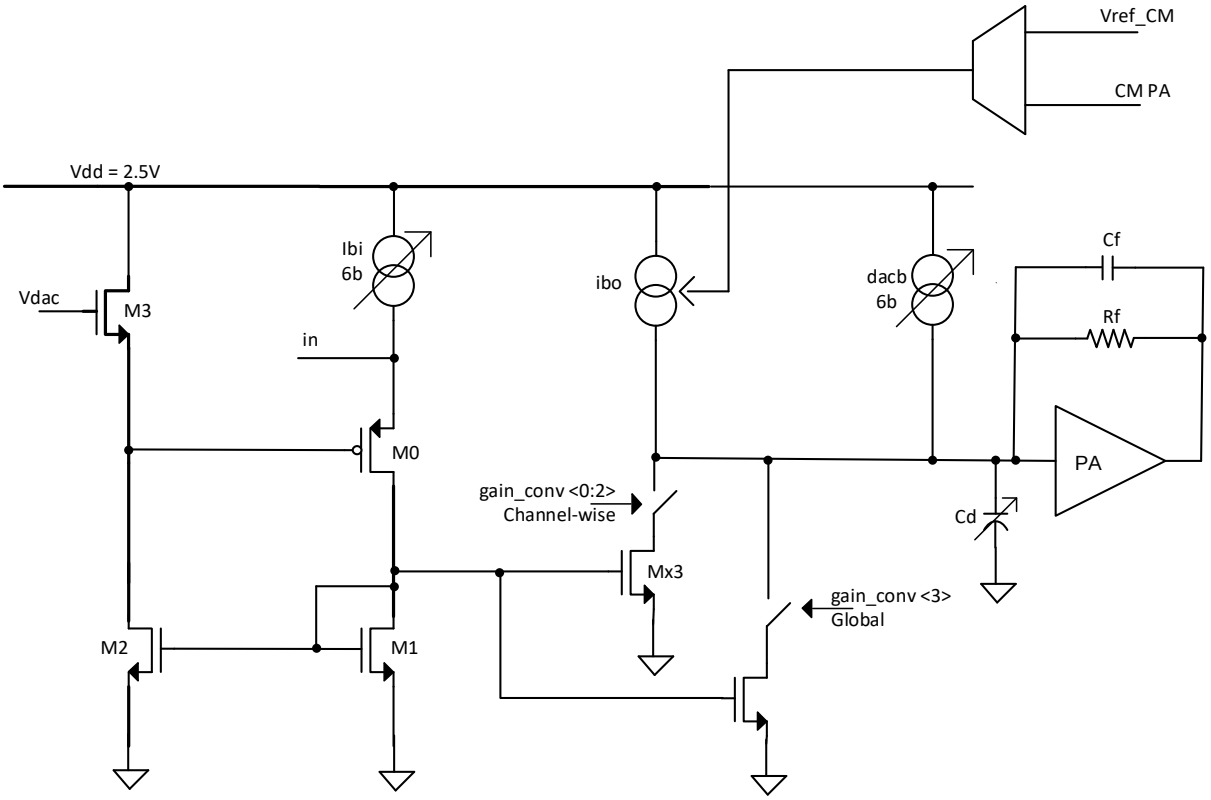


Figure 1.4: Current Conveyor architecture

The preamplifier output voltage is equal to the preamplifier input voltage if there is no residual current flowing through the feedback resistor R_f. In order to optimize the dynamic range at the preamplifier output, the biasing of the current conveyor has to be done so that any residual current is minimized at the conveyor output.

The biasing of the VFE stage can be made with two different method:

- ON_backup=1. This method is as the one used with H2CROC2. Both ibi and ibo currents are generated from current mirrors: ibi can be tuned over 6 bits half-wise in order to adapt the input impedance, ibo is set equal to ibi so that the current does not go to the preamplifier and does not change the preamplifier DC output voltage (R_f*$I_{residual}$). In practice it is not possible to achieve this condition and the dacb DAC is used to compensate for any current eaqual to ibi – ibo.
- ON_backup=0. This new method, implemented in H2GCROC3, generates ibo from an OTA. This OTA forces the output of a CM channel to a desired value (Vref_CM) and adapt the ibo current in consequence whatever the current conveyor gain. Then ibo current is distributed over the entire half in all channels. This method allows to minimize residual current without using the dacb DAC. The dacb DAC will be used only to compensate for leakage current coming from the sensor.

The current conveyor gain is tuneable over 4 bits, getting a gain from 0.025 to 0.375. The most significant bit is tuneable globally and the other 3 bits could be tuned channel-wise. The Vref_CM voltage could be tuned with values 150mV, 200mV or 250mV. On image below there is an example of the preamplifier output after injecting 1pC at the conveyor input for each conveyor gains.

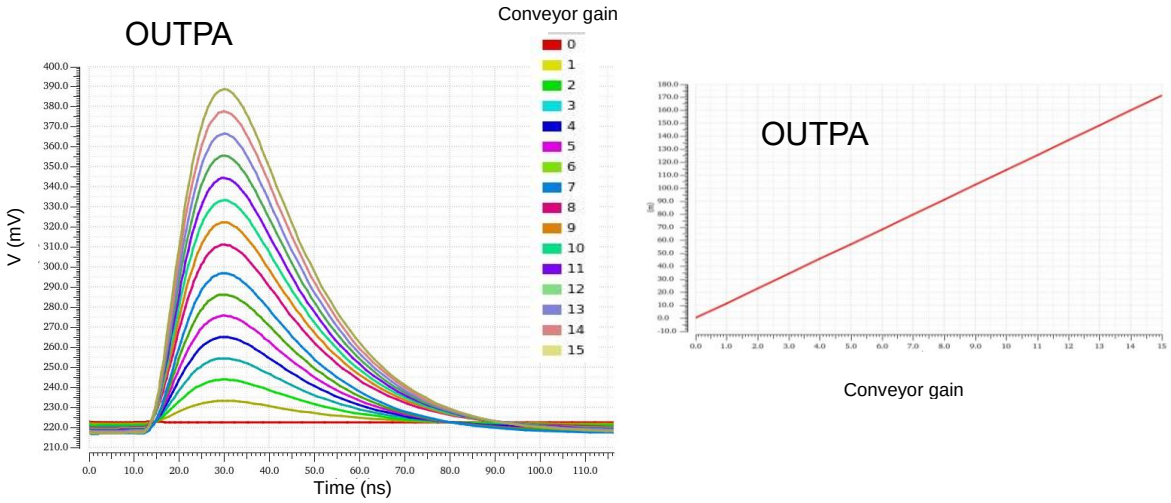


Figure 1.5: OUTPA for different conveyor gain

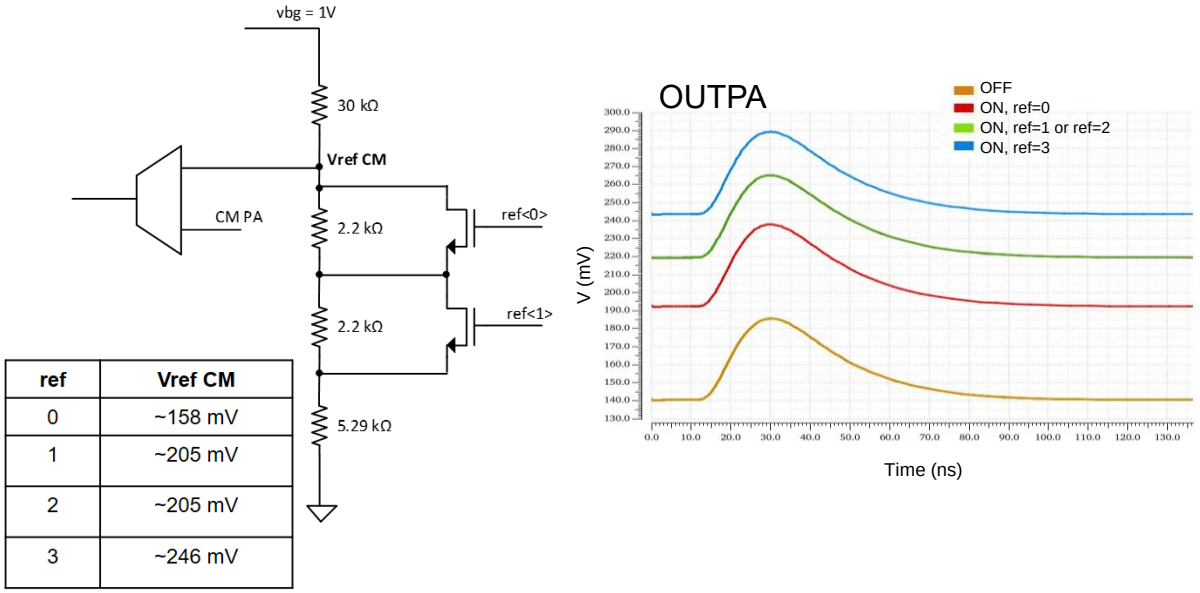
A 6b-DAC is connected channel-wise to the conveyor input to tune the DC level and adjust the SiPM bias voltage.

The preamplifier gain can be adjusted by changing the C_f and C_{f_comp} values. The duration of the signal can be adjusted by changing R_f. In order to adjust the stability at the preamplifier output, C_d can be adapted as well.

Cd (pF)	5, 10, 20	At the conveyor output and at the preamp input. To ensure the preamp stability.
Rf (Ω)	25K, 50K, 66.66K, 100K	In parallel, these resistors provide 15 values to be adjusted with the Cf and Cf_comp values to get the desired decay time constant.
Cf (fF)	50, 100, 200, 400	Combined with the Cf_comp capacitors, provide the gain of the preamplifier.
Cf_comp (fF)	50, 100, 200, 400	Same purpose than Cf capacitors but connected differently to improve the preamplifier stability. From gain point-of-view can be considered in parallel with Cf capacitors.

Table 1.1: Values for Rf, Cf and Cd

As described in the figure below, Vref_CM can be adjusted.



ref	Vref CM
0	~158 mV
1	~205 mV
2	~205 mV
3	~246 mV

Figure 1.10: Compensation with common mode channel

In the table below, all the parameters concerning the preamplifier are described

Preamplifier parameters

Name	# bits	I2C Sub-block / sub-address	Description
Dacb<5:0>	6	Channel-wise	Current dac for preamp DC current compensation.
Sign_Dacb	1	Channel-wise	Sign for Current dac for preamp DC current conmpensation.
Inputdacb<5:0>	6	Channel-wise	Input dac for SiPM Biasing adjustement
Probe_pa	1	Channel-wise	Preamplifier output probe
LowRange	1	Channel-wise	0.5pF injection cap
HighRange	1	Channel-wise	8pF injection cap
Channel_off	1	Channel-wise	"1" = preamplifier input tied to ground
Conv_gain<2:0>	3	Channel-wise	Conveyor Gain: **<0>=0.025, <1>=0.05, <2>=0.1**
Vb_conv<2:0>	3	Global-analog	Conveyor current bias: ibi *bits <5:3> connected to Vdd 2.5V
Vbi_pa<5:0>	6	Global-analog	Preamp current bias
ON_pa	1	Global-analog	"1" = enable preamplifier bias
Sw_super_conv	1	Global-analog	"1" = enable "super conveyor"
Conv_gain<3>	1	Global-analog	Conveyor Gain: **<3>=0.2**
Cd<2:0>	3	Global-analog	Input preamp cap.: **<0> = 5pF, <1> = 10pF, <2>=20pF** In //
Cf<3:0>	4	Global-analog	Preamp feedback cap.: **<0> = 50fF, <1> = 100fF, <2> = 200fF, <3> = 400fF** In //
Cf_comp<3:0>	4	Global-analog	Preamp feedback comp. cap.: **<0> = 50fF, <1> = 100fF, <2> = 200fF, <3> = 400fF** In //

Rf<3:0>	4	Global-analog	Preamp feedback Res.: **<3> = 25K, <2> = 50K, <1> = 66.66K, <0> = 100K** In //
ON_conv	1	Global-analog	"1" = enable current conveyor.
ON_dac_trim	1	Global-analog	"1" = enable dac_trim.
ON_input_dac	1	Global-analog	"1" = enable input dac
Ref_pa_cm<1:0>	2	Global-analog	Configuration of the reference voltage in the compensation common mode channel: **<00> = ~158mV, <01> = ~205mV, <10> = ~205mV, <11> = ~246mV**
ON_backup	1	Global-analog	"1" = enable compensation with common mode channel.
Dacb_dynran_config<1:0>	2	Global-analog	Configuration of dacb dynamic range (See following table)
Calib_dac<11:0>	12	Voltage references	Calibration DAC value
IntCtest	1	Voltage references	Selection of the Calibration DAC
ExtCtest	1	Voltage references	Selection of the external pulse test
Calib_dac_2V5<11:0>	12	Voltage references	Calibration DAC 2.5V value
ExtCtest_2V5	1	Voltage references	Selection of the external pulse test with 2.5V
Choice_cinj	1	Voltage references	
Cmd_120p	1	Voltage references	

Table 1.3: Preamplifier parameters

1.1.2 Input DAC (leakage current compensation)

It is expected an increasing leakage current of the sensor while it is irradiated. This current will actually flow from the preamplifier output to the input through the feedback resistor. As a consequence, the DC level of the preamplifier output follows exactly the leakage current and its dynamic range is so limited by this effect. To withdraw the natural shift of the DC level due to the leakage, the dacb DAC is needed for providing the current compensation at the price of a degraded noise performance. As this DAC is needed to maintain the optimal preamplifier dynamic range (and so the ADC range), it has to be designed so that the noise performance are kept as good as possible.

The leakage compensation ability goes from 270µA to 1.31mA, as shown in Table 3.
The architecture was modified to be able to configure the dynamic range of the dacb. This allows to compensate more leakage current. The dynamic range is configured using 2 bits ('Dacb_dynran_config' on image below).

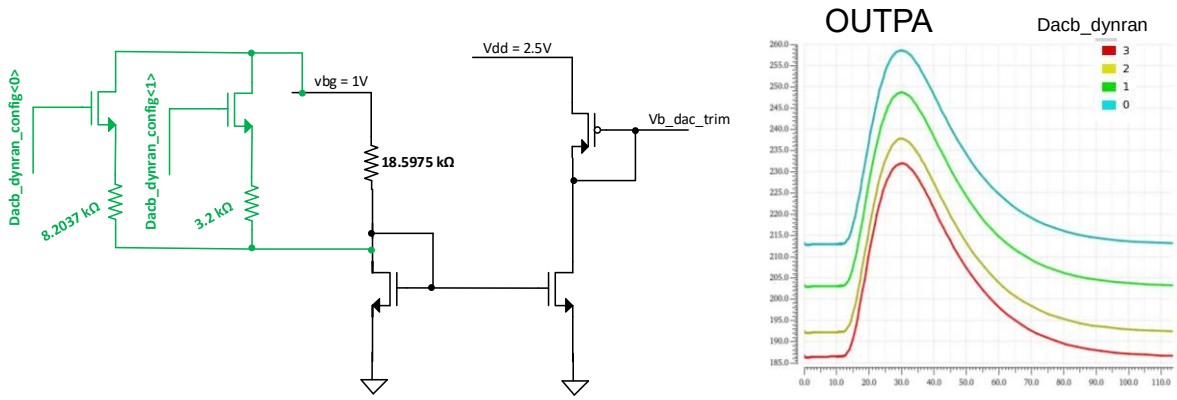


Figure 1.12: Dacb dynamic range configuration

Dacb_dynran_config <1:0>	Leakage current Correction range	Max dacb current	OUTPA pedestal step
0	0 to ~270 µA	~21 µA	~6.45mV
1	0 to ~660 µA	~54 µA	~16.4mV
2	0 to ~1.09 mA	~90 µA	~27.2mV
3	0 to ~1.31 mA	~109 µA	~32.96mV

Table 1.4: Dacb dynamic range configuration

1.1.3 Shapers

The shaper is divided in three stages:

- A Sallen-Key (S-K) shaper
- A gain 1.1 amplifier and a buffer to drive the ADC

The user has to set the Inv_vref and Noinv_vref 10b-DACs to globally set the DC levels of respectively the inverter and non_inverter 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 6b-DAC.

Optimisation of the « ADC range »

A simple circuit has been introduced in order to automatically find out the best combination of both Inv_vref and Noinv_vref 10b DACs. The principle is to force one of the ADC input to 0.6V by setting the corresponding shaper output in high impedance (HZ_inv or HZ_noinv) and applying a 0.6 V to the corresponding ADC input, perform a scan of the Vref DAC of the other branch and set it to the value giving the code 256 (266 in fact to have some margin), and then redo the same operation but for the other branch. By construction, this procedure allows the optimization of both the pedestal and the dynamic range.

With Vref (= ~ 1V) the reference voltage of the ADC, Vdd (=1,2 V) the power supply value, Inv and Noinv the inverted and non-inverted shaper outputs respectively, we have:

$$Adc = \left(1 - \frac{Inv - Noinv}{Vref}\right) * \frac{1024}{2}$$

Shaper parameters

Name	# bits	I2C Sub-block / sub-address	Description
Probe_noinv	1	Channel-wise	Non inverter shaper output probe (“1” = selected)
Probe_inv	1	Channel-wise	Inverter shaper output probe (“1” = selected)
Ref_dac_inv<5:0>	6	Channel-wise	Local 6b-TrimDAC for ADC pedestal tuning
HZ_noinv	1	Channel-wise	x
HZ_inv	1	Channel-wise	x
ON_rtr	1	Global-analog	“1” = enable shaper amplifiers bias
Ibi_sk<1:0>	2	Global-analog	S-K amplifier input stage current
Ibo_sk<5:0>	6	Global-analog	S-K amplifier output stage current
S_sk<2:0>	3	Global-analog	S-K amp Miller cap. <0> = 50fF, <1> = 100fF, <2> = 200fF
Ibi_inv<1:0>	2	Global-analog	Inverter amplifier input stage current
Ibo_inv<5:0>	6	Global-analog	Inverter amplifier output stage current
S_inv<2:0>	3	Global-analog	Inverter amp Miller cap. <0> = 50fF, <1> = 100fF, <2> = 200fF
Ibi_noinv<1:0>	2	Global-analog	Non Inverter amplifier input stage current
Ibo_noinv<5:0>	6	Global-analog	Non Inverter amplifier output stage current
S_noinv<2:0>	3	Global-analog	Non Inverter amp Miller cap. <0> = 50fF, <1> = 100fF, <2> = 200fF
Ibi_inv_buf<1:0>	2	Global-analog	Inverter buffer input stage current
Ibo_inv_buf<5:0>	6	Global-analog	Inverter buffer output stage current
S_inv_buf<2:0>	3	Global-analog	Inverter buffer Miller cap. <0> = 100fF, <1> = 200fF, <2> = 400fF
Ibi_noinv_buf<1:0>	2	Global-analog	Non Inverter buffer input stage current
Ibo_noinv_buf<5:0>	6	Global-analog	Non Inverter buffer output stage current
S_noinv_buf<2:0>	3	Global-analog	Non Inverter buffer Miller cap. <0> = 100fF, <1> = 200fF, <2> = 400fF
Inv_vref<9:0>	10	Voltage references	Inverter shaper global reference
Noinv_vref<9:0>	10	Voltage references	Non Inverter shaper global reference

Table 1.5: Shaper parameters

1.1.4 Discriminators

There are two discriminators per channel, one for the TOT measurement, the other for the TOA. The TOT discriminator is connected on the outpa output of the preamplifier (160 – 200 mV). The TOA discriminator is connected to the out_time_pa output of the preamplifier (~ 600 mV). Two global 10b-DACs allow the user to adjust the thresholds of the discriminators and two local trimming 6b-DACs

allow to reduce the dispersion per channel. The 10b-DAC have typically 1 mV LSB, the trimming 6b-DAC have typically 0.25 mV LSB.

- Toa_Threshold = Toa_vref<9:0> - Trim_dac_toa<4:0>
- Tot_Threshold = Tot_vref<9:0> - Trim_dac_tot<4:0>

There are two external trigger inputs available; typically, in the case when the user wants to calibrate the TOT and TOA, he can send a trigger for the TOA discriminator and the other for the TOT discriminator. He can also send a trigger for a channel, and the other for a neighbouring channel.
The two discriminators' outputs can be masked per channel as well.
The following figures shows 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 others must be masked.

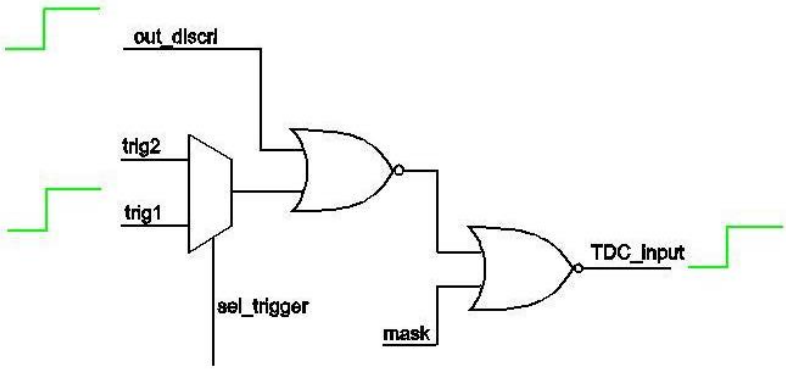


Figure 1.13: Output discriminator configuration

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

Discriminators' parameters

Name	# bits	I2C Sub-block / sub-address	Description
Ref_dac_toa<5:0>	6	Channel-wise	Local 6b-DAC for TOA threshold tuning
Ref_dac_tot<5:0>	6	Channel-wise	Local 6b-DAC for TOT threshold tuning
Mask_toa	1	Channel-wise	TOA discri output mask ("1" = masked)
Sel_trigger_toa	1	Channel-wise	External trigger selection for TOA ("0" = Ext Trig1; "1" = Ext Trig2)
Sel_trigger_tot	1	Channel-wise	External trigger selection for TOT ("0" = Ext Trig1; "1" = Ext Trig2)
Mask_tot	1	Channel-wise	TOT discri output mask ("1" = masked)
Probe_tot	1	Channel-wise	TOT discri output probe
Probe_toa	1	Channel-wise	TOA discri output probe
ON_toa	1	Global-analog	"1" = enable TOA discri bias
ON_tot	1	Global-analog	"1" = enable TOT discri bias
En_hyst_tot	1	Global-analog	"1" = enable the TOT discri hysteresis
Tot_vref<9:0>	10	Voltage references	TOT threshold global value

Toa_vref<9:0>	10	Voltage references	TOA threshold global value

Table 1.5: Discriminator parameters

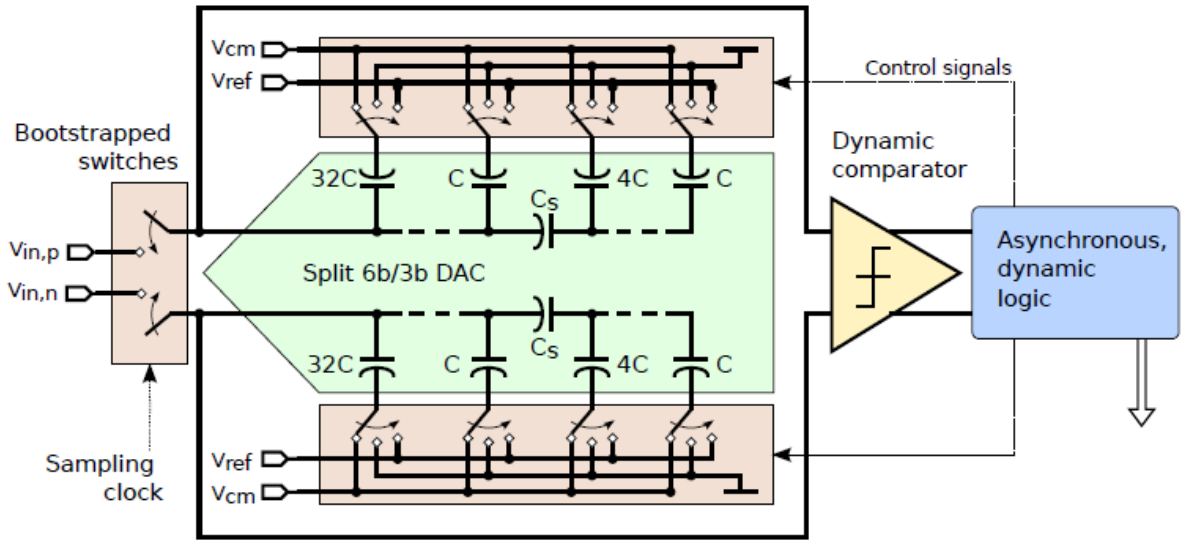
1.1.5 Bias

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.

1.2 Mixed-signal blocks

1.2.1 10 bits ADC

The H2GCROC3 contains a 10b SAR ADC, designed by AGH in Krakow. The ADC's vrefm reference voltage is tied to ground.



The 10 bits data provided by the ADC are sent to an Align Buffer to align them to the TOT and TOA data. A mask can be applied channel-wise to the ADC which stops the clock. Half-wise the asynchronous conversion delays can be adjusted over 3 bits.

ADC parameters

Name	# bits	I2C Sub-block / sub-address	Description
Mask_adc	1	Channel-wise	"1" = ADC clock off
maskAlignBuffer	1	Channel-wise	"1" = AlignBuffer clock off
Adc_pedestal<7:0>	8	Channel-wise	ADC pedestal value (use in Trigger path)
ExtData<9:0>	10	Channel-wise	Forced ADC data (enable ExtData by SelExtADC parameter: bit 3, register 10, "Global Analog"
Clr_ShaperTail	1	Global-analog	Force ADC to 0 for 2 BXs after the TOT to remove the undershoot
SelRisingEdge	1	Global-analog	"1" = AlignBuffer provides data on rising edge
SelExtADC	1	Global-analog	"1" = Forced ADC data send to the DRAM
Clr_ADC	1	Global-analog	Force ADC to 0 when TOT signal @ 1

Ref_adc<1:0>	2	Global-analog	Input stage current of the Ref ADC OTA
Delay40<2:0>	3	Global-analog	Delay tuning for bits <4:0> "000" = faster conversion
Delay65<2:0>	3	Global-analog	Delay tuning for bits <6:5> "000" = faster conversion
Delay87<2:0>	3	Global-analog	Delay tuning for bits <8:7> "000" = faster conversion
Delay9<2:0>	3	Global-analog	Delay tuning for bit <9> "000" = faster conversion
ON_ref_adc	1	Global-analog	"1" = enable ADC ref OTA

Table 1.6: ADC parameters

1.2.2 TOT and TOA TDCs

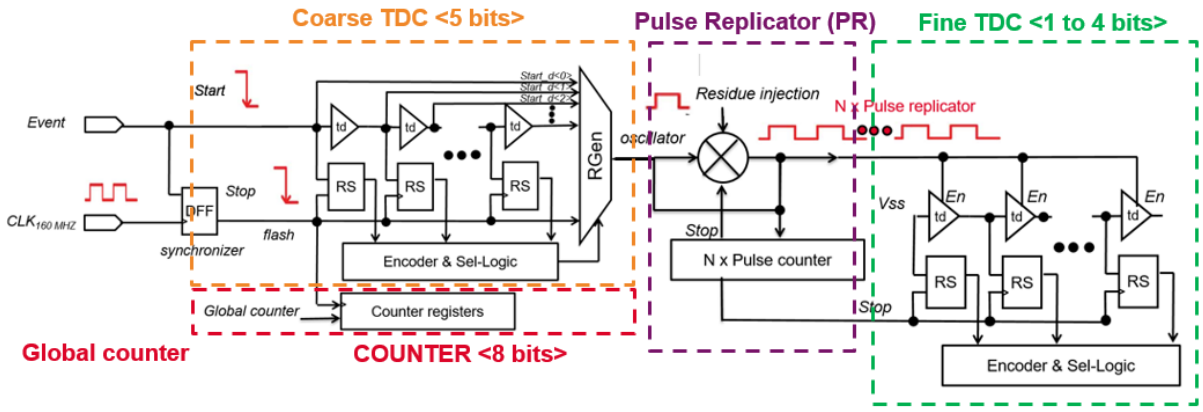
One TDC block handles the TOA and TOT measurements. It was designed by the CEA IRFU group in Saclay. The two following tables give the specifications respectively for the TOA and the TOT.

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

TDC ToT specifications	
Resolution	< 50 ps RMS
Range	12 bits over 2-200 ns
Min time between hits	25 ns
Power consumption	< 2 mW / channel
Fixed latency	12 clock periods
Technology	TSMC 130 nm
Area	Pitch 120 µm
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.



TDCs parameters

Name	# bits	I2C Sub-block / sub-address	Description
DIS_TDC	1	Channel-wise	Disable TDC channel (enable by default)
DAC_CAL_CTDC_TOA <5:0>	6	Channel-wise	Tune the fine gain of the TOA CTDC: 5 bits DAC <4:0> (1kΩ x 32) by the BIAS_CAL_DAC_P and the sign 1 bit <5> VD_P CAL = VD_P + (sign <5> x <4:0> x 1kΩ x BIAS_CAL_DAC_P)
DAC_CAL_FTDC_TOA <5:0>	6	Channel-wise	Tune the fine gain of the TOA FTDC: 5 bits DAC <4:0> (1kΩ x 32) by the BIAS_CAL_DAC_P and the sign 1 bit <5> VD_P CAL = VD_P + (sign <5> x <4:0> x 1kΩ x BIAS_CAL_DAC_P)
DAC_CAL_CTDC_TOT <5:0>	6	Channel-wise	Tune the fine gain of the TOT CTDC: 5 bits DAC <4:0> (1kΩ x 32) by the BIAS_CAL_DAC_P and the sign 1 bit <5> VD_P CAL = VD_P + (sign <5> x <4:0> x 1kΩ x BIAS_CAL_DAC_P)
DAC_CAL_FTDC_TOT <5:0>	6	Channel-wise	Tune the fine gain of the TOA FTDC: 5 bits DAC <4:0> (1kΩ x 32) by the BIAS_CAL_DAC_P and the sign 1 bit <5> VD_P CAL = VD_P + (sign <5> x <4:0> x 1kΩ x BIAS_CAL_DAC_P)
IN_FTDC_ENCODER_TOA<5:0>	6	Channel-wise	Adjust the ToA FTDC offset by 5 bits <4:0> and the sign 1 bit <5> OFFSET = FTDC_TDC<5:0> + (sign <5> x DATA<4:0>)
IN_FTDC_ENCODER_TOT<5:0>	6	Channel-wise	Adjust the ToT FTDC offset by 5 bits <4:0> and the sign 1 bit <5> OFFSET = FTDC_TDC<5:0> + (sign <5> x DATA<4:0>)
GLOBAL_TA_SELECT_GAIN_TOT<3:0>	4	Master TDC	ToT pulse replicator gain : '0001' : 2 pulses (TDC LSB : 97,6 ps) **'0011' : 4 pulses (TDC LSB : 48,8 ps) (Default)** '0111' : 8 pulses (TDC LSB : 24,4 ps) '1111' : 16 pulses (TDC LSB : 12,2 ps)
GLOBAL_TA_SELECT_GAIN_TOA<3:0>	4	Master TDC	ToA pulse replicator gain : '0001' : 2 pulses (TDC LSB : 97,6 ps) '0011' : 4 pulses (TDC LSB : 48,8 ps) **'0111' : 8 pulses (TDC LSB : 24,4 ps) (Default)** '1111' : 16 pulses (TDC LSB : 12,2 ps)
GLOBAL_SEU_TIME_OUT	1	Master TDC	Activate the SEU time-out in the control logic channel
GLOBAL_MODE_FTDC_TOA_S1 & S0 <S1:S0>	2	Master TDC	ToA pulse replicator gain : S0 & S1 for the digital substation ToT – ToA <S0:S1>

			'00' : 2 pulses (TDC LSB : 97,6 ps) '01' : 4 pulses (TDC LSB : 48,8 ps) **'10' : 8 pulses (TDC LSB : 24,4 ps) (Default)** '11' : 16 pulses (TDC LSB : 12,2 ps)
GLOBAL_LATENCY_TIME<3:0>	4	Master TDC	Configure the latency time windows for data output synchronization with 40 MHz CLK '0000': direct 0 s **'1010': 10 CLK latency** '1111': 15 CLK latency
GLOBAL_MODE_NO_TOT_SUB	1	Master TDC	Disable the digital substation ToT – ToA ToT output = ToT without substation
GLOBAL_MODE_TOA_DIRECT_OUTPUT	1	Master TDC	The output of the ToA TDC are directly connected to the ToA & ToT output
GLOBAL_DISABLE_TOT_LIMIT	1	Master TDC	Disable ToT auto limit max to '1' for the ToT test
GLOBAL_FORCE_EN_TOT	1	Master TDC	Force the ToT flag event to '1' (always read ToT)
GLOBAL_FORCE_EN_OUTPUT_DATA	1	Master TDC	Force the En flag event to '1' (always read ToA)
GLOBAL_FORCE_EN_CLK	1	Master TDC	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)
EN_REF_BG	1	Master TDC	Activate the BANDGAP source 1V to the calibration part '0' : External PAD 'EXT_REF_TDC' **'1' : Internal 1V Bandgap 'BG_1V'**
CALIB_CHANNEL_DLL	1	Master TDC	Activate the automatic calibration of all TDC channel
START_COUNTER	1	Master TDC	Activate the global gray counter for the TDC channel (rising edge for the 40 MHz 160 MHz synchronization)
INV_FRONT_40MHZ	1	Master TDC	Inverse the 40 MHz front for the gray counter and the SEU_TIME_OUT
BIAS_FOLLOWER_CAL_P_CTDC_EN	1	Master TDC	Enable the VTC follower for the channel CTDC calibration follower bias P
BIAS_FOLLOWER_CAL_P_CTDC_D<3:0>	4	Master TDC	Configure the VTC follower for the channel CTDC calibration follower bias P current : Current = DATA<3:0> Weight addition **'base' : 50 nA** <0>: 100 nA <1>: 500 nA <2>: 2,5 µA <3>: 5 µA
BIAS_FOLLOWER_CAL_P_FTDC_EN	1	Master TDC	Enable the VTC follower for the channel FTDC calibration follower bias P
BIAS_FOLLOWER_CAL_P_FTDC_D<3:0>	4	Master TDC	Configure the VTC follower for the channel FTDC calibration follower bias P current : Current = DATA<3:0> Weight addition **'base' : 50 nA** <0>: 100 nA <1>: 500 nA <2>: 2,5 µA <3>: 5 µA
EN_MASTER_CTDC_DLL	1	Master TDC	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)**
EN_MASTER_CTDC_VOUT_INIT	1	Master TDC	Activate the voltage load in "VD_CTDC_P" of the master CTDC DLL pump charge output :

			'0' : High Z pump charge (Default) '1' : CTDC_P INIT voltage load « VD_CTDC_P_DAC MODE»
VD_CTDC_P_DAC_EN	1	Master TDC	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>"
VD_CTDC_P_D<4:0>	5	Master TDC	5 bits DAC for the VD_CTDC_P voltage : '00000' = 0,2 V '11111' = 0,75 V
CTDC_CALIB_FREQENCY<0:5>	6	Master TDC	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
FOLLOWER_CTDC_EN	1	Master TDC	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>**
BIAS_I_CTDC_D<0:5>	6	Master TDC	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
VD_CTDC_N_DAC_EN	1	Master TDC	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>"
VD_CTDC_N_D<0:4>	5	Master TDC	5 bits DAC for the VD_CTDC_N voltage : '00000' = 0,6 V '11111' = 1 V
VD_CTDC_N_FORCE_MAX	1	Master TDC	**Force the VD_CTDC_N voltage to 1,2 V**
GLOBAL_EN_BUFFER_CTDC	1	Master TDC	Enable the BUFFER of the calibration CTDC channel **'0' : OFF** '1' : 'VTC 6 bits "BIAS_CAL_DAC_CTDC_P_D<0:3>'
BIAS_CAL_DAC_CTDC_P_EN	1	Master TDC	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>'
BIAS_CAL_DAC_CTDC_P_D<0:3>	4	Master TDC	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
GLOBAL_INIT_DAC_B_CTDC	1	Master TDC	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
CTRL_IN_SIG_CTDC_P_EN	1	Master TDC	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>"
CTRL_IN_SIG_CTDC_P_D<0:4>	5	Master TDC	5 bits DAC for the CTRL_SIG_CTDC calibration voltage : '00000' = 0,2 V '11111' = 0,8 V
CTRL_IN_REF_CTDC_P_EN	1	Master TDC	Activate the DAC for CTRL_REF_CTDC calibration voltage : **'0' : DAC OFF 0 V output** '1' : 5 bits DAC "CTRL_IN_REF_CTDC_P_D<0:4>"
CTRL_IN_REF_CTDC_P_D<0:4>	5	Master TDC	DAC 5 bits for the CTRL_REF_CTDC calibration voltage : '00000' = 0,2 V '11111' = 0,8 V
EN_MASTER_FTDC_DLL	1	Master TDC	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)**
EN_MASTER_FTDC_VOUT_INIT	1	Master TDC	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»
VD_FTDC_P_DAC_EN	1	Master TDC	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>"
VD_FTDC_P_D<0:4>	5	Master TDC	5 bits DAC for the VD_FTDC_P voltage : '00000' = 0,4 V '11111' = 0,9 V
FTDC_CALIB_FREQENCY<0:5>	6	Master TDC	FTDC 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
FOLLOWER_FTDC_EN	1	Master TDC	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>**
BIAS_I_FTDC_D<0:5>	6	Master TDC	Configure the VTC the bias FTDC BIAS_N_FTDC 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
VD_FTDC_N_DAC_EN	1	Master TDC	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>"

VD_FTDC_N_D<0:4>	5	Master TDC	5 bits DAC for the VD_FTDC_N voltage : '00000' = 0,4 V '11111' = 0,9 V
VD_FTDC_N_FORCE_MAX	1	Master TDC	**Force the VD_FTDC_N voltage to 1,2 V**
GLOBAL_EN_BUFFER_FTDC	1	Master TDC	Enable the BUFFER of the calibration FTDC channel **'0' : OFF** '1' : 'VTC 6 bits "BIAS_CAL_DAC_FTDC_P_D<0:3>'
CTRL_IN_SIG_FTDC_P_EN	1	Master TDC	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>"
CTRL_IN_SIG_FTDC_P_D<0:4>	5	Master TDC	5 bits DAC for the CTRL_SIG_FTDC calibration voltage : '00000' = 0,2 V '11111' = 0,8 V
CTRL_IN_REF_FTDC_P_EN	1	Master TDC	Activate the DAC for CTRL_REF_FTDC calibration voltage : **'0' : DAC OFF 0 V output** '1' : 5 bits DAC "CTRL_IN_REF_FTDC_P_D<0:4>"
CTRL_IN_REF_FTDC_P_D<0:4>	5	Master TDC	DAC 5 bits for the CTRL_REF_FTDC calibration voltage : '00000' = 0,2 V '11111' = 0,8 V
GLOBAL_EN_TOT_PRIORITY	1	Master TDC	
GLOBAL_EN_TUNE_GAIN_DAC	1	Master TDC	
Sel_clk_reg<1:0>	2	Master TDC	

Table 1.7 : Master TDC parameters

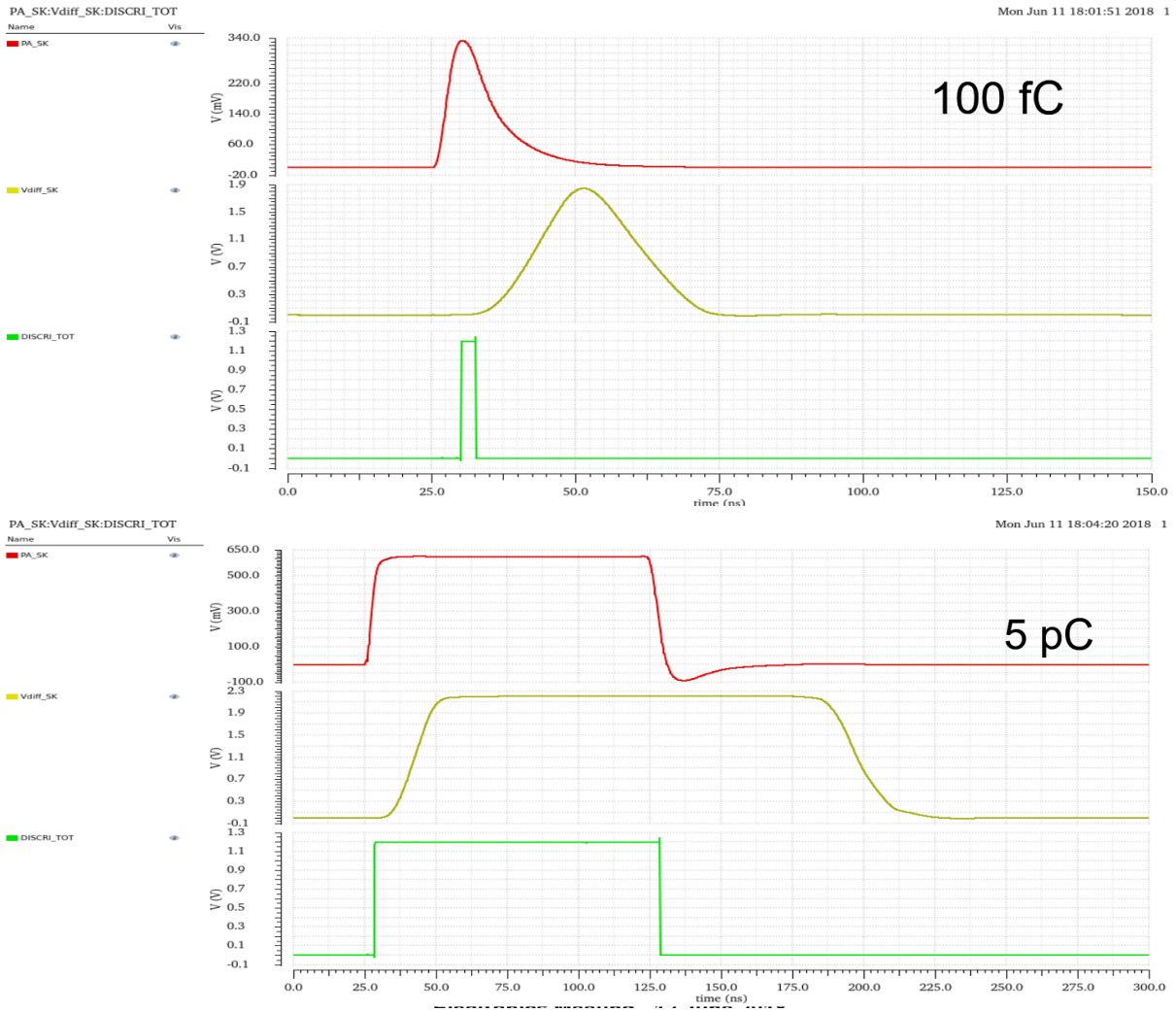
1.2.2.1 The Time of Arrival (ToA) TDC

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 HGROC 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.

The ToA information does not participate to the L1 trigger process. In principle, it can be available immediately after the conversion is finished. However, in the proposed architecture, it is required for the ToT measurement. To simplify overall design it is stored in the on-chip RAM memory when the ToT measurement is over.

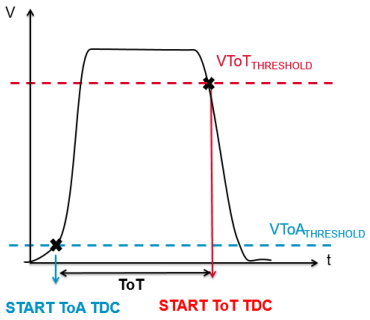
1.2.2.2 The Time over Threshold (ToT) TDC

To measure ToT with a TDC, the analog input signal is compared with a programmable threshold (higher than the one used for the TOA) by the on-chip FEE. The pulse width of the resulting digital signal carries information about the deposited charge in the channel. The two following figures illustrate the ToT pulses respectively for 100 fC and 5 pC input charges.



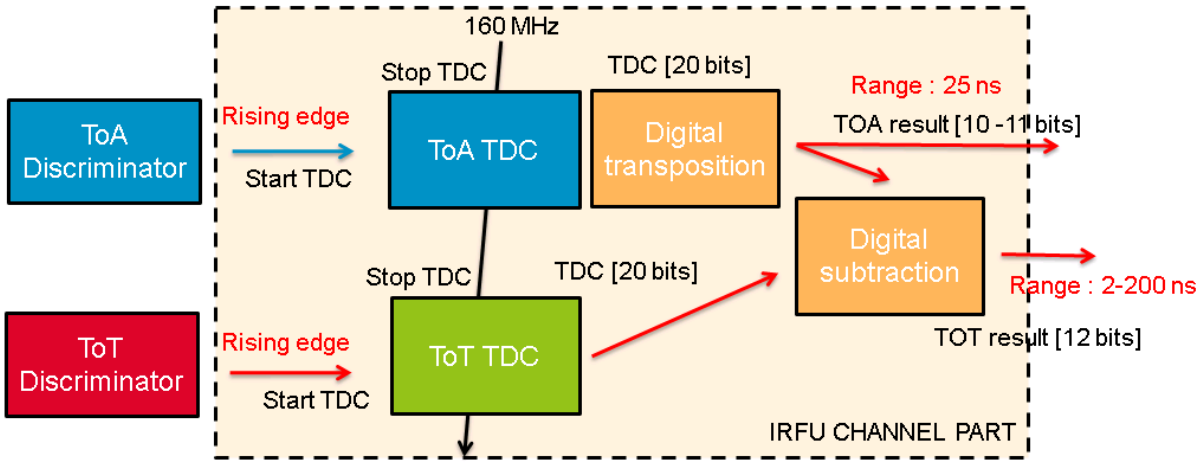
The shortest ToT interval to convert is about 2 ns. With an input charge up to 10 pC the duration of the longest ToT pulse is 200 ns. The conversion time of the ToA TDC is an order of magnitude longer than the specified 2 ns. Therefore, the same TDC cannot be used to convert arrival times of the leading and trailing edges of the ToT pulse.

The proposed solution for the ToT is to use the ToA measurement to mark the ToT rising edge and to use a dedicated TDC to measure the time of the falling edge of the ToT signal. The difference between the two times will give the ToT duration. The following figure shows the proposed architecture for the ToA and ToT measurements:



Clearly, this architecture can be used only for a step rising edges of the input analog signals for which the time difference between the crossing points of the ToA and ToT thresholds can be neglected. According to the frontend simulations, this is the case for the typical charge deposits that generate ToT signals.

The rising edge of the ToA discriminator initiates ToA measurements. The second TDC, named ToT TDC, is activated only when the ToT discriminator detects the falling edge of the analog signal. Upon an active high signal from the discriminator, it measures the time of the falling edge of the signal delivered by the ToT discriminator.



The ToA value is obtained directly from the ToA TDC measurement after a digital transposition in 10 or 11 bits. The ToT value is obtained from the digital subtraction of the ToT TDC measurement from the ToA TDC measurement. Eventual overflows are detected and marked with all bits set in the ToT results.

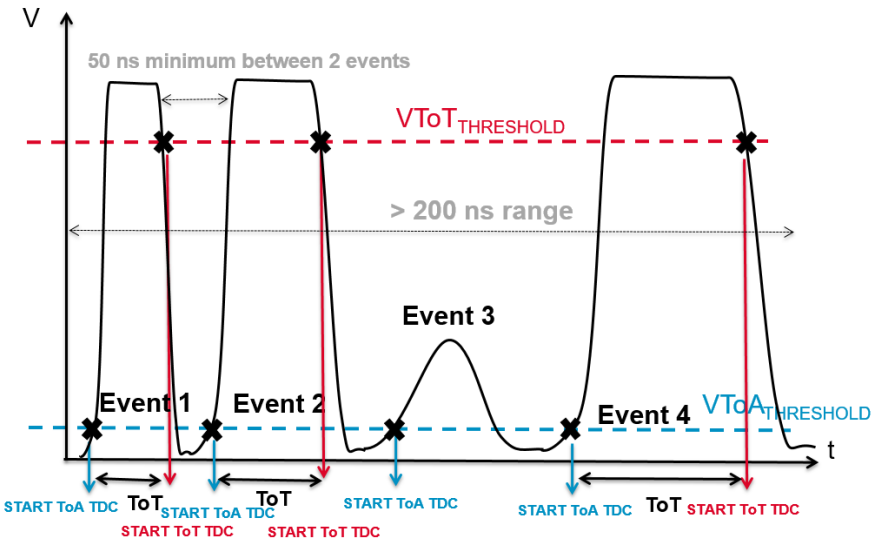
The ToT result is valid during one 40 MHz clock period and is written to the on-chip RAM. For non-ToT events, the RAM is filled with the all-zero code.

1.2.2.3 Synchronization

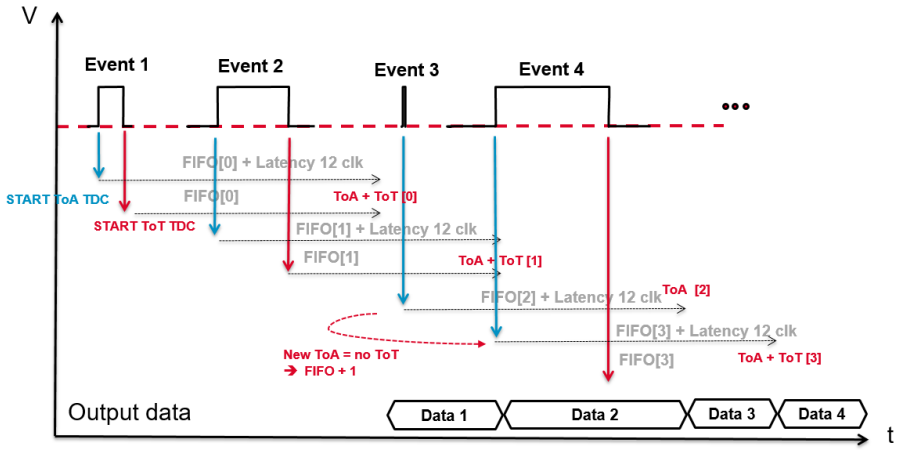
The ToT measurement participates in the L1 trigger process along with the signal amplitude measurement via the channel ADC. The ADC values are available immediately after conversion, as are those of the ToA TDC. However, as described above, the ToT interval may vary from 2 to 200 ns. The on-chip Trigger unit requires the ADC and the ToT measurements to be present simultaneously at the same rising edge of the on-chip PLL 40 MHz clock. To achieve the synchronization, the ToT logic guaranties that the ToT measurement will be available after a fixed number of clock periods starting from the ToA discriminator rising edge. With today's assessment of the ToT logic performance, the fixed ToT latency is determined to be **200 ns + 100 ns corresponding to 12 (8+4) clock periods**.

1.2.2.4 Pileup effect

Taking into account the discharge time of the analog part, the minimal time between two consecutive ToT events can be as low as 50 ns. During the 300 ns ToT latency, several other charge deposits may occur within the same channel. The design guarantees that the ToT will be measured for all of them.



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 12 40 MHz clock cycles. This creates a so-called ToA latency window to wait for a possible ToT discriminator signal, to measure its arrival time and to store it into the corresponding ToT FIFO. Note that this latency can be tuned by SC (0 to 15 clock cycles).



At the end of the 12-cycle count, the ToT is obtained reading the ToA and ToT FIFOs and subtracting one from another. Per-channel subtraction logic is based on asynchronous gates that accomplish the operation within a 5-ns delay. After that, the ToA and ToT results are sent to the on-chip RAM where they are associated with the channel ADC measurement.

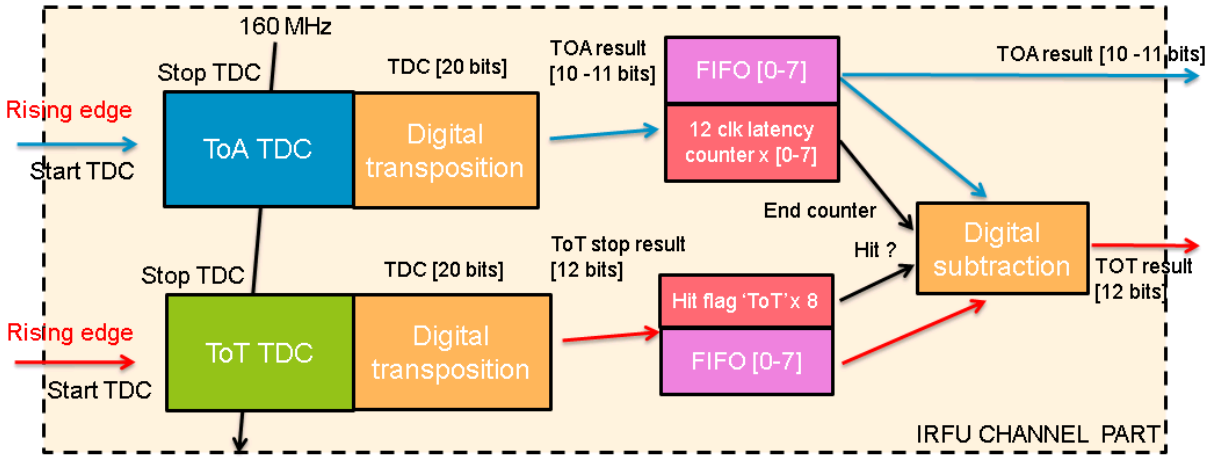
If there is no ToT hit during the ToA latency window, the ToT measurement logic is not activated and the ToA data is accompanied by a “0” code to indicate the absence of the ToT information.

If a new ToA/ToT 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.

If, within an active ToA[n-1] latency window, a ToA[n-1] hit is followed by another ToA[n] hit, rather than the ToT hit , this means that the charge deposit within the event [n-1] was not high enough to trigger the ToT discriminator. The “0” ToT code is appended to the ToT FIFO and the new ToA[n] latency window is started.

The ToA and ToT FIFO depth is determined by the fixed ToT latency of 300 ns and the 25 ns minimum time between the ToA: 300 ns / 25 ns = 12. Ideally, the FIFOs need to store 12 ToA and ToT measurements and there should be 12 counters in the pool. However, with the expected channel occupancies, a long train of consecutive events has sufficiently low probability to reduce the FIFO depth. For simplification, a power of two logic is used and the 8-entry deep FIFOs implemented together with 8 counters. In case of overflows, events are lost without blocking of the ToA /ToT TDC circuitry.

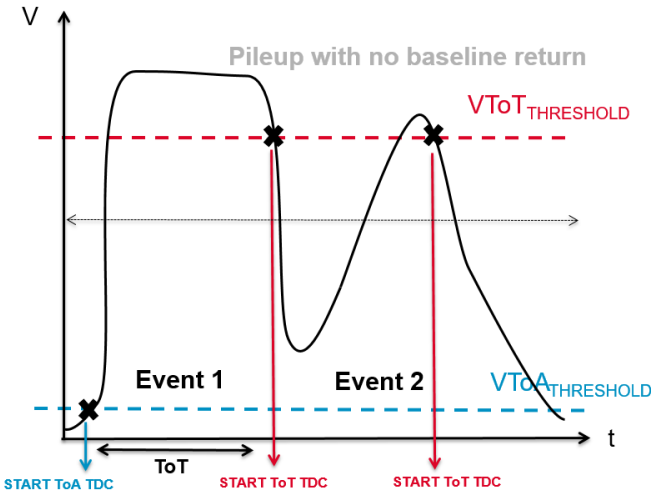
The block diagram of the latency counter and FIFO architecture for the ToA and the ToT measurements is presented in the next figure:



In addition, a data ToA/ToT debug mode is implemented. It allows reading of the 20-bit "raw" (non-transposed) data of the ToA and ToT TDCs.

1.2.2.5 Two consecutives ToT events:

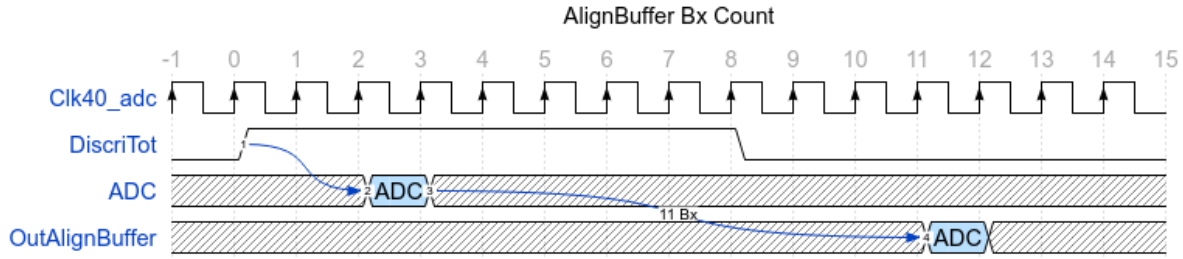
In case of high rate pileup, the preamplifier output signal may not return to its baseline after a ToT event when a new charge arrives. In this case, a second ToT may appear without the ToA threshold being crossed as shown on the next figure.



In the current implementation the second ToT event, which does not have corresponding to its ToA event, will be ignored.

1.2.3 *Align Buffer*

The 10 bits data provided by the ADC are sent to an Align Buffer to align them to the TOT and TOA data. Indeed, the TDCs providing the TOT measurement introduce a latency due to the duration of the TOT itself (this latency is tuneable in the TDCs). The ADC + Align Buffer have a fixed latency of 11 bunch crossings: sampling at BC=1, ADC data available at BC=2, data at the Align Buffer outputs at BC=11 (see next chronogram).



The AlignBuffer block provides:

- The ADC raw data for the DAQ path
- The Tot_in_progress flag for the DAQ path. It is equal to 1 as long as the analog shaper output is corrupted by a TOT event: saturation and undershoot.
- The pedestal subtracted ADC data for the Trigger path. The pedestal is adjustable channel-wise.

The user can choose to force ADC data to 0 when the TOT pulse is at 1 (Clr_ADC="1"), otherwise it provides the actual ADC values.

As after a TOT the shaper returns to the pedestal after a given time, the user can also choose to force ADC data to 0 for two next bunch crossing after the end of the TOT pulse (Clr_ShaperTail="1").

A mask is also available channel-wise to stop the clock and so the digital activity of the AlignBuffer.

The edges of the "analog" 40 MHz clock can be selected and so the phase to send AlignBuffer data to the RAM1.

External ADC data can be forced, half-wise, configurable by slow-control, this mode is in particular interest for the DRAM tests.

The TDC block provides also the aligned TOA and TOT data with an internal align buffer.

1.3 Digital blocks

The main specifications of the digital part of the ASIC are described in the table below.

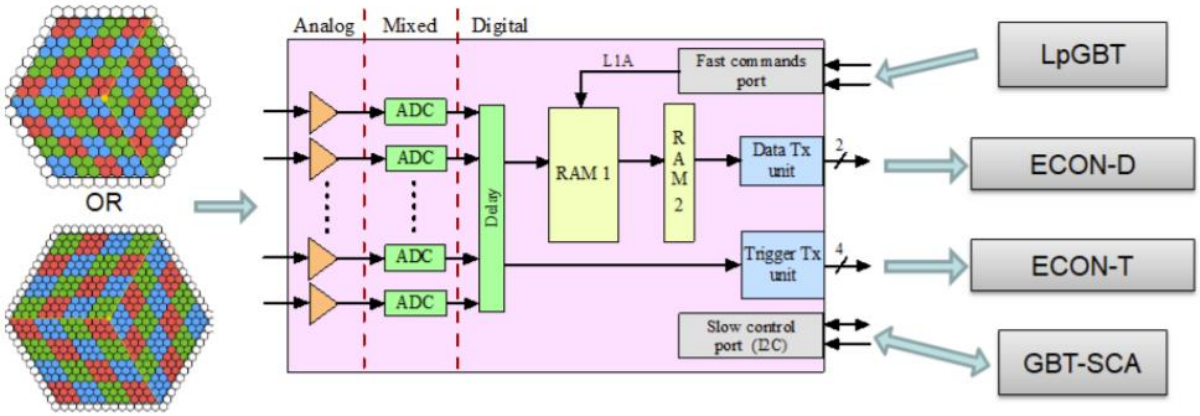
Analog channels	1. 72 analog "normal" channels 2. 4 for common mode channels 3. 2 for calibration
Acquisition mode	Continuous @ Bx rate (40 MHz)
L1 functionality	With derandomizer
L1-Trigger rate	1 MHz (max 32 consecutive)

DAQ readout mode	• Triggered by L1A • Normal or Tests modes
Trigger data type	7-bit sums (4 or 9 channels)
DAQ data type	32 bits / channels (40 words/frame)
Trigger output port	4 x CLPS @ 1280 Mbps (2 used for 9ch sums)
DAQ output port	2 x CLPS @ 1280 MBps

Table 1.8 : Main specifications of digital part

HGCROC3 integrates 72 channels to readout

- 192 channels sensor with a 64-ch configuration
- 432 channels sensor with a 72-ch configuration

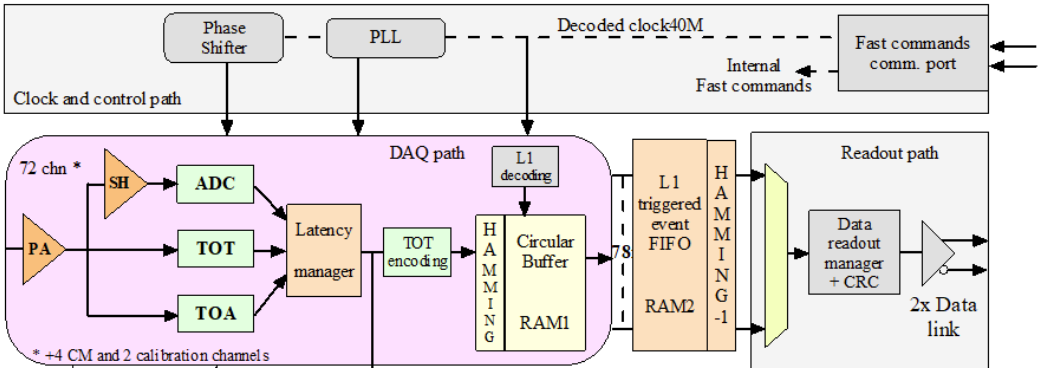


1.3.1 Data path

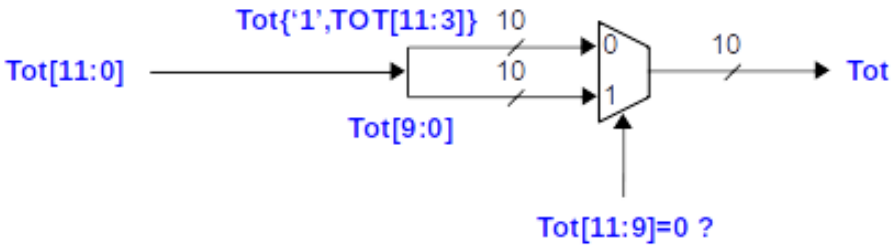
All the data coming from the ADC and the two TDCs are continuously memorized into a circular memory based on a DRAM architecture. The circular memory is 512 length and only L1-triggered data are read out.

Common mode channels provide only the ADC data.

The figure below describes the functionality of the DAQ path. The Latency manager represents both the AlignBuffer and a part of the TDC digital processing: at its output, ADC, TOA and TOT are aligned and sent on the DRAM with the phase-adjustable 40 MHz clock.



The ADC and the TOA data are over 10 bits each and the TOT is natively in 12 bits. A first step of the digital processing is to compress the 12b TOT to 10b TOT. The figure below shows the algorithm of the compression.



For each channel, the 30 bits of the Data path content is populated in the RAM1 with values shown in the next table. Three digitized values are selected depending on the crossed threshold for the charge.

	ADC (BX-1)	ADC (BX)	TOT (BX)	TOA (BX)	Charge collection	Data type
1	x	x		x (=0)	Q < TOA_thr AN	Normal
2	x	x		x	Q < TOT_thr AN	Normal
3	x		x	x	Q > TOT_thr AN	Normal
4		x	x	x		"Characterization"

The "Characterization" mode is for debug purpose, in this mode the RAM1 is filled with ADC, TOA and TOT data of the same event.

Two flags are added (MSB positions) to the 30 bits in order to remove some ambiguities which can occurs in the data path:

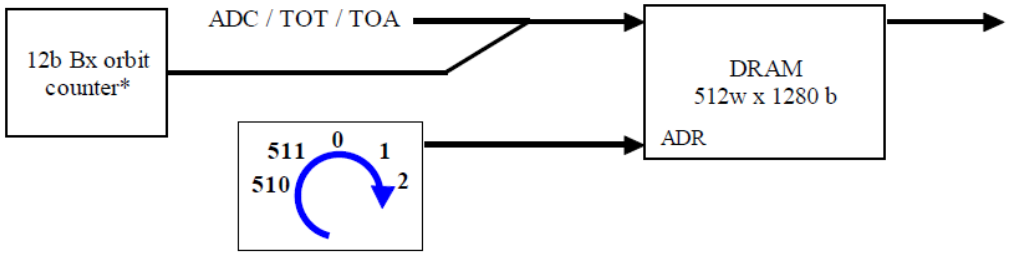
1. TOT-In-Progress, Tp : A TOT occurred in a previous BX and the ADC value can be "corrupted" (saturation or undershoot)
2. TOT-Complete, Tc : the second 10 bits packet corresponds to TOT, not ADC.

Note : The ADC values in the data path are without pedestal subtraction.

Tc and Tp Interpretation:

- 0b00 : The TOT is not in operation (not busy), normal behaviour with ADC data
- 0b01 : The TOT is busy (integration or undershoot), Tp highlights the fact that provided ADC correspond to saturation or undershoot
- 0b10 : must not appear => we only output value when TOT is busy
- 0b11 : The TOT value is outputted, normal behaviour with TOT data (ADC value is between saturation and undershoot)

All these data are stored at 40 MHz continuously into the DRAM after having been encoded with the Hamming encoding. The DRAM memory is controlled as a circular memory. After receiving a L1 trigger, a readout of the selected BX data is activated (the offset of the L1 trigger is configurable by slow control).



For the DAQ path, the ASIC integrates 2 memories. The first one, RAM1, handles the storage of the data until the ASIC receives a L1A command. The second one, RAM2, is managed like a FIFO memory and stores the data extracted from RAM1 by the L1A command.

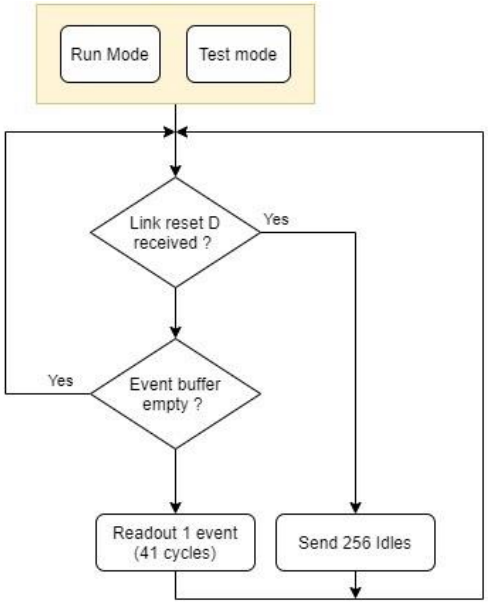
When the chip receives this L1A command, the data of the event which occurred at a programmed number of bunch-crossing earlier (value set by slow control: typically 12.5 µs) is copied into the RAM2 with the value of the time tag and event counters. Then, the latter one (EC) is incremented.

As the internal architecture is pipelined, the system is able to handle consecutive L1A.

As long as RAM2 is not empty, the chip sends out the data else it sends out the IDLE pattern. The RAM2 is handled like a FIFO circular buffer.

As described below, when the chip receives a Link-Reset-ROC-D fast command, it starts the link reset procedure. If an event readout is in progress, the link reset procedure is delayed after the end of the event readout.

If another Link-Reset-ROC-D command is received during the execution of a previous one, it’s skipped.

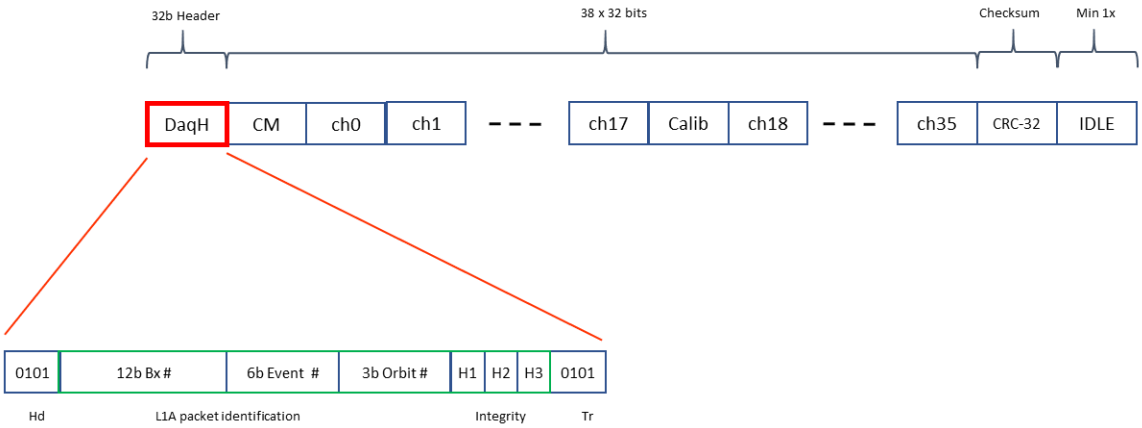


The DAQ frame

# DAQ link	2 (no sharing, 1 per side)
Idle word	4b header + 28 bits configurable by SC
# 32b-word send by single L1	39 + 1 CRC-32
# of Idle packet after each « frame »	1
Bits order	MSB first
Word order / half chip	1w with header / counters 1w with 2xCM 18w with channels 0 to 17 1w Calib 18w with channels 18 to 35 1w CRC-32
Header	0101 + BxCounter + EventCounter + OrbitCounter + HammingErrors + 0101
Checksum	CRC-32bits
Link type / speed	CLPS @ 1280 Mbps

The table above summarizes the data content of the DAQ frame.

An idle packet is continuously sent out when no L1 trigger is activated. This Idle packet is configurable by slow control. The data format is described in the figure below.



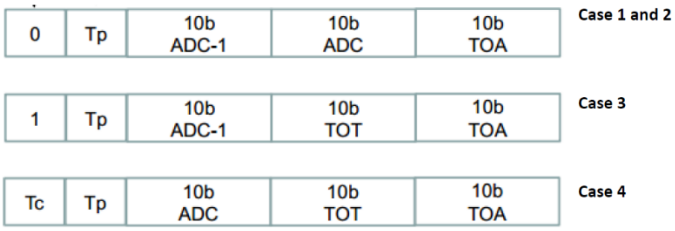
The first 32 bit word is the header composed as per below:

- Bx#: Value of the Bunch Crossing Counter (BC) on 12 bits.
- Event#: Value of the Event Counter (EC) on 6 bits, to detect if a L1 trigger has been sent but related data wasn't saved (RAM2 full)
- Orbit#: Value of the Orbit Counter (OB) on 3 bits.
- H1: Error during hamming decoding in Header.
- H2: Error during hamming decoding in channels from CM to CH17 (1st Quarter).
- H3: Error during hamming decoding in channels from Calib to CH35 (2nd Quarter).

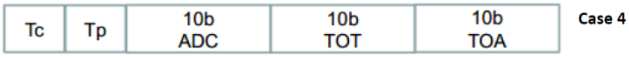
The common mode channels' data content is as per below:

10	10b " 00...00 "	Adc CM0	Adc CM1

The normal channels data content are as per below :



The calibration channels data content are always in case 4:



The packet integrity is checked by the CRC (Cyclic Redundancy Check) on 32 bits data width.

This module takes 39 packets of data in input (Header, CM, Chn0, …, Chn17, Calib, Chn18, …, Chn35) and sends the CRC packet at the end of the frame (packet 40) to the serializer.

The polynomial to apply is 0x04C11DB7:

$$X^{32}+X^{26}+X^{23}+X^{22}+X^{16}+X^{12}+X^{11}+X^{10}+X^{8}+X^{7}+X^{5}+X^{4}+X^{2}+X+1$$

For instance, if the first three words are 0x56370C85, 0x000ABCDE, 0xA30C2BEC, the CRC code at the end of the third word will be 0x2CCB229A, following the code below:

```
Entrée [3]: vals = ['56','37','0c','85','00','0a','bc','de','a3','0c','2b','ec']
            crc = crcmod.mkCrcFun(0x104c11db7, initCrc=0,xorOut=0, rev=False)
            ## initial Value
            crc_val = 0x0
            for i, val in enumerate(vals):
                crc_val = crc(codecs.decode(val,'hex'), crc_val)
                if ((i+1)%4) == 0:
                    print(f'{i=}: {val=} {hex(crc_val)=}')

          i=3: val='85' hex(crc_val)='0x74d92f51'
          i=7: val='de' hex(crc_val)='0x84f0a9b0'
          i=11: val='ec' hex(crc_val)='0x2ccb229a'
```

The word 40 is made of the CRC code obtained after the 39th word.

One IDLE is attached at the end of the frame.

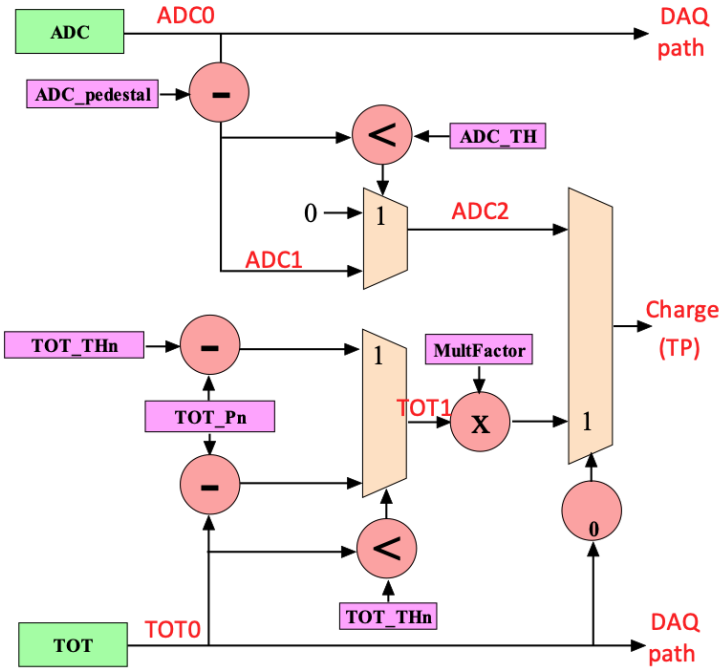
1.3.2 Trigger Path

Following the AlignBuffer block, the 2 pieces of channel information (ADC and TOT) are fed into the trigger path. There follows a Charge Linearization block and a summing block to create Trigger Sum cells.

The data processing for the trigger path is composed as per below:

- Charge linearization over ADC/TOT range.
- Sum of 4 or 9 channels depending on the sensor.
- Charge compression to fit the bandwidth.

The Charge linearization block treats the ADC and TOT charge in different ways. The two paths are shown below. It is a scheme of principle as actually the pedestal subtraction is made in the AlignBuffer.



Referring to figure above, the ADC charge processing has 2 main steps.

Step	Process name	Parameter	Action

1	Pedestal subtraction	ADC_pedestal (8 bits per channel)	If ADC0 > (ADC_pedestal + ADC_TH) then
			ADC2 = ADC0 - ADC_pedestal
			Else
			ADC2 = 0

2	Noise cancellation	ADC_TH (5 bits global per ROC)	

The constants are to be measured during a calibration phase and loaded via slow control. ADC_Pedestal is a dc offset per channel and ADC_TH is used for a noise cut.

The TOT charge processing and selection has 4 main steps.

Step	Process name	Parameter	Actions

1	TOT pedestal subtraction	TOT_Pn (7 bits per group of 9 channels)	If TOT0 > TOT_THn then
			TOT1 = TOT0 - TOT_Pn
			Else
			TOT1 = TOT_THn - TOT_Pn

2	Plateau or linear selection	TOT_THn (8 bits per group of 9 channels)	
3	Convert time to charge (ADC units)	Multifactor (5 bits global, default value = 25)	

4	Selection between ADC and TOT		If TOT0 ≠ 0 then
			Charge = TOT1 x Multfactor
			Else
			Charge = ADC2

TOT_pedestal, TOT_threshold and Multiplication factor are parameters to be measured during a calibration phase and loaded via slow control.

TOT0 is a 12 bit value obtained from the TOT block.

TOT pedestal, defines the TOT linear fit offset.

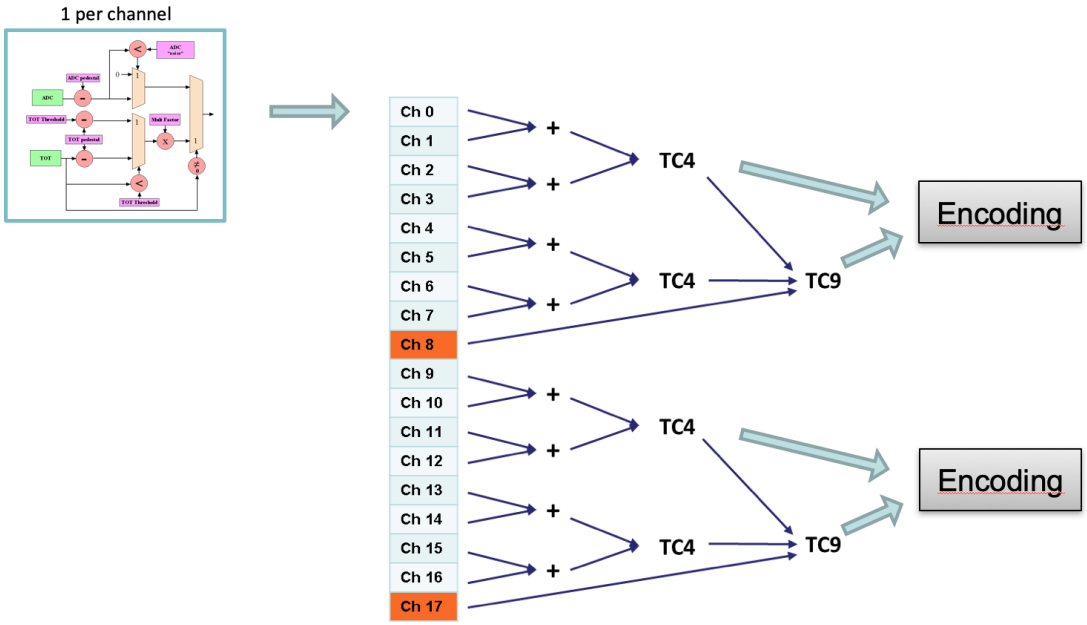
TOT threshold defines the lower limit of the TOT linear part

Multiplication factor = Ratio between the TOT and the ADC LSB

Finally there is the selection between charge from the ADC or TOT.

The user can define the chip to sum charge in groups or 4 or 9 channels to obtain trigger sums.

The selection between the sum by 4 (TC4) or 9 (TC9) is done by the ASIC parameter “SelTC4”.



* “orange” channels only when sum of 9 channels

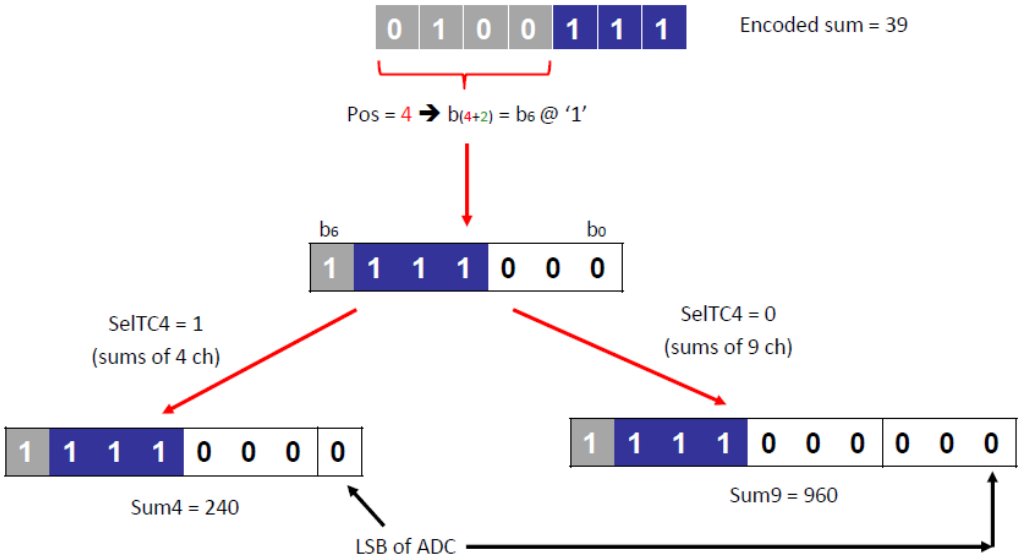
The last step of the trigger data processing is the compression over 7 bits.

Internally, the sum by 4 gives 19b and 21 bits for the sum by 9. We are using the encoding over 7 bits with 4 bits for the position of the MSB and the 3 next bits after the MSB. This maximum number of bits is limited to 18 (b_{17} to b_0). So the sums are truncated by 1 LSB and 3 LSB before the encoding respectively for the sum by 4 and the sum by 9. Finally, we have a word over 18 bits and the encoding follows this logic:

- If sum <= 7 ➔ Position = “0000” and next 3 bits = b_2 b_1 b_0

- Else if MSB @1 is b_n ➔ Position = $(n-2)_2$ and next 3 bits = b[n-1:n-3]
  - example, if b_{17} is 1 then Pos = “1111”

And so the decoding can be described as per below.



The Trigger frame

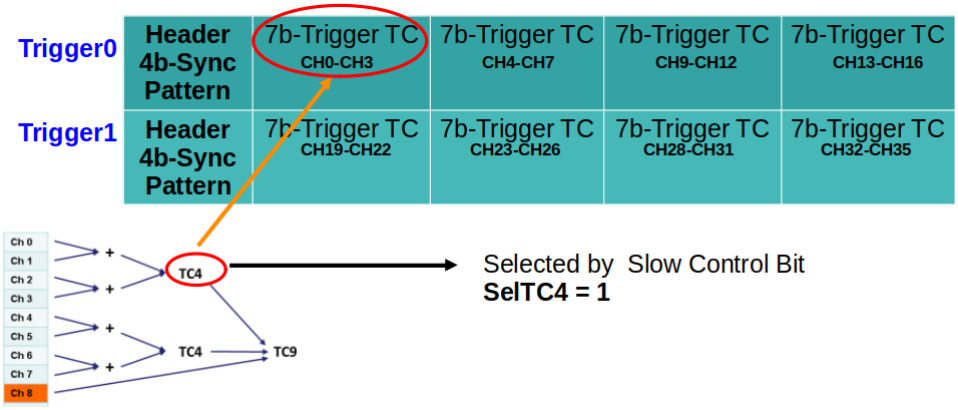
# Trig-link	2 or 4 (configurable)
Link electrical standard	CLPS
Link bitrate	1280 Mbps
Possible to switch-off unused link	Yes (partially)
Serialisation factor	32 (i.e. 32 bits per 25 ns)
Bits order	MSB first
Packet composition	header (4 bit) + payload (28 bits) 4 header = packet [31:28] 28 payload = packet[27:0]
Header	- During startup : 4'b1010 on all links (enabled or disabled) - During run phase : - LINKRESETROCT procedure: 4'h9 @ BCT else 4'hA on all links (enabled or disabled) - else : 4'h9 @ BCT else 4'hA for enabled links, 4'h0 for disabled links
Payload	- During startup: configurable idle word (default 28'hCCC_CCCC) on all links (enabled or disabled) - During run phase: - LINKRESETROCT procedure: configurable idle word (default 28'hCCC_CCCC) on all links (enabled or disabled) - else: 4 consecutive Trigger Cells (TC) for enabled links, 28'h000_0000 for disabled links
Trigger cell (TC) encoding	Floating point with 4 bits exponent and 3 bits mantissa Exponent= TC[6:3] Mantissa= TC[2:0]

The data transmission is MSB first.

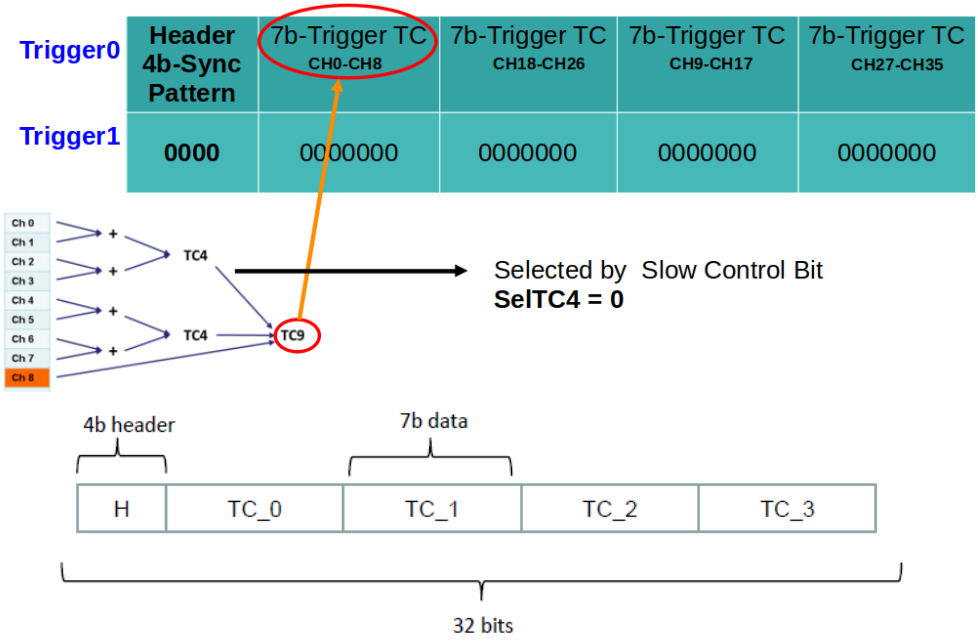
The sum by 4 or 9 is set by slow control.

The Calibration and common mode channels are not in Trigger data.

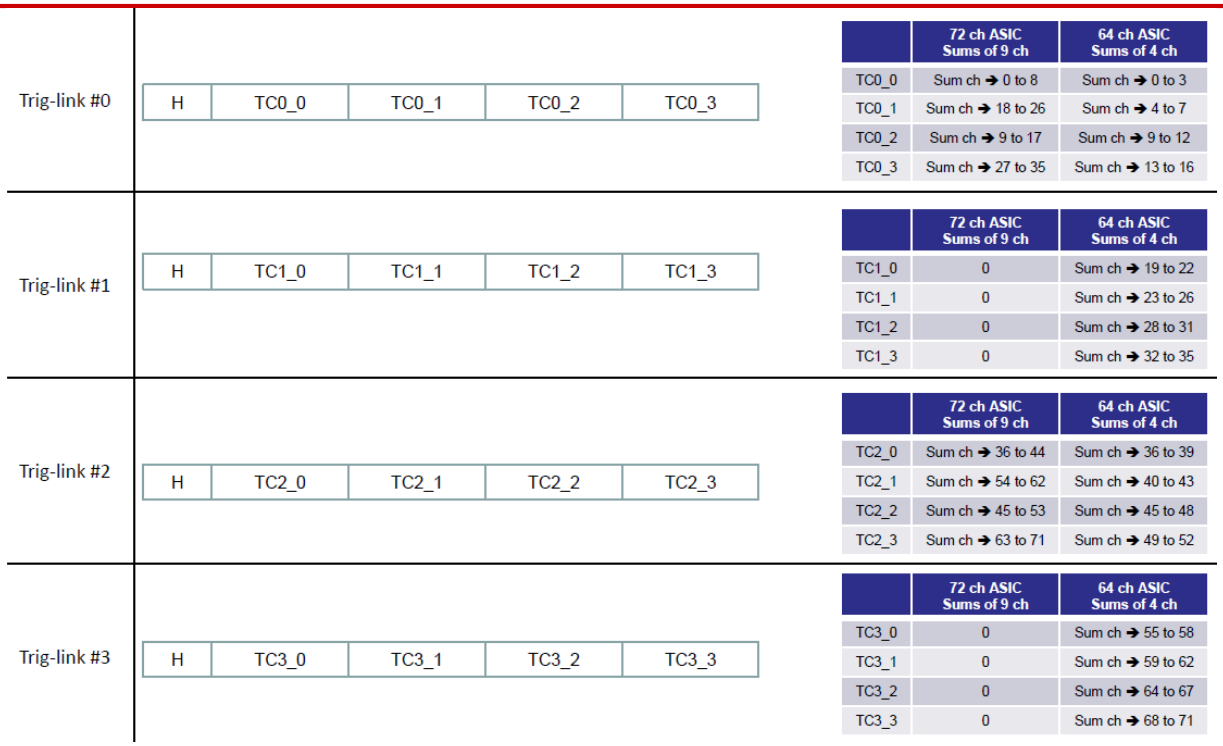
The figure below shows the dataframe for the sum by 4 (half chip).



The figure below shows the dataframe for the sum by 9 (except during link reset T procedure and startup).



The content of the trigger data is depending on the SelTC4 setting.



Trig-link #0	H	TC0_0	TC0_1	TC0_2	TC0_3
Trig-link #1	H	TC1_0	TC1_1	TC1_2	TC1_3
Trig-link #2	H	TC2_0	TC2_1	TC2_2	TC2_3
Trig-link #3	H	TC3_0	TC3_1	TC3_2	TC3_3

	72 ch ASIC Sums of 9 ch	64 ch ASIC Sums of 4 ch
TC0_0	Sum ch ➔ 0 to 8	Sum ch ➔ 0 to 3
TC0_1	Sum ch ➔ 18 to 26	Sum ch ➔ 4 to 7
TC0_2	Sum ch ➔ 9 to 17	Sum ch ➔ 9 to 12
TC0_3	Sum ch ➔ 27 to 35	Sum ch ➔ 13 to 16

	72 ch ASIC Sums of 9 ch	64 ch ASIC Sums of 4 ch
TC1_0	0	Sum ch ➔ 19 to 22
TC1_1	0	Sum ch ➔ 23 to 26
TC1_2	0	Sum ch ➔ 28 to 31
TC1_3	0	Sum ch ➔ 32 to 35

	72 ch ASIC Sums of 9 ch	64 ch ASIC Sums of 4 ch
TC2_0	Sum ch ➔ 36 to 44	Sum ch ➔ 36 to 39
TC2_1	Sum ch ➔ 54 to 62	Sum ch ➔ 40 to 43
TC2_2	Sum ch ➔ 45 to 53	Sum ch ➔ 45 to 48
TC2_3	Sum ch ➔ 63 to 71	Sum ch ➔ 49 to 52

	72 ch ASIC Sums of 9 ch	64 ch ASIC Sums of 4 ch
TC3_0	0	Sum ch ➔ 55 to 58
TC3_1	0	Sum ch ➔ 59 to 62
TC3_2	0	Sum ch ➔ 64 to 67
TC3_3	0	Sum ch ➔ 68 to 71

H is always a 4-bit “1010” except when crossing a trigger value (BXCpt=BCT) where it is “1001”.

Noisy channel

To not sum noisy channels, ADC and TOT data can be forced to 0 per channel.

1.3.3 Digital parameters

The table below gives all the parameters of the digital block for the data and trigger paths, thus there are two of them in the chip as there are two digital blocks for the both sides of the chip.

Name	# bits	Comment
SelTC4	1	1: sum by 4 / 0: sum by 9
Top_P_Add	1	
SC_testRAM_cmd	1	
CalibrationSC	1	
IdleFrame	28	Default 28 LSB “1100---1100” of idle DAQ/T frame
L1Offset	9	L1 offset corresponding to L1 latency
Adc_TH	4	Threshold corresponding to noise in ADC count
MultFactor	5	TOT vs ADC ratio for linearization (default ~25)
Tot_P0, Tot_P1, Tot_P2, Tot_P3	7	TOT pedestal used in TP (common to 9 channels)
Tot_TH0, Tot_TH1, Tot_TH2, Tot_TH3	8	TOT threshold used in TP (common to 9 channels)
ClrAdcTot_Trig	36	
SC_testRAM_data	32	

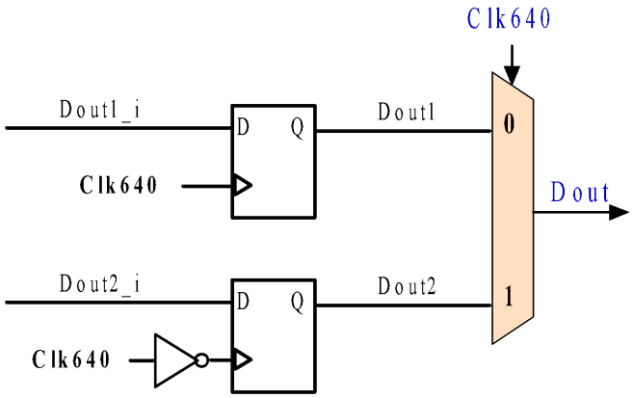
Bx_trigger	8	
Bx_offsetA, Bx_offsetB, Bx_offsetC	12	

SelTCA, namely Select Trigger Cell of 4 channels, allows the user to select the sum mode: set to 1 to sum 4 channels, otherwise sum by 9.

SelTC4		# channels used	# ch in each TC	# Daq-link @ 1,28G	# Trig-link @ 1,28G	unused channels
0		72	9	2	2	-
1 (default)		64	4	2	4	(8, 17, 18, 27) (44, 53, 54, 63)

1.4 E-links

The output differential links 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 1280 MHz. It provides also a 8-bit counter value which counts the number of bit shift.



In the table below, the electrical specifications of the driver are given.

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.

1.5 Interface blocks

1.5.1 Fast Command

The Fast Command unit (FCU) is used to generate internal fast commands and to create 2 triplicated 40 MHz clocks (1 free running + 1 synchronized with the incoming bitstream). The following table describes each command and their latency. The Latency is defined from the last bit received in the 320Mbps bitstream to the associated actions taken inside the ASIC. The Latency is given related to the synchronized 40 MHz clock (clk_fc). The latency of the FCU, common to all fast commands, is 3 clocks cycles (= 59.375ns ± 3,125ns) to decode a command since the arrival of the last serial bit of command.

Command	Code	Description	Total latency (including FCU)
CMD_IDLE	00110110	Permit to lock the phase to generate the main clock 40MHz of digital.	
CMD_PREL1A	11001100	Send a L1A with a delay configured by the PreL1AOffset slow control's parameter.	1 additional clock cycle @40 per PreL1AOffset
CMD_L1A	01001011	The L1A validate Event in the RAM1 Circular buffer. It will send by link throw RAM2 FIFO.	**- if EB empty,** L1A to serial link latency (from last bit of fast cmd to first bit of Header): 10 clock cycle at 40MHz. - **if EB not empty,** from last bit of fast cmd to EB Writing: 4 clock cycle
CMD_L1A_PREL1A	11100001	Send a L1A then a delayed L1A.	L1A and PreL1A latency.
CMD_L1A_NZS	00001111	Same behaviour as L1A	L1A latency.
CMD_L1A_NZS_PREL1A	00101011	Send a NZS and L1A signal, then a delayed L1A.	L1A and PreL1A latency.
CMD_L1A_BCR	01110001	Send a L1A and reset the Bx counter.	L1A and BCR latency.
CMD_L1A_BCR_PREL1A	10100101	Send a L1A and reset the Bx counter, then a delayed L1A.	L1A, BCR and PreL1A latency.
CMD_L1A_CALPULSEINT	00111001	Send a L1A and a Calibration Pulse of 32 Bx (800ns)	CalPulseInt and L1A latency.
CMD_L1A_CALPULSEEXT	10000111	Send a L1A and a Calibration Pulse of 4 Bx (100ns)	CalPulseExt and L1A latency.
CMD_L1A_CALPULSEINT_PREL1A	11100010	Send a L1A and a Calibration Pulse of 32 Bx, then a delayed L1A.	CalPulseInt L1A and PreL1A latency.

CMD_L1A_CALPULSEEXT_PREL1A	11110000	Send a L1A and a Calibration Pulse of 4 Bx, then a delayed L1A.	CalPulseExt L1A and PreL1A latency.
CMD_BCR	00011101	Reset the Bx counter to its default value (parameter).	3 clock cycle at 40MHz.
CMD_BCR_PREL1A	10100011	Reset the Bx counter, then a delayed L1A.	BCR and PreL1A latency.
CMD_BCR_OCR	10010101	Reset the Bx and Orbit counter.	BCR and OCR latency.
CMD_CALPULSEINT	00101101	Send a Calibration Pulse of 32 Bx.	1 additional clock cycle at 40MHz to latch the pulse.
CMD CALPULSEEXT	01111000	Send a Calibration Pulse of 4 Bx.	1 additional clock cycle at 40MHz to latch the pulse.
CMD CALPULSEINT PREL1A	01010101	Send a Calibration Pulse of 4 Bx, then a delayed L1A.	CalPulseInt and PreL1A latency.
CMD CALPULSEEXT PREL1A	10010011	Send a Calibration Pulse of 32 Bx, then a delayed L1A.	CalPulseExt and PreL1A latency.
CMD CHIPSYNC	11010010	Like a ReSb but only affects the digital part. Does not reset analog part (PLL) and the FCU	4 clock cycles at 40 MHz to reset Startup FSM to idle state.
CMD EBR	11010001	Reset the buffer pointer of RAM2 (FIFO). Do not perform ECR	4 clock cycle at 40MHz.
CMD ECR	10101001	Reset the Event counter.	5 clock cycles 40MHz.
CMD LINKRESETROCT	10011001	Force links to send 260 IDLES.	5 clock cycles at 40MHz.
CMD LINKRESETROCD	10011010	Force links to send 260 IDLES after the current sending packet ends.	The link is reset immediately if the readout is finished.

- An EBR followed by a L1A is acceptable for HGCROC. The EBR let the current event to be complete. It clears events from memory. After the L1A, we have the next event on the Daq output link.
- CalPulseInt: send STROBE pulse to the internal calibration DAC. The STROBE pulse length is 32 Bx at 40 MHz = 800ns. The STROBE pulse phase depends on the main digital clock from Fast Commands module (clk40_fc).
- CalPulseExt: send STROBE pulse to the external pin SiPM_calibration. The STROBE pulse length is 4 Bx at 40 MHz = 100ns. The STROBE pulse phase depends on the main digital clock from Fast Commands module (clk40_fc).
- The phase of the STROBE pulse can be set by slow-control.
- Sending Fast Command CalPulse{Int/Ext} does not change the data content mode.
- LinkResetRocD: Send 260 IDLES immediately if the Buffer (RAM2) is empty.

The FCU decodes the 40MHz clock 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.

1.5.2 I2C

I2C	Detail	Comments
Chip addressing	4 bits	MSB of I2C first byte
Number of Direct-access registers	8	Reg0 to Reg7
Direct-access register adressing	3 bits	LSB of I2C first byte
Indirect-access register adressing	16 bits	
"Burst" writing / reading	Yes	Accessible through R3 (Reg3)
Multi-Byte writing / reading*	Yes	*Multi-Byte reading in **HGCROC3A only**
I2C speed	100k - 1M	
General call	No	
Broadcast write	No	

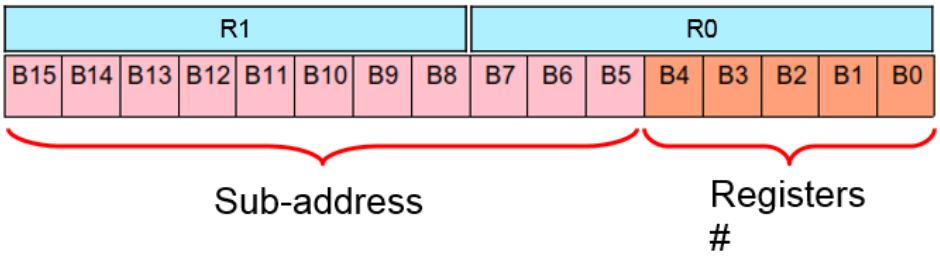
As general call and broadcast write features are not implemented, we can access the asic only with its chipid.

The I2C circuits of the chip has 8 internal registers whose the use is described in the table below:

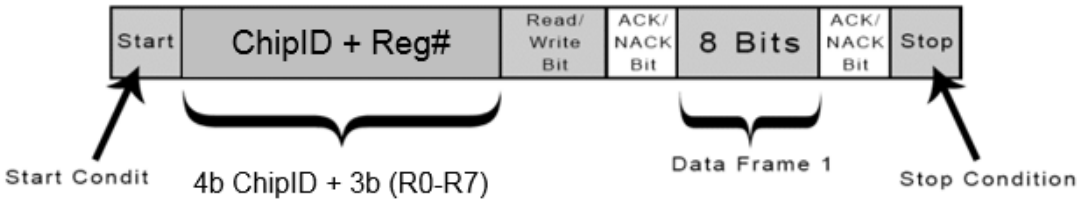
I2C @	Register	Comments
R0	ASIC parameter address (LSB)	Indirect @
R1	ASIC parameter address (MSB)	Indirect @
R2	Data	
R3	Data with auto @++	Increment indirect @ after each access
R4-R5-R6	Direct access SC-register	
R7	Status register (error, parity)	Read-only

To cope with the large number of parameters, extended addressing is used:

- 512 sub_address can be addressed (B15, B14 not used and have to be set to 0)
- Each sub_address has max 32 configuration parameters
- Extended addressing realized through 2 direct access registers: R0 and R1

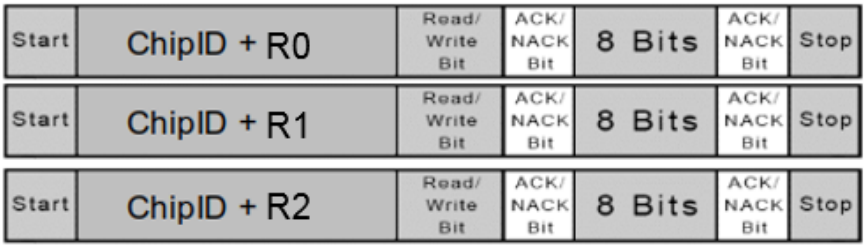


The frame of the I2C protocol is always the same:

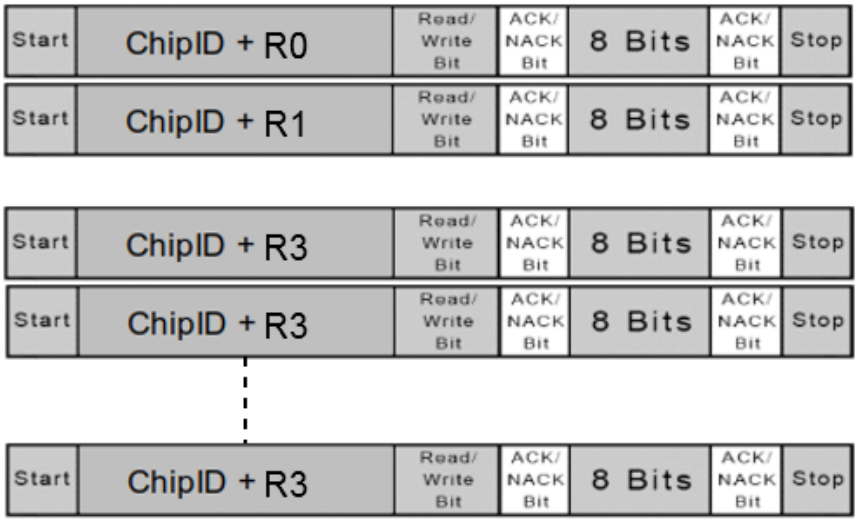


To write, set R/W bit to 0 and to read set R/W to 1.

For instance, to set a specific 8b word of the chip, the user has to write into the R0 register then R1 register to select the good parameters register address, and then write the data into the R2 register.



The user can also write into consecutive parameters register addresses: rather to write the parameters into the R2 register, he has to write successively into the R3 register.



The I2C circuitry has been fully checked:

- digital verification of the I2C block
- analog verification of the I2C block
    - Write and read transactions from two tri-states inputs, with the bi-directionnal pads and the i2c block
- analog verification of the chip's full registers: write and read of dedicated pattern in chosen registers on both halves.

1.6 Operating mode

1.6.1 Resets

There are three way to reset the chip.

- The hard reset pin ReHb (CMOS input, active low) resets everything inside the ASIC (PLL, I2C, FSM, counters, configuration)
- The soft reset pin ReSb (CMOS input, active low) resets everything except the slow-control parameters.
- The fast command ChipSync resets the digital part. Does not reset analog part, the PLL and the FCU, and the configuration.

A Power-On-Reset has been implemented which performs a hard reset ReHb 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 (I2C_rstb).

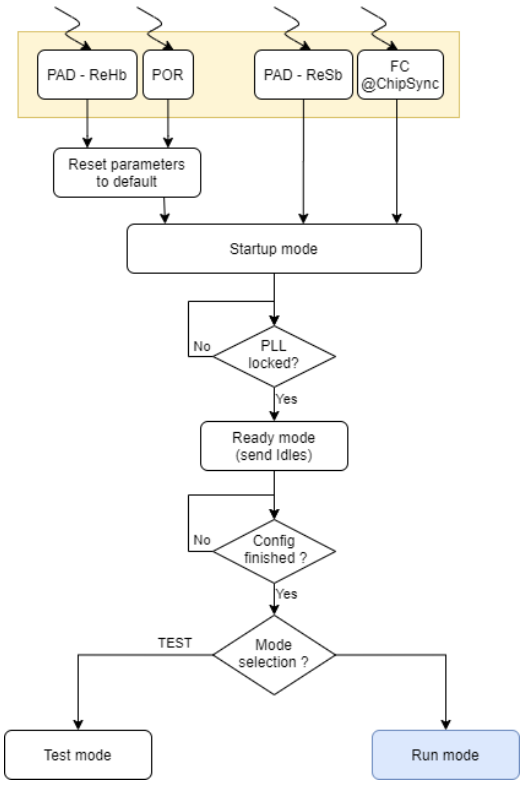
Others fast command allow to reset some parts of the chip. They are described below:

- Bunch Crossing counter reset (BCR): from the fast command receiver, it resets the Bunch Crossing counter to its default value (BX_offset parameter)
- Event Counter reset (ECR): from the fast command receiver, it resets the Event Counter to 1
- Orbit Counter reset (OCR): from the fast command receiver, it resets the Orbit Counter to 0

- Event Buffer Reset (EBR): from the fast command receiver, it resets only the pointers of the Event Buffer (DRAM2). The ASIC is able to accept a new L1A on the next clock cycle. If there is an event readout in progress, it finishes normally.

1.6.2 Startup sequence

The FCU decodes the 40MHz clock 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.



When the chip is powering on, the Power-On-Reset (POR) block applies a reset (like the ReHb). The POR block can be disabled by a pin (Power-On disable).

There are 4 main states in this sequence:

- Startup mode: This is the state just after a reset and the ASIC state is defined by the default parameters. It waits there until the PLL is locked
- Ready mode: In this state the PLL is locked and the user should load the ASIC parameters through I2C. To get out of this state, a special “config_OK” parameter should be written (RunR and RunL for both halves). When all the parameters related to the serial links are loaded, the ASIC starts sending a forced 32b IDLE pattern through the 4 trigger links (4b header forced to 1010).
- Then depending of the chosen mode (ASIC parameter) loaded in the previous state, we enter in:
    - Run mode: this is the normal operational mode where the ASIC starts writing into DRAM physics data.

- Test mode: this mode enables an extensive test of the DRAM (retention time). When tests are finished, the user should apply an hard reset to re-initiate the startup flow and go to the “Run mode”.

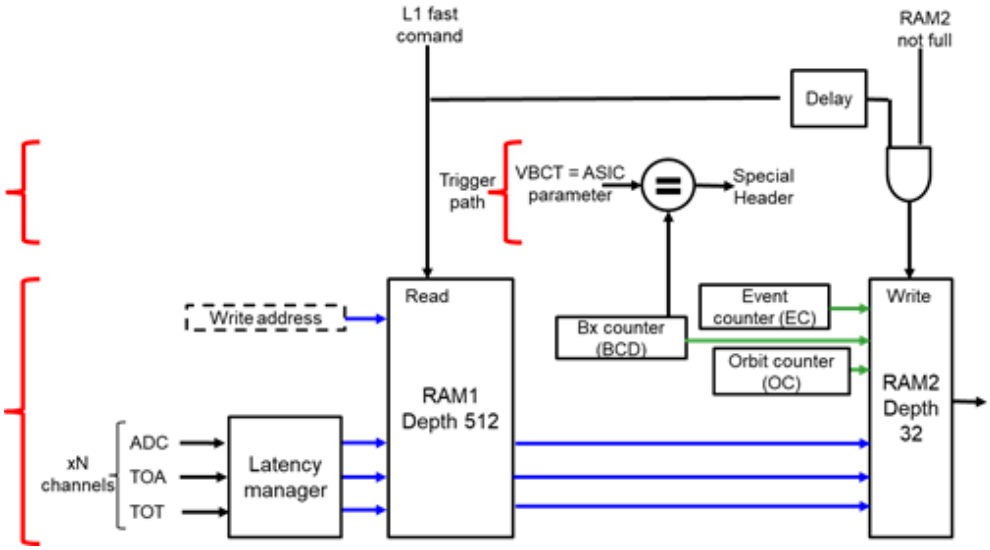
Low Power mode disables the analog part of the chip: preamplifier, shaper, discriminators. But since the PLL is ON, the ADC, TDC, AlignBuffer will be ON.

1.6.3 Definition of the Time Tag Counter

The HGCROC chip has 3 internal counters to tag the data:

- Bunch crossing counter (BCD)
- Orbit counter (OC)
- Event counter (EC)

They are attached to the data extracted from RAM1 and copied to RAM2. BCD is also used to generate the special header in the trigger path when its value crosses a programmed one (slow control parameter on 12 bits). A global scheme of these counter is shown below:



The 3 counters are detailed below:

- BCD counter (BCD): 12-bit counter incremented every cycle (25 ns). Its reset value is given by a 12-bit programmable parameter (BX_offset parameter, value 1 by default). Its range is from 1 to 3564 with a wrapping when the upper limit is reached. This counter is also used to generate the special header (binary 4 bits: 1001) when a programmed value (VBCT/BxTrigger) is crossed. This header occurs on output at the same time in all serial links (trigger or data).
- Event Counter (EC): 6-bit counter incremented every L1A (after data processing). Its reset value is 1 with a range from 0 to 63 (wrapping when the upper limit is reached). After a reset, the first event readout will have an event counter tag equal to 1.
- Orbit Counter (OC): 3-bit counter incremented every BCD wrap. Its reset value is 0 with a range from 0 to 7 (wrapping when the upper limit is reached).

1.7 Ancillary blocks

1.7.1 PLL and clocks distribution

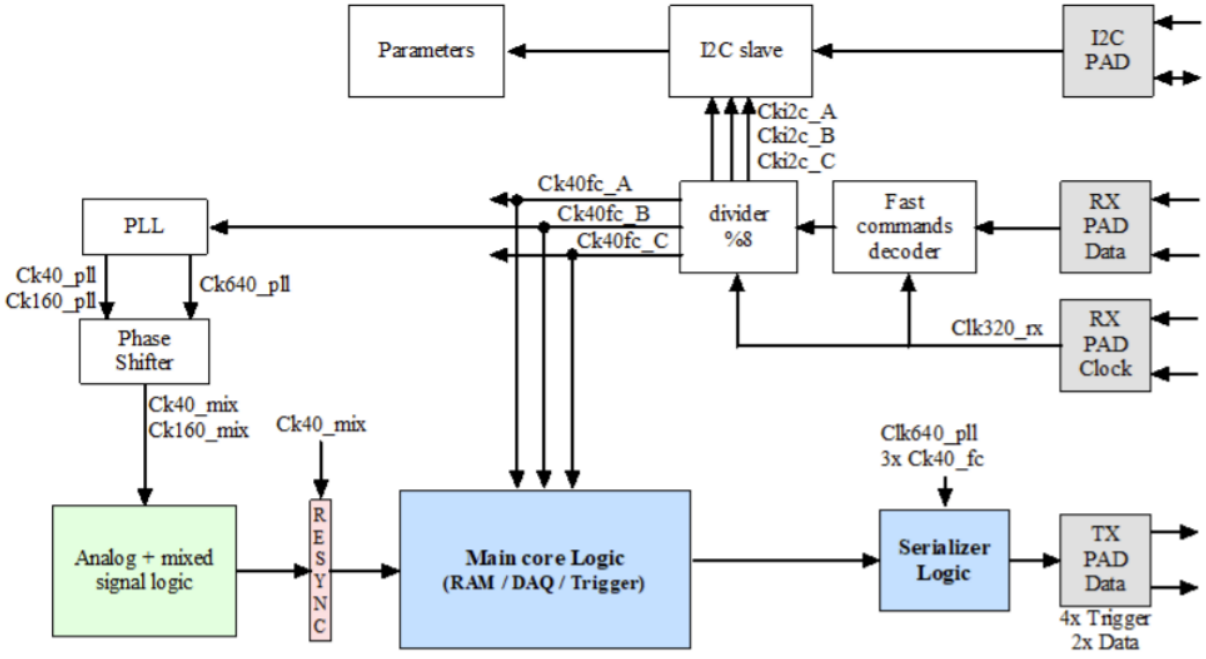
The main specifications of the PLL are described in the following table:

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.

During the normal operation the 40 MHz PLL input clock is derived by the on-chip Fast Command unit receiving a 320 MHz clock on its fast command interface. For test purposes, the pinout of the DV1 version of the HgRoc ASIC foresees a pair of input pins dedicated to a differential 40 MHz clock. The chip can be programmed to feed this external clock to the PLL. Also the chip pinout fixes a dedicated differential signaling output in order to evaluate the quality of the PLL generated clocks. For power supply, the PLL must have its own power supplies and reserved pads (VDD and VSS).

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 the 40 MHz clock in phase with the LHC. This 40 MHz clock is used for the I2C block, all the FSMs, the counters, the two RAMs, the wr/rd pointers and is the PLL reference clock. Only hard resets (ReHb and RESb) allows to stop or restart this clock by resetting the FastCommand block.



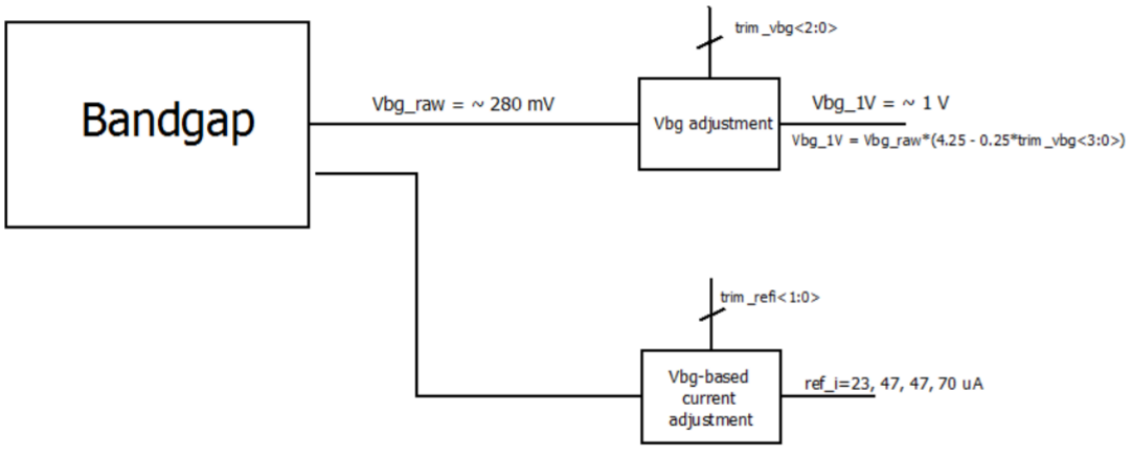
The FCU decodes the 40MHz clock from the 320MHz links. The 40MHz clock for the I2C is directly divided from the 320MHz clock input, so that the chip can be configurable even if it does not receive any fast commands. The 40MHz clock for all the others parts (digital, PLL, mixed) are decode from both 320MHz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

The PLL takes as reference the 40 MHz clock provided by the FastCommand block. The PLL can only be reset by the hard resets (ReHb and ReSb). 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.

1.7.2 Bandgap and voltage references

Typically, the bandgap provides an output voltage of around 280 mV. We need to multiply this value to get a usable voltage reference around 1 V. From the bandgap voltage, a Vbg-based current is provided as well.



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.

Four global 10 bits DACs provide voltage references for the analog part: the two discriminator thresholds, the inverted and non-inverted references for the shaper. As the circuit is symmetrical, there are two bandgaps and two sets of voltage references with DACs for the two right and left sub-parts.

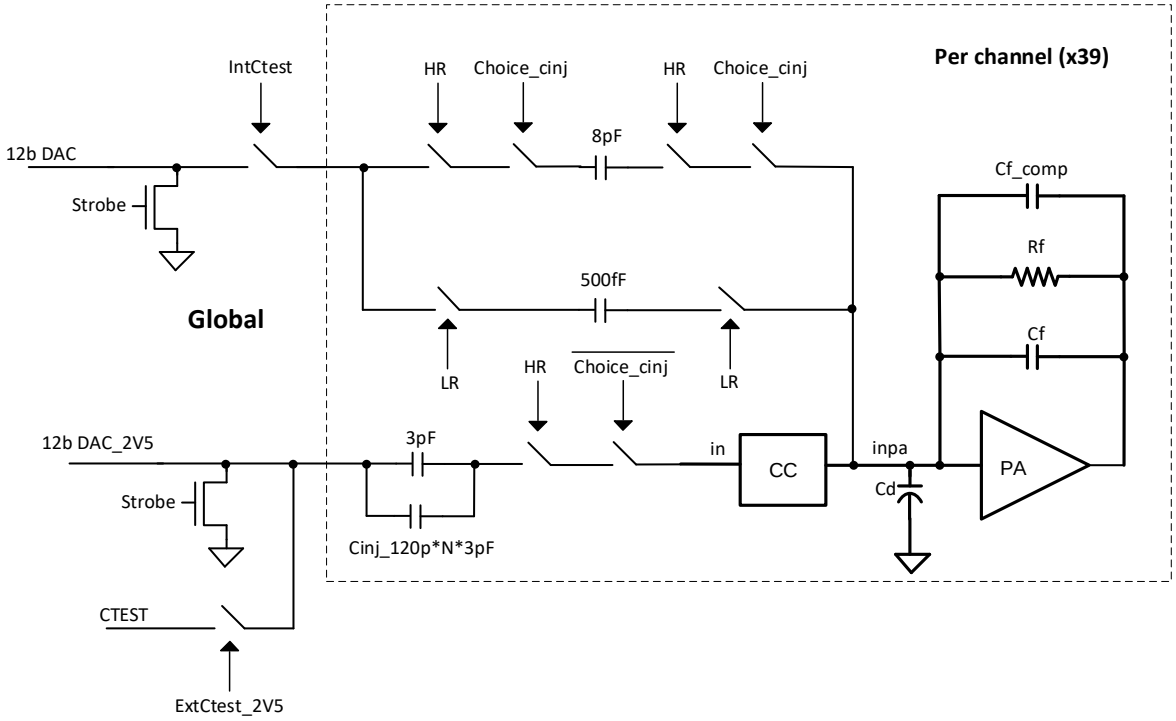
The voltage references are fabricated from the bandgap (ref_i), the 10b DAC and the preamplifier output of a common mode channel (channels CM<1> and CM<3>) so that the chip is not sensitive to the temperature: ref_volt (inv, noinv, toa and tot) = DC_CM<1/3> - R1*i_ref + DAC*i_ref*R2. So all the references follow the preamplifier input DC value.

Note that since the reference voltages are made from the preamplifier output of CM<1> and CM<3>, these two channels are requested to be ON (the channel_off parameters cannot be set to 1 for instance for these channels).

1.7.3 Calibration circuit

There are two **12b-DACs** available (**12b-DAC** and **12b-DAC_2V5**), one to inject at the input of the Current Conveyor and another one to inject directly to the input of the preamplifier.
For preamplifier calibration and assuming the 12b-DAC provides a voltage up to around 1 V (the bandgap value), the user may set 0.5pF or/and 8pF injection capacitances in the chosen channels, to study the 0 – 0,5 pC range, the 0 – 8 pC range or, by setting both capacitances, the 0 – 8,5 pC range.

For charge injection at the conveyor input, the user may set the 3pF capacitor alone of a desired channel or inject in the same channel but by using the 3pF capacitor of each channel (3pF * 39 = 117 pF).



The following tables show how to select the different configurations for internal/external calibration:

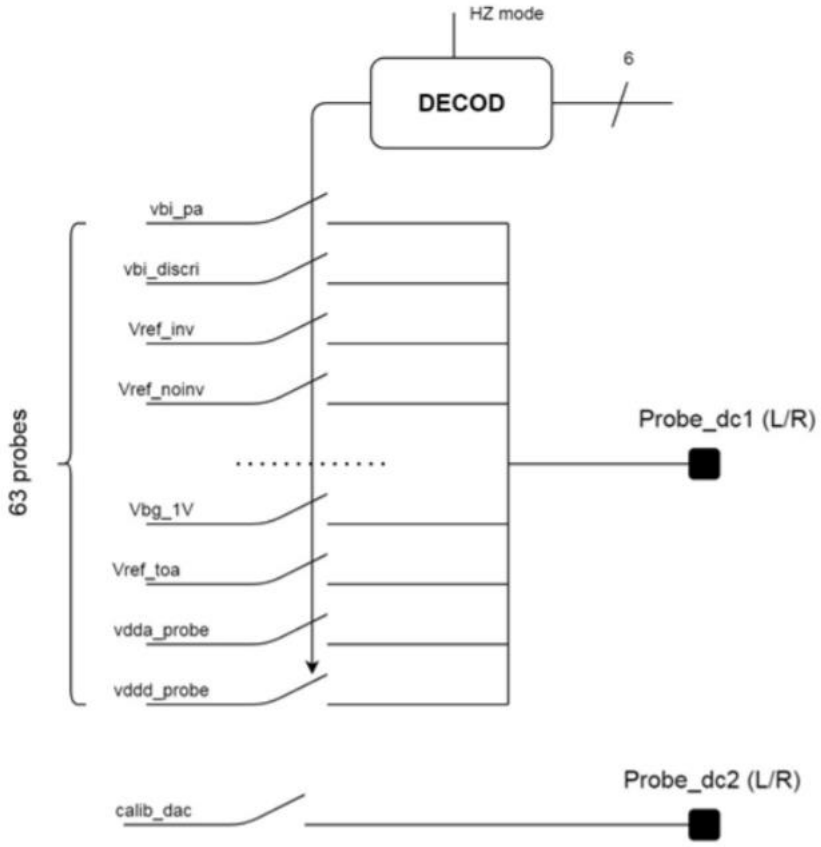
	DAC LR	DAC HR	CINJ 3pF	CINJ 3pF*N	CTEST 3pF	CTEST 3pF*N
HighRange	0	1	1	1	0	0
LowRange	1	0	0	0	0	0
Choice_cinj	X	1	0	0	0	0
Cinj_120p	0	0	0	1	0	1
IntCtest	1	1	0	0	0	0
ExtCtest_2V5	0	0	0	0	1	1
Calib INPA	0 - 500fC	0 - 8pC	-	-	-	-
Calib IN	-	-	0 – 3pC	0 – 3pC*N	-	-
Calib Ext Inj	-	-	-	-	CTEST * (0 – 3pC)	CTEST * (0 – 3pC*N)

Notes:

- For HighRange and LowRange (LR) see Register #4 of Section 3.2
- For choice_cinj, cinj_120p, IntCtest, ExtCtest_2V5, Calib_dac and Calib_dac_2V5 see Registers #6, #7, #9 and #10 of Section 3.4

To correctly calibrate the ADC/TOT behaviour, it is important to have an overlap between the available charge ranges.
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_2V5.

1.7.4 Monitoring



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 64 available monitoring points.
If cmd_probe_dc1<0> is set, the next 32 voltages are accessible:

address	Probe point	Nominal value	Comment
0	Vbi_pa_mon		
1	Ib_sc_mon		
2	Ib_input_dac		
3	Ib_dac_trim		
4	Vbo_pa		Tunable over 3bits (dac_vbo_pa)
5	Ib_conv		
6	Vbi_discri_tot		
7	Vbm_discri_tot		
8	Vbo_discri_tot		
9	Vbi_discri_toa		

10	Vcasc_discri_toa		
11	Vbm1_discri_toa		
12	Vbm2_discri_toa		
13	Vbo_discri_toa		
14	EXT_REF_TDC		
15	Probe_Vref_time		Equal to the TOA discri input DC
16	Vcn		
17	VD_FTDC_P_EXT		
18	VD_CTDC_P_EXT		
19	Probe_VrefPa		Equal to the preamp output (DC)
20	Vcp		
21	VD_FTDC_N_EXT		
22	VD_CTDC_N_EXT		
23	Vb_hyst_tot		
24	Vb_suiv1_pa		
25	Vb_suiv2_pa		
26	vbiN_sk		Tunable over 2bits (ibi_sk)
27	vbiP_sk		Tunable over 2bits (ibi_sk)
28	vbFCN_sk		Tunable over 6bits (dac_sk)
29	vbFCP_sk		Tunable over 6bits (dac_sk)
30	vbiN_noinv		Tunable over 2bits (ibi_noinv)
31	vbiP_noinv		Tunable over 2bits (ibi_noinv)

If cmd_probe_dc1<1> is set, the next 32 voltages are accessible:

address	Probe point	Nominal value	Comment
0	vbFCN_noinv		Tunable over 6bits (dac_noinv)
1	vbFCP_noinv		Tunable over 6bits (dac_noinv)
2	vbiN_inv		Tunable over 2bits (ibi_inv)
3	vbiP_inv		Tunable over 2bits (ibi_inv)
4	vbFCN_inv		Tunable over 6bits (dac_inv)
5	vbFCP_inv		Tunable over 6bits (dac_inv)
6	vbiN_noinv_buf		Tunable over 2bits (ibi_noinv_buf)
7	vbiP_noinv_buf		Tunable over 2bits (ibi_noinv_buf)
8	vbFCN_noinv_buf		Tunable over 6bits (dac_noinv_buf)
9	vbFCP_noinv_buf		Tunable over 6bits (dac_noinv_buf)
10	vbiN_inv_buf		Tunable over 2bits (ibi_inv_buf)
11	vbFCP_inv_buf		Tunable over 6bits (dac_inv_buf)
12	vbiP_inv_buf		Tunable over 2bits (ibi_inv_buf)
13	vbFCN_inv_buf		Tunable over 6bits (dac_inv_buf)
14	Vb_5bdac_out_inv		Tunable range over 3bits(idac_inv)
15	Vb_5bdac_tot		Tunable range over 3bits(idac_tot)
16	Vb_5bdac_toa		Tunable range over 3bits(idac_toa)

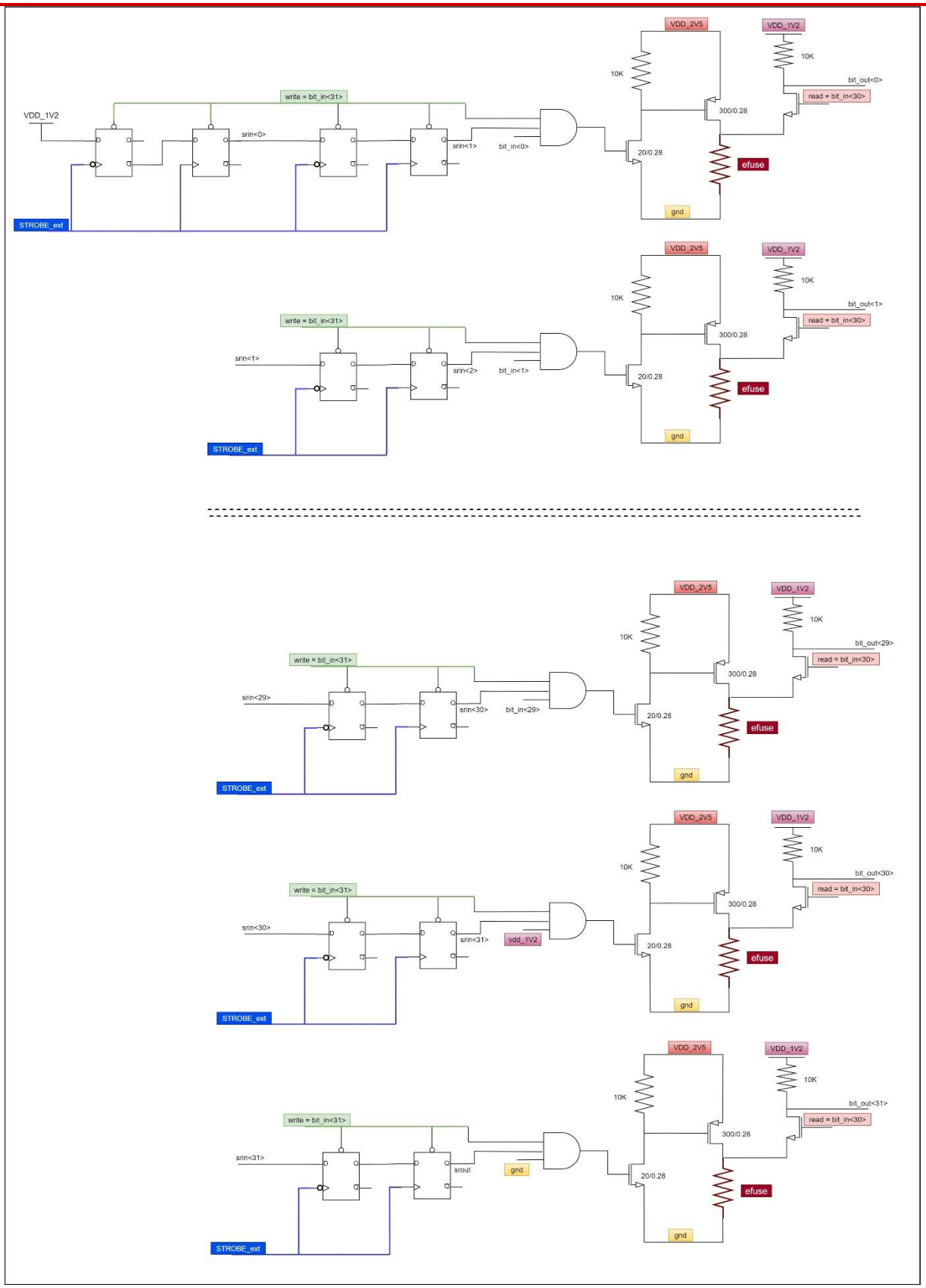
17	Vcm_0p6_inv		
18	Vcm_0p6_noinv		
19	Vrefp_adc		
20	Vcm_adc		
21	Vref_sk		
22	Vref_noinv		Tunable over 10bits (Vref_noinv)
23	Vref_inv		Tunable over 10bits (Vref_inv)
24	Vref_tot		Tunable over 10bits (Vref_tot)
25	Vref_toa		Tunable over 10bits (Vref_toa)
26	Vbg_1V		Tunable over 3bits (Vbg_1V)
27	Probe_vdd_pa		
28	Probe_vddd		
29	Ibo_ref_adc		
30	Vbm2_pa		
31	Probe_center		VOUT_INIT_EXT / EXT_REF_PLL (half 1/0)

If cmd_probe_dc1<1:0> 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 is set, the calibration dac output is sent to probe_dc2.

1.7.5 Efuse

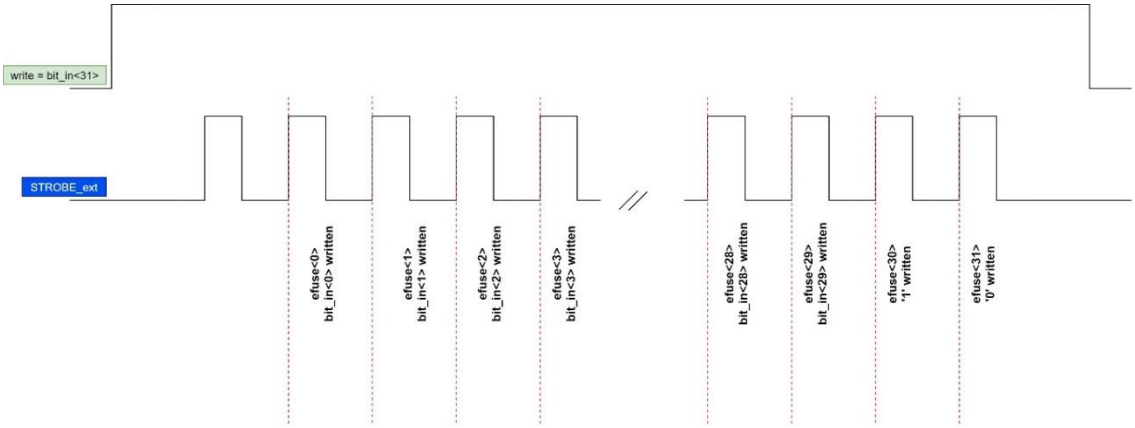
Serial number over 31 bits can be written inside each chip by burning e-fuses. Actually there are 32 e-fuses inside the chip, but the two last MSB, bit<31> and bit<30>, are hard-wired to 0 and to 1 respectively, furthermore the MSB, bit<31>, cannot be read. So in fact the range of the serial number goes 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.



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.

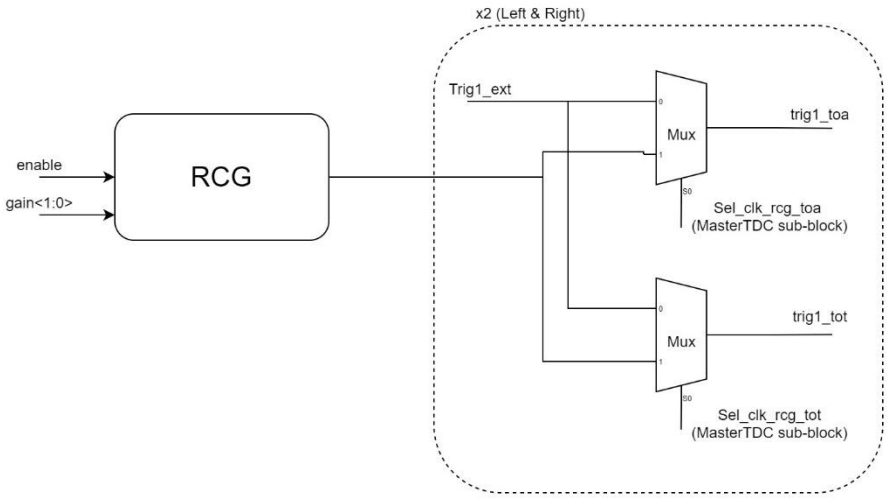


In order to read the serial number back, R15[6] of Top sub-block has to be set to 1 to enable the *read* permission, and then each bit is connected, LSB first, into R8[0..7], R10[0..7], R12[0..7] and R14[0..6]. No matter the value on the *efuse* pin: 0V, 2.5V or floating. The bit R14[7] gives the state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

1.7.6 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.

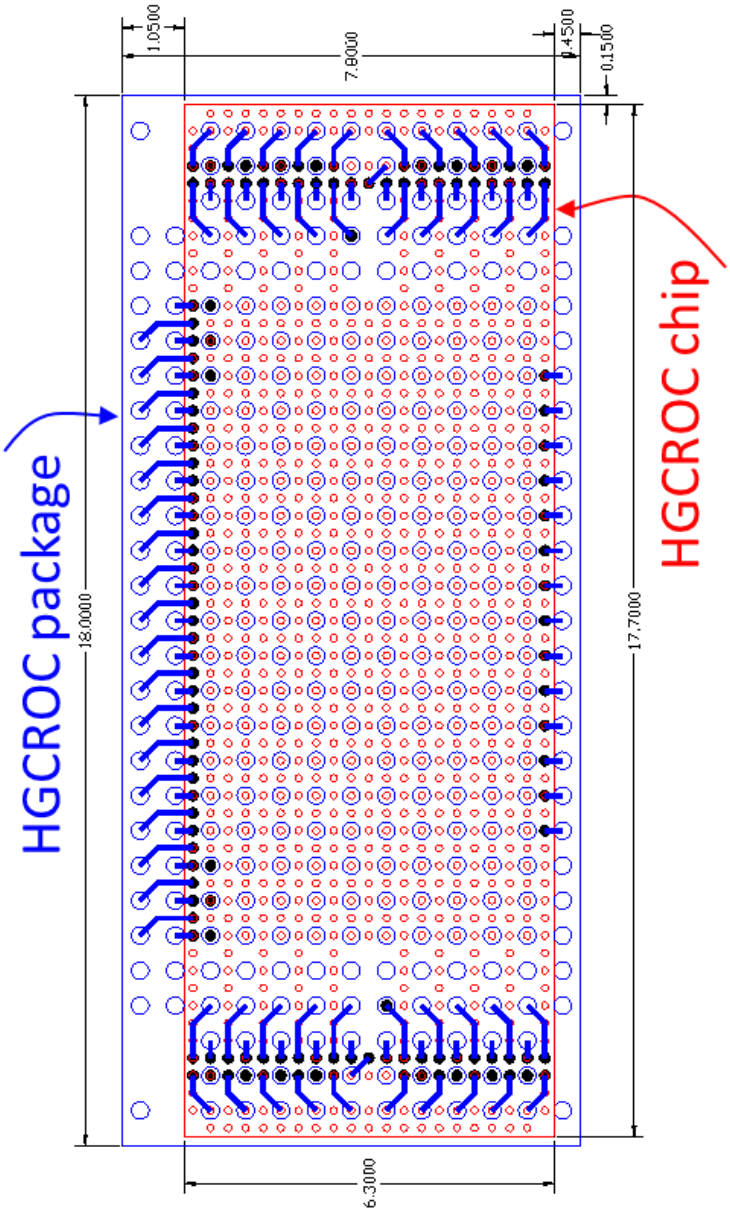
By default, the RCG is disabled and off.



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).

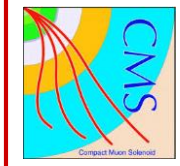
2 Packaging, I/Os and powering scheme

The chip pinout has been defined by the hexaboard constraints [Tommaso et al.] and the BGA package that will be used to house the chips, as shown in figure below.



2.1 Pin maps

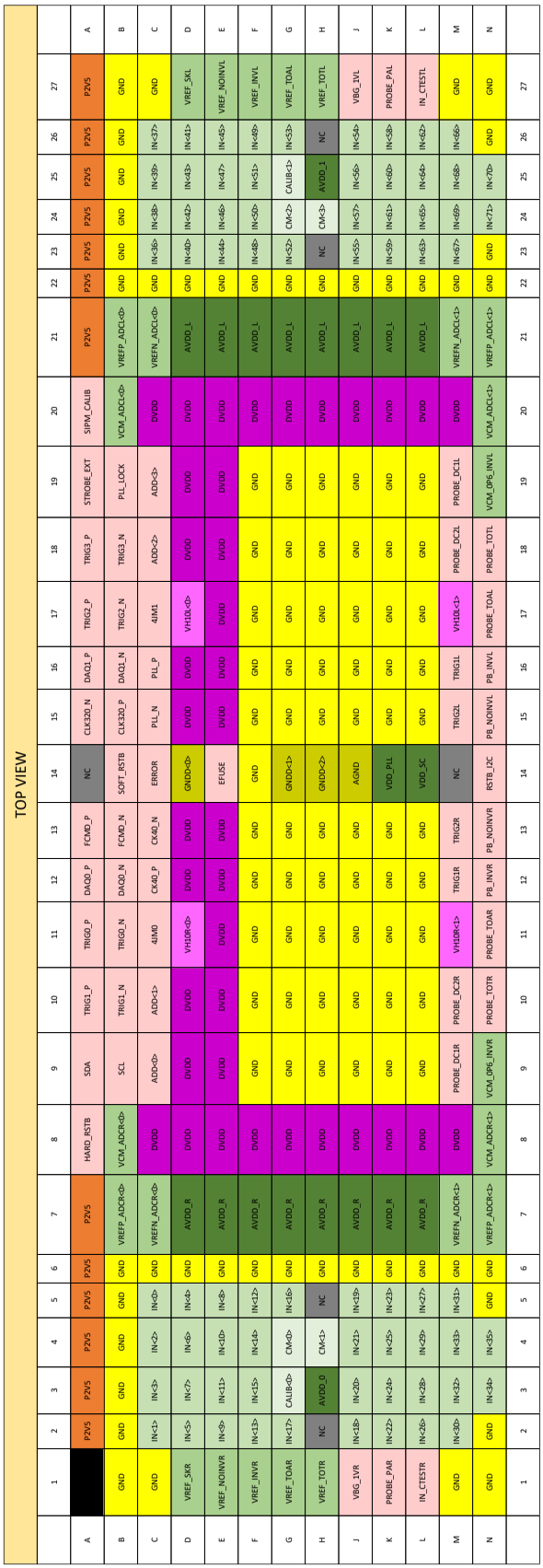
The two figures below show the left view and the right view of the die. Can be noticed the common column 25 in both figures. TO BE UPDATED FOR ROC3



	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32	31	30	29	28	27	26	25
A		gnd_dacL	gnd_paL	in<37>	in<36>	gnd_paL	gnd_skL	gnd_bufL	gnd_adcL	vrefp_adcL	gnd_tdcL	NC	gnddL	sipm_calib	pll_lock	strobe_ext	trig3_n	trig3_p	trig2_n	trig2_p	daq1_n	daq1_p	clk320_p	clk320_n	soft_rstb
B	vssa	vdd_dacL		in<39>	in<38>							vcm_adcL	vdddL	NC	add<3>	NC	add<2>	NC	4JM1	VHI10_2L	pll_p	NC	pll_n	NC	error
C	vdd_2v5L	gnd_dacL	gnd_paL	in<41>	in<40>	vdd_paL	vdd_skL	vdd_bufL	vdd_adcL	gnd_adcL	vdd_tdcL	gnddL	gnddL	vss	vss	vss	vss	vss	vss	vss	vss	vss	vss	vss	NC
D	vref_skL	NC		in<43>	in<42>							vcm_adcL	vdddL	vddd2L	vddd2L	vddd2L	vddd2L	vddd2L	VHI10-2L	vddd2L	vddd2L	vddd2L	vddd2L	vddd1	gndd1
E	vdd_2v5L	gnd_dacL	gnd_paL	in<45>	in<44>	gnd_paL	gnd_skL	gnd_bufL	gnd_adcL	vrefp_adcL	gnd_tdcL	gnddL	gnddL	vss	vss	vss	vss	vss	vss	vss	vss	vss	vss	vss	NC
F	vref_noinvL	NC		in<47>	in<46>							vcm_adcL	vdddL	vddd2L	vddd2L	vddd2L	vddd2L	vddd2L	vddd2L	vddd2L	vddd2L	vddd2L	vddd2L	vddd1	efuse
G	vdd_2v5L	gnd_dacL	gnd_paL	in<49>	in<48>	vdd_paL	vdd_skL	vdd_bufL	vdd_adcL	gnd_adcL	vdd_tdcL	gnddL	gnddL	vss	vss	vss	vss	vss	vss	vss	vss	vss	vss	vss	NC
H	vref_invL	NC		in<51>	in<50>							vcm_adcL	vdddL	vss	vss	vss	vss	vss	vss	vss	vss	vss	vss	vss	NC
J	vdd_2v5L	gnd_dacL	gnd_paL	in<53>	in<52>	gnd_paL	gnd_skL	gnd_bufL	gnd_adcL	vrefp_adcL	gnd_tdcL	gnddL	gnddL	vddd2L	vddd2L	vddd2L	vddd2L	vddd2L	vddd2L	vddd2L	vddd2L	vddd2L	vddd2L	vddd1	NC
K	vref_toaL	vdd_dacL		calib<1>	CM<2>							vcm_adcL	vdddL	vss	vss	vss	vss	vss	vss	vss	vss	vss	vss	vss	gndd1
L		gnd_dacL	gnd_paL	gnd_paL	gnd_paL	vdd_paL	vdd_skL	vdd_bufL	vdd_adcL	gnd_adcL	vdd_tdcL	gnddL	gnddL	vddd2L	vddd2L	vddd2L	vddd2L	vddd2L	vddd2L	vddd2L	vddd2L	vddd2L	vddd2L	vddd1	NC
M	vref_totL	vdd_dacL		NC	CM<3>							vcm_adcL	vdddL	vss	vss	vss	vss	vss	vss	vss	vss	vss	vss	vss	gndd1
N	vdd_2v5L	gnd_dacL	gnd_paL	in<54>	in<55>	gnd_paL	gnd_skL	gnd_bufL	gnd_adcL	vrefp_adcL	gnd_tdcL	gnddL	gnddL	vddd2L	vddd2L	vddd2L	vddd2L	vddd2L	vddd2L	vddd2L	vddd2L	vddd2L	vddd2L	vddd1	NC
P	vbg_1VL	NC		in<56>	in<57>							vcm_adcL	vdddL	vss	vss	vss	vss	vss	vss	vss	vss	vss	vss	vss	gnd_pll
R	vdd_2v5L	gnd_dacL	gnd_paL	in<58>	in<59>	vdd_paL	vdd_skL	vdd_bufL	vdd_adcL	gnd_adcL	vdd_tdcL	gnddL	gnddL	vddd2L	vddd2L	vddd2L	vddd2L	vddd2L	vddd2L	vddd2L	vddd2L	vddd2L	vddd2L	vddd1	gnd_pll
T	probe_paL	NC		in<60>	in<61>							vcm_adcL	vdddL	vss	vss	vss	vss	vss	vss	vss	vss	vss	vss	vss	vdd_pll
U	vdd_2v5L	gnd_dacL	gnd_paL	in<62>	in<63>	gnd_paL	gnd_skL	gnd_bufL	gnd_adcL	vrefp_adcL	gnd_tdcL	gnddL	gnddL	vddd2L	vddd2L	vddd2L	vddd2L	vddd2L	vddd2L	vddd2L	vddd2L	vddd2L	vddd2L	vddd1	vdd_pll
V	in_ctestL	NC		in<64>	in<65>							vcm_adcL	vdddL	vss	vss	vss	vss	vss	vss	vss	vss	vss	vss	vss	vdd_sc
W	vdd_2v5L	gnd_dacL	gnd_paL	in<66>	in<67>	vdd_paL	vdd_skL	vdd_bufL	vdd_adcL	gnd_adcL	vdd_tdcL	gnddL	gnddL	vss	vss	vss	vss	vss	vss	vss	vss	vss	vss	vss	vdd_sc
Y	vssa	vdd_dacL		in<68>	in<69>							vcm_adcL	vdddL	NC	probe_dc1L	NC	probe_dc2L	NC	VHI10_1L	NC	trig1L	NC	trig2L	NC	NC
Z		gnd_dacL	gnd_paL	in<70>	in<71>	gnd_paL	gnd_skL	gnd_bufL	gnd_adcL	vrefp_adcL	gnd_tdcL	NC	gnddL	NC	vcm_0p6_inv	NC	probe_totL	NC	probe_toaL	NC	probe_invL	NC	probe_noinvL	NC	rstb_i2c
	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32	31	30	29	28	27	26	25

25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	
soft_rstb	fcmd_p	fcmd_n	daq0_p	daq0_n	trig0_p	trig0_n	trig1_p	trig1_n	sda	scl	hard_rstb	gnddR	NC	gnd_tdcR	vrefp_adcR	gnd_adcR	gnd_bufR	gnd_skR	gnd_paR	in<0>	in<1>	gnd_paR	gnd_dacR		A
error	NC	clk40_p	NC	clk40_n	VHI10_2R	4JM0	NC	add<1>	NC	add<0>	NC	vdddR	vcm_adcR							in<2>	in<3>		vdd_dacR	vssa	B
NC	vss	vss	vss	vss	vss	vss	vss	vss	vss	vss	vss	gnddR	gnddR	vdd_tdcR	gnd_adcR	vdd_adcR	vdd_bufR	vdd_skR	vdd_paR	in<4>	in<5>	gnd_paR	gnd_dacR	vdd_2v5R	C
gndd1	vddd1	vddd2R	vddd2R	vddd2R	vddd2R	VHI10_2R	vddd2R	vddd2R	vddd2R	vddd2R	vddd2R	vddd	vcm_adcR							in<6>	in<7>		vdd_dacR	vref_skR	D
NC	vss	vss	vss	vss	vss	vss	vss	vss	vss	vss	vss	gnddR	gnddR	gnd_tdcR	vrefp_adcR	gnd_adcR	gnd_bufR	gnd_skR	gnd_paR	in<8>	in<9>	gnd_paR	gnd_dacR	vdd_2v5	E
efuse	vddd1	vddd2R	vddd2R	vddd2R	vddd2R	vddd2R	vddd2R	vddd2R	vddd2R	vddd2R	vddd2R	vdddR	vcm_adcR							in<10>	in<11>		vdd_dacR	vref_noinvR	F
NC	vss	vss	vss	vss	vss	vss	vss	vss	vss	vss	vss	gnddR	gnddR	vdd_tdcR	gnd_adcR	vdd_adcR	vdd_bufR	vdd_skR	vdd_paR	in<12>	in<13>	gnd_paR	gnd_dacR	vdd_2v5R	G
NC	vss	vss	vss	vss	vss	vss	vss	vss	vss	vss	vss	vddd	vcm_adcR							in<14>	in<15>		vdd_dacR	vref_invR	H
NC	vddd1	vddd2R	vddd2R	vddd2R	vddd2R	vddd2R	vddd2R	vddd2R	vddd2R	vddd2R	vddd2R	gnddR	gnddR	gnd_tdcR	vrefp_adcR	gnd_adcR	gnd_bufR	gnd_skR	gnd_paR	in<16>	in<17>	gnd_paR	gnd_dacR	vdd_2v5R	J
gndd1	vss	vss	vss	vss	vss	vss	vss	vss	vss	vss	vss	vdddR	vcm_adcR							CM<0>	calib<0>		vdd_dacR	vref_toaR	K
NC	vddd1	vddd2R	vddd2R	vddd2R	vddd2R	vddd2R	vddd2R	vddd2R	vddd2R	vddd2R	vddd2R	gnddR	gnddR	vdd_tdcR	gnd_adcR	vdd_adcR	vdd_bufR	vdd_skR	vdd_paR	gnd_pa	gnd_pa	gnd_paR	gnd_dacR		L
gndd1	vss	vss	vss	vss	vss	vss	vss	vss	vss	vss	vss	vddd	vcm_adcR							CM<1>	NC		vdd_dacR	vref_totR	M
NC	vddd1	vddd2R	vddd2R	vddd2R	vddd2R	vddd2R	vddd2R	vddd2R	vddd2R	vddd2R	vddd2R	gnddR	gnddR	gnd_tdcR	vrefp_adcR	gnd_adcR	gnd_bufR	gnd_skR	gnd_paR	in<19>	in<18>	gnd_paR	gnd_dacR	vdd_2v5R	N
gnd_pll	vss	vss	vss	vss	vss	vss	vss	vss	vss	vss	vss	vdddR	vcm_adcR							in<21>	in<20>		vdd_dacR	vbg_1VR	P
gnd_pll	vddd1	vddd2R	vddd2R	vddd2R	vddd2R	vddd2R	vddd2R	vddd2R	vddd2R	vddd2R	vddd2R	gnddR	gnddR	vdd_tdcR	gnd_adcR	vdd_adcR	vdd_bufR	vdd_skR	vdd_paR	in<23>	in<22>	gnd_paR	gnd_dacR	vdd_2v5R	R
vdd_pll	vss	vss	vss	vss	vss	vss	vss	vss	vss	vss	vss	vddd	vcm_adcR							in<25>	in<24>		vdd_dacR	probe_paR	T
vdd_pll	vddd1	vddd2R	vddd2R	vddd2R	vddd2R	vddd2R	vddd2R	vddd2R	vddd2R	vddd2R	vddd2R	gnddR	gnddR	gnd_tdcR	vrefp_adcR	gnd_adcR	gnd_bufR	gnd_skR	gnd_paR	in<27>	in<26>	gnd_paR	gnd_dacR	vdd_2v5R	U
vdd_sc	vss	vss	vss	vss	vss	vss	vss	vss	vss	vss	vss	vdddR	vcm_adcR							in<29>	in<28>		vdd_dacR	in_ctestR	V
vdd_sc	vss	vss	vss	vss	vss	vss	vss	vss	vss	vss	vss	gnddR	gnddR	vdd_tdcR	gnd_adcR	vdd_adcR	vdd_bufR	vdd_skR	vdd_paR	in<31>	in<30>	gnd_paR	gnd_dacR	vdd_2v5R	W
NC	NC	trig2R	NC	trig1R	NC	VHI10_1R	NC	probe_dc2R	NC	probe_dc1R	NC	vdddR	vcm_adcR							in<33>	in<32>		vdd_dacR	vssa	Y
rstb_i2c	NC	probe_noinvR	NC	probe_invR	NC	probe_toaR	NC	probe_totR	NC	vcm_0p6_inv	NC	gnddR	NC	gnd_tdcR	vrefp_adcR	gnd_adcR	gnd_bufR	gnd_skR	gnd_paR	in<35>	in<34>	gnd_paR	gnd_dacR		Z
25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	

The figure below shows the top view of the BGA map as can be seen on the board.



2.2 Pin list

	Pin Name	Pin type		Comments	Default value
POWERS/GROUNDS	P2V5	POWER		analog pad ring power supply	2,5V
	AVDD_0	POWER		analog power supply	1,2V
	AVDD_1	POWER		analog power supply	1,2V
	GND	GROUND		Analog & digital grounds	0V
	DVDD	POWER		digital power supply	1,2V
	VDD_PLL	POWER		PLL power supply	1,2V
	GND_PLL	GROUND		PLL ground	0V
	VDD_SC	POWER		Slow Control power supply	1,2V
	VHI10_R<0>	ANALOG		RAM2, 10 nF	1V
	VHI10_L<0>	ANALOG		RAM2, 10 nF	1V
	VHI10_R<1>	ANALOG		RAM1, 10 nF	1V
	VHI10_L<1>	ANALOG		RAM2, 10 nF.	1V
Half part (ch. 36 to 71)	Vref_SK_R	ANALOG	Input/Output	Reference voltage of the Sallen-Key amplifier, value close to the preamp input voltage. By default must be kept free. Can be monitored/checked.	200mV
	Vref_noinv_R	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.	230mV
	Vref_inv_R	ANALOG	Input/Output	Reference voltage of the inverted shaper. Value given by an internal 10b_DAC. Can be checked/monitored externally. By default must be kept free.	460mV
	Vref_Toa_R	ANALOG	Input/Output	Reference voltage of the TOA threshold. Value given by an internal 10b_DAC. Can be checked/monitored externally. By default must be kept free.	500mV
	Vref_Tot_R	ANALOG	Input/Output	Reference voltage of the TOT threshold. Value given by an internal 10b_DAC. Can be checked/monitored	400mV

				externally. By default must be kept free.	
	VBG_1V_R	ANALOG	Input/Output	BANDGAP voltage. Value can be tuned by slow-control. Default value = 1V	1V
	Probe_PA_R	ANALOG		preamplifier analog probe. Channel-wise	
	IN_Ctest_R	ANALOG	Input	By default, external pin test, but possibility to connect the calibration DAC output to this pin by slow-control.	
	VrefP_ADC_R	ANALOG	Input/Output	Positive reference voltage of the ADC.	1V
	Vcm_ADC_R	ANALOG	Input/Output	Common mode reference voltage of the ADC.	500mV
	Vcm_0p6_inv_R	ANALOG	Input/Output		600mV
	Probe_DC1_R	ANALOG	Input/Output	DC analog probe (bias, ref.)	
	Probe_DC2_R	ANALOG	Input/Output	DC analog probe (bias, ref.)	
	Probe_Toa_R	CMOS	Output	TOA discri output probe. Channel-wise	
	Probe_Tot_R	CMOS	Output	TOT discri output probe. Channel-wise	
	Trig1_R	CMOS	Input	External trigger 1	
	Trig2_R	CMOS	Input	External trigger 2	
	ADCP_R	ANALOG	Input/Output	Analog non-inverted probe; positive ADC input	100mV
	ADCN_R	ANALOG	Input/Output	Analog inverted probe; negative ADC input	1,1V
	IN<35:0>	ANALOG	Channel input		200mV
	CALIB<0>	ANALOG	Channel input		200mV
	CM<1:0>	ANALOG	Channel input		200mV
	Hard_resetb	CMOS	Input	on power-on-reset, active low	1.2V
	Soft_resetb	CMOS	Input		1.2V
Common pins	Rstb_I2C	CMOS	Input/Output	Hard_resetb output ; debug	1.2V
	SDA	I2C signal		I2C data	
	SCL	I2C clk		I2C clock	
	Error	Open drain		Open collector; external resistor must be added. OR of all the slow-control cells' error signals	
	ADD<3:0>	I2C chip adress		I2C chip address	
	4JM0	CMOS	Input	I2C parameter	
	4JM1	CMOS	Input	I2C parameter	
	Strobe_Ext	CMOS	Input	External calibration signal selectable by slow control E-fuse writing	0V

	SiPM_Calib	CMOS	Output	Calibration signal for SiPM	
	Rstb	CMOS	Input	General reset; ACTIVE LOW	
	EFUSE	ANALOG		pin connected to an internal resistor. HGCROC3 => 100KΩ. HGCROC3A => 2KΩ. H2GCROC3 => 10KΩ. Also 2V5 for e-fuse burning	
	PLL_lock	CMOS	Output		
	Clk320_p	CLPS	Input	Fast command 320 MHz clock	
	Clk320_n	CLPS	Input		
	Fcmd_p	CLPS	Input	Fast command data	
	Fcmd_n	CLPS	Input		
	Clk40_p	CLPS	Input	40MHz clock for debug purpose	
	Clk40_n	CLPS	Input		
	PLL_p	CLPS	Output	PLL output probe	
	PLL_n	CLPS	Output		
	Daq0_p	CLPS	Output	Data 0 link	
	Daq0_n	CLPS	Output		
	Daq1_p	CLPS	Output	Data 1 link	
	Daq1_n	CLPS	Output		
	Trig0_p	CLPS	Output	Trigger 0 link	
	Trig0_n	CLPS	Output		
	Trig1_p	CLPS	Output	Trigger 1 link	
	Trig1_n	CLPS	Output		
	Trig2_p	CLPS	Output	Trigger 2 link	
	Trig2_n	CLPS	Output		
	Trig3_p	CLPS	Output	Trigger 3 link	
	Trig3_n	CLPS	Output		
Half part (ch. 0 to 35)	Vref_SK_L	ANALOG	Input/Output	Reference voltage of the Sallen-Key amplifier, value close to the preamp input voltage. By default must be kept free. Can be monitored/checked.	200mV
	Vref_noinv_L	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.	230mV
	Vref_inv_L	ANALOG	Input/Output	Reference voltage of the inverted shaper. Value given by an internal 10b_DAC. Can be checked/monitored	460mV

				externally. By default must be kept free.	
	Vref_Toa_L	ANALOG	Input/Output	Reference voltage of the TOA threshold. Value given by an internal 10b_DAC. Can be checked/monitored externally. By default must be kept free.	500mV
	Vref_Tot_L	ANALOG	Input/Output	Reference voltage of the TOT threshold. Value given by an internal 10b_DAC. Can be checked/monitored externally. By default must be kept free.	400mV
	VBG_1V_L	ANALOG	Input/Output	BANDGAP voltage. Value can be tuned by slow-control. Default value = 1V	1V
	Probe_PA_L	ANALOG		preamplifier analog probe. Channel-wise	
	IN_Ctest_L	ANALOG	Input	By default, external pin test, but possibility to connect the calibration DAC output to this pin by slow-control.	
	VrefP_ADC_L	ANALOG	Input/Output	Positive reference voltage of the ADC.	1V
	Vcm_ADC_L	ANALOG	Input/Output	Common mode reference voltage of the ADC.	500mV
	Vcm_0p6_inv_L	ANALOG	Input/Output		600mV
	Probe_DC1_L	ANALOG	Input/Output	DC analog probe (bias, ref.)	
	Probe_DC2_L	ANALOG	Input/Output	DC analog probe (bias, ref.)	
	Probe_Toa_L	CMOS	Output	TOA discri output probe. Channel-wise	
	Probe_Tot_L	CMOS	Output	TOT discri output probe. Channel-wise	
	Trig1_L	CMOS	Input	External trigger 1	
	Trig2_L	CMOS	Input	External trigger 2	
	ADCP_L	ANALOG	Input/Output	Analog non-inverted probe; positive ADC input	100mV
	ADCN_L	ANALOG	Input/Output	Analog inverted probe; negative ADC input	1,1V
	IN<71:36>	ANALOG	Channel input		200mV
	CALIB<1>	ANALOG	Channel input		200mV
	CM<3:2>	ANALOG	Channel input		200mV

3 ASIC parameters

As described in section **1.4.2 I2C,** the I2C circuit has 8 internal registers; the four first are dedicated to write/read the slow-control parameters. Registers 4, 5 and 6 are direct access registers and the register 7 is a status register in read-only mode.

In the tables below, the content of the direct access registers is described.

I2C Register R4			
Bit	**Name**	**Default**	**Description**
0	AutoReload	"0"	Allows to rewrite the value after a SEU
1	EdgeSel	"0"	Selection of the clock edge of the 320MHz clock
2	NA	"0"	
3	NA	"0"	
4	NA	"0"	
5	NA	"0"	
6	NA	"0"	
7	NA	"0"	

I2C Register R5			
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"	

I2C Register R6			
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"	

I2C Register R7 (Status , read only)			
Bit	**Name**	**Default**	**Description**
0	Error		OR of all the slow-control cells' error signals
1	Parity		Parity of all the slow-control cells' values
2	Pll_Lock		Pll lock flag. "1" pll is locked
3	NA		

4	NA		
5	NA		
6	NA		
7	NA		

3.1 I2C Addressing

The four first I2C registers, R0-3, allow to define the address and the data to write or read into the internal 8 bits registers distributed into the chip sub-parts. R0 and R1 define the address of internal 8 bits registers. R2 defines the data to write into the chosen internal 8 bits register. R3 allows the user to access successive register address.

The chip, from I2C protocol point of view, is divided in sub-blocks containing maximum 32 registers each. In consequence, the address in 16 bits (given in R0 and R1) is composed of two sub-address:

- The 11 MSB bits code the address of the sub-block
- The 5 LSB bits code the address of the register of the sub-block

The table below gives the address, the name and a short description of all the sub-blocks.

Sub-block name	Sub-block address	Description
CM_2	0	Registers described in "**channel wise**" I2C parameters table
CM_3	1	
Channel_36	2	
Channel_37	3	
Channel_38	4	
Channel_39	5	
Channel_40	6	
Channel_41	7	
Channel_42	8	
Channel_43	9	
Channel_44	10	
Channel_45	11	
Channel_46	12	
Channel_47	13	
Channel_48	14	
Channel_49	15	
Channel_50	16	
Channel_51	17	
Channel_52	18	
Channel_53	19	
Channel_54	20	
Channel_55	21	
Channel_56	22	
Channel_57	23	
Channel_58	24	
Channel_59	25	

Channel_60	26	
Channel_61	27	
Channel_62	28	
Channel_63	29	
Channel_64	30	
Channel_65	31	
Channel_66	32	
Channel_67	33	
Channel_68	34	
Channel_69	35	
Channel_70	36	
Channel_71	37	
CALIB_1	38	
No sub-block		
Reference_Voltage_1	40	Registers described in "**Reference Voltage**" table
Global_Analog_1	41	Registers described in "**Global analog**" table
Master_TDC_1	42	Registers described in "**Master TDC**" table
Digital_Half_1	43	Registers described in "**Digital half**" table
HalfWise_1	44	Registers described in "**channel wise**" I2C parameters table
Top	45	Registers described in "**Top sub-block**" table
CM_0	46	Registers described in "**channel wise**" I2C parameters table
CM_1	47	
Channel_0	48	
Channel_1	49	
Channel_2	50	
Channel_3	51	
Channel_4	52	
Channel_5	53	
Channel_6	54	
Channel_7	55	
Channel_8	56	
Channel_9	57	
Channel_10	58	
Channel_11	59	
Channel_12	60	
Channel_13	61	
Channel_14	62	
Channel_15	63	

Channel_16	64	
Channel_17	65	
Channel_18	66	
Channel_19	67	
Channel_20	68	
Channel_21	69	
Channel_22	70	
Channel_23	71	
Channel_24	72	
Channel_25	73	
Channel_26	74	
Channel_27	75	
Channel_28	76	
Channel_29	77	
Channel_30	78	
Channel_31	79	
Channel_32	80	
Channel_33	81	
Channel_34	82	
Channel_35	83	
CALIB_0	84	
No sub-block		
Reference_Voltage_0	86	Registers described in "**Reference Voltage**" table
Global_Analog_0	87	Registers described in "**Global analog**" table
Master_TDC_0	88	Registers described in "**Master TDC**" table
Digital_Half_0	89	Registers described in "**Digital half**" table
HalfWise_0	90	Registers described in "**channel wise**" I2C parameters table

3.2 **"Channel-wise"** I2C parameters

Register # 0			
Bit	**Name**	**Default**	**Description**
0	Inputdac<0>	"0"	Input DAC value
1	Inputdac<1>	"0"	
2	Inputdac<2>	"0"	
3	Inputdac<3>	"0"	
4	Inputdac<4>	"0"	

5	Inputdac<5>	"0"	
6	Gain_conv<0>	"0"	Conveyor Gain: **<0> = 0.025, <1> = 0.05**
7	Gain_conv<1>	"0"	

Register # 1			
Bit	**Name**	**Default**	**Description**
0	mask_toa	"0"	TOA discri output mask ("1" = masked)
1	sel_trig_toa	"0"	External trigger selection for TOA ("0" = Ext Trig1; "1" = Ext Trig2)
2	trim_toa<0>	"0"	Local 5b-DAC for TOA threshold tuning
3	trim_toa<1>	"0"	
4	trim_toa<2>	"0"	
5	trim_toa<3>	"0"	
6	trim_toa<4>	"0"	
7	trim_toa<5>	"0"	

Register # 2			
Bit	**Name**	**Default**	**Description**
0	NA	"0"	
1	NA	"0"	
2	trim_tot<0>	"0"	Local 5b-DAC for TOT threshold tuning
3	trim_tot<1>	"0"	
4	trim_tot<2>	"0"	
5	trim_tot<3>	"0"	
6	trim_tot<4>	"0"	
7	trim_tot<5>	"0"	

Register # 3			
Bit	**Name**	**Default**	**Description**
0	probe_inv	"0"	Inverter amplifier output probe
1	probe_noinv	"0"	Non inverter amplifier output probe
2	trim_inv<0>	"0"	Local 5b-DAC for ADC pedestal tuning
3	trim_inv<1>	"0"	
4	trim_inv<2>	"0"	
5	trim_inv<3>	"0"	
6	trim_inv<4>	"0"	
7	trim_inv<5>	"0"	

Register # 4			
Bit	**Name**	**Default**	**Description**
0	probe_pa	"0"	Preamplifier output probe
1	LowRange	"0"	0.5pF injection cap
2	HighRange	"0"	8pF injection cap
3	Channel_off	"0"	"1" = preamplifier input tied to ground
4	sel_trig_tot	"0"	External trigger selection for TOT ("0" = Ext Trig1; "1" = Ext Trig2)

5	mask_tot	“0”	TOT discri output mask (“1” = masked)
6	probe_tot	“0”	TOT discri output probe
7	probe_toa	“0”	TOA discri output probe

Register # 5			
Bit	**Name**	**Default**	**Description**
0	DAC_CAL_CTDC_TOT<0>	“0”	Tune the fine gain of the TOT CTDC: 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_TOT<1>	“0”	
2	DAC_CAL_CTDC_TOT<2>	“0”	
3	DAC_CAL_CTDC_TOT<3>	“0”	
4	DAC_CAL_CTDC_TOT<4>	“0”	
5	DAC_CAL_CTDC_TOT<5>	“0”	
6	NA	“0”	
7	mask_adc	“0”	“1” = ADC clock off

Register # 6			
Bit	**Name**	**Default**	**Description**
0	DAC_CAL_CTDC_TOA<0>	“0”	Tune the fine gain of the TOA CTDC: 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	HZ_noinv	“0”	x
7	HZ_inv	“0”	x

Register # 7			
Bit	**Name**	**Default**	**Description**
0	DAC_CAL_FTDC_TOT<0>	“0”	Tune the fine gain of the TOT 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_TOT<1>	“0”	
2	DAC_CAL_FTDC_TOT<2>	“0”	
3	DAC_CAL_FTDC_TOT<3>	“0”	
4	DAC_CAL_FTDC_TOT<4>	“0”	
5	DAC_CAL_FTDC_TOT<5>	“0”	
6	NA	“0”	
7	NA	“0”	

Register # 8			
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<0>	“0”	
2	DAC_CAL_FTDC_TOA<0>	“0”	
3	DAC_CAL_FTDC_TOA<0>	“0”	
4	DAC_CAL_FTDC_TOA<0>	“0”	
5	DAC_CAL_FTDC_TOA<0>	“0”	
6	NA	“0”	

7	NA	"0"	

Register # 9			
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	"0"	
7	NA	"0"	

Register # 10			
Bit	**Name**	**Default**	**Description**
0	IN_FTDC_ENCODER_TOT<0>	"0"	Adjust the ToT 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_TOT<1>	"0"	
2	IN_FTDC_ENCODER_TOT<2>	"0"	
3	IN_FTDC_ENCODER_TOT<3>	"0"	
4	IN_FTDC_ENCODER_TOT<4>	"0"	
5	IN_FTDC_ENCODER_TOT<5>	"0"	
6	NA	"0"	
7	DIS_TDC	"0"	"1" = TDC clocks off

Register # 11			
Bit	**Name**	**Default**	**Description**
0	ExtData<8>	"0"	Forced ADC data, bits 9 & 8
1	ExtData<9>	"0"	
2	NA	"0"	
3	NA	"0"	
4	NA	"0"	
5	NA	"0"	
6	NA	"0"	
7	mask_AlignBuffer	"0"	"1" = AlignBuffer clock off

Register # 12			
Bit	**Name**	**Default**	**Description**
0	Adc_pedestal<0>	"0"	ADC pedestal value
1	Adc_pedestal<1>	"0"	
2	Adc_pedestal<2>	"0"	
3	Adc_pedestal<3>	"0"	
4	Adc_pedestal<4>	"0"	
5	Adc_pedestal<5>	"0"	
6	Adc_pedestal<6>	"0"	
7	Adc_pedestal<7>	"0"	

Register # 13			
Bit	**Name**	**Default**	**Description**
0	ExtData<0>	"0"	Forced ADC data, bits 7 downto 0
1	ExtData<1>	"0"	
2	ExtData<2>	"0"	
3	ExtData<3>	"0"	
4	ExtData<4>	"0"	
5	ExtData<5>	"0"	
6	ExtData<6>	"0"	
7	ExtData<7>	"0"	

Register # 14			
Bit	**Name**	**Default**	**Description**
0	dacb<0>	"0"	Current dac for preamp DC current compensation.
1	dacb<1>	"0"	
2	dacb<2>	"0"	
3	dacb<3>	"0"	
4	dacb<4>	"0"	
5	dacb<5>	"0"	
6	sign_dac	"0"	Sign for Current dac for preamp DC current conmpensation.
7	Gain_conv<2>	"0"	Conveyor Gain: **<2> = 0.1**

3.3 **"Global analog"** I2C parameters

Register # 0			
Bit	**Name**	**Default**	**Description**
0	ON_dac_trim	"1"	"1" = enable DC trimming bias
1	ON_input_dac	"1"	"1" = enable input DAC bias
2	ON_conv	"1"	"1" = enable conveyor bias
3	ON_pa	"1"	"1" = enable preamp bias
4	Idac_tot<0>	"0"	x
5	Idac_tot<1>	"0"	
6	Idac_tot<2>	"1"	
7	Gain_conv<3>	"1"	Conveyor Gain: **<3> = 0.2**

Register # 1			
Bit	**Name**	**Default**	**Description**
0	ON_rtr	"1"	"1" = enable shaper amplifiers bias
1	Sw_super_conv	"1"	"1" = enable preamp fast output bias
2	Idac_inv<0>	"0"	x
3	Idac_inv<1>	"0"	
4	Idac_inv<2>	"1"	
5	Dacb_vb_conv<0>	"0"	Conveyor current bias
6	Dacb_vb_conv<1>	"1"	
7	Dacb_vb_conv<2>	"1"	

Register # 2			
Bit	**Name**	**Default**	**Description**
0	ON_toa	“1”	“1” = enable TOA discri bias
1	ON_tot	“1”	“1” = enable TOT discri bias
2	Dacb_vbi_pa<0>	“0”	6b-DAC for preamp input stage current tuning
3	Dacb_vbi_pa<1>	“0”	
4	Dacb_vbi_pa<2>	“0”	
5	Dacb_vbi_pa<3>	“0”	
6	Dacb_vbi_pa<4>	“0”	
7	Dacb_vbi_pa<5>	“1”	

Register # 3			
Bit	**Name**	**Default**	**Description**
0	Ibi_sk<0>	“0”	S-K amplifier input stage current
1	Ibi_sk<1>	“0”	
2	Ibo_sk<0>	“0”	S-K amplifier output stage current
3	Ibo_sk<1>	“1”	
4	Ibo_sk<2>	“0”	
5	Ibo_sk<3>	“1”	
6	Ibo_sk<4>	“0”	
7	Ibo_sk<5>	“0”	

Register # 4			
Bit	**Name**	**Default**	**Description**
0	Ibi_inv<0>	“0”	Inverter amplifier input stage current
1	Ibi_inv<1>	“0”	
2	Ibo_inv<0>	“0”	Inverter amplifier output stage current
3	Ibo_inv<1>	“1”	
4	Ibo_inv<2>	“0”	
5	Ibo_inv<3>	“1”	
6	Ibo_inv<4>	“0”	
7	Ibo_inv<5>	“0”	

Register # 5			
Bit	**Name**	**Default**	**Description**
0	Ibi_noinv<0>	“0”	Non Inverter amplifier input stage current
1	Ibi_noinv<1>	“0”	
2	Ibo_noinv<0>	“0”	Non Inverter amplifier output stage current
3	Ibo_noinv<1>	“1”	
4	Ibo_noinv<2>	“0”	
5	Ibo_noinv<3>	“1”	
6	Ibo_noinv<4>	“0”	
7	Ibo_noinv<5>	“0”	

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

Register # 7			
Bit	**Name**	**Default**	**Description**
0	Ibi_noinv_buf<0>	"0"	Non Inverter buffer input stage current
1	Ibi_noinv_buf<1>	"1"	
2	Ibo_noinv_buf<0>	"0"	Non Inverter buffer output stage current
3	Ibo_noinv_buf<1>	"1"	
4	Ibo_noinv_buf<2>	"1"	
5	Ibo_noinv_buf<3>	"0"	
6	Ibo_noinv_buf<4>	"0"	
7	Ibo_noinv_buf<5>	"1"	

Register # 8			
Bit	**Name**	**Default**	**Description**
0	Cd<0>	"1"	Internal capacitors in // : <0>=5pF; <1>=10pF; <2>=20pF
1	Cd<1>	"1"	
2	Cd<2>	"1"	
3	En_hyst_tot	"1"	"1" = enable the TOT discri hysteresis
4	Cf_comp<0>	"0"	Preamp feedback comp. cap. <0> = 50fF; <1> = 100fF; <2> = 200fF; <3> = 400fF
5	Cf_comp<1>	"1"	
6	Cf_comp<2>	"0"	
7	Cf_comp<3>	"1"	

Register # 9			
Bit	**Name**	**Default**	**Description**
0	Cf<0>	"0"	Preamp feedback cap. <0> = 50fF, <1> = 100fF, <2> = 200fF, <3> = 400fF
1	Cf<1>	"1"	
2	Cf<2>	"0"	
3	Cf<3>	"1"	
4	Rf<0>	"0"	Preamp feedback Res. <0> = 100K, <1> = 66.66K, <2> = 50K, <3> = 25K
5	Rf<1>	"0"	
6	Rf<2>	"0"	
7	Rf<3>	"1"	

Register # 10			
Bit	**Name**	**Default**	**Description**

0	Ref_pa_cm<0>	“1”	Configuration of the reference voltage in the compensation common mode channel: <00> = ~158mV, <01> = ~205mV, <10> = ~205mV, <11> = ~246mV
1	Ref_pa_cm<1>	“1”	
2	ON_backup	“1”	“1” = disable compensation with common mode channel. ➔ **error**
3	SelExtADC	“1”	“1” = Forced ADC data send to the DRAM ➔ **error**
4	Clr_ADC	“0”	
5	S_sk<0>	“0”	S-K amp Miller cap. <0> = 50fF, <1> = 100fF, <2> = 200fF
6	S_sk<1>	“1”	
7	S_sk<2>	“0”	

Register # 11			
Bit	**Name**	**Default**	**Description**
0	Clr_ShaperTail	“0”	
1	SelRisingEdge	“0”	“1” = AlignBuffer provides data on rising edge ➔**error**
2	S_inv<0>	“0”	Inverter amp Miller cap. <0> = 50fF, <1> = 100fF, <2> = 200fF
3	S_inv<1>	“1”	
4	S_inv<2>	“0”	
5	S_noinv<0>	“0”	Non Inverter amp Miller cap. <0> = 50fF, <1> = 100fF, <2> = 200fF
6	S_noinv<1>	“1”	
7	S_noinv<2>	“0”	

Register # 12			
Bit	**Name**	**Default**	**Description**
0	Dacb_dynran<0>	“1”	Configuration of dacb dynamic range (See following table)
1	Dacb_dynran<1>	“1”	
2	S_inv_buf<0>	“0”	Inverter buffer Miller cap. <0> = 100fF, <1> = 200fF, <2> = 400fF
3	S_inv_buf<1>	“1”	
4	S_inv_buf<2>	“0”	
5	S_noinv_buf<0>	“0”	Non Inverter buffer Miller cap. <0> = 100fF, <1> = 200fF, <2> = 400fF
6	S_noinv_buf<1>	“1”	
7	S_noinv_buf<2>	“0”	

Register # 13			
Bit	**Name**	**Default**	**Description**
0	ref_adc<0>	“0”	Input stage current of the Ref ADC OTA
1	ref_adc<1>	“0”	
2	Delay40<0>	“0”	Delay tuning for bits <4:0> “000” = faster conversion
3	Delay40<1>	“0”	
4	Delay40<2>	“0”	
5	Delay65<0>	“0”	Delay tuning for bits <6:5> “000” = faster conversion
6	Delay65<1>	“0”	

7	Delay65<2>	“0”	

Register # 14			
Bit	**Name**	**Default**	**Description**
0	ON_ref_adc	“1”	“1” = enable ADC ref OTA
1	NA	“1”	
2	Delay87<0>	“0”	Delay tuning for bits <8:7> “000” = faster conversion
3	Delay87<1>	“0”	
4	Delay87<2>	“0”	
5	Delay9<0>	“0”	Delay tuning for bit <9> “000” = faster conversion
6	Delay9<1>	“0”	
7	Delay9<2>	“0”	

3.4 “Reference Voltage” I2C parameters

Register # 0			
Bit	**Name**	**Default**	**Description**
0	NA	“0”	
1	NA	“0”	
2	Refi<0>	“1”	Bandgap current tuning
3	Refi<1>	“1”	
4	Vbg_1v<0>	“1”	1V bandgap ref. tuning
5	Vbg_1v<1>	“0”	
6	Vbg_1v<2>	“1”	
7	ON_dac	“1”	“1” = enable DACs

Register # 1			
Bit	**Name**	**Default**	**Description**
0	Noinv_vref<0>	“0”	Non Inverter shaper global reference <1:0>
1	Noinv_vref<1>	“0”	
2	Inv_vref<0>	“0”	Inverter shaper global reference <1:0>
3	Inv_vref<1>	“0”	
4	Toa_vref<0>	“0”	TOA threshold global value <1:0>
5	Toa_vref<1>	“0”	
6	Tot_vref<0>	“0”	TOT threshold global value <1:0>
7	Tot_vref<1>	“0”	

Register # 2			
Bit	**Name**	**Default**	**Description**
0	Tot_vref<2>	“0”	TOT threshold global value
1	Tot_vref<3>	“0”	
2	Tot_vref<4>	“1”	
3	Tot_vref<5>	“1”	
4	Tot_vref<6>	“0”	
5	Tot_vref<7>	“1”	
6	Tot_vref<8>	“1”	

7	Tot_vref<9>	“0”	

Register # 3			
Bit	**Name**	**Default**	**Description**
0	Toa _vref<2>	“0”	TOA threshold global value
1	Toa _vref<3>	“0”	
2	Toa _vref<4>	“1”	
3	Toa _vref<5>	“1”	
4	Toa _vref<6>	“1”	
5	Toa _vref<7>	“0”	
6	Toa _vref<8>	“0”	
7	Toa _vref<9>	“0”	

Register # 4			
Bit	**Name**	**Default**	**Description**
0	Inv_vref<2>	“0”	Inverter shaper global reference
1	Inv_vref<3>	“0”	
2	Inv_vref<4>	“0”	
3	Inv_vref<5>	“0”	
4	Inv_vref<6>	“0”	
5	Inv_vref<7>	“0”	
6	Inv_vref<8>	“1”	
7	Inv_vref<9>	“0”	

Register # 5			
Bit	**Name**	**Default**	**Description**
0	Noinv_vref<2>	“1”	Non Inverter shaper global reference
1	Noinv_vref<3>	“1”	
2	Noinv_vref<4>	“1”	
3	Noinv_vref<5>	“1”	
4	Noinv_vref<6>	“0”	
5	Noinv_vref<7>	“0”	
6	Noinv_vref<8>	“1”	
7	Noinv_vref<9>	“0”	

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

Register # 7			
Bit	**Name**	**Default**	**Description**
0	Calib<8>	"0"	Calibration DAC value
1	Calib<9>	"0"	
2	Calib<10>	"0"	
3	Calib<11>	"0"	
4	NA	"0"	
5	NA	"0"	
6	IntCtest	"0"	Selection of the Calibration DAC
7	ExtCtest	"0"	Selection of the external pulse test

Register # 8			
Bit	**Name**	**Default**	**Description**
0	probe_dc<0>	"0"	Value for probe dc selected
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"	Selection of probe dc1_0
6	probe_dc1_1	"0"	Selection of probe dc1_1
7	probe_dc2	"0"	Selection of probe dc2

Register # 9			
Bit	**Name**	**Default**	**Description**
0	Calib_2V5<0>	"0"	
1	Calib_2V5<1>	"0"	
2	Calib_2V5<2>	"0"	
3	Calib_2V5<3>	"0"	
4	Calib_2V5<4>	"0"	
5	Calib_2V5<5>	"0"	
6	Calib_2V5<6>	"0"	
7	Calib_2V5<7>	"0"	

Register # 10			
Bit	**Name**	**Default**	**Description**
0	Calib_2V5<8>	"0"	
1	Calib_2V5<9>	"0"	
2	Calib_2V5<10>	"0"	
3	Calib_2V5<11>	"0"	
4	NA	"0"	
5	ExtCtest_2V5	"0"	
6	choice_cinj	"0"	
7	cmd_120p	"0"	

3.5 “Master TDC” I2C parameters

Register # 0			
Bit	**Name**	**Default**	**Description**
0	GLOBAL_TA_SELECT_GAIN_TOA<0>	“1”	
1	GLOBAL_TA_SELECT_GAIN_TOA<1>	“1”	
2	GLOBAL_TA_SELECT_GAIN_TOA<2>	“1”	
3	GLOBAL_TA_SELECT_GAIN_TOA<3>	“0”	
4	GLOBAL_TA_SELECT_GAIN_TOT<0>	“1”	
5	GLOBAL_TA_SELECT_GAIN_TOT<1>	“1”	
6	GLOBAL_TA_SELECT_GAIN_TOT<2>	“0”	
7	GLOBAL_TA_SELECT_GAIN_TOT<3>	“0”	

Register # 1			
Bit	**Name**	**Default**	**Description**
0	GLOBAL_MODE_NO_TOT_SUB	“0”	
1	GLOBAL_LATENCY_TIME<0>	“0”	
2	GLOBAL_LATENCY_TIME<1>	“1”	
3	GLOBAL_LATENCY_TIME<2>	“0”	
4	GLOBAL_LATENCY_TIME<3>	“1”	
5	GLOBAL_MODE_FTDC_TOA_S0	“0”	
6	GLOBAL_MODE_FTDC_TOA_S1	“1”	
7	GLOBAL_SEU_TIME_OUT	“1”	

Register # 2			
Bit	**Name**	**Default**	**Description**
0	BIAS_FOLLOWER_CAL_P_CTDC_D<0>	“0”	
1	BIAS_FOLLOWER_CAL_P_ CTDC_D<1>	“0”	
2	BIAS_FOLLOWER_CAL_P_ CTDC_D<2>	“0”	
3	BIAS_FOLLOWER_CAL_P_ CTDC_D<3>	“0”	
4	BIAS_FOLLOWER_CAL_P_CTDC_EN	“0”	
5	INV_FRONT_40MHZ	“0”	
6	START_COUNTER	“1”	
7	CALIB_CHANNEL_DLL	“0”	

Register # 3			
Bit	**Name**	**Default**	**Description**
0	VD_CTDC_P_D<0>	“0”	
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”	
6	EN_MASTER_CTDC_VOUT_INIT	“0”	
7	EN_MASTER_CTDC_DLL	“1”	

Register # 4			
Bit	**Name**	**Default**	**Description**
0	BIAS_CAL_DAC_CTDC_P_D<0>	"0"	
1	BIAS_CAL_DAC_CTDC_P_D<1>	"0"	
2	CTDC_CALIB_FREQUENCY<0>	"0"	
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"	
1	BIAS_I_CTDC_D<0>	"0"	
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"	

Register # 6			
Bit	**Name**	**Default**	**Description**
0	GLOBAL_EN_BUFFER_CTDC	"0"	
1	VD_CTDC_N_FORCE_MAX	"1"	
2	VD_CTDC_N_D<0>	"0"	
3	VD_CTDC_N_D<1>	"0"	
4	VD_CTDC_N_D<2>	"0"	
5	VD_CTDC_N_D<3>	"0"	
6	VD_CTDC_N_D<4>	"0"	
7	VD_CTDC_N_DAC_EN	"0"	

Register # 7			
Bit	**Name**	**Default**	**Description**
0	CTRL_IN_REF_CTDC_P_D<0>	"0"	
1	CTRL_IN_REF_CTDC_P_D<1>	"0"	
2	CTRL_IN_REF_CTDC_P_D<2>	"0"	
3	CTRL_IN_REF_CTDC_P_D<3>	"0"	
4	CTRL_IN_REF_CTDC_P_D<4>	"0"	
5	CTRL_IN_REF_CTDC_P_EN	"0"	
6	BIAS_CAL_DAC_CTDC_P_D<2>	"0"	
7	BIAS_CAL_DAC_CTDC_P_D<3>	"0"	

Register # 8			
Bit	**Name**	**Default**	**Description**
0	CTRL_IN_SIG_CTDC_P_D<0>	“0”	
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”	
6	GLOBAL_INIT_DAC_B_CTDC	“0”	
7	BIAS_CAL_DAC_CTDC_P_EN	“0”	

Register # 9			
Bit	**Name**	**Default**	**Description**
0	VD_FTDC_P_D<0>	“0”	
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”	
6	EN_MASTER_FTDC_VOUT_INIT	“0”	
7	EN_MASTER_FTDC_DLL	“1”	

Register # 10			
Bit	**Name**	**Default**	**Description**
0	NA	“0”	
1	BIAS_FOLLOWER_CAL_P_FTDC_EN	“0”	
2	FTDC_CALIB_FREQUENCY<0>	“0”	
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”	

Register # 11			
Bit	**Name**	**Default**	**Description**
0	EN_REF_BG	“1”	
1	BIAS_I_FTDC_D<0>	“0”	
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”	

Register # 12			
Bit	**Name**	**Default**	**Description**

0	GLOBAL_EN_BUFFER_FTDC	“0”	
1	VD_FTDC_N_FORCE_MAX	“1”	
2	VD_FTDC_N_D<0>	“0”	
3	VD_FTDC_N_D<1>	“0”	
4	VD_FTDC_N_D<2>	“0”	
5	VD_FTDC_N_D<3>	“0”	
6	VD_FTDC_N_D<4>	“0”	
7	VD_FTDC_N_DAC_EN	“0”	

Register # 13			
Bit	**Name**	**Default**	**Description**
0	CTRL_IN_SIG_FTDC_P_D<0>	“0”	
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”	
6	BIAS_FOLLOWER_CAL_P_FTDC_D<0>	“0”	
7	BIAS_FOLLOWER_CAL_P_FTDC_D<1>	“0”	

Register # 14			
Bit	**Name**	**Default**	**Description**
0	CTRL_IN_REF_FTDC_P_D<0>	“0”	
1	CTRL_IN_REF_FTDC_P_D<1>	“0”	
2	CTRL_IN_REF_FTDC_P_D<2>	“0”	
3	CTRL_IN_REF_FTDC_P_D<3>	“0”	
4	CTRL_IN_REF_FTDC_P_D<4>	“0”	
5	CTRL_IN_REF_FTDC_P_EN	“0”	
6	BIAS_FOLLOWER_CAL_P_FTDC_D<2>	“0”	
7	BIAS_FOLLOWER_CAL_P_FTDC_D<3>	“0”	

Register # 15			
Bit	**Name**	**Default**	**Description**
0	GLOBAL_DISABLE_TOT_LIMIT	“0”	
1	GLOBAL_FORCE_EN_CLK	“0”	
2	GLOBAL_FORCE_EN_OUTPUT_DATA	“0”	
3	GLOBAL_FORCE_EN_TOT	“0”	
4	GLOBAL_EN_TOT_PRIORITY	“0”	
5	GLOBAL_EN_TUNE_GAIN_DAC	“0”	
6	sel_clk_rcg<0>	“0”	
7	sel_clk_rcg<1>	“0”	

See Annexe A for more details about TDCs parameters.

3.6 “Digital half” I2C parameters

Register # 0			
Bit	**Name**	**Default**	**Description**
0	ClrAdcTot_trig<0>	"0"	
1	ClrAdcTot_trig<1>	"0"	
2	ClrAdcTot_trig<2>	"0"	
3	ClrAdcTot_trig<3>	"0"	
4	ClrAdcTot_trig<4>	"0"	
5	ClrAdcTot_trig<5>	"0"	
6	ClrAdcTot_trig<6>	"0"	
7	ClrAdcTot_trig<7>	"0"	

Register # 1			
Bit	**Name**	**Default**	**Description**
0	ClrAdcTot_trig<8>	"0"	
1	ClrAdcTot_trig<9>	"0"	
2	ClrAdcTot_trig<10>	"0"	
3	ClrAdcTot_trig<11>	"0"	
4	ClrAdcTot_trig<12>	"0"	
5	ClrAdcTot_trig<13>	"0"	
6	ClrAdcTot_trig<14>	"0"	
7	ClrAdcTot_trig<15>	"0"	

Register # 2			
Bit	**Name**	**Default**	**Description**
0	ClrAdcTot_trig<16>	"0"	
1	ClrAdcTot_trig<17>	"0"	
2	ClrAdcTot_trig<18>	"0"	
3	ClrAdcTot_trig<19>	"0"	
4	ClrAdcTot_trig<20>	"0"	
5	ClrAdcTot_trig<21>	"0"	
6	ClrAdcTot_trig<22>	"0"	
7	ClrAdcTot_trig<23>	"0"	

Register # 3			
Bit	**Name**	**Default**	**Description**
0	ClrAdcTot_trig<24>	"0"	
1	ClrAdcTot_trig<25>	"0"	
2	ClrAdcTot_trig<26>	"0"	
3	ClrAdcTot_trig<27>	"0"	
4	ClrAdcTot_trig<28>	"0"	
5	ClrAdcTot_trig<29>	"0"	
6	ClrAdcTot_trig<30>	"0"	
7	ClrAdcTot_trig<31>	"0"	

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

0	ClrAdcTot_trig<32>	“0”	
1	ClrAdcTot_trig<33>	“0”	
2	ClrAdcTot_trig<34>	“0”	
3	ClrAdcTot_trig<35>	“0”	
4	Top_P_Add	“0”	
5	SC_testRAM	“0”	
6	CalibrationSC	“0”	
7	SelTC4	“1”	

Register # 5			
Bit	**Name**	**Default**	**Description**
0	Tot_TH3<0>	“0”	TOT threshold used in TP (common to 9 channels)
1	Tot_TH3<1>	“0”	
2	Tot_TH3<2>	“0”	
3	Tot_TH3<3>	“0”	
4	Tot_TH3<4>	“0”	
5	Tot_TH3<5>	“0”	
6	Tot_TH3<6>	“0”	
7	Tot_TH3<7>	“0”	

Register # 6			
Bit	**Name**	**Default**	**Description**
0	Tot_TH2<0>	“0”	TOT threshold used in TP (common to 9 channels)
1	Tot_TH2<1>	“0”	
2	Tot_TH2<2>	“0”	
3	Tot_TH2<3>	“0”	
4	Tot_TH2<4>	“0”	
5	Tot_TH2<5>	“0”	
6	Tot_TH2<6>	“0”	
7	Tot_TH2<7>	“0”	

Register # 7			
Bit	**Name**	**Default**	**Description**
0	Tot_TH1<0>	“0”	TOT threshold used in TP (common to 9 channels)
1	Tot_TH1<1>	“0”	
2	Tot_TH1<2>	“0”	
3	Tot_TH1<3>	“0”	
4	Tot_TH1<4>	“0”	
5	Tot_TH1<5>	“0”	
6	Tot_TH1<6>	“0”	
7	Tot_TH1<7>	“0”	

Register # 8			
Bit	**Name**	**Default**	**Description**
0	Tot_TH0<0>	“0”	TOT threshold used in TP (common to 9 channels)
1	Tot_TH0<1>	“0”	

2	Tot_TH0<2>	“0”	
3	Tot_TH0<3>	“0”	
4	Tot_TH0<4>	“0”	
5	Tot_TH0<5>	“0”	
6	Tot_TH0<6>	“0”	
7	Tot_TH0<7>	“0”	

Register # 9			
Bit	**Name**	**Default**	**Description**
0	Tot_P3<0>	“0”	TOT pedestal used in TP (common to 9 channels)
1	Tot_P3<1>	“0”	
2	Tot_P3<2>	“0”	
3	Tot_P3<3>	“0”	
4	Tot_P3<4>	“0”	
5	Tot_P3<5>	“0”	
6	Tot_P3<6>	“0”	
7	NA	“0”	

Register # 10			
Bit	**Name**	**Default**	**Description**
0	Tot_P2<0>	“0”	TOT pedestal used in TP (common to 9 channels)
1	Tot_P2<1>	“0”	
2	Tot_P2<2>	“0”	
3	Tot_P2<3>	“0”	
4	Tot_P2<4>	“0”	
5	Tot_P2<5>	“0”	
6	Tot_P2<6>	“0”	
7	NA	“0”	

Register # 11			
Bit	**Name**	**Default**	**Description**
0	Tot_P1<0>	“0”	TOT pedestal used in TP (common to 9 channels)
1	Tot_P1<1>	“0”	
2	Tot_P1<2>	“0”	
3	Tot_P1<3>	“0”	
4	Tot_P1<4>	“0”	
5	Tot_P1<5>	“0”	
6	Tot_P1<6>	“0”	
7	NA	“0”	

Register # 12			
Bit	**Name**	**Default**	**Description**
0	Tot_P0<0>	“0”	TOT pedestal used in TP (common to 9 channels)
1	Tot_P0<1>	“0”	
2	Tot_P0<2>	“0”	
3	Tot_P0<3>	“0”	

4	Tot_P0<4>	“0”	
5	Tot_P0<5>	“0”	
6	Tot_P0<6>	“0”	
7	NA	“0”	

Register # 13			
Bit	**Name**	**Default**	**Description**
0	MultFactor<0>	“1”	
1	MultFactor<1>	“0”	
2	MultFactor<2>	“0”	
3	MultFactor<3>	“1”	
4	MultFactor<4>	“1”	
5	NA	“0”	
6	NA	“0”	
7	NA	“0”	

Register # 14			
Bit	**Name**	**Default**	**Description**
0	Adc_TH<0>	“0”	
1	Adc_TH<1>	“0”	
2	Adc_TH<2>	“0”	
3	Adc_TH<3>	“0”	
4	Adc_TH<4>	“0”	
5	NA	“0”	
6	NA	“0”	
7	L1Offset<8>	“0”	

Register # 15			
Bit	**Name**	**Default**	**Description**
0	L1Offset<0>	“0”	
1	L1Offset<1>	“0”	
2	L1Offset<2>	“0”	
3	L1Offset<3>	“1”	
4	L1Offset<4>	“0”	
5	L1Offset<5>	“0”	
6	L1Offset<6>	“0”	
7	L1Offset<7>	“0”	

Register # 16			
Bit	**Name**	**Default**	**Description**
0	IdleFrame<0>	“0”	
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 # 17			
Bit	**Name**	**Default**	**Description**
0	IdleFrame<8>	“0”	
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 # 18			
Bit	**Name**	**Default**	**Description**
0	IdleFrame<16>	“0”	
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 # 19			
Bit	**Name**	**Default**	**Description**
0	IdleFrame<24>	“0”	
1	IdleFrame<25>	“0”	
2	IdleFrame<26>	“1”	
3	IdleFrame<27>	“1”	
4	NA	“0”	
5	NA	“0”	
6	NA	“0”	
7	NA	“0”	

Register # 20			
Bit	**Name**	**Default**	**Description**
0	sc_testRAM<0>	“0”	
1	sc_testRAM<1>	“0”	
2	sc_testRAM<2>	“1”	
3	sc_testRAM<3>	“1”	
4	sc_testRAM<4>	“0”	
5	sc_testRAM<5>	“0”	
6	sc_testRAM<6>	“1”	
7	sc_testRAM<7>	“1”	

Register # 21			
Bit	**Name**	**Default**	**Description**
0	sc_testRAM<8>	"0"	
1	sc_testRAM<9>	"0"	
2	sc_testRAM<10>	"1"	
3	sc_testRAM<11>	"1"	
4	sc_testRAM<12>	"0"	
5	sc_testRAM<13>	"0"	
6	sc_testRAM<14>	"1"	
7	sc_testRAM<15>	"1"	

Register # 22			
Bit	**Name**	**Default**	**Description**
0	sc_testRAM<16>	"0"	
1	sc_testRAM<17>	"0"	
2	sc_testRAM<18>	"1"	
3	sc_testRAM<19>	"1"	
4	sc_testRAM<20>	"0"	
5	sc_testRAM<21>	"0"	
6	sc_testRAM<22>	"1"	
7	sc_testRAM<23>	"1"	

Register # 23			
Bit	**Name**	**Default**	**Description**
0	sc_testRAM<24>	"0"	
1	sc_testRAM<25>	"0"	
2	sc_testRAM<26>	"1"	
3	sc_testRAM<27>	"1"	
4	sc_testRAM<28>	"0"	
5	sc_testRAM<29>	"0"	
6	sc_testRAM<30>	"1"	
7	sc_testRAM<31>	"1"	

Register # 24			
Bit	**Name**	**Default**	**Description**
0	Bx_trigger<0>	"1"	
1	Bx_trigger<1>	"1"	
2	Bx_trigger<2>	"1"	
3	Bx_trigger<3>	"1"	
4	Bx_trigger<4>	"0"	
5	Bx_trigger<5>	"0"	
6	Bx_trigger<6>	"0"	
7	Bx_trigger<7>	"0"	

Register # 25

Bit	Name	Default	Description
0	Bx_offset<0>	"1"	
1	Bx_offset<1>	"0"	
2	Bx_offset<2>	"0"	
3	Bx_offset<3>	"0"	
4	Bx_offset<4>	"0"	
5	Bx_offset<5>	"0"	
6	Bx_offset<6>	"0"	
7	Bx_offset<7>	"0"	

Register # 26			
Bit	**Name**	**Default**	**Description**
0	Bx_offset<8>	"0"	
1	Bx_offset<9>	"0"	
2	Bx_offset<10>	"0"	
3	Bx_offset<11>	"0"	
4	Bx_trigger<8>	"0"	
5	Bx_trigger<9>	"0"	
6	Bx_trigger<10>	"0"	
7	Bx_trigger<11>	"0"	

3.7 "Top sub-block" I2C parameters

Register # 0			
Bit	**Name**	**Default**	**Description**
0	RunR	"0"	
1	RunL	"0"	
2	TestMode	"0"	
3	n_counter_rst	"1"	
4	EdgeSel_T1	"0"	
5	in_inv_cmd_rx	"0"	
6	PreL1AOffset<0>	"0"	
7	PreL1AOffset<1>	"0"	

Register # 1			
Bit	**Name**	**Default**	**Description**
0	EN_PLL	"1"	Active the PLL to provide the different internal clks : '0' : OFF : No clk "power off channels" **'1' : ON : PLL activated**
1	DIV_PLL<0>	"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_PL<1>	"1"	
3	FOLLOWER_PLL_EN	"1"	
4	sel_strobe_ext	"0"	
5	sel_40M_ext	"0"	
6	sel_error	"0"	

7	sel_lock	“0”	

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”	
7	VOUR_INIT_EN	“0”	

Register # 3			
Bit	**Name**	**Default**	**Description**
0	EN_HIGH_CAPA	“1”	
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**)
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”	

Register # 4			
Bit	**Name**	**Default**	**Description**
0	INIT_DAC_EN	“0”	
1	INIT_D<0>	“0”	
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”	
7	rcg_gain<1>	“0”	

Register # 5			
Bit	**Name**	**Default**	**Description**
0	EN1	“1”	Current value of the CLPS drivers (Data/Trigger)
1	EN2	“1”	
2	EN3	“0”	
3	EN-pE0	“0”	Current value of the CLPS pre-emphasis driver (Data/Trigger)
4	EN-pE1	“0”	
5	EN-pE2	“0”	
6	S0	“0”	Delay value of the CLPS pre-emphasis driver (Data/Trigger)
7	S1	“0”	

Register # 6			
Bit	**Name**	**Default**	**Description**
0	EN_LOCK_CONTROL	“1”	
1	ERROR_LIMIT_SC<0>	“0”	
2	ERROR_LIMIT_SC<1>	“1”	
3	ERROR_LIMIT_SC<2>	“0”	
4	NA	“0”	
5	PLL_Locked_sc	“0”	
6	EN_probe_pll	“0”	
7	En_PhaseShift	“1”	

Register # 7			
Bit	**Name**	**Default**	**Description**
0	phase_ck<0>	“0”	
1	phase_ck<1>	“0”	
2	phase_ck<2>	“0”	
3	phase_ck<3>	“0”	
4	phase_strobe<0>	“0”	
5	phase_strobe<1>	“0”	
6	phase_strobe<2>	“0”	
7	phase_strobe<3>	“0”	

Register # 8			
Bit	**Name**	**Default**	**Description**
0	bit_out<0>		
1	bit_out<1>		
2	bit_out<2>		
3	bit_out<3>		
4	bit_out<4>		
5	bit_out<5>		
6	bit_out<6>		
7	bit_out<7>		

Register # 9			
Bit	**Name**	**Default**	**Description**
0	bit_in<0>	“0”	
1	bit_in<1>	“0”	
2	bit_in<2>	“0”	
3	bit_in<3>	“0”	
4	bit_in<4>	“0”	
5	bit_in<5>	“0”	
6	bit_in<6>	“0”	
7	bit_in<7>	“0”	

Register # 10

Bit	Name	Default	Description
0	bit_out<8>		
1	bit_out<9>		
2	bit_out<10>		
3	bit_out<11>		
4	bit_out<12>		
5	bit_out<13>		
6	bit_out<14>		
7	bit_out<15>		

Register # 11			
Bit	**Name**	**Default**	**Description**
0	bit_in<8>	“0”	
1	bit_in<9>	“0”	
2	bit_in<10>	“0”	
3	bit_in<11>	“0”	
4	bit_in<12>	“0”	
5	bit_in<13>	“0”	
6	bit_in<14>	“0”	
7	bit_in<15>	“0”	

Register # 12			
Bit	**Name**	**Default**	**Description**
0	bit_out<16>		
1	bit_out<17>		
2	bit_out<18>		
3	bit_out<19>		
4	bit_out<20>		
5	bit_out<21>		
6	bit_out<22>		
7	bit_out<23>		

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

Register # 14			
Bit	**Name**	**Default**	**Description**
0	bit_out<24>		

1	bit_out<25>		
2	bit_out<26>		
3	bit_out<27>		
4	bit_out<28>		
5	bit_out<29>		
6	bit_out<30>		
7	srout		

Register # 15			
Bit	**Name**	**Default**	**Description**
0	bit_in<24>	"0"	
1	bit_in<25>	"0"	
2	bit_in<26>	"0"	
3	bit_in<27>	"0"	
4	bit_in<28>	"0"	
5	bit_in<29>	"0"	
6	bit_in<30>	"0"	
7	bit_in<31>	"0"	

Register # 16			
Bit	**Name**	**Default**	**Description**
0	statusR<0>		
1	statusR<1>		
2	statusR<2>		
3	statusR<3>		
4	statusL<0>		
5	statusL<1>		
6	statusL<2>		
7	statusL<3>		

Register # 17			
Bit	**Name**	**Default**	**Description**
0	lock_count<0>		
1	Lock_count<1>		
2	lock_count<2>		
3	lock_count<3>		
4	lock_count<4>		
5	lock_count<5>		
6	lock_count<6>		
7	lock_count<7>		

Register # 18			
Bit	**Name**	**Default**	**Description**
0	fc_error_count<0>		
1	fc_error_count<1>		
2	fc_error_count<2>		

3	fc_error_count<3>		
4	fc_error_count<4>		
5	fc_error_count<5>		
6	fc_error_count<6>		
7	fc_error_count<7>		

Register # 19			
Bit	**Name**	**Default**	**Description**
0	err_countL<0>		
1	err_countL<1>		
2	err_countL<2>		
3	err_countL<3>		
4	err_countL<4>		
5	err_countL<5>		
6	err_countL<6>		
7	err_countL<7>		

Register # 20			
Bit	**Name**	**Default**	**Description**
0	err_countR<0>		
1	err_countR<1>		
2	err_countR<2>		
3	err_countR<3>		
4	err_countR<4>		
5	err_countR<5>		
6	err_countR<6>		
7	err_countR<7>		

4 HGCROC3 design-review reports

Analog parts design review on Monday 30th November 2020:

https://indico.cern.ch/event/975335/

Digital parts design review on Tuesday 1 December 2020:

https://indico.cern.ch/event/972739/