RD53 pixel chips for ATLAS and CMS

Bergen - Sep 20th 2022

TWEPP 2022 Topical Workshop on Electronics for Particle Physics
https://indico.cern.ch/event/1127562

Roberto Beccherle, INFN - Pisa, **On behalf of the RD53 Collaboration**

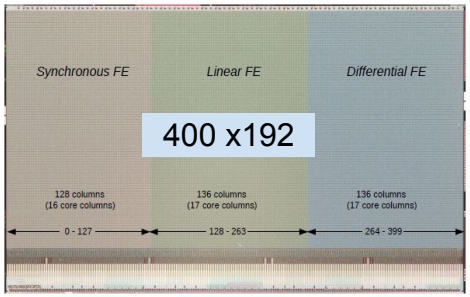
RD53 collaboration

It is a joint effort between ATLAS and CMS to develop readout chips for the HL-LHC pixel detectors:

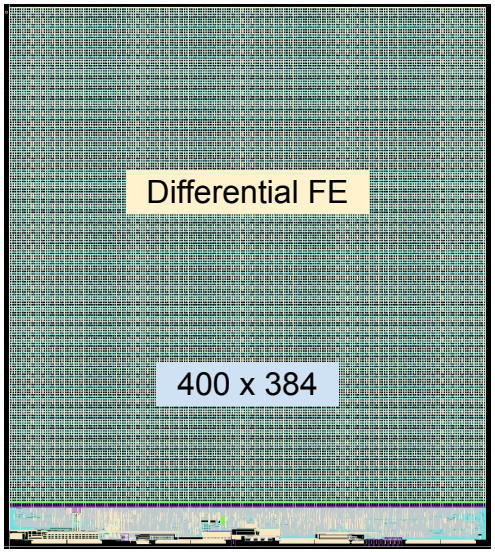
1. Characterization of chosen 65nm CMOS technology in radiation environment.
2. Design of a rad-hard IP library (Analog front-ends, DACs, ADCs, CDR/PLL, high-speed serializers, RX/TX, ShuntLDO, ...)
3. Design and characterization of half-size pixel chip demonstrator (**RD53A**) with design variations to investigate Analog front-ends and digital core architecture.
4. Design of pre-production (**RD53B**) and production (**RD53C**) pixel readout chips: ATLAS and CMS chips are two instances of the same common design, having different size and Analog Front-Ends, according to specific requirements by the experiments. Size (number of Cores) is a parameter in the RTL code
   Analog Front-End are two IP blocks with same size for both experiments

	ATLAS/CMS
Chip size	20x21mm²/21.6x18.6mm²
Pixel size	50x50 µm²
Hit rate	3 GHz/cm²
Trigger rate	1 MHz/750kHz
Trigger latency	12.5 us
Min. threshold	600 e-
Radiation tolerance	500 Mrad @-15C
Power	< 1W/cm²

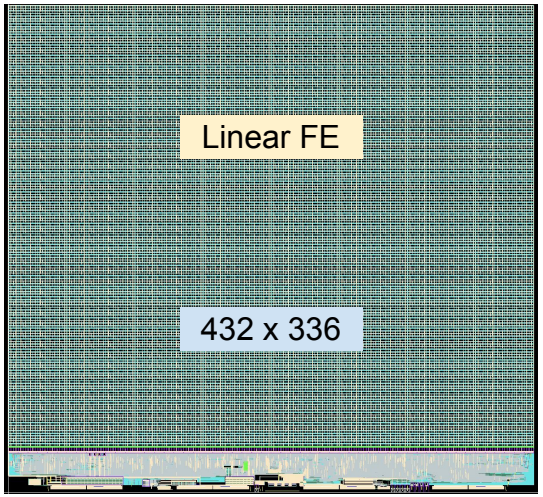
24 Institutes (Europe - USA)
Started in 2013



RD53A
Submitted: Aug 2017
Size: 20 x 11.5 mm^2



RD53B-ATLAS (**ItkPix** V1)
Submitted: Mar 2020
Size: 20 x 21 mm^2



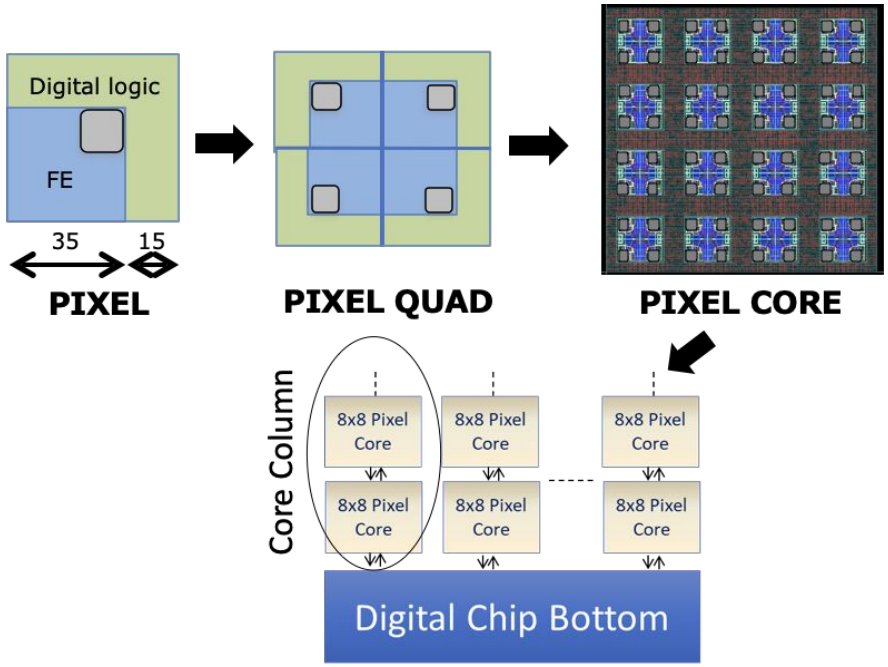
RD53B-CMS (**CROC** V1)
Submitted: Mar 2021
Size: 21.6 x 18.6 mm^2

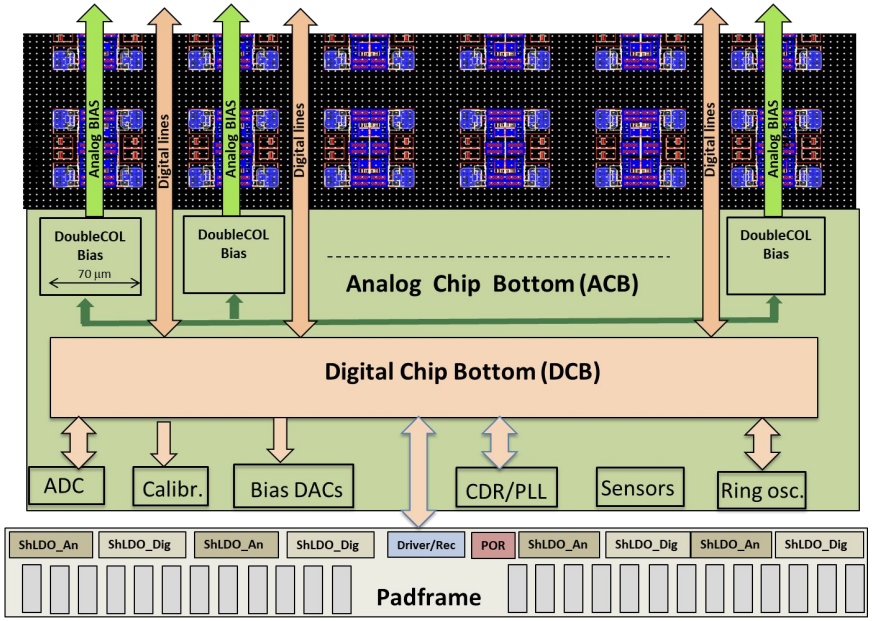
Final submissions:

RD53C - ATLAS
(**ItkPix** V2)
Q4 2022

RD53C - CMS
(**CROC** V2)
Q1 2023

Floorplan





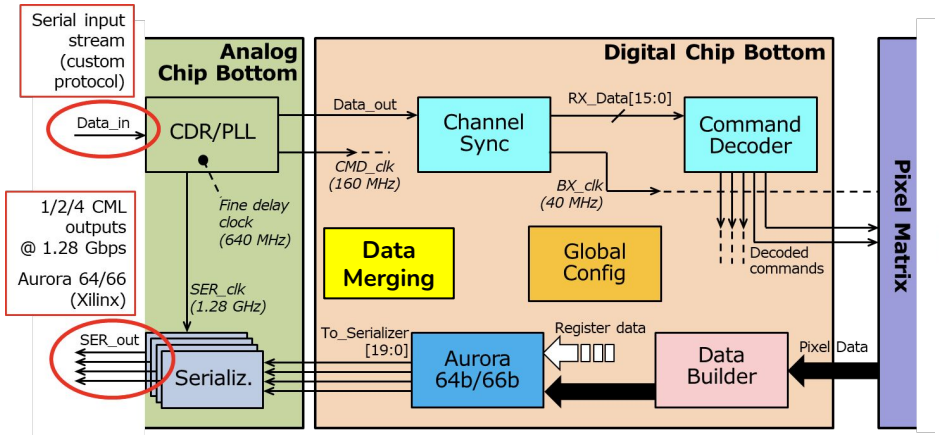
Pixel array:

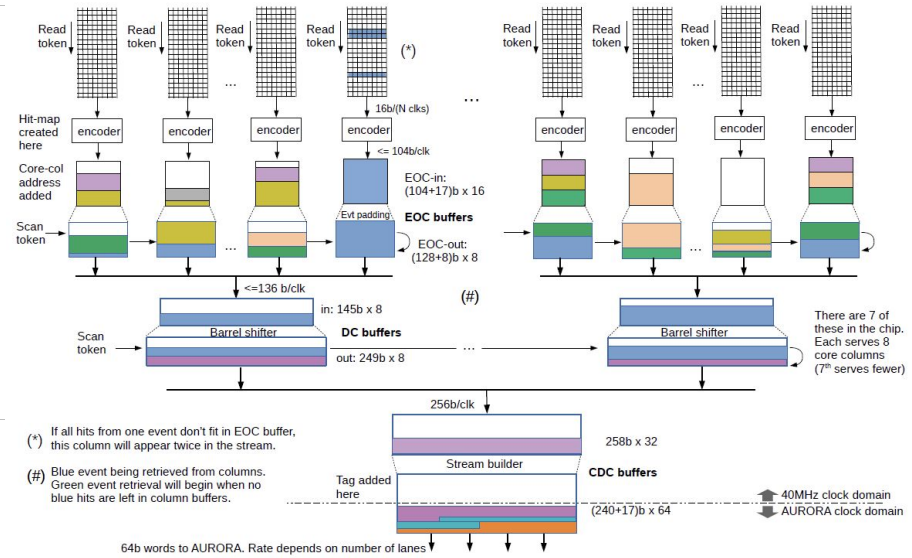
- Starting from a single FE we build a Pixel Core (8 x 8 FE's).
- All digital logic needed to operate the FE is shared in a single Core.
- Digital circuitry allows for local memory sharing.
- All Cores are identical: Simplifies verification and Physical implementation.
- Cores are placed by abutment (no extra routing) and global signals are eventually regenerated inside each Core.
- One digital readout block for each CoreColumn.

Chip periphery: Analog IP's and Digital Chip Bottom

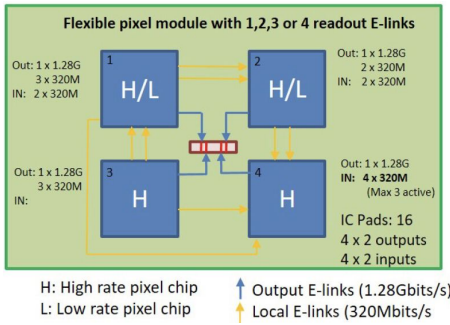
- Analog Chip Bottom: Contains all analog and mixed signal building blocks, like Calibration, Bias, Monitoring, Clock/Data recovery, …
- Digital Chip Bottom: All synthesized digital logic containing communication to/from the chip and the readout of the Pixel matrix and global configuration.
- Padframe: All I/O blocks with ESD protections, ShuntLDO for serial powering.

Data flow architecture





- **Command Control and Timing**:
  One single 160 Mbit/s differential control link with a custom developed protocol driving up to 15 chips.
- **CDR/PLL** recovers Data and Clock.
- **Readout**: 4 x 1.28 Gb/s using the Aurora 64/66 encoding.
- **Multi chip data merging** (for low rate outer layers):
  One chip can be configured as "primary" to aggregate serial data coming from one or more "secondary" chips and merge them with its own output.
  Allows to reduce the number of links to read out data.



- **Hits in the Pixel matrix are stored locally** upon Hit arrival and **associated to a timestamp**.
- 6-bit counter with only 4-bits sent out from the matrix.
- **Each pixel has 8x4 ToT bit memory**.
  - Option for 6 to 4 bit mapping (dual slope)
  - Counting with 40 MHz or 80 MHz clock
- Time stamp **memory shared between 4 pixels**.
- **Token based readout** of hits in parallel foreach CoreColumn as soon as a valid Trigger is received.
- **Multiple level of data processing**, event building, data buffering and formatting.
- **Service data** collection and formatting
- **Building an Aurora frame** based data stream to be send to the high speed serializers

Analog building blocks

In the chip there are many different Analog IPs. There is one slide for each of them in the backup section.

Analog FE:

- There are two FE's, one for ATLAS (differential Front-End) and one for CMS (linear Front-End).

CDR/PLL:

- Recovers the 160 MHz clock from the input line and generates internal clocks (160 MHz, 64 MHz, 640 MHz, 1.28 GHz).

Shunt LDO for serial powering :

- ShuntLDO regulators (1 for Analog and 1 for Digital), constant input current shared between chips, modules on serial chains.

Bias circuit:

- Based on Bandgap reference circuits, to provide a reference voltage/current with low sensitivity to temperature variations.

Calibration circuit:

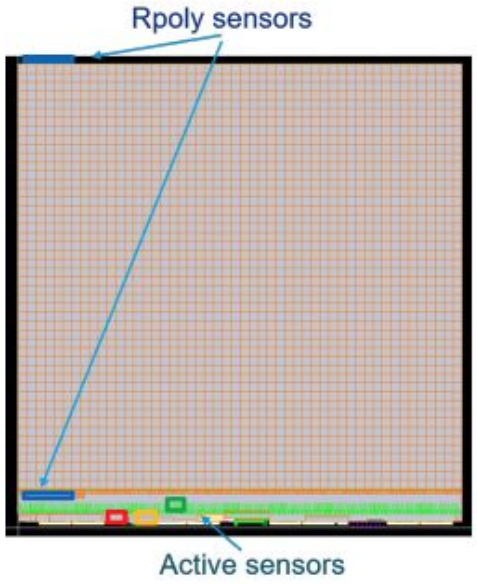
- Each pixel is equipped with a calibration injection circuit for test and calibration purpose.

Monitoring block:

- Current mux, a voltage mux and a 12-bit ADC to digitize and read out internal parameters.

Temperature and Radiation sensors:

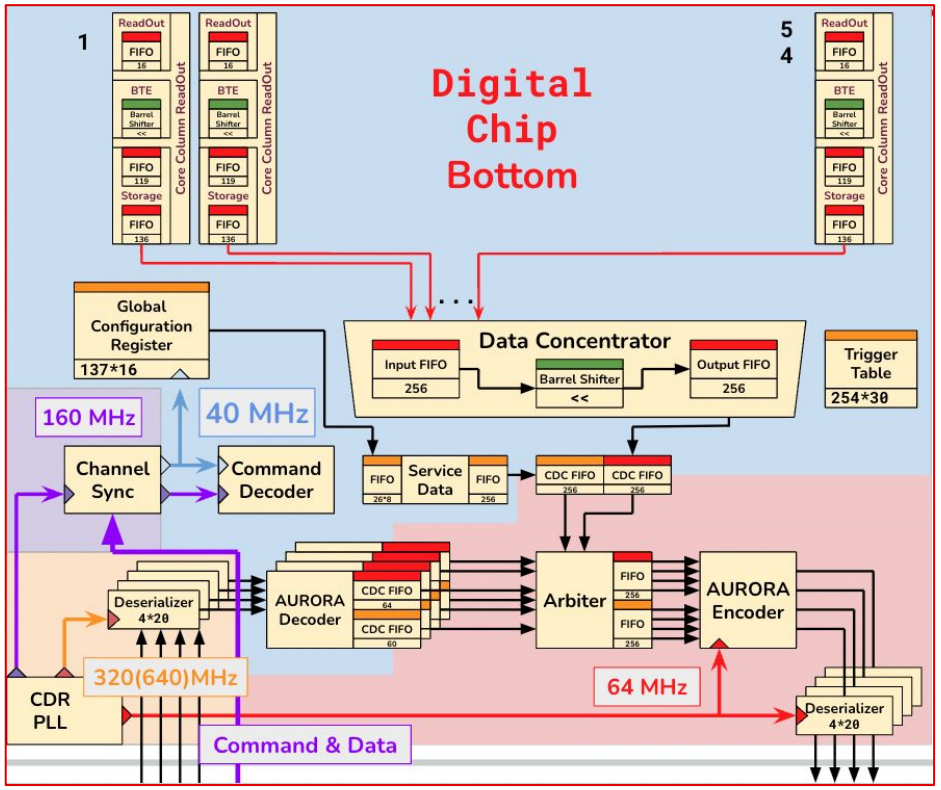
- They are implemented as diode-connected BJTs and NMOS transistors.
- There are three Radiation sensors distributed on chip.
  Two are implemented as diode-connected BJTs sensors
  One is made with a parallel structure of several minimum size PMOS devices.
- In addition there are two polysilicon resistor temperature sensors.
- All sensors are readout via the monitoring block.



Clock and Reset

Clock generation and distribution:

- **CDR/PLL** recovers the **160 MHz clock** from the input line.
- It generates internal clocks (**160**, **64**, **640**, **1280** MHz).
- **Channel Synchronizer** generates **40 MHz master clock**.
  This clock is sent to Digital Chip Bottom and Pixel Matrix.
- 4 different clock domains are dealt with using standard CDC design techniques and checked using formal tools.
- Not used blocks, and Pixel Matrix, use clock gating techniques to reduce power consumption.



Reset:

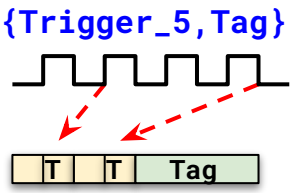
- The chip has no reset pin and the **reset is fully synchronous**.
  No reset line triplication is therefore foreseen.
- There is no **PoR** built in the Chip and **no Reset pin**.
- Global and Pixel configuration have hardcoded default values. At power up they are selected by a multiplexer.
  Register values can be overwritten by **WrRegister** commands and then one has to switch to non default configuration.
- All digital blocks have individual reset signals that can be sent via a dedicated **GlobalPulse** command.
- There is a Fast command, called **Clear**, used to fully and only resets the datapath and control state machines inside the chip.
- Due to serial powering of modules in the detector, in order to avoid power cycling, the system is built in such a way that a **Clear** command should always bring the chip back to its operational state.
- Nevertheless there is a safety mechanism that allows to fully reset the CDR/PLL block that acts as a PoR.
  In absence of commands to the chip for a certain amount of time (~1 μs) a reset signal is sent to initialize the CDR/PLL block.
  Only once the clock is generated the reset signal is deasserted.

Input command protocol

Protocol definition: [Downlink is running **@160 Mbit/s**, **clock** and **data** encoded on the same lane]

- It is a custom serial protocol with a maximum run length of 4, a built in command priority and a 4+1 bit command addressing.
- It defines 54 different 8-bit **Symbols** and one unique synchronization **Frame**. [32 symbols are used to encode 5-bit of data].
- All **Symbols** have a hamming distance of 2. This allows for Error detection but not correction.
- Two consecutive **Symbols** form a 16-bit **Frame** which is the minimum data unit the user can send to the chip.
- Commands are defined using a certain number of 16-bit **Frames**.
- One **Frame** is sent every four 40MHz clock cycles or every sixteen 160MHz clock cycles (**100 ns**).
- There are two classes of commands:
  - **Fast commands**: Single frame commands with **high priority**. They last 4 clock cycles, 100 ns.
  - **Slow commands**: Multiple frame commands with variable length that have **low priority**. They can last many ms!
- Commands sent to the chip have a built in priority that allows the user to always interleave a fast command while executing a slow command, without interrupting it.
  Due to this feature **continuous reconfiguration of the chip** in ~100 ms is possible writing a routine that reconfigures the chip with its current values and as soon as a **Trigger** or a **Sync** has to be sent one simply interrupts current command sequence, inserts the desired fast command, and then resumes the slow command that was executing, without the need of restarting it.

Available commands (for a detailed list of implementation refer to backup slides)**:**



Sync, **PllLock**: Used for channel synchronization and to ease the lock of the PLL at startup.

Cal : Calibration command, used to perform FE calibration.

Clear : Clears the whole data path leaving configuration (Global and Pixel) untouched.

Trigger_NN : 15 trigger commands covering 4 clock cycles (up to 216 consecutive triggers). Each event has a Tag attached to it.

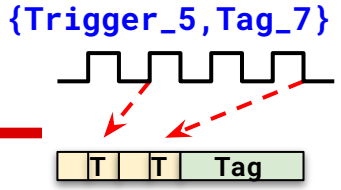
ReadTrigger: Used when 2 lever triggering is enabled. Allows to select which of the Triggered data we want to read out.

WrReg : Allows to write a register specifying address and contents. (Special version for speeding up FE register writing.)

RdReg: Allows to read back the contents of a configuration register (Global of Pixel) specifying its address.

GlobalPulse: A general purpose command that can be configured by the user to perform dedicated actions.
It is used for example to reset certain digital blocks, or to initiate a Monitor block readout.

Trigger and Tags



The chip implements a **zero suppressed triggered system**.

Hits are stored in the Pixel matrix with a 4-bit ToT and a timestamp associated to them.

Hits that are Triggered will be read out, cleared otherwise. A Tag will be associated to all read out data.

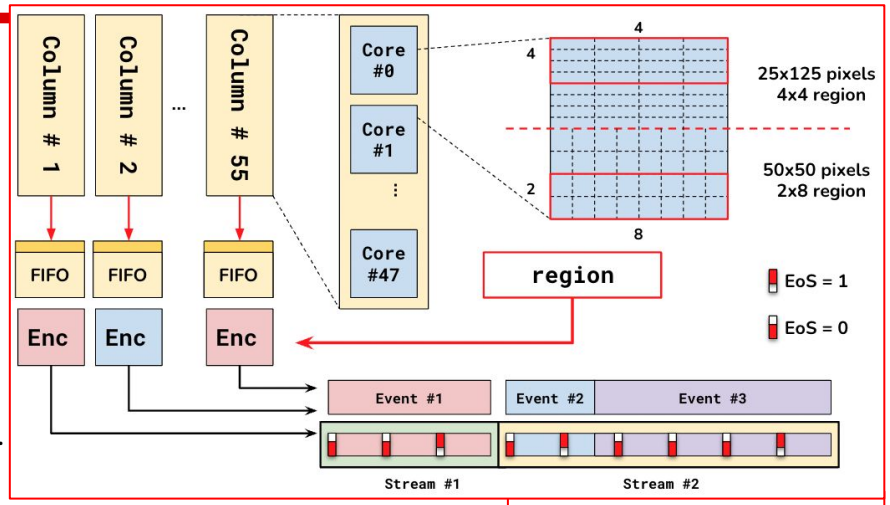
Trigger command: it is defined as {Trigger_NN,Tag_N}

- A trigger is a fast command built using one of the 15 Trigger Symbols followed by one of the 54 symbols (Tag).
- As any command spawns 4 clock cycles we have to be able to tell the chip to which of the 4 clock the trigger belongs.
- In addition each Trigger has a 6-bit Tag (i.e. a label) associated to it. Up to 54 Tags are possible.
- In the depicted example a single Trigger_5 command corresponds to two different Events to be read out.
- The Command Decoder will automatically generate two trigger entries to be stored in the TriggerTable and extend the Tag associated to them in order to uniquely label each event. (The extended Tags will be **`000111.00`** and **`000111.10`**)
- As soon as a Trigger is detected it will be stored in a global TriggerTable and a data readout sequence will be initiated.
- The TriggerTable will handle the data requests to the PixelMatrix based on the incoming Tag order. (Order is preserved)
- As soon as a readout request is issued, all data associated to that Tag will be requested from the Pixels sending a token with the corresponding TimeStamp (current TimeStamp - fixed Latency).
- When all data for a specific Tag is collected Event Building will start and that Tag will be made available again to the daq.
- The event, associated to a particular Tag, will be compressed and formatted and the Tag itself will be part of it.
- All Hit data, stored in the Pixels, for timestamps not matching the requested ones will be cleared in the pixels.
- If a Trigger with a Tag is sent to the chip before that event is readout the Trigger will be rejected.
- This mechanism allows up to 216 (54*4) consecutive Triggers to be sent to the chip.
- In addition there's an user selectable two level trigger mechanism built in that allows to only request a subset of triggered events via a **ReadTrigger** command. If an Event is not requested within ~35 μs all stored data is automatically cleared.

Event building and formatting

Hit data readout and compression:

- Token based readout of Hits is performed in parallel foreach Corecolumn as soon as a Trigger is received.
- Data is stored in a Fifo at the end of each column.
- Data is compressed via a Binary Tree Encoder (up to a factor 2)

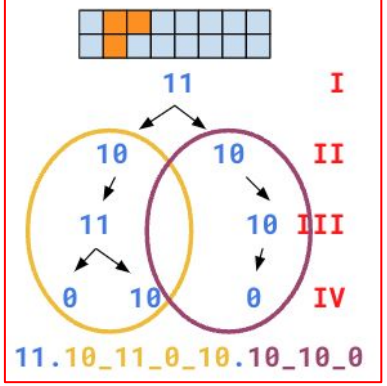


Event building:

- Data is collected from all CoreColumns and formatted in order to form a full event. This is done in a multi level event building stage.
- Each event is full contained and referenced by the associated Tag.
- All events are built preserving trigger order (there will be NO event mixing).
- Events are of variable length due to Binary Tree encoding, and zero suppression.

Stream encapsulation:

- All built events are encapsulated into a variable length Stream, a container used to efficiently split events over 64-bit wide Aurora frames. This is very effective in case of many small events.
- A Stream might contain up to 32 different events. This is a user configurable option.



Format of events inside a stream:

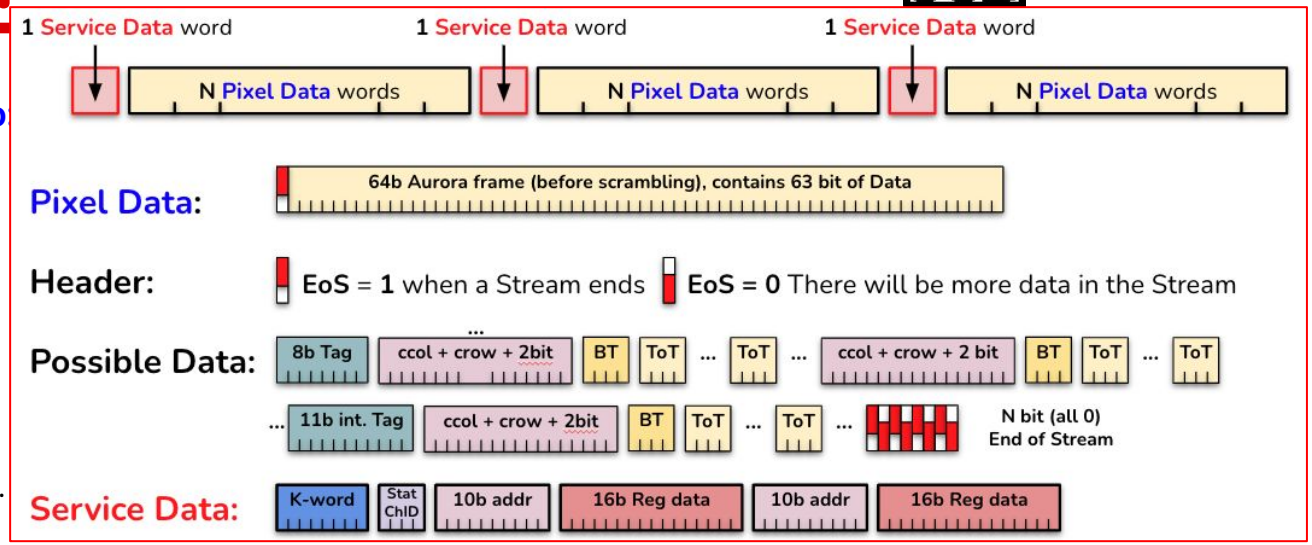
- Each Stream is splitted into 64-bit data words. The first bit will tell the daq if the stream ends with this Frame or not (EoS).
- The first 8-bit of data are the Tag that identifies the Event currently being processed.
- It is followed by the 6 bit address of the chip column (ccol) and the 6 bit of the 8x8 Core address (crow).
- Core data is then divided in 2x8 regions and encoded using the Binary Tree Encoder. [4 to 30 bits long].
- The encode address of all hit Pixels of that region is followed by all the 4-bit ToT fields of the hits. [4 to 64-bit long].
- Once the event is finished we close the Stream padding with zeros or insert a new event separating it from the previous one with a three bit separator (**111**) that never can occur in event data.

Tag+Nc*(ccol+Nr*(crow+enc+Nh*ToT))+Ne*(Sep+Tag+Nc*(ccol+Nr*(crow+enc+Nh*ToT)))+padding

Aurora output

We use a subset of Aurora64/66b

- Output data stream is sent over 1, 2, 3 or 4 simplex Aurora channel, depending on location.
- Each Aurora Channel might contain PixelData, ServiceData or NoData.
- **PixelData**: Contains the data stream.
- **ServiceData**: Contains monitoring, error/warning and register readbacks.
- **NoData**: Standard Aurora Idle Frame
- Aurora scrambles the frames and adds a 2 bit header (10, 01) to the 64-bit scrambled frame.



PixelData [01 header]:

- Data is generated inside the chip and put in variable length streams and is then split into 64 bit wide Aurora Frames.
- The first bit of each Aurora Data Frame contains one bit (EoS) that tells if this frame will terminate or not current data stream.
- When a Data Stream contains no more data it is padded with zeros. One Stream can contain multiple events (user selectable).

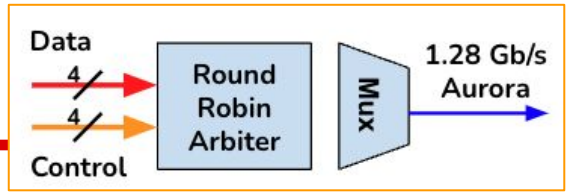
ServiceData [10 header]:

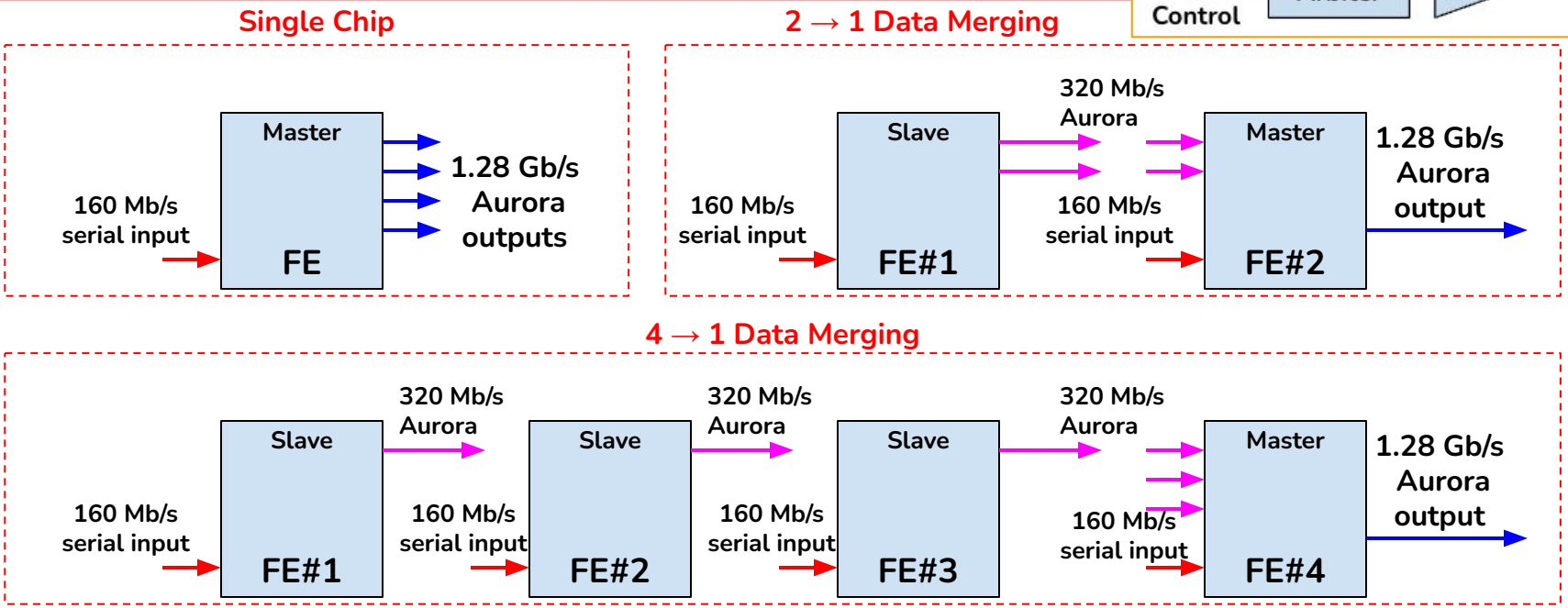
- A single Aurora Frames that contains the value of two RD53 registers (Global or Pixel) 16-bit fields with its address.
- ServiceData is generated once every N Aurora frames. N is a user configurable parameter (N = 51→ ~100Mb/s of ServiceData).
- RdRegister commands store the contents of the read register in a dedicated Fifo.
- When ServiceData is generated the contents of the Fifo are fetched until the Fifo becomes empty.
- If the Fifo is empty contents of up to 16 user defined registers will be fetched instead.

NoData [10 header]:

- Contains standard Aurora Idle, Clock compensation or Line alignment frames.

Data merging





Need for Data Merging:

- For outer layers a single 1.28 Gb/s output lane would be under utilized as there is not enough data to completely fill the available bandwidth.
- Therefore we introduced the ability to merge data coming from multiple chips locally at the module level.
- Data merging needs strongly depend on chip location in the detector where a 2→ 1 or a 4→ 1 merging capability is needed.
- All links between chips will be 320 Mb/s using an Aurora 64/666b single lane, or two lanes with strict alignment.
- Different operation modes are available: Single primary, one secondary and one primary, three secondary and one primary.
- Each incoming link needs a dedicated receiver instance in order to correctly align, unscramble and decode the incoming Aurora stream(s). For each link PixelData and ServiceData are stored separately.
- 1.28 GHz clock from Clock-Data-Recovery IP used to sample data from incoming link.
- Receivers are “dumb”, i.e. they do not have to know contents of decoded data. Data (with a 2-bit address) is stored in FIFO’s.
- A simple Round Robin Arbiter circuit will forward incoming data to the Primary Aurora encoder and sent it out @ 1.28 GHz/s

Special digital features

In the chip there are many different digital features that one can eventually enable.
There is one slide for each of them in the backup section.

Hit sampling mode:

- User can select to sample hit data with an edge or level sensitive input.

Precision ToT and ToA :

- PTOT module can be used for high resolution Time over Threshold and Time of Arrival (640 MHz clock, 1.5625 ns resolution).

SelfTrigger:

- Flexible auto trigger function, based on a Hit-OR network from the pixel array.

Ring oscillators:

- Multiple ring oscillators built with 8 different digital cells allow to precisely measure gate delay degradation due to radiation.

Data limiting options to save output bandwidth:

- Define a maximum number of allowed hits per CoreColumn: All hits above threshold will be discarded.
- Define a maximum time to read out an entire event: After timeout is reached remaining hits of that event are cleared.
- Isolated Hit Removal: One can decide to remove hits considered as isolated on a CoreColumn level.

SEU counting (both in Pixel matrix and in GlobalConfiguration registers):

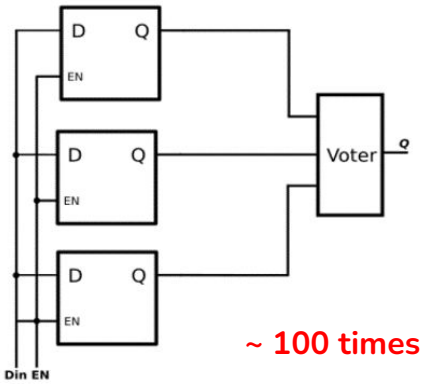
- There is an option for detecting and counting SEU events. Counter values are stored in a dedicated configuration register.

Different output formats (mostly for debugging and detector tuning):

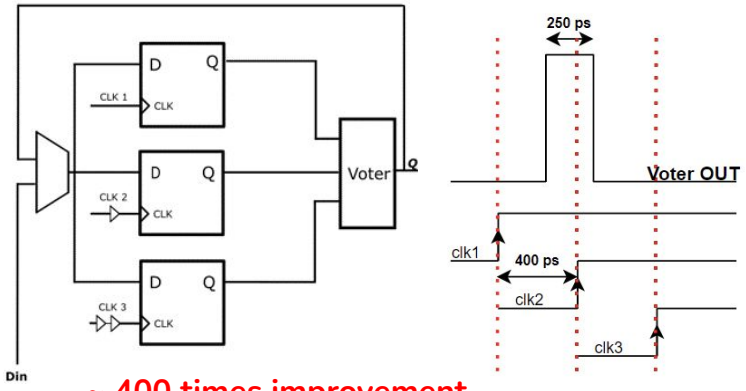
- User can select to disable Event compression, perform binary readout, add BCId, Lv1Id information and add full CRC info.

All these options, requested by experiments, have considerably complicated the verification of the chip.

Radiation hardness



~ 100 times improvement



~ 400 times improvement

Need for protection and possible solutions:

- The chip has **~150.000 pixels**. Each Pixel has 8 configuration latches that need to be protected.
- Digital Chip Bottom contains Globalconfiguration and all state machines and storage that deal with event building.
- For the innermost layer the foreseen **SEU rate is ~100 Hz per chip**, so SEU robustness is absolutely mandatory.
- Not the whole chip can be triplicated, especially in the Core region due to area and power constraints.
- Perfect protection will not be available so we have to cope with failures. Detected SEU's are counted and can be read out.
- The main requirement is that the chip should never go, due to an SEU, in a stuck state that requires power cycling.
- We need to ensure that it will always recover data taking by means of a single **Clear** command, even if in such cases some data and/or events will be lost.
- See Jelena Lalic talk tomorrow for more details on SEU/SET tests results and verifications. [Link to presentation]

Pixel array:

- Due to area constraints we managed to triplicate only 5 out of 8 configuration latches with a single majority voter circuit. There is no feedback circuitry.
- We have to rely on continuous configuration of the whole matrix to mitigate data corruption as much as possible.
- This capability has been built in our command protocol.

Digital Chip Bottom:

- Here we decided to fully triplicate all configuration and control logic.
- There is a feedback circuitry that rewrites corrected data.
- In addition we have clock skewing (400 ps) to be able to filter out short SET glitches that might occur.
- GlobalConfiguration will be continuously reconfigured.

Conclusions

RD53 collaboration:

Very big collaboration (24 institutions from Europe and USA) lasting for almost 10 years!

Very rare example of engineers and physicists of two experiments working together.

Allowed us to address many design challenges of such a big and complex chip.

Highlighted also some criticalities of such long term projects (manpower and complexity).

Overall chip architecture:

A brief overview of the chip architecture has been presented.

More detailed digital architecture:

A small insight of most important points of the chosen digital architecture:
- Input/Output protocol.
- Clock and reset design choices.
- Event building.
- Radiation hardening approach to the design.

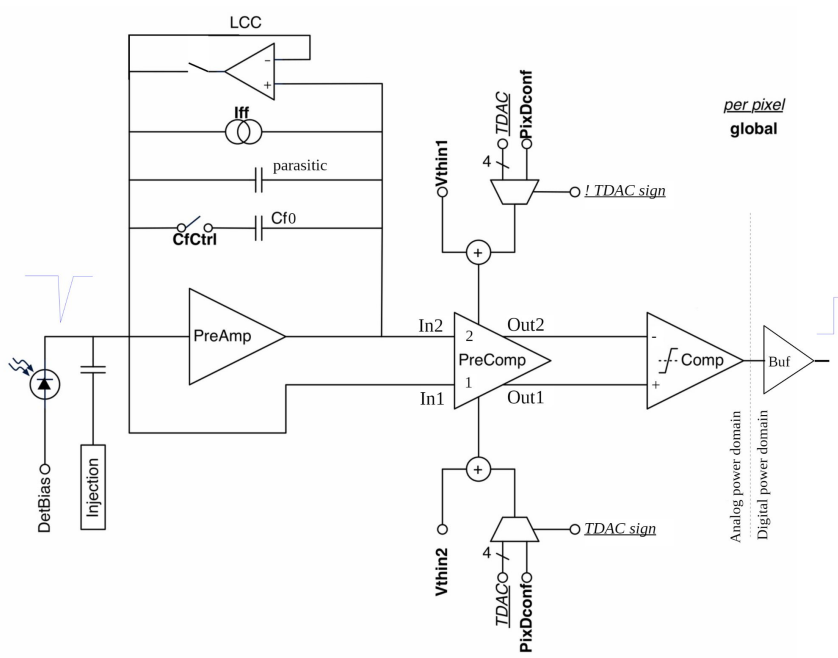
User configurable options and list of Analog IP's:

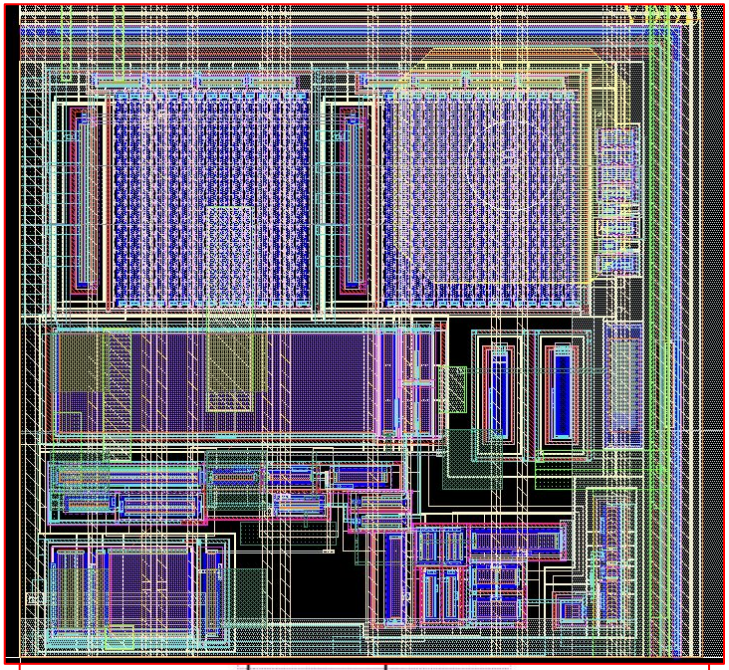
Such a big and complex design has a lot of user defined options to operate the chip and many different analog IP's not covered in the presentation but that are present in the backup slides.

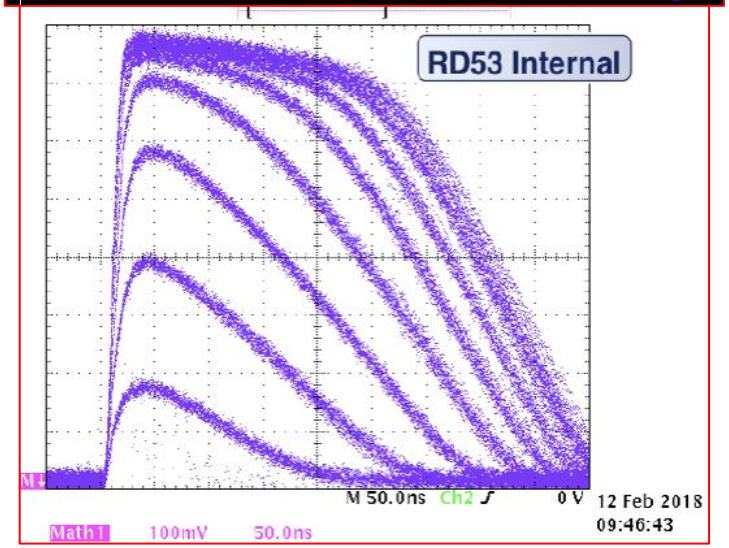
Many thx for your attention!

Backup Slides

Differential FE (RD53B-ATLAS)



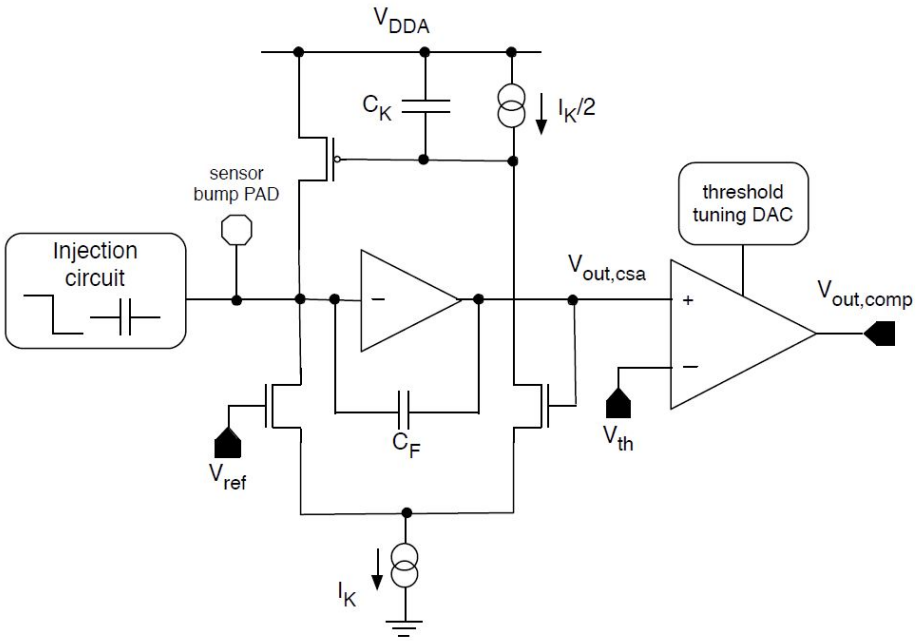


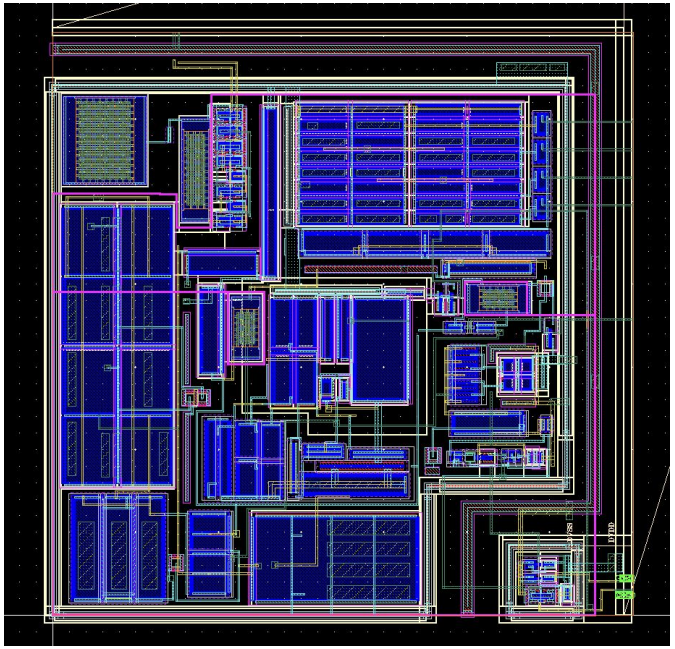


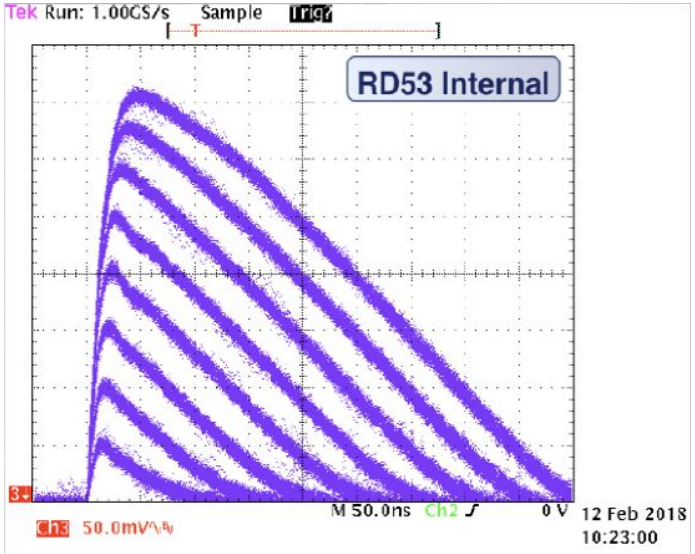
Differential Front-End (used by ATLAS):

- Charge sensitive amplifier.
- Leakage current compensation circuit.
- Continuous reset integrator, with tunable feedback current (global setting).
- DC-coupled pre-comparator stage:
  - 10-bit DAC for global threshold.
  - 4+1 bit local trimming DAC for threshold tuning.
- Fully differential input comparator.

Linear FE (RD53B-CMS)





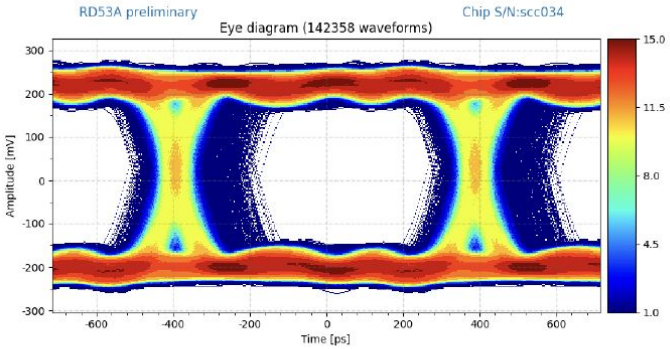


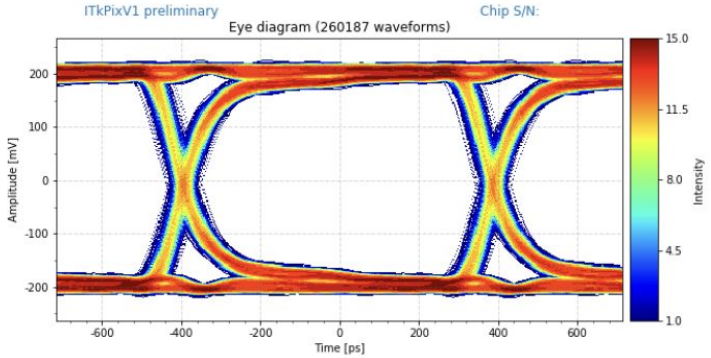
Linear Front-End (used by CMS):

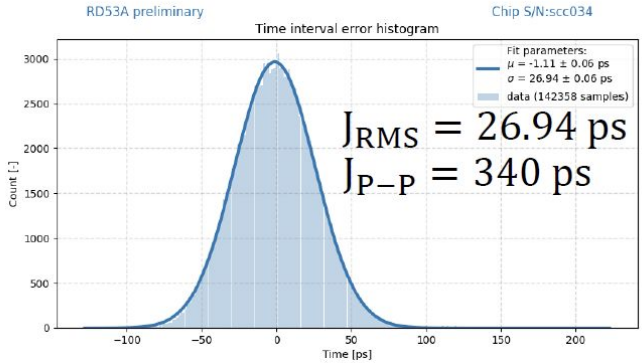
- Charge sensitive amplifier.
- Krummenacher feedback for return to baseline and leakage current compensation.
- Comparator:
  - 10-bit DAC for global threshold.
  - 5-bit local trimming DAC for threshold tuning.

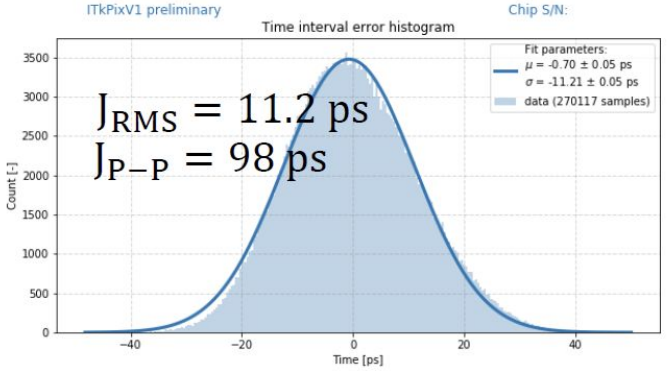
CDR/PLL

- CDR/PLL greatly improved compared to RD53A in terms of jitter and start-up reliability.
- Aurora output link stable with good quality.
- Input jitter = 5ps rms
  - **RD53A**
  - Input: Threshold scan
  - Output: Aurora (1.28 Gbps)
  - **ITkPixV1**
  - Input: PRBS5
  - Output: Aurora (1.28 Gbps)





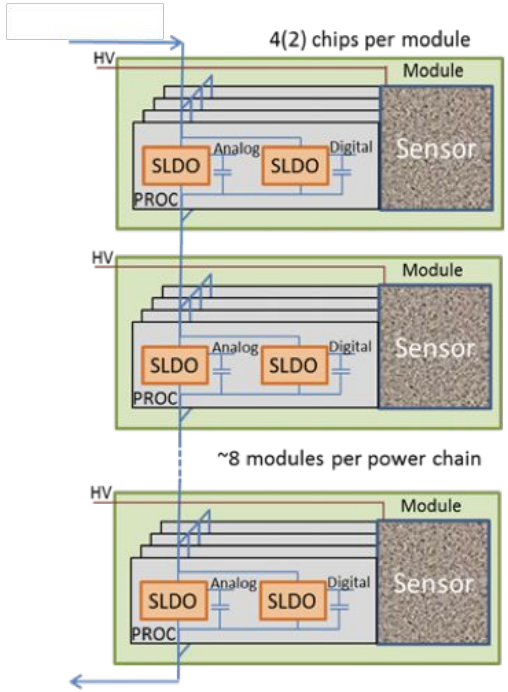


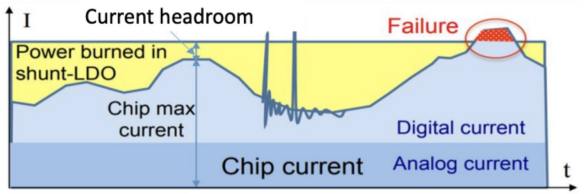


ShuntLDO for serial powering

ATLAS and CMS will adopt for the upgrade pixel detectors a **serial powering scheme**:

- **ShuntLDO** regulators in the readout chips (1 for Analog, 1 for Digital domain).
- **Constant input current Iin** is shared among chips (2÷4) on the same module (less cables)
- Modules are in serial chains: “recycle” current from one module to another.
- I_{in} dimensioned to satisfy the highest load, with ~20% headroom for stable operation, absorbed by the Shunt device
- In case of chip failure, its current can be absorbed by the other chips of the module
- Not sensitive to voltage drops (low mass cables)
- On-chip regulated supply voltages, low noise
- Radiation hardness (> 500 Mrad) silicon proven in RD53A and next prototypes

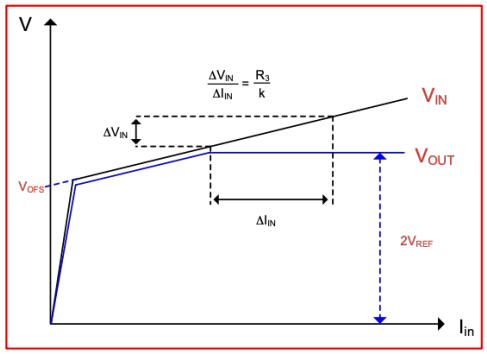




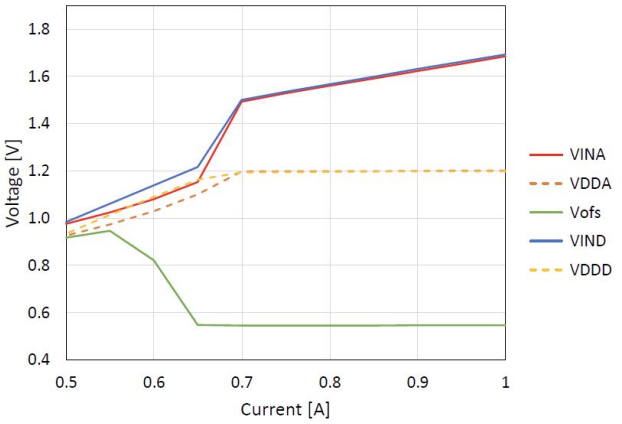
Example current consumption of one readout chip

Protections:

- Over-voltage protection: V_{IN} clamped to 2 V
- Under-shunt protection: V_{OUT} decreased in case shunt current goes below a certain threshold (due to excess load current)

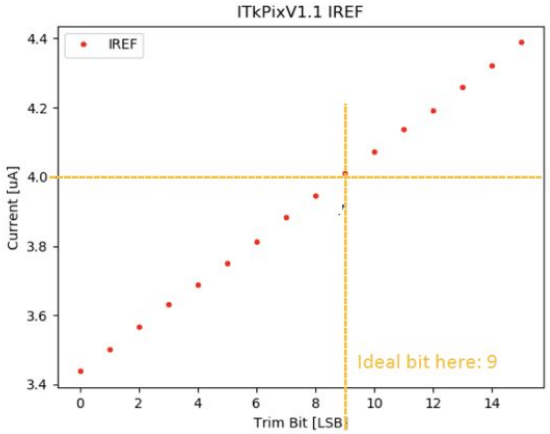


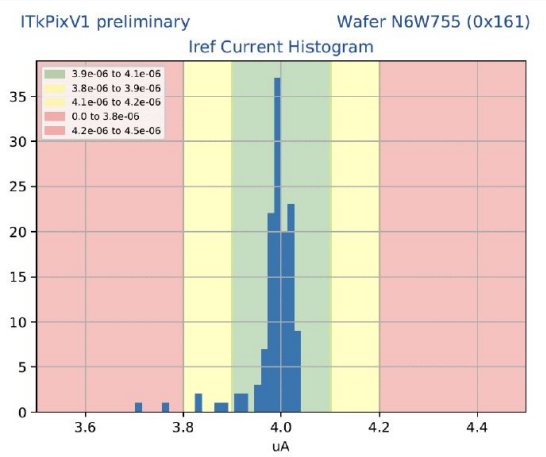
- **V/I** curve parameters (Voffset, slope) defined by external resistors
- V_{OUT} tunable by chip configuration

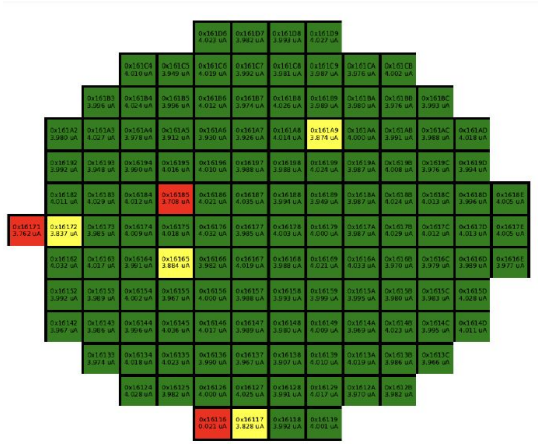


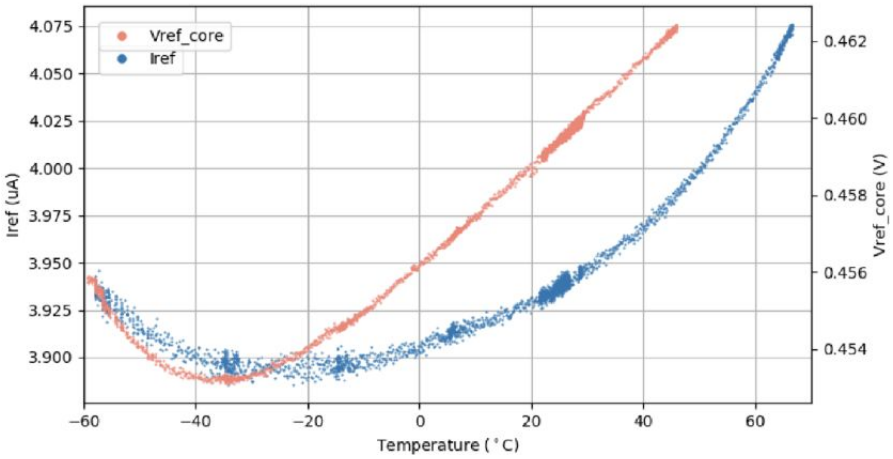
Bias circuit

- BIAS network is based on Bandgap reference circuits, to provide a reference voltage/current with low sensitivity to temperature variations.
- Tuning by means of 4 wire-bond trimming pads (no risk of SEU bit flips), whose optimal value is found during wafer probing.
- The tuned current I_{ref} is replicated and used as reference to 23 Digital-to-Analog converters to bias the analog Front-end, the CDR and other IPs.

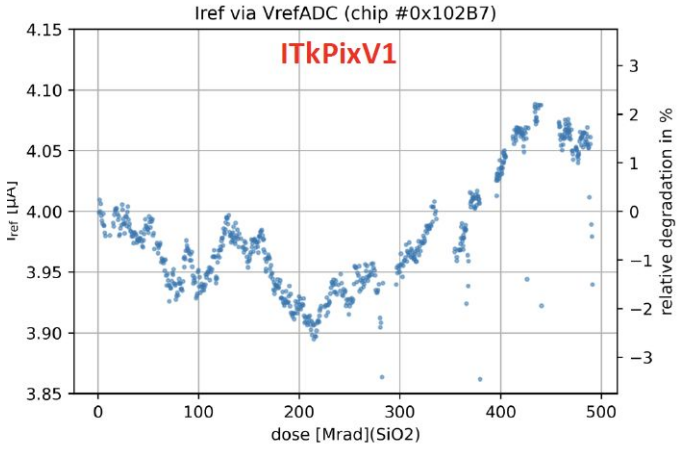






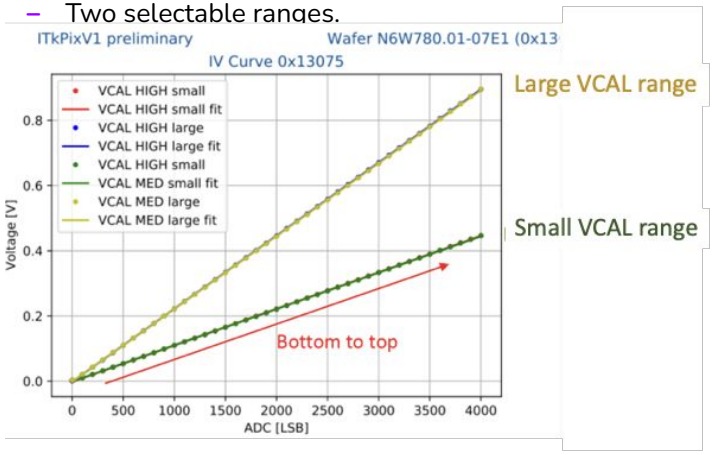


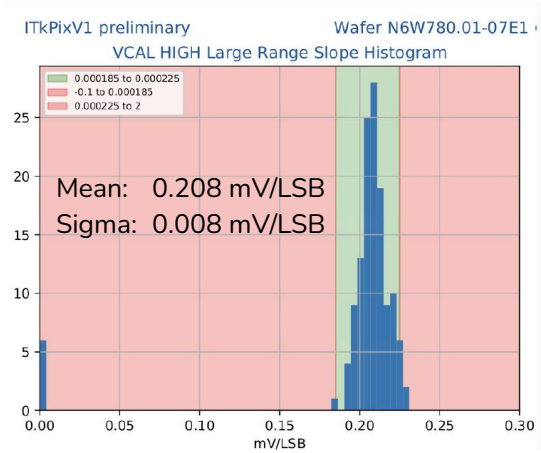
4% difference over 120^{0}C temp range

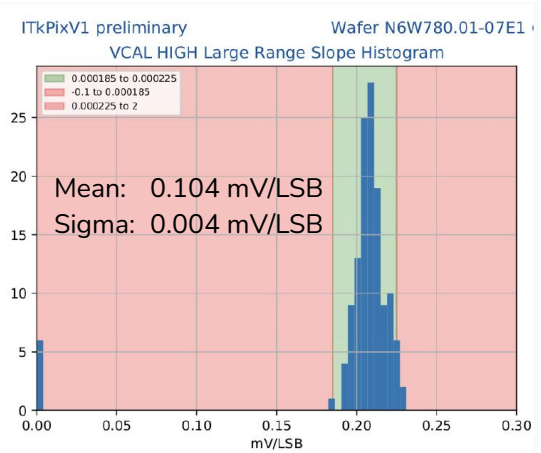


Calibration circuit

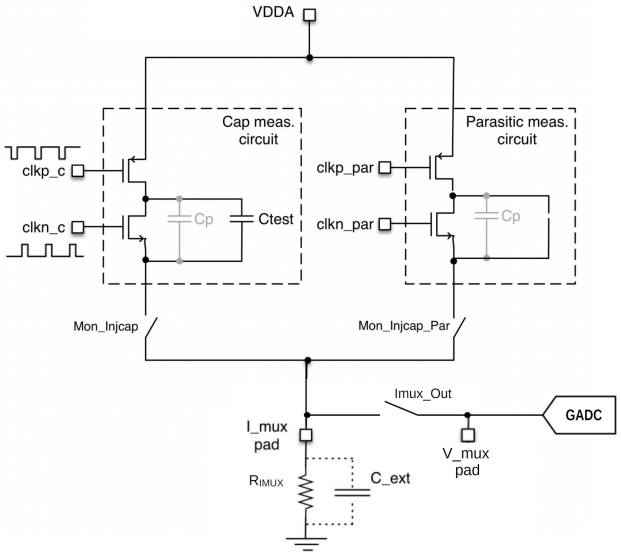
- Each pixel is equipped with a calibration injection circuit for test and calibration purpose.
- The analog injection uses two distributed voltages, provided by two 12-bit voltage DACs, to generate a precise voltage step fed to an injection capacitor.
- Two selectable ranges.

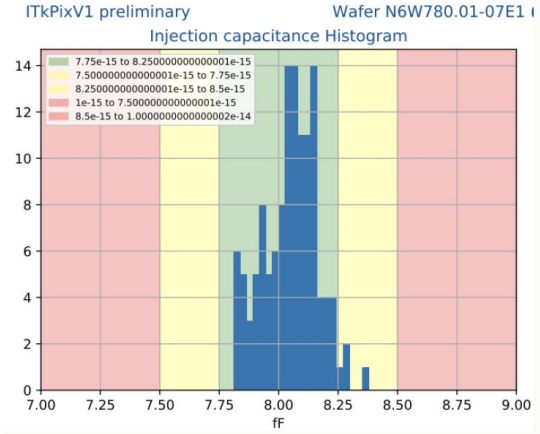






- Possibility to measure the value of injection capacitor using a dedicated circuit.

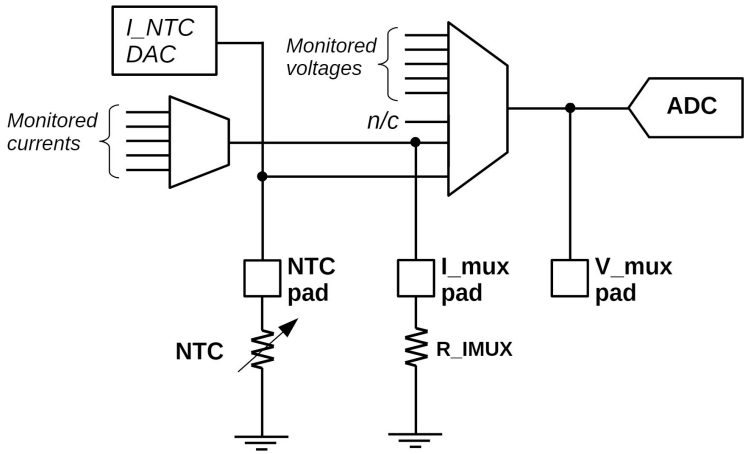


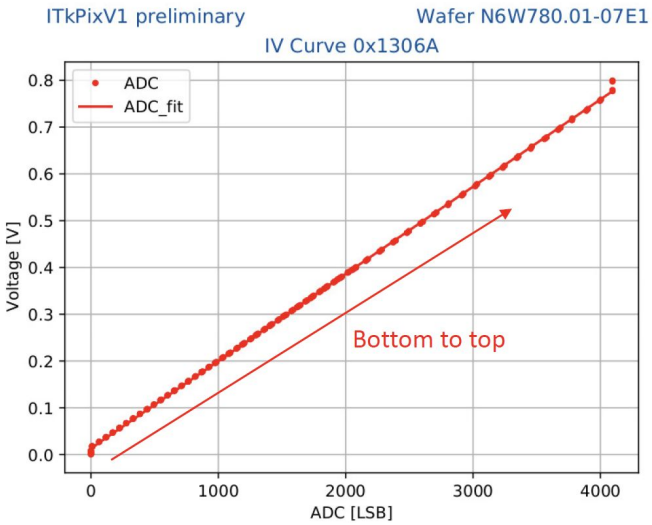


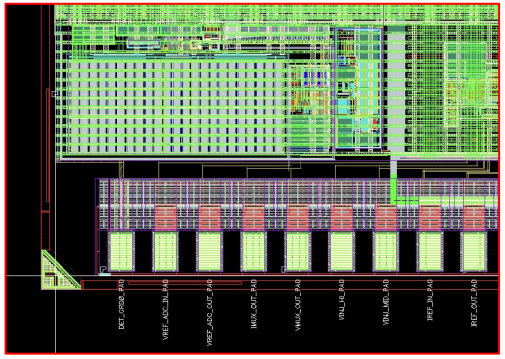
Mean: 8.05 fF (expected 8.02 fF)
Sigma: 0.11 fF

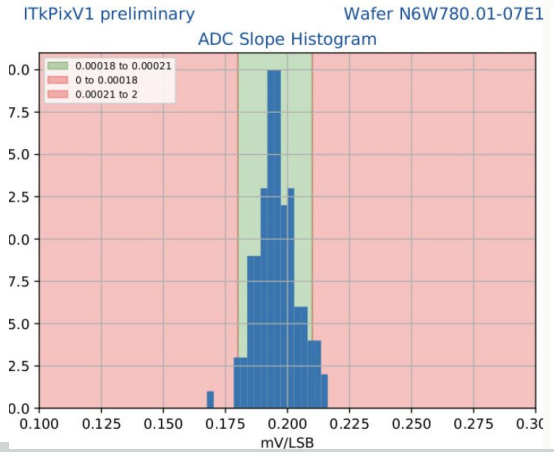
Monitoring block

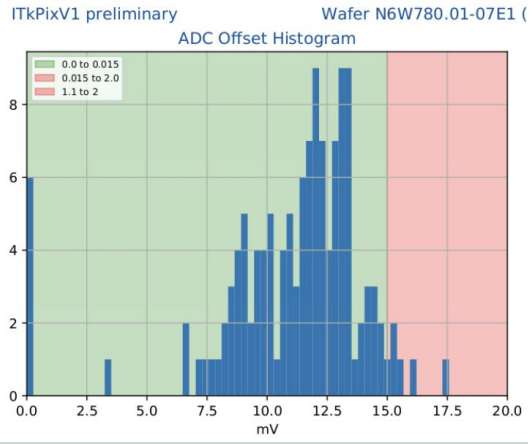
- The Monitoring block enables digitization and readout of internal parameters (T, voltages and currents from different parts of the chip).
- Consists of a current mux, a voltage mux and a 12-bit Analog to Digital Converter (ADC).
- Monitoring can be performed at any time, also during data-taking, via the normal data output links.
- There are 29 internal voltages and 29 internal currents that can be monitored.
- Ring Oscillators → measurements of digital cells speed degradation with TID.
- Monitoring sequence: Enable, select channel, send ADC start of conversion pulse, wait for Enf of Conversion pulse, readout data from dedicated Register









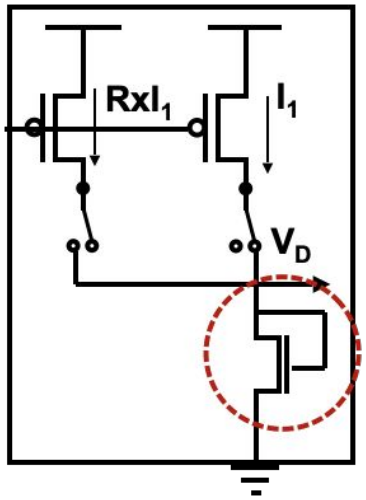


Temperature and Radiation sensors

Temperature sensor principle:

- The voltage is measured across sensor following 2 steps :
  - Sensor biased at 1 x I_1 (V_{D1})
  - Sensor biased at R x I_1 (V_{D2})
  - $\Delta V_D = V_{D2} - V_{D1}$ is determined offline
  - Default ideality factor value :
  - BJT: Nf = 1.008
  - NMOS : Nf = 1.225

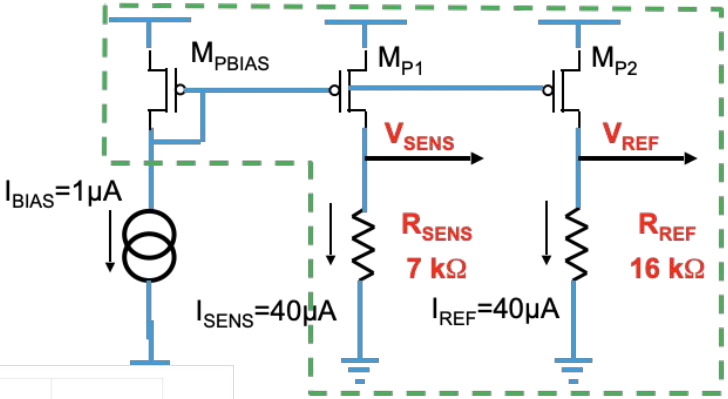
$$T_{ab} = \frac{q}{N_f \times k_B \times \ln R} \times \Delta V_D$$



R-Poly based temperature sensors

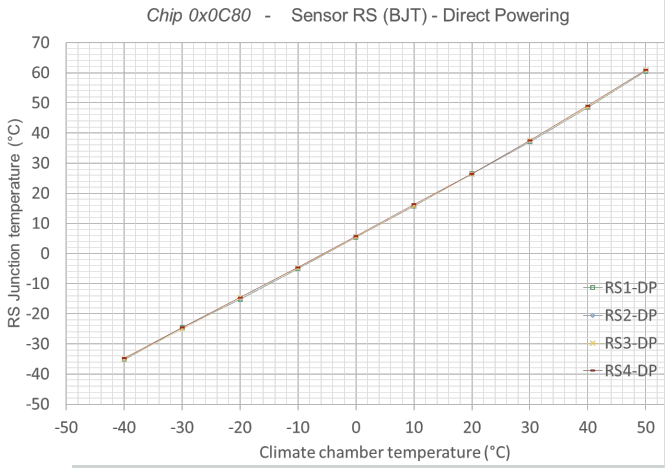
To reduce circuit size RD53B implements a temperature sensor based on R-Poly resistors.

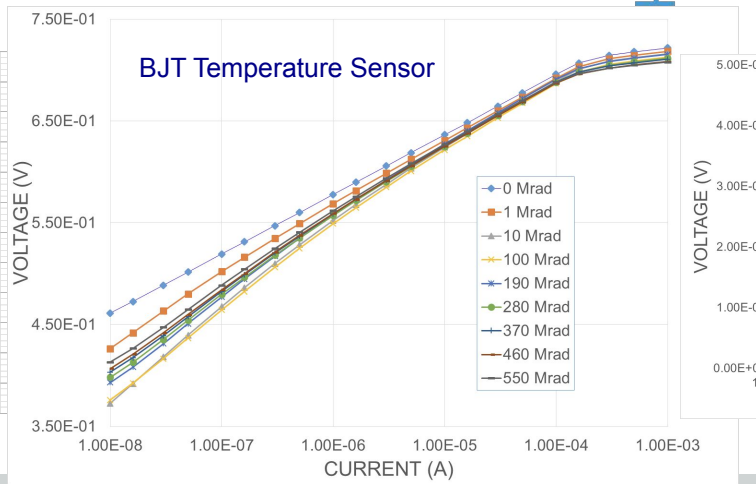
- RSENS = 7 kΩ → RPPOLY with Positive temperature coefficient: TC=+0.22% per °C
- RREF = 16 kΩ → RPPOLYWO with Negative temperature coefficient: TC =-0.03% per °C
- The global TC is +0.25% per °C

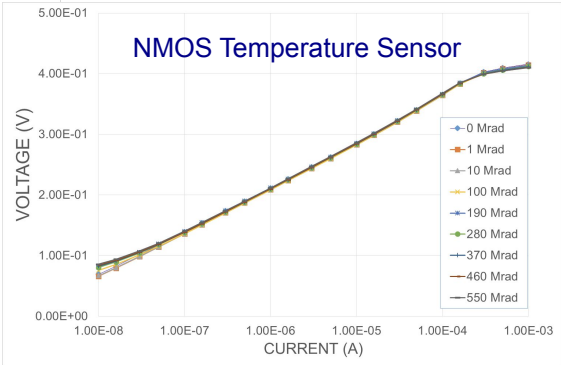


ADC:

- Radiation tolerant (architecture)
- The conversion gain is invariant versus the TID
- Offset : shift of 2 mV at 550 Mrad



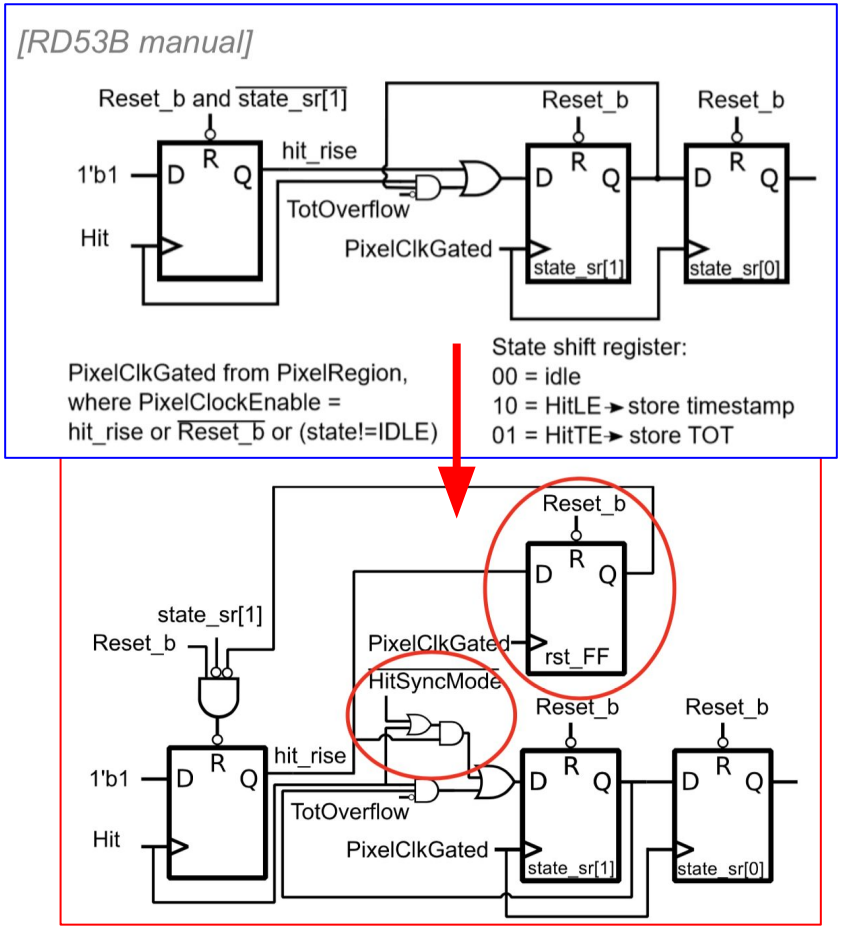




Hit sampling mode:

User selectable sampling mode:

- It allows to have both **edge** and **level** sensitive inputs for the Hits coming from the Front-End.
- A configuration bit allows to select the desired sampling mode [HitSampleMode].



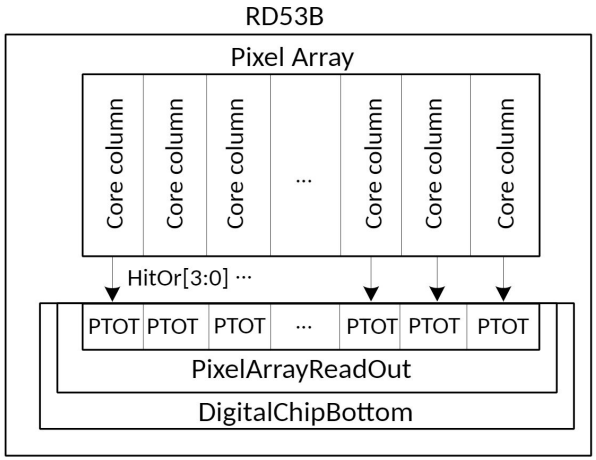
Changes to allow both edge and level inputs:

- Small additional combinatorial logic and one flip-flop per pixel to be able to reset `hit_rise` signal when in level-sensitive mode.
- No change to the clock tree power optimization.
- Same clock gating schema.
- Area impact after synthesis ~1%.
- Small power increase (<5%).

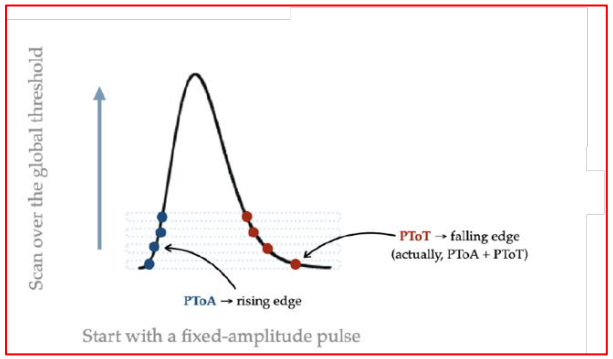
Implementation:

- It has been implemented and fully simulated with detailed mixed mode simulations.

Precision ToT and ToA

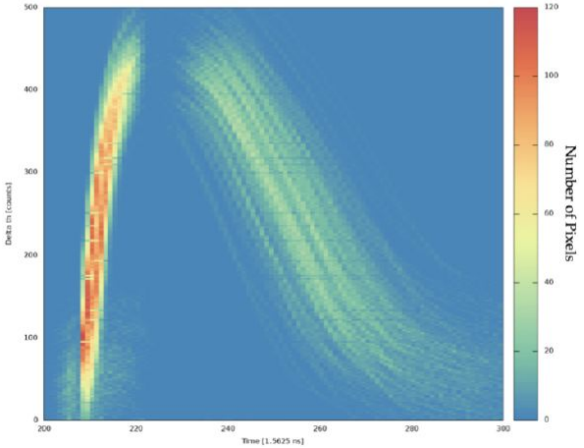


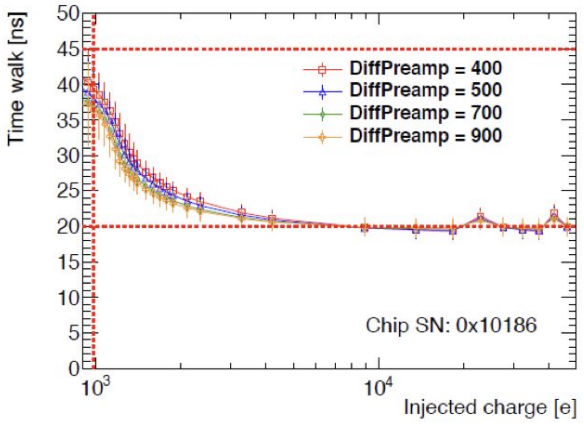
- PTOT module can be used for high resolution Time over Threshold and Time of Arrival measurement of the HitOR lines, using 640 MHz counting clock (1.5625 ns resolution)
  - 11-bit PToT counters
  - 5-bit PToA counters, measuring the phase difference from HitOr leading edges and next BX clock rising edge.
- Each Core Column is equipped with a PTOT module. Can be triggered for readout via the normal path, just like a Pixel Core
- Can be used to make precision measurements of analog front-end, like time walk, and also as a workaround for the bug in the pixel ToT memory
- Allows to reconstruct the amplifier output waveform (sort of oscilloscope)



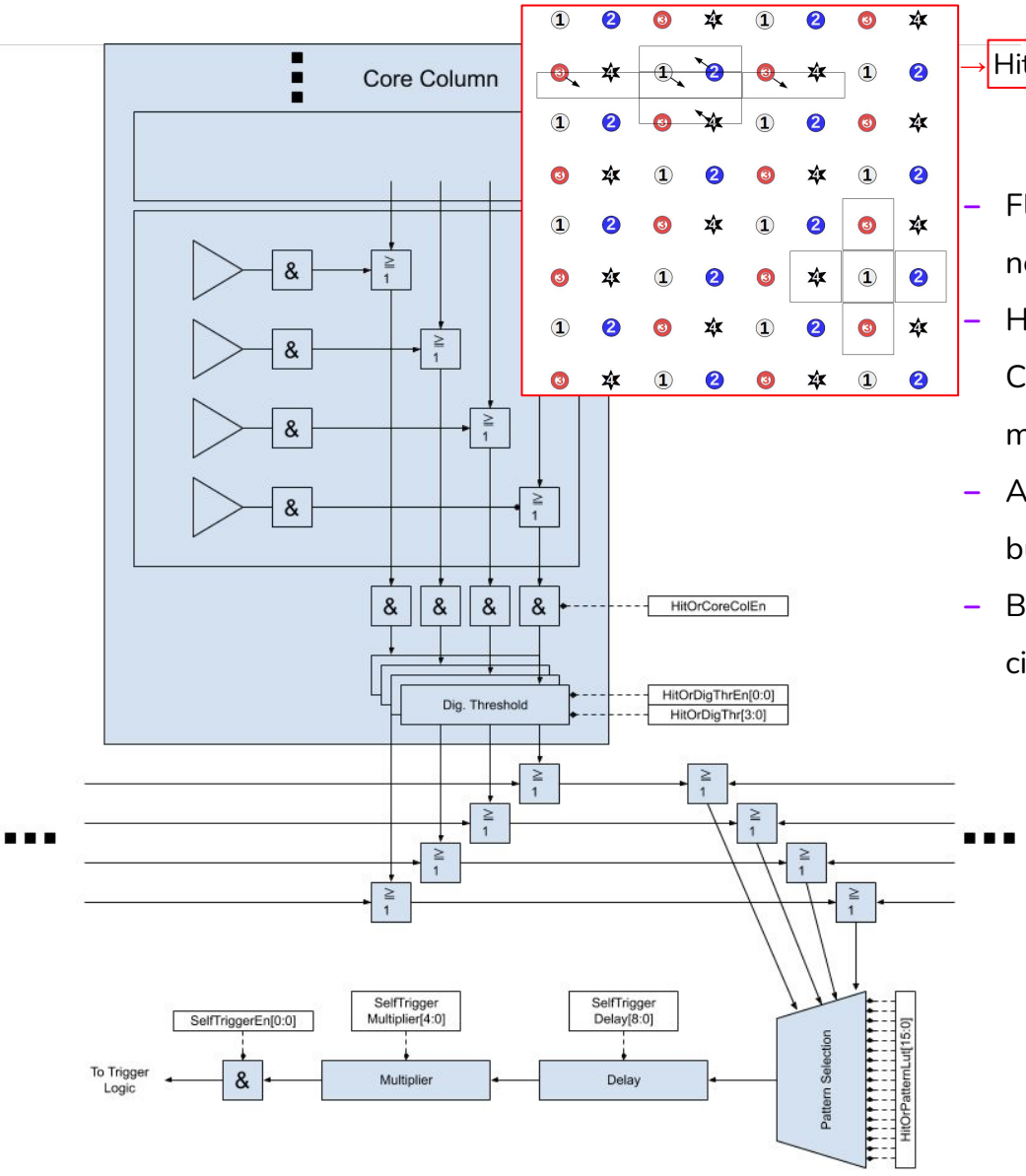
Procedure:

- Inject fixed calibration pulse.
- Scan the Threshold.
- Sample PToT and ToA at each step.



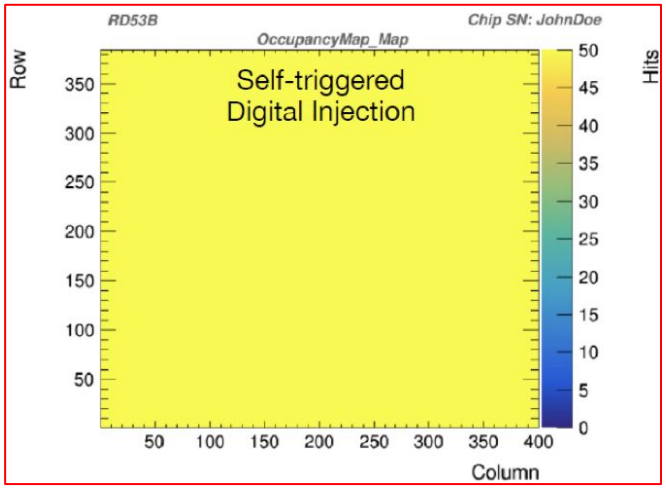


Self Trigger

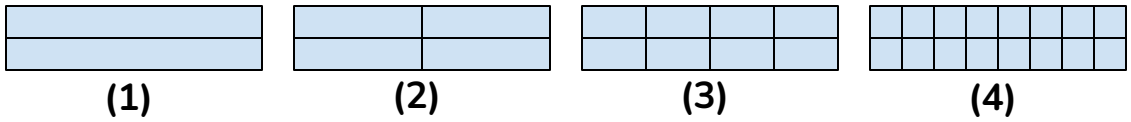


→ Hit-Or network definition

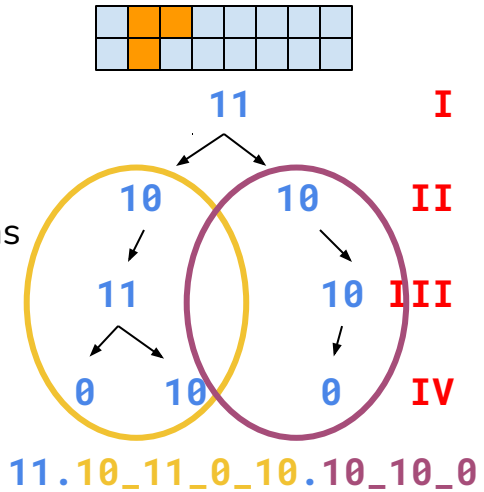
- Flexible auto trigger function, based on a Hit-OR network from the pixel array.
- Hit-OR network consists of 4 OR lanes per Core Column, with a mapping such that neighbor pixels are mapped on different lines (see figure).
- At the end of Core Column, the 4 lanes are combined to build the global Hit-OR with programmable patterns.
- Basic testing with digital injection show that this circuitry is **working as expected**.

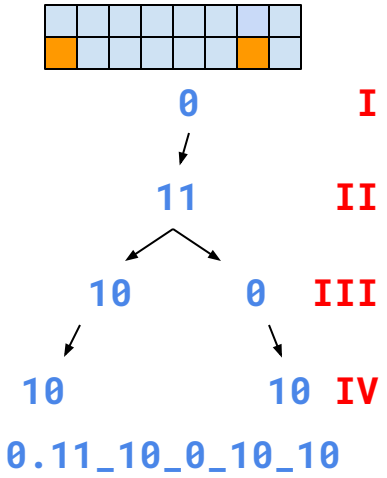


Binary Tree Encoding:



- Developed by Cornell group for CMS, it will be the data formatting used in RD53 for both ATLAS and CMS.
- **How it works**:
  1. TheCore Column (**ccol**) is a 6 bit wide mask that tells which column has data. No compression is done on this part of data.
  2. CoreRow address (**crow**) is encoded with some compression.
  3. Once a 2x8 or 4x4 pixel region has been addressed a binary search is performed.
  4. The encoding is performed in 4 subsequent steps [I, II, III, IV]
  5. ToT information for all the Hits of the region is added after encoded data.





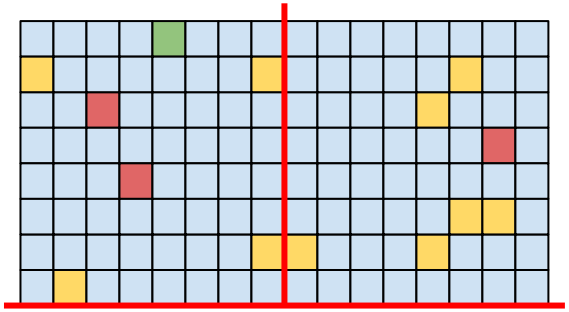
- The binary tree encoding has variable length, ranging from 4 to 30 bits.
  Cluster of hits (which are more probable) use less bits than isolated ones.
- A data reduction factor close to two can be obtained.
  This has been proven using MC data provided by the experiments.

Data reduction and Debug options

Data truncation: In case of very long events we have the ability to truncate events, before they get encoded in two ways.

- CoreColumn level truncation: The maximum number of events per CoreColumn can be programmed.
- Chip level truncation: After Trigger is received a programmable timeout will clear all Data still in the Matrix.

Isolated hit removal: Will be performed, if enabled, inside the EoC.



- Depending on the sensor (50 μm x 50 μm or 25 μm x 100 μm) the definition of neighbour pixel changes so two 16-bit masks are needed.
- There is a programmable ToT threshold.
- All hits at the L/R border are NOT isolated by construction.
- Therefore the removal efficiency of the algorithm is 75%.
- Hit removal is performed independently for each Trigger.
- This allows to monitor the background level and the effectiveness of the on-chip filtering.
- In the depicted example:
  - Red Hits are removed as considered isolated,
  - Orange Hits are kept because they are clusters or they lay on the border.
  - Green Hits can be isolated or not, depending on data coming next in the core column.

Binary readout:

- There is also an option to remove ToT values from Data Stream saving up to 30% of the bandwidth.

Data without Binary Tree Encoding:

- For debugging reasons one can choose to remove BTE and replace hit information with a 2x8 raw hitmap.

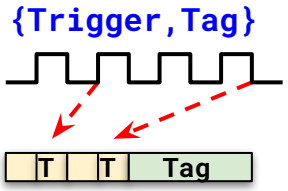
CRC of PixelData [Used polynomial is: $G(X)=X^{32}+X^{26}+X^{23}+X^{22}+X^{16}+X^{12}+X^{11}+X^{10}+X^{8}+X^{7}+X^{5}+X^{4}+X^{2}+X+1$]:

- An option allows to append to each Stream of PixelData a 32-bit wide CRC, calculated on all Aurora frames of the Stream.

BCId and LV1Id:

- Events are read out with Tags, but one can choose to add a 16-bit wide word containing BCId and/or LV1Id information.

Command definition



Sync command:

- {**1000_0001**,**0111_1110**} Special command, to be sent periodically, to lock the channel.

Protocol defines 54 different 8-bit symbols that are used to define all commands:

- **Clear, PllLock, Trig_00, ..., Trig_15, Cal, GlobalPulse, RdReg, ReadTrigger, WrReg, Data_00, ..., Data_31**

Frames:

- Two symbols are combined together, to form a single Frame that lasts 4x40MHz clock cycles (100 ns)

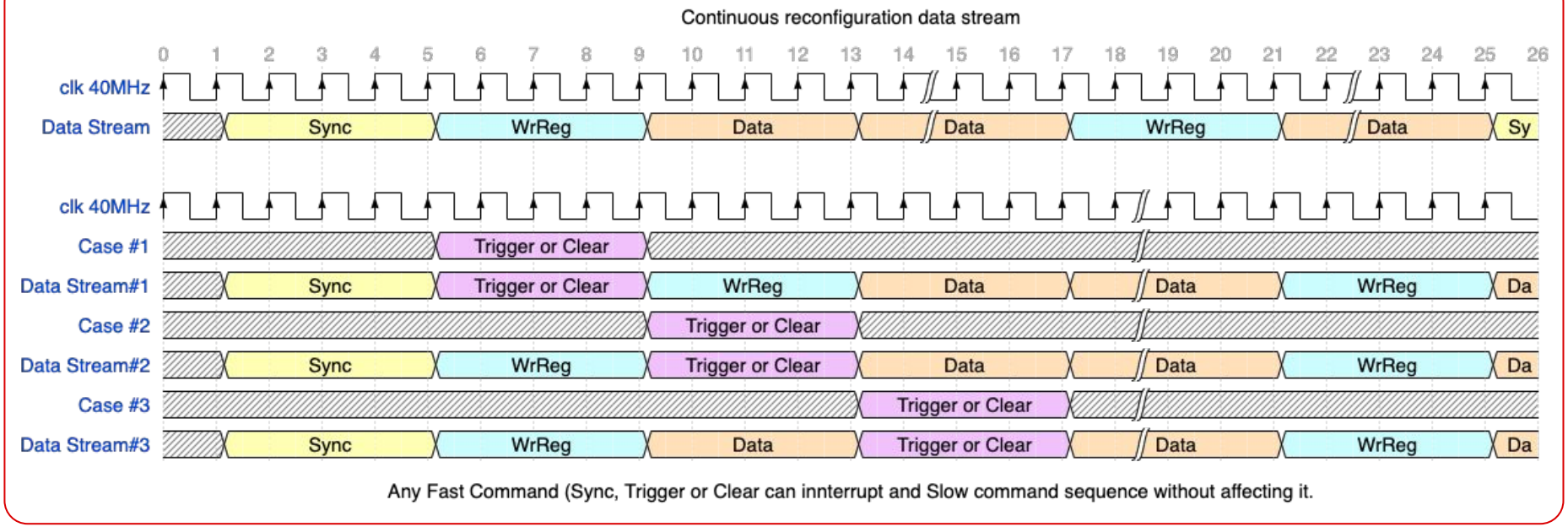
Commands:

- All commands are built using Frames. A single command lane can address multiple chips and most commands contain a **ChipId** data field that allows to select to which chip the command is meant to be sent.
- Single frame commands are **Fast Commands** and have high priority.
- Multiple frame commands are **Slow Commands** and have low priority.
- Slow commands can alway be interrupted by any number of Fast commands and then resumed from where the slow command was interrupted, allowing for an easy continuous reconfiguration routine.

In command definition each **{...}** pair is a Frame.

- **Clear**: {Clear,**ChipId**} A command to clear the data path and reinitialize state machines
- **PllLock**: {PllLock,PllLock} A command used during initialization to ease PLL lock.
- **Trig_NN**: {Trig_NN,Tag} A Tag is any of the 54 symbols that is used to “label” the trigger
- **GlobalPulse**: {GlobalPulse,**ChipId**} A command that can be programmed to perform different tasks
- **Cal**: {Cal, **ChipId**},{Calibration type},{Calibration Data} A command to calibrate single Pixels
- **ReadTrigger**: {ReadTrigger,ReadTrigger},{8-bit Tag} To be used for 2 level triggering
- **RdReg**: {RdReg,**ChipId**},{Addr},{Addr} Used to readout configuration registers
- **WrReg0**: {WrReg,**ChipId**},{Type=0,Addr},{Data},{Data} Used to write Global configuration
- **WrReg1**: {WrReg,**ChipId**},{Type=1,Addr=0},{Data0},...,{DataN} Used to write FE configuration

Continuous reconfiguration



The custom developed protocol allows to insert inside any Slow command sequence any number (and type) of Fast commands without affecting the original sequence.

This allows **DAQ** to preload in memory the whole configuration sequence of the desired Chip status, containing all needed Sync and Clear commands, and play it back to the chip.

As soon as a **Trigger** (or **Clear** or **Sync**) request arrives **DAQ** temporarily suspends the actual sequence, executes all the required commands, and resumes the sequence from where it was left.

A full Chip reconfiguration can be done in ~100 ms and therefore, even with a Trigger occupancy of the available bandwidth of 50%, we are able to reconfigure the whole chip at least 5 times per second, without any external intervention.

This mechanism should almost eliminate any **SEU** problem with respect to system configuration bits.

SEU protection: Pixels and GlobalConfiguration

Pixel Memory triplication:

- Each Pixel has 8 local configuration bits stored in 8 distinct latches.
- For radiation hardness we triplicated, with majority voting circuit **5 out of 8 latches**.
- Due to space constraints in the Core, the majority voted outputs are NOT fed back to the latches.
- Therefore if there are two upsets between two rewrites of Pixel configuration the majority voting circuit will fail. → We rely on continuous reconfiguration to solve this potential issue.

CONF. BIT		TMR	SEU-COUNT
0	HIT_EN	YES	YES
1	CAL_EN	NO	NO
2	HIT-OR_EN	NO	NO
3	TDAC_CORE[0]	~~YES~~ NO	NO
4	TDAC_CORE[1]	YES	NO
5	TDAC_CORE[2]	YES	YES
6	TDAC_CORE[3]	YES	YES
7	TDAC_CORE[4]	YES	YES

SEU detection:

- The CROC has an SEU detection circuitry only for **4 triplicated latches**.
- If the SEU Counting option is enabled in the chip, all detected SEU pulses are routed, via the HitOr bus (4 distinct buses per Column), to the chip periphery.
- At the end of the column these asynchronous signals are sampled and a single, one clock cycle lasting pulse is generated.
- All these pulses are counted and mapped to a 16-bit wide ReadOnly configuration register.

GlobalConfiguration Memory triplication:

- Global Configuration configuration bits are almost all triplicated. (All but 64 to measure SEU's)
- Each register has majority voting and SEU detection circuitry.
- Corrected values are always fed back to the register.
- Detected SEU's are counted and mapped to a 16-bit wide ReadOnly configuration register.