EICROC2 digital readout

2nd Apr 2026

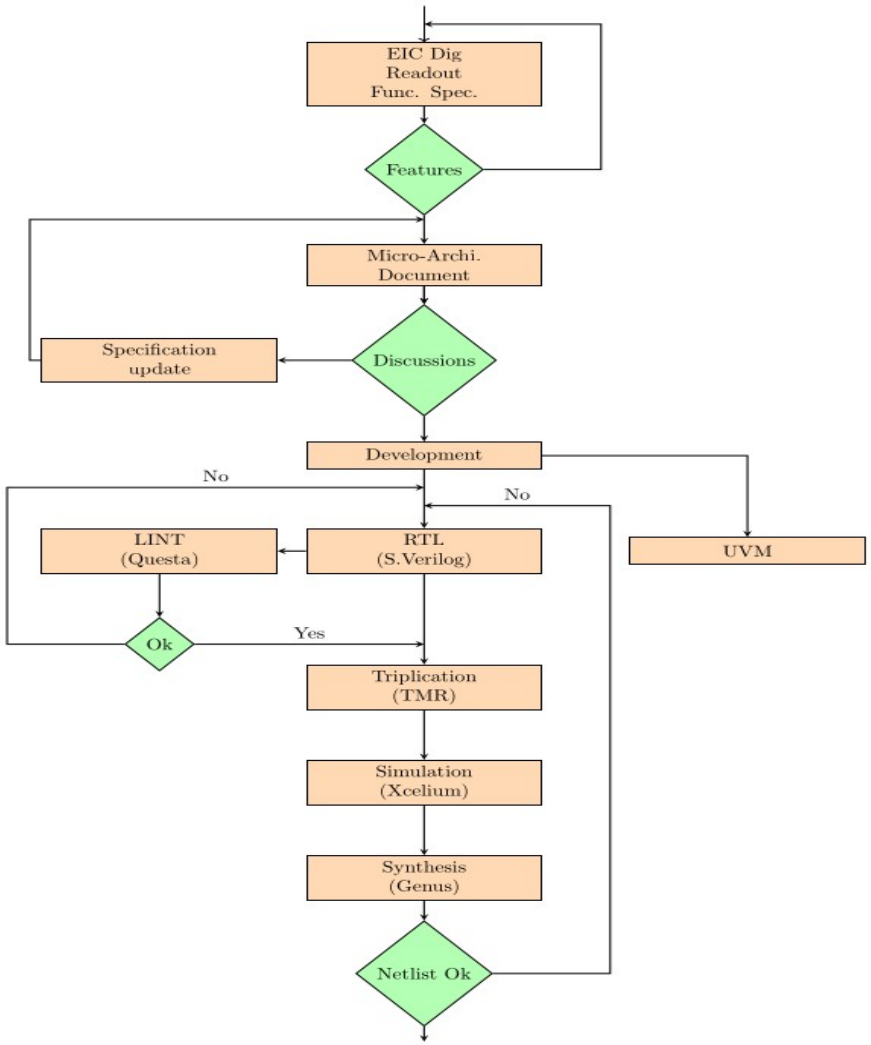
Presentation plan

- Front-end digital steps:
  - Architecture updates
  - RTL updates
  - Simulation updates
  - Synthesis updates
- Back-end digital steps:
  - Floorplan updates
  - Timing updates
- Questions

Presentation plan

- Front-end digital steps:
  - Architecture updates
  - RTL updates
  - Simulation updates
  - Synthesis updates
- Back-end digital steps:
  - Floorplan updates
  - Timing updates
- Questions

Front-end digital design steps



Architecture updates

- DONE :

- Clusters to digital periph Interface (IF) **update** => **read_enable, cloumn_adress** (to save pins)
- **Optimization** of the clusters' **addresses decoding** in the digital periphery
- Implementation specification update => Version **V5 of the digital micro-architecture** document is available

- To Be DONE :

- Specification of the additional features for the digital readout => hit bit, timeout
- Specification of the missing modules : Clocks & resets, Data formating, Serializer, I2C, FC, 8b/10b encoding , zero suppress
- Proposal of the AFE IF (analog front-end interface) => Might be valid for 2 ADC types : ramp and SAR

RTL updates

- DONE :

- RTL code **synthesis-driven cleaning**
- RTL style update : signal naming **homogenization** using standard conventions
- **Triplication** : Cluster => on going
- Pixels registers IF generation using script from N. Arveuf => will also generate UVC for easier registers' verification in UVM
- Cluster + Column + Digital Periphery **intermediate** RTL release on IN2P3 git
- To Be DONE :

- RTL coding of the additional features for the digital readout => hit bit, timeout
- RTL & TMR of the missing modules : Clocks and resets, Data formating, Serializer, I2C, FC, 8b/10b encoding, zero suppress
- RTL coding of AFE IF
- Triplication of Column + Digital Periphery
- RTL Lint checking (Questa)

Simulation updates

- DONE :

- RTL (Cluster + Column + Digital Periphery) simulation using **basic test bench** (One TP, 2 TPs, 1024 TPs, double hit) => **Pass**
- **Triplicated** RTL (Cluster + Column + Digital Periphery) simulation using basic test bench => **Pass**
- **UVM** setup started: first test available => **simple** randomized hit injection (Cf. backup slide : UVM Simulation- First test)

- To Be DONE :

- Simulation of the additional features for the digital readout => ADC counter, hit bit, timeout
- Simulation of the missing modules : Clocks and resets, Data formating, Serializer, I2C, FC, 8b/10b encoding, zero suppress
- Exhaustive test plan elaboration
- UVM additional tests : configuration tests, other tests,…

Synthesis updates

- DONE :
- Column and digital periphery **first** synthesis => implemented optimizations of the RTL (**area** & **timing** pb resolved)
- TMR Cluster final/**clean** synthesis on progress => fixing/analyzing synthesis check_design messages (undriven, assigns, …)

- To Be DONE :
- TMR Column and digital periphery final/clean synthesis
- Synthesis of the other upcoming digital modules

Global progression – Digital front-end implementation

Block	RTL	TMR	Synth
Cluster	🟡	🟡	🟡
Column	🟡	🟡	🟡
Digital periphery	🟡	🟡	🟡
Top	🟡	🟡	🟡
I2C	🔴	🔴	🔴
Fast Commands	🔴	🔴	🔴
Data Formatting	🔴	🔴	🔴
Serializer	🔴	🔴	🔴

- 🟢 Done
- 🟡 On Going
- 🔴 Not started

Global progression – Verification (UVM)

- Setup of UVM environment just started
- Only one test is available : simple hits injection
- Code coverage

Task	Status
Verification plan	🔴
Injection tests	🟡
Configuration tests	🔴
Other tests	🔴
Regressions	🔴
Coverage	🟡
Target : RTL	🟡
Target : Triplicated RTL	🔴
Target : Post layout netlists	🔴

Presentation plan

- Front-end digital steps:
  - Architecture updates
  - RTL updates
  - Simulation updates
  - Synthesis updates
- Back-end digital steps:
  - Floorplan updates
  - Timing updates
- Questions

Physical Implementation updates

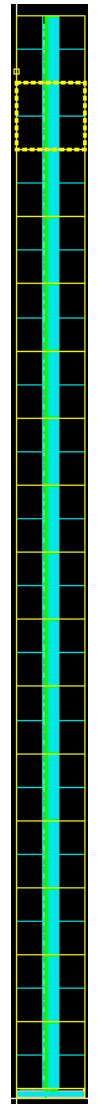
- Update floorplan of cluster/column
  - ➔ Fix some misalignment issues causing lots of DRC violations
  - ➔ Pins positions updated for easier routing

- Timing : hold fixing

Huge skew difference in STA between capture/launch clocks

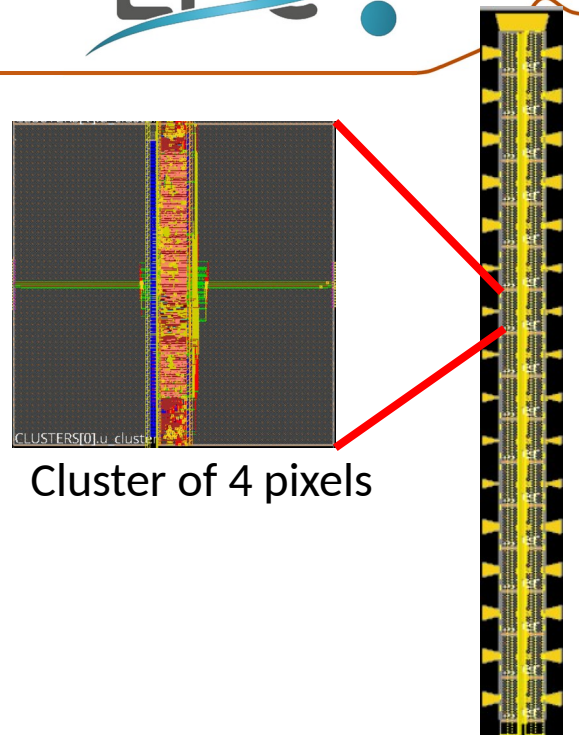
  - ➔ Added a buffer for each triplicated clock at cluster/column entrance
  - ➔ Update NDR for clocks :
    - Increase minimum width
    - Increase minimum spacing

- Digital blocks now available on SOS



Global progression – Physical Implementation

- Digital on top flow for the matrix and digital periphery
- 3-step implementation : cluster / column / top
- First implementation of cluster and column
  - Check that it fits in allocated area using TSMC130 nm
- Setup of power analyses on going, preliminary results available



Cluster of 4 pixels

Column

Block	Floorplan	CTS	Route	TimingAnalyses	DRC/LVS	PowerAnalyses
Cluster	🟡	🟡	🟡	🟡	🟡	🟡
Column	🟡	🟡	🟡	🟡	🟡	🔴
Top	🔴	🔴	🔴	🔴	🔴	🔴

Presentation plan

- Front-end digital steps:
  - Architecture updates
  - RTL updates
  - Simulation updates
  - Synthesis updates
- Back-end digital steps:
  - Floorplan updates
  - Timing updates
- Questions

Questions

- TSMC130 discontinuation of MPW runs

  => No change, keep going with TSMC130

- Gray counter : 1 per pixel/1 per cluster ? Triplicated ? reset before hit?

  => No gray counter implemented yet

- Interface analog/digital, which signals and their function

  => Laurent can look into it but needs pixel analog block shared

- Clock(s) and resets specs
- Authorization for slave I2C
- Fast Command block

  => Check with Fred once we're implementing it

Backup

Simulation UVM – First Test

- First test has been setup : randomized hit injection
- Completely rewrite pixel_analog model for UVM, might need some analog experts inputs at some point
- Test is doing the following :
  - Randomize hit frequency
  - Randomize pixel hit (only one pixel at a time for now)
  - Randomize TDC/ADC values on pixel hit
    - Neighbors pixel only produce a randomized ADC value which is 20-80% of initial pixel hit
  - Checking : Regroup pixel with same BCID into event (usually 9 pixels per event)
  - Check that injected event is the same as event read in the periphery
    - Number of pixels, pixelIDs, ADC/TDC values

Dead time

What is the deadtime?

Dead time corresponds to the time a pixel is **blocked** and will not acquire new hits : pixel is **blind** during that time

- Analog dead time corresponds to the **ADC conversion time**
- Digital dead time is the time to get the **data from the pixel to the periphery**

- With the following hypothesis :
  - Event randomly distributed
  - Asynchronous system

  Efficiency can be defined as $$R = \frac{1}{\lambda\delta + 1}$$

  Where λ is the event rate and δ the dead time

- From Alex Jentsch : « Event rate of 100Hz for the 10% pixels closest to the beam (beam halo), 10Hz for other pixels »
  - Event rate used for analyses is 20Hz per pixel as default, and 200Hz as worst case (considering double pixels hit)
- Dead time comes from simulation results of our architecture

Dead time

- Dead time also highly depends on the number of pixels blocked when a hit occurs
- Plots show how dead time impacts data loss for different blocking cases
- **Specifications** :
  Data loss due to dead time : **0.001% / 0.05%** (99.999 % / 99.95 % efficiency)
- **First estimation of data loss with such readout :**

Considering an **average** analog dead time (ADC converstion time) of 400ns

Data loss for Clusters, 3x3 readout	Rate = 20Hz	Rate = 200Hz
Default (1 hit = 1 TP, 8 TL1)	0.028%	0.28%
Multiple (1 hit = 2 TP, 10 TL1)	0.032%	0.32%
Worst case (1 hit = 4 TP, 12 TL1)	0.040%	0.40%

<0.05% data loss is achievable
0.001% data loss not possible for ADC with 800ns conversion time