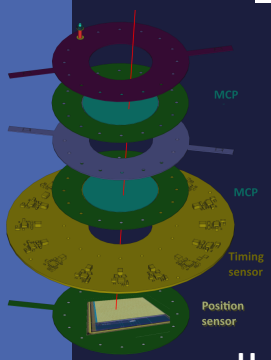
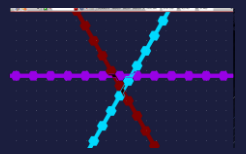
PICMIC0 : Tri-axis 5µm hexagon pixel-strip matrix combining 3*852 current comparator in a 180nm node





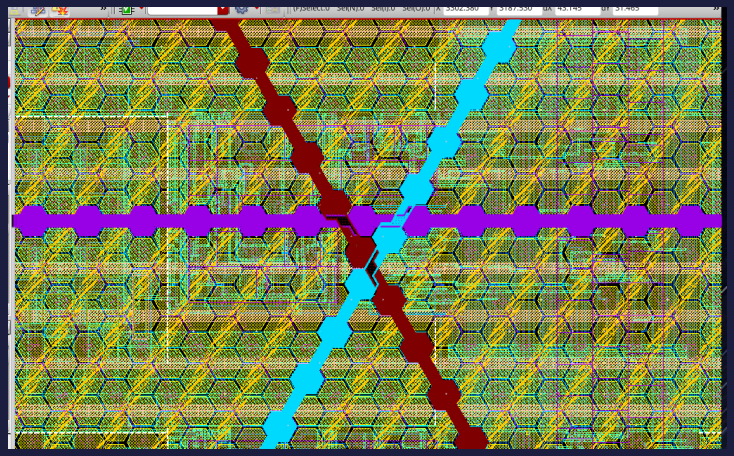
H. Abreu[(1)], E. Bechetoille[(1,4)],G. Bertolone[(2,4)], G. Claus[(2)], C. Colledani[(2,4)], C. Combaret[(1)], G. Doziere[(2,4)], C. Hu-Guo[(2,4)], I. Laktineh[(1)], H. Mathez[(1,4)], H. Pham[(2,4)], M. Specht[(2)], I. Valin[(2,4)], L. Zhang[(3)], Y.Zhao[(2,4)]

(1) IP2I, Institut de Physique des 2 infinis, CNRS/IN2P3

(2) IPHC, Institut Hubert Curien, CNRS/IN2P3

(3) SDU : Shandong University, China

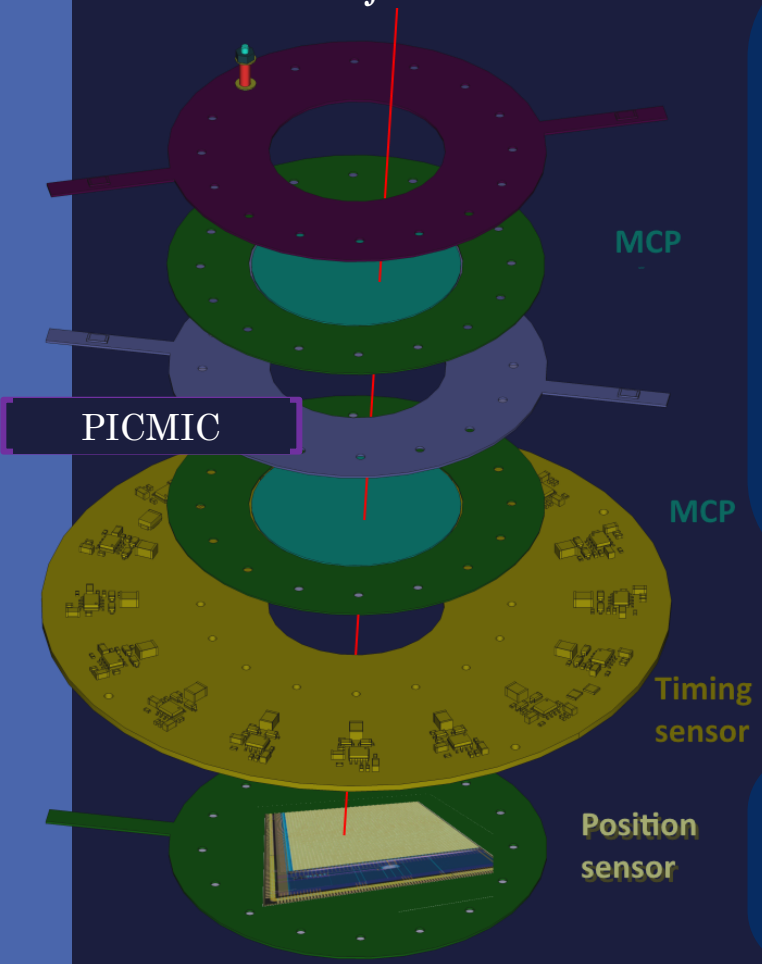
(4) Mi2i: Microélectronique des deux infinis (CNRS/IN2P3)



Plan

- Overall Project : Picmic in numbers and animation
- Detail of the PICMIC-0 ASIC
  - Layout 3D views
  - Analog part
  - Digital part
- Experimental results
  - Description of the test bench (HW + FW)
  - Static Comparator threshold S-curves
  - Dynamic Comparator Hat-curves
  - 5um position measure of two injection probes
- Today's setup
- Conclusion and perspectives

PICMIC Project in numbers



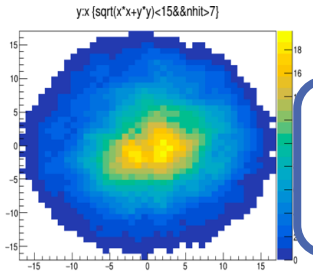
Started in 2019.

- IN2P3 R&T funded for 3 years @ 25k€/years
- CNRS pre-maturation funding : 150k€ for 2 years.
- 2 ANR (French national funding) started to go from MCP to NCP (Micro to Nano Channel Plate)
- Patent protected
- Industry company are waiting for the result of the PoC (Proof of Concept)
- The project paid a part of the TJ180nm foundry in 2021 : 60k€ (including dicing)
- Human resources for the PICMIC-0 ASIC : 10 persons for 2 years, which represent **3 Full-Time-Equivalent for 2 years**.

- Main interest : gain a factor 100 on the state of art

- Principle : Coincidence between a timing sensor and a position sensor in order to measure at the **PICo-seconde** level, the passage of an incident particle together with a precision at the **MICro-meter** level.

Patent : PCT/EP2020/050058



3ps Time precision + 1mm position estimation

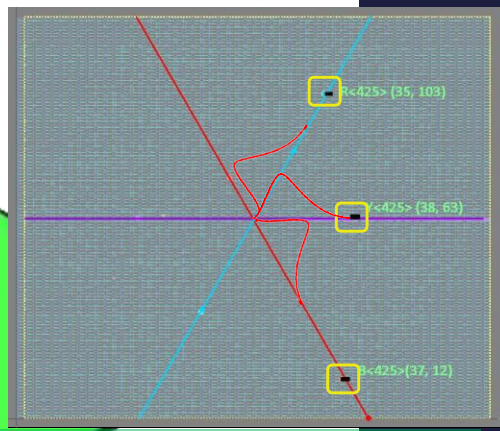
t_i-t_j =>

SAMPIC HW

MCP

(tracker-like) Timing sensor

(electron absorber) Position sensor



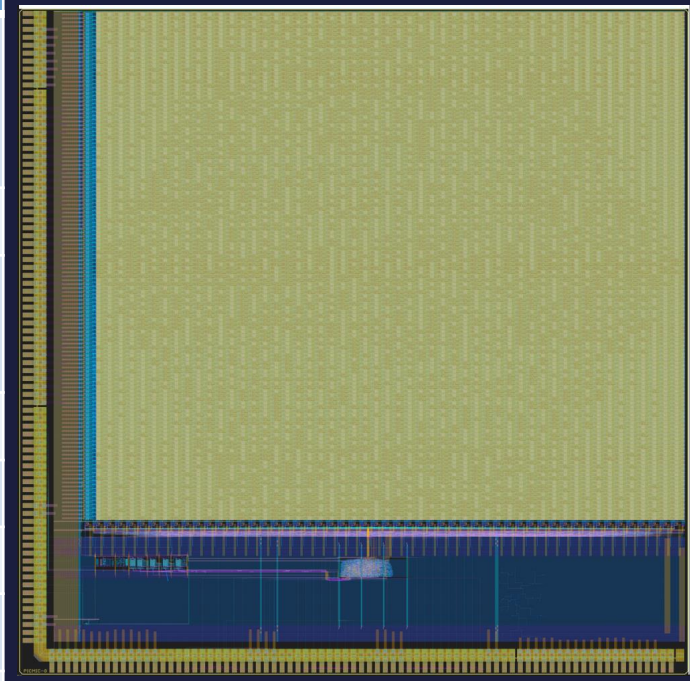
(Row,Col)(35,103)(38,63)(37,12)

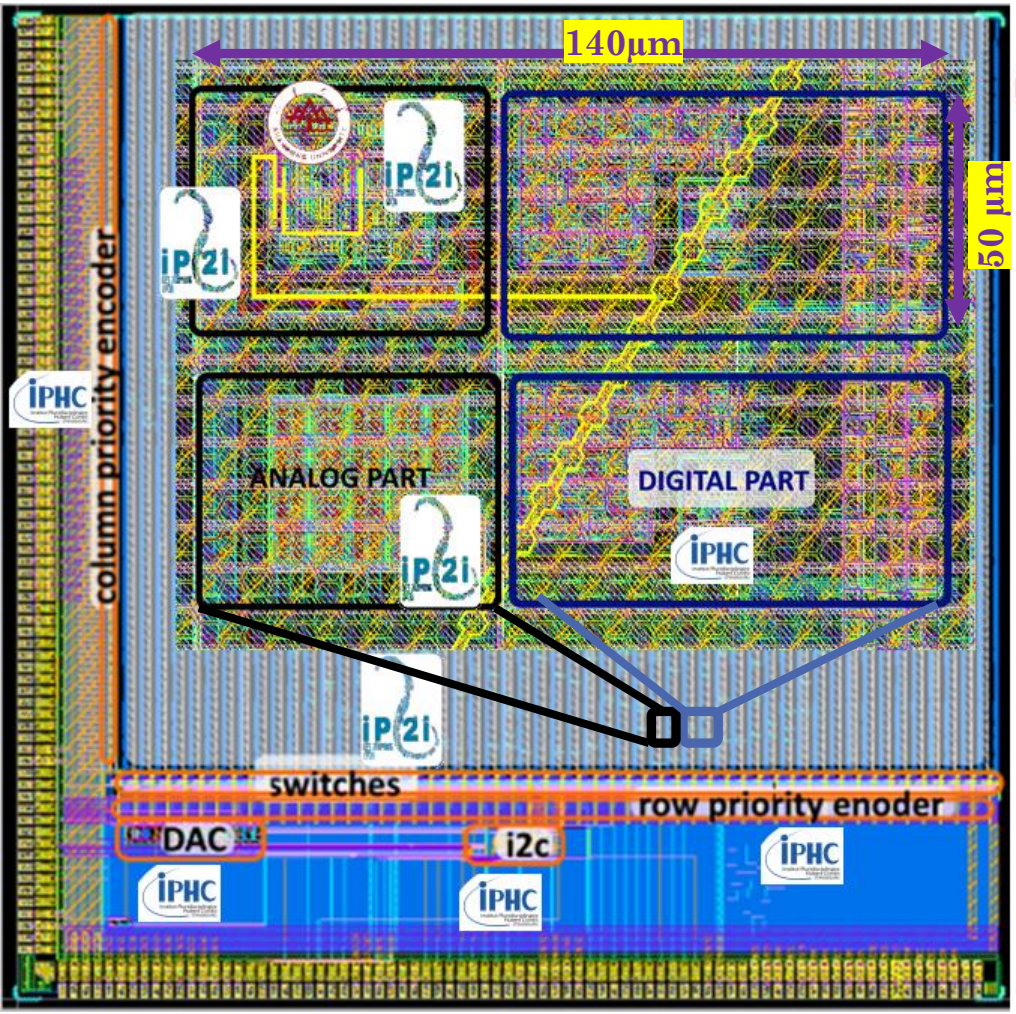
SAMPIC FirmWare

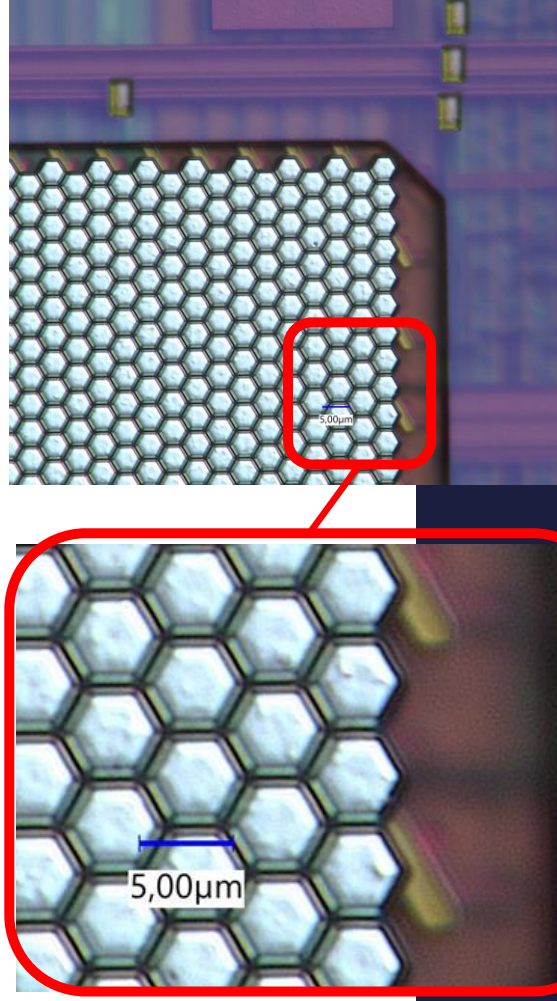
5µm position, in 400ns frame resolution

PICMIC-0 : Specifications

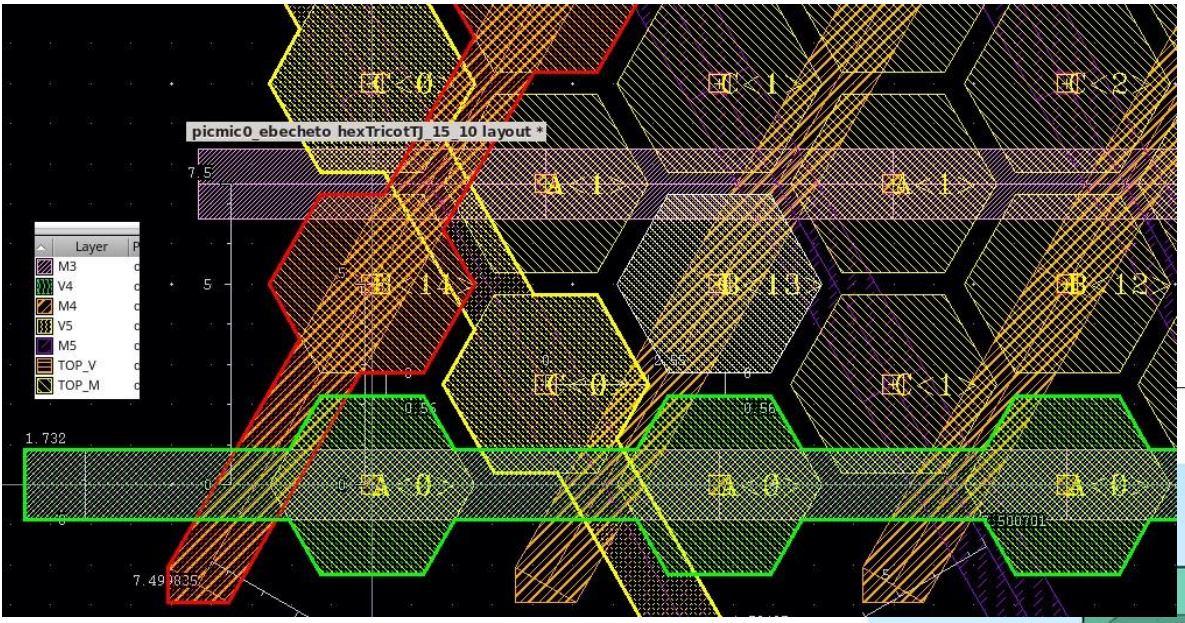
Circuit size	8.4 mm * 8.4 mm
Pixel matrix size number of row * number of column active cell dimension active readout cells	7.4 mm * 6.4 mm 53*128 140µm x 50µm 3*852=2556 canals
maximal power consumption	256mA*1.8V
Triangle /\ *input* signal range [current x pulse width]	between [13uA x 2ns] (13fC) and [1.8mA x 0.4ns] (360fC)
Maximum hit rate	16 cells touched during 400ns
Internal readout clock	40MHz by priority encoder
External readout clock	40MHz of 14-bit in parallel
Pitch (positional quantisation step)	5µm (height of an hexagon)
total number of hexagons	2 077 200 (~2 Million)



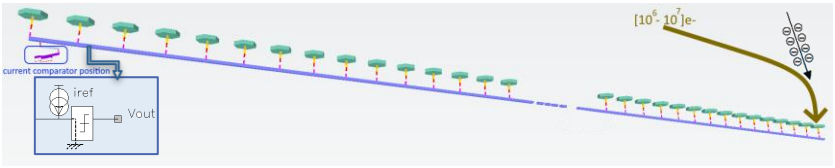


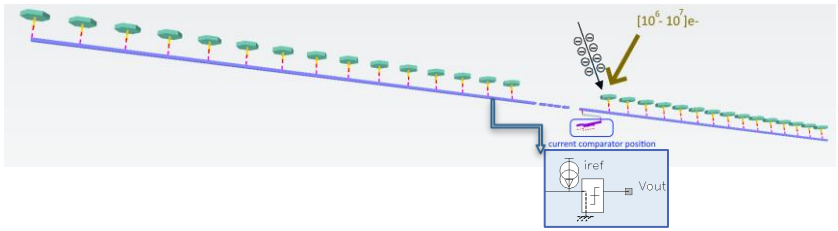


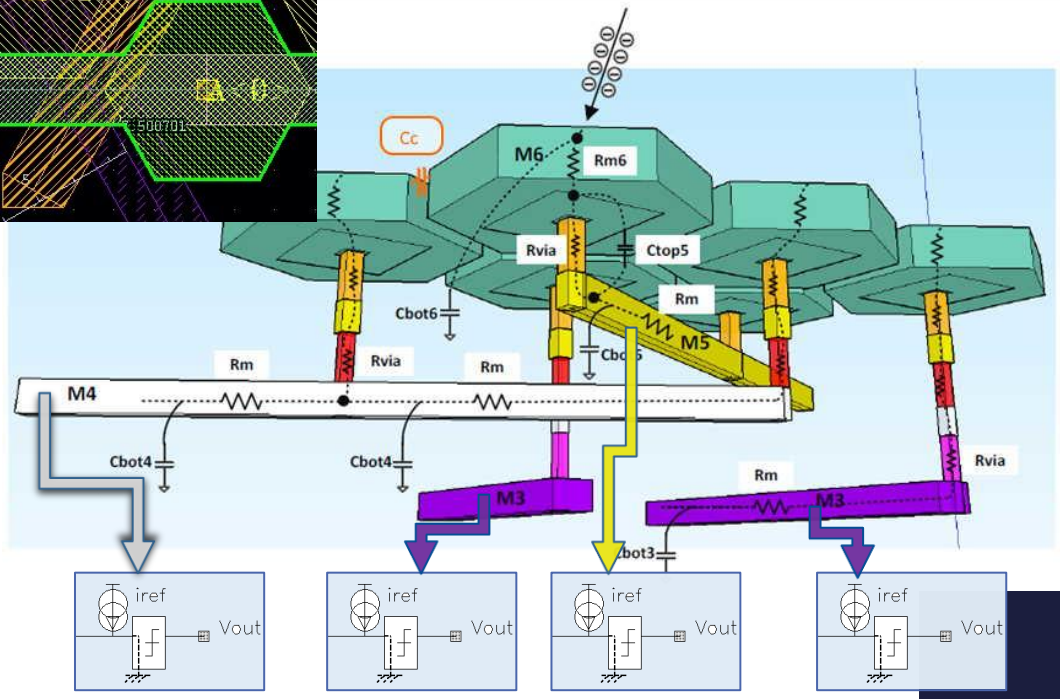
3 directions * 852 hexagon-strip-lines, in TOP metal interconnecter in Metal-543
[53,128] matrix of Analog+Digital comparator readout in M1 an M2
each line gets a comparator



3 directions * 852 lines,
each line gets a comparator

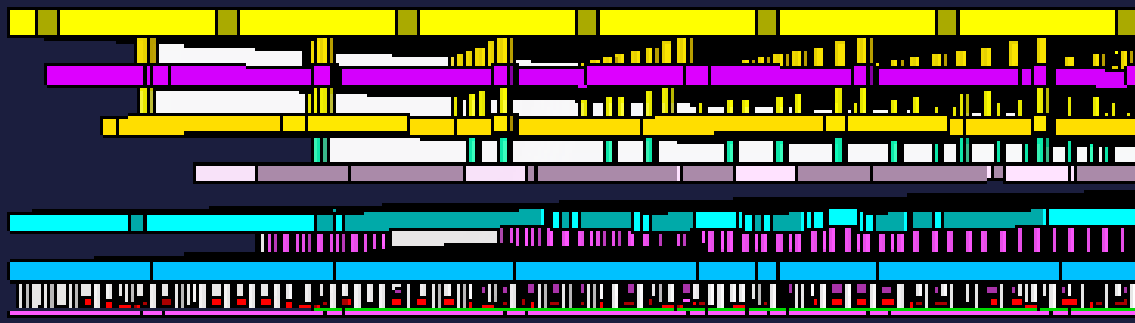


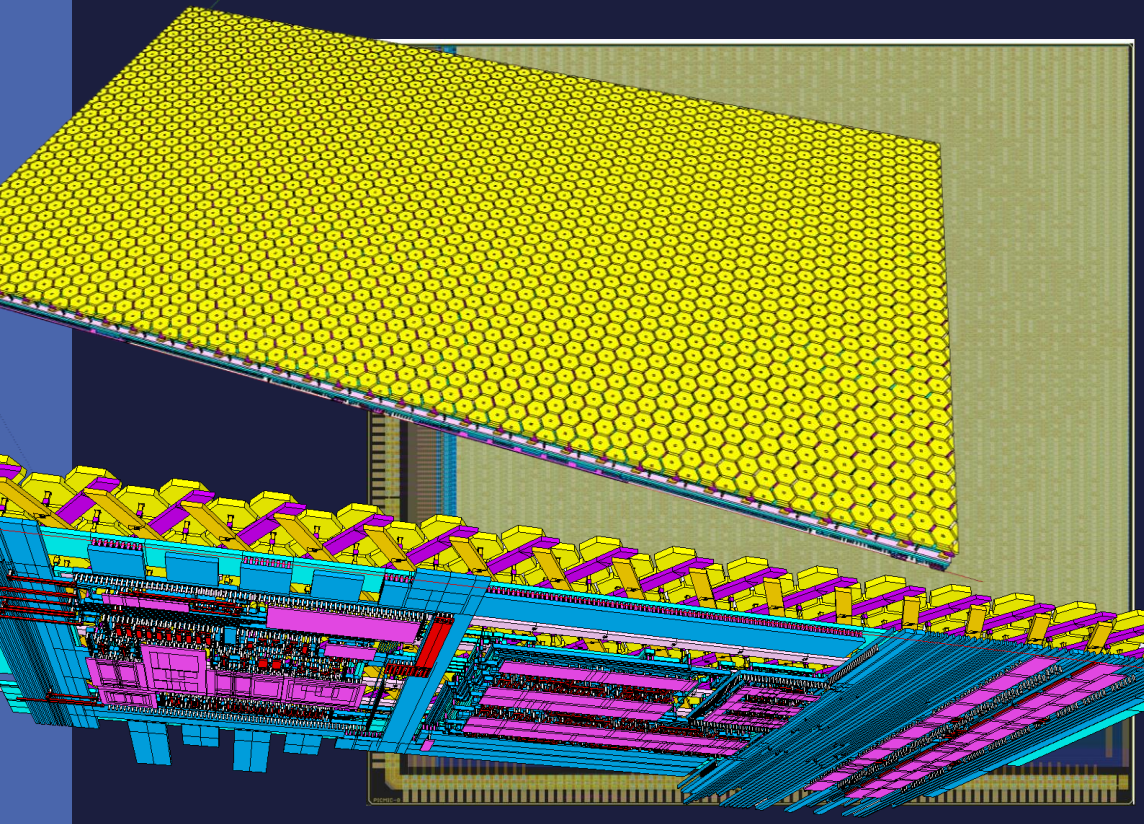


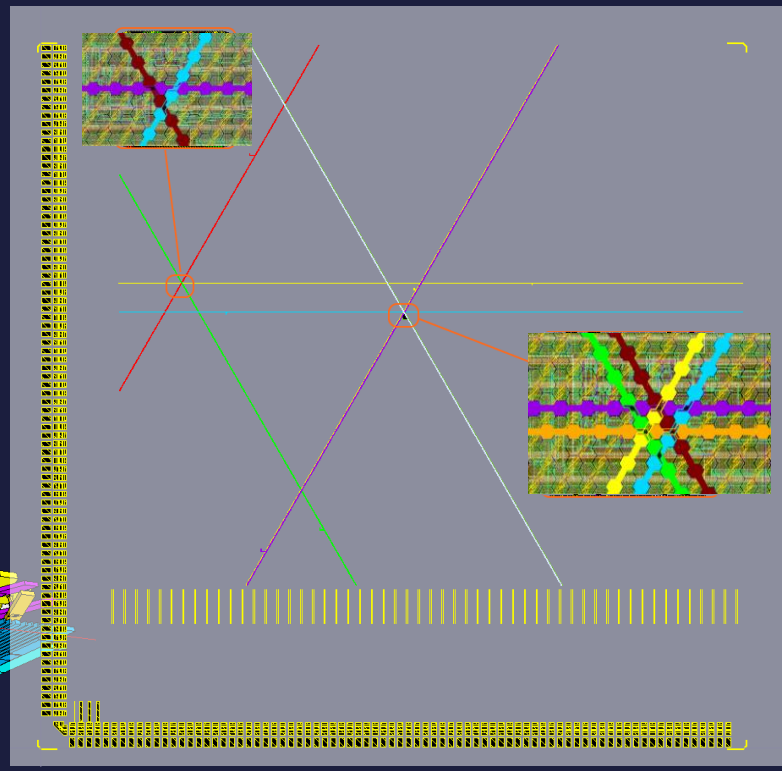


Depending on the algorithm, the readout-cell is somewhere between one side or another
Depending on the position of the shower impact, the comparator could be below or at the opposite side.

hexagone	MTOP
interco -60°	M5
interco +60°	M4
interco 0°	M3
Metal 2	M2
Metal 1	M1
Transistors	Poly

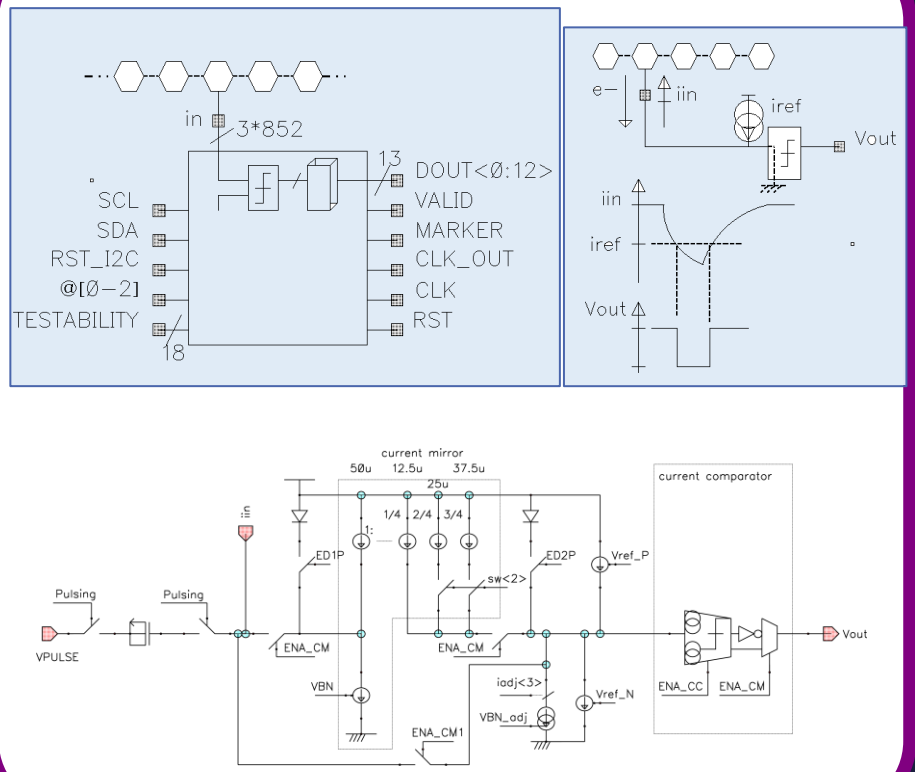




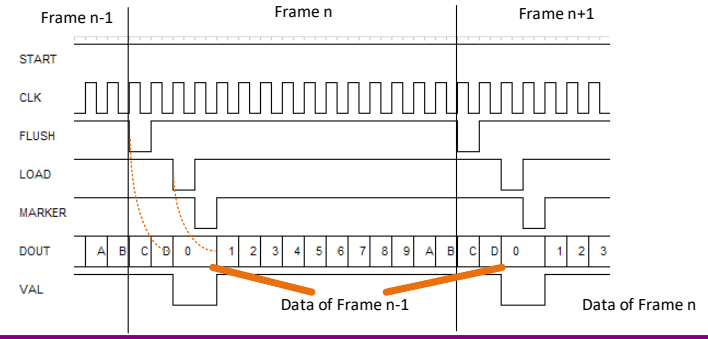


Synoptic : current comparator matrix element

❖Analog readout



❖Digital readout



Very front-end:

- ❖Current mirror + current comparator
- ❖ injection MOScap inside front-end
- ❖Current reference: global 8-bit tuning DAC + 3-bit local
- ❖ protection diodes

i2c Slow control :

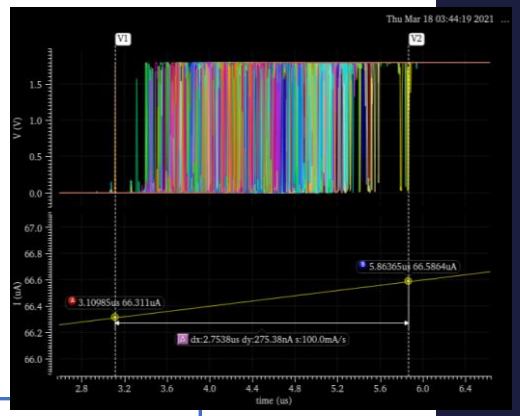
- ❖ global registers + each readout-cell locally
- ❖ Column readout by priority encoder

❖ Parallel readout :

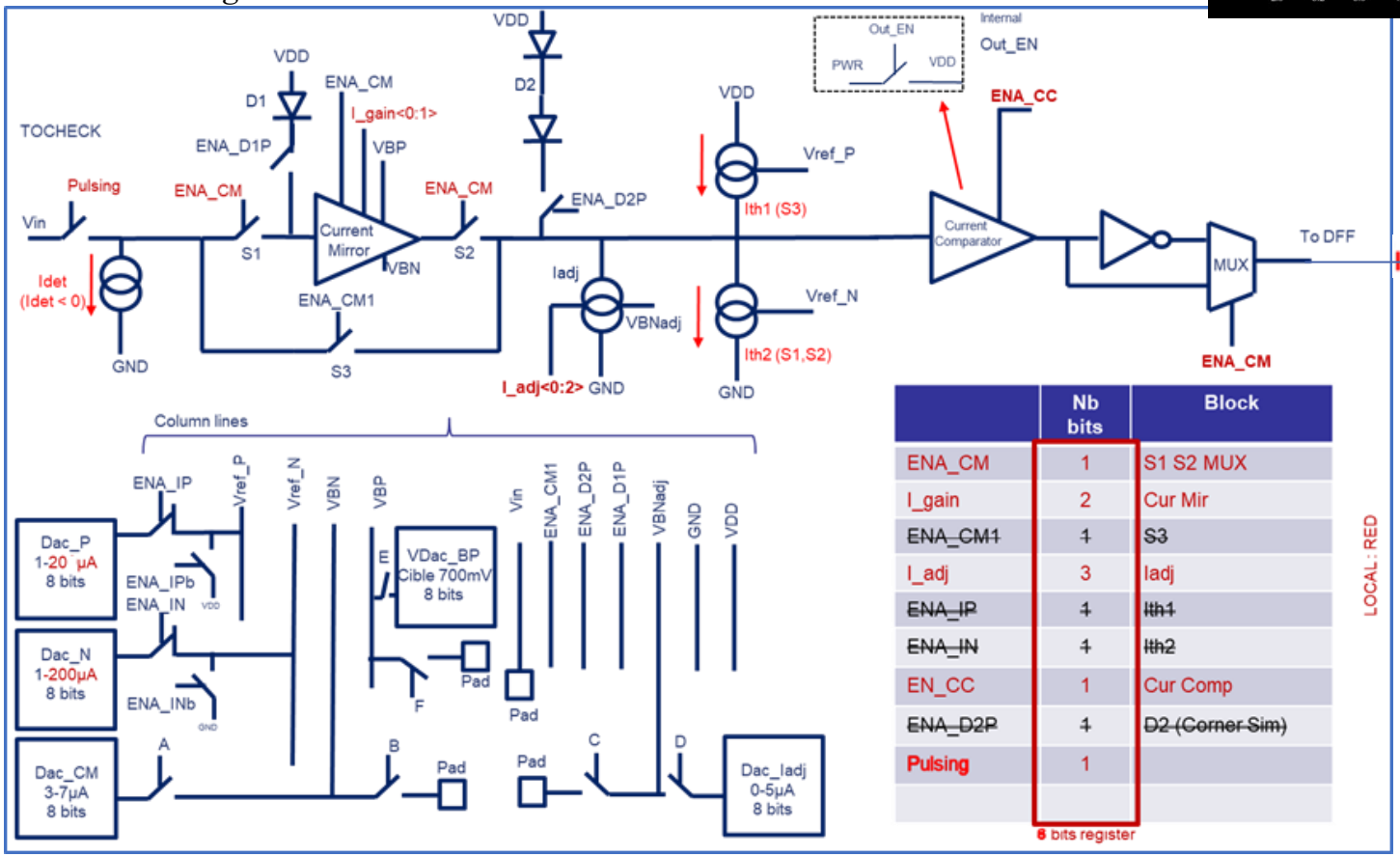
- ❖ 13-bit data (touched pixel address) + Marker
- ❖ 40MHz // readout speed
- ❖ 16 sample by frame : 16* 25ns => 400ns frame

Very Front end synoptique

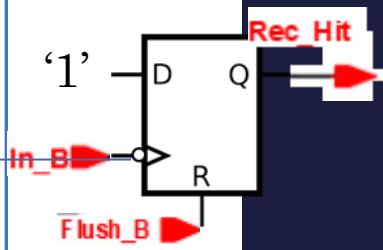
- Common global configuration
  - DAC references,
  - diod protection,
  - current-mirror mode on/off
  - Current comparator is a typical Traff-cell
- Transient noise threshold :
  140 nA_{RMS}



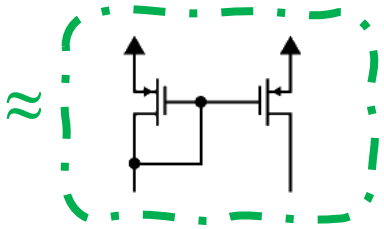
- 7-bit local registers

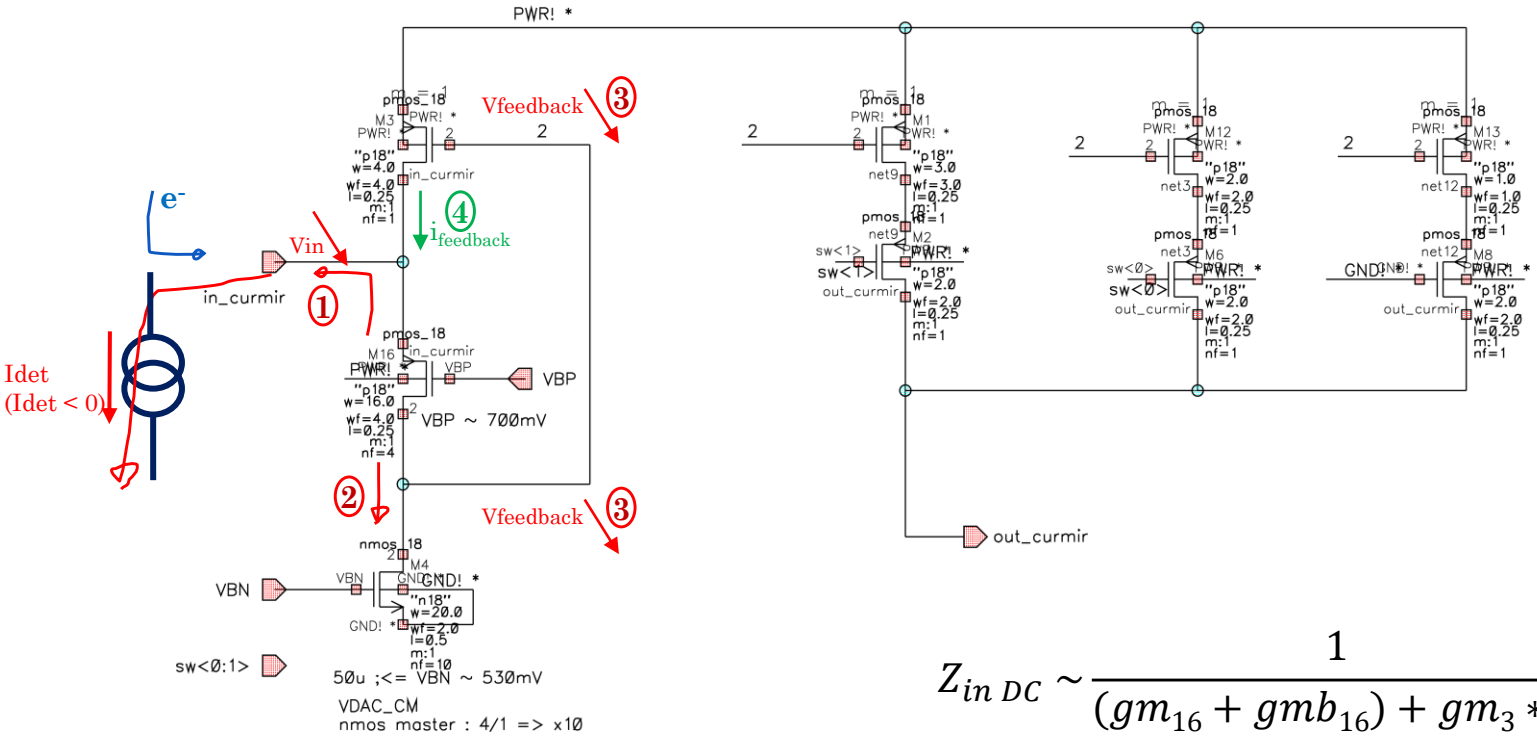


	Nb bits	Block
ENA_CM	1	S1 S2 MUX
I_gain	2	Cur Mir
~~ENA_CM1~~	~~1~~	~~S3~~
I_adj	3	Iadj
~~ENA_IP~~	~~1~~	~~Ith1~~
~~ENA_IN~~	~~1~~	~~Ith2~~
EN_CC	1	Cur Comp
~~ENA_D2P~~	~~1~~	~~D2 (Corner Sim)~~
Pulsing	1	



By-passable current mirror



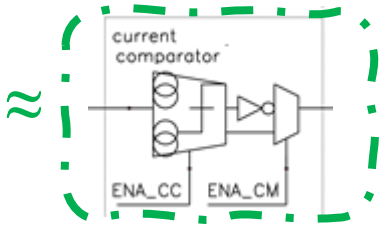


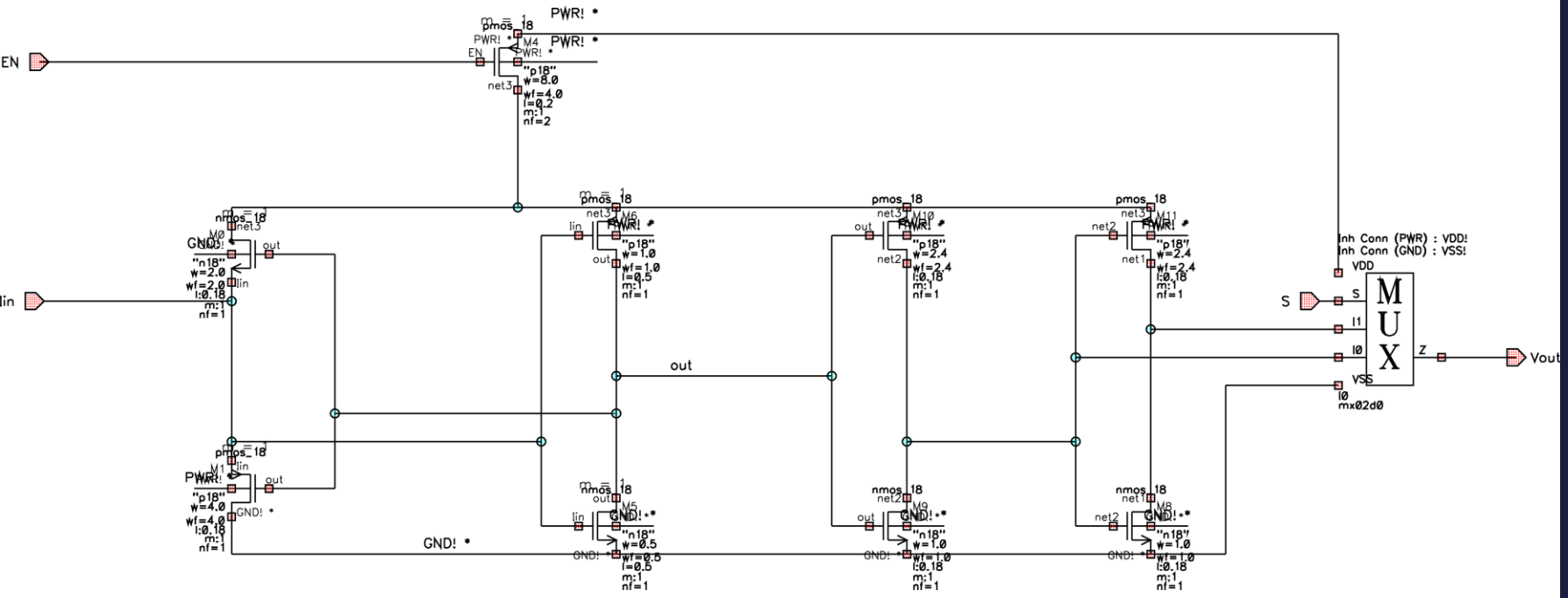
$$Z_{in\ DC} \sim \frac{1}{(gm_{16} + gmb_{16}) + gm_3 * A}$$

$$A(SD\ M_{16}) \sim 1 + \frac{gm_{16} + gmb_{16}}{gds_{16}}$$

① & ④ steps are opposed by feedback to stabilize input impedance

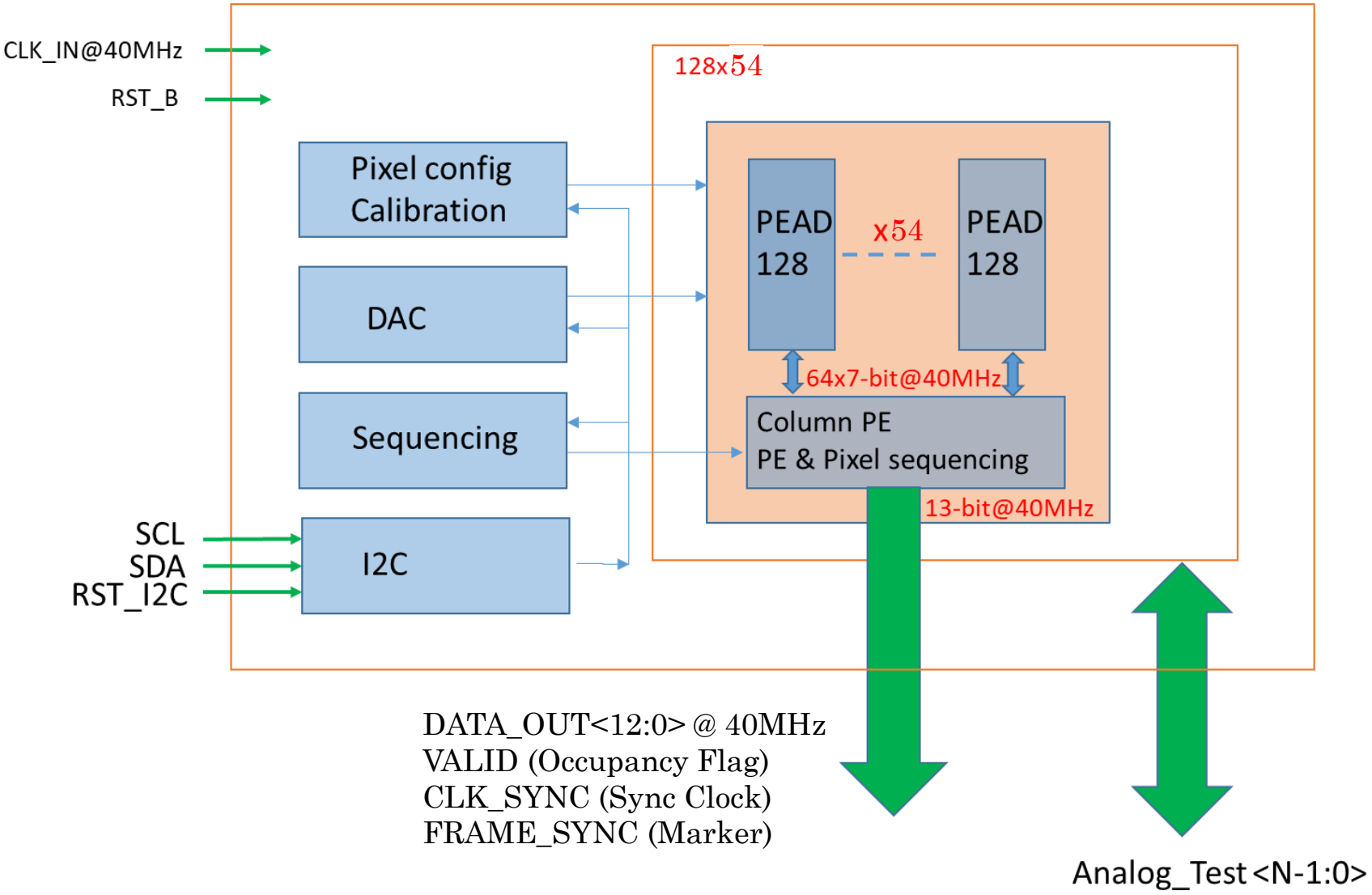
Current comparator is a typical Traff-cell



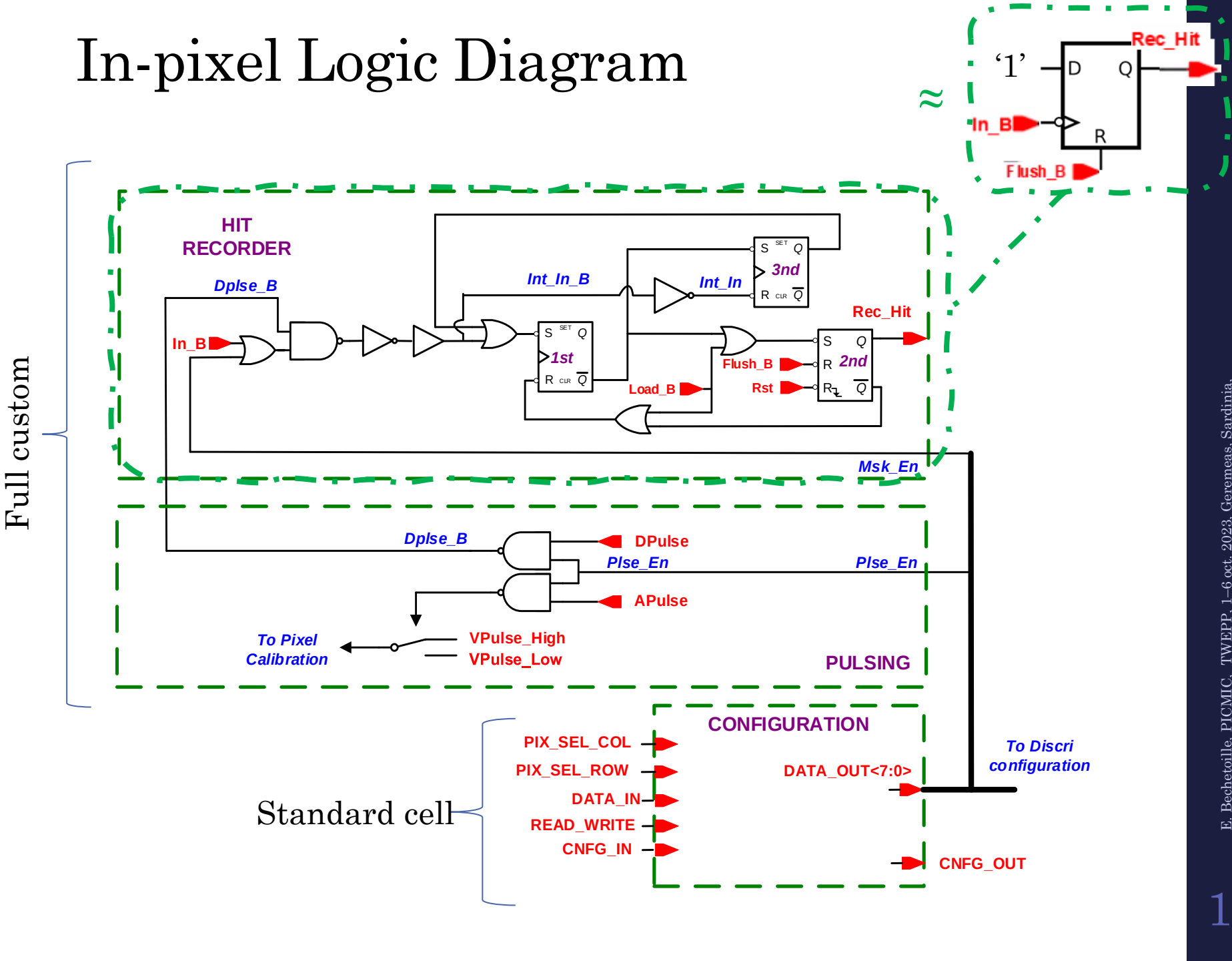


i_threshold $10uA_{DC}$

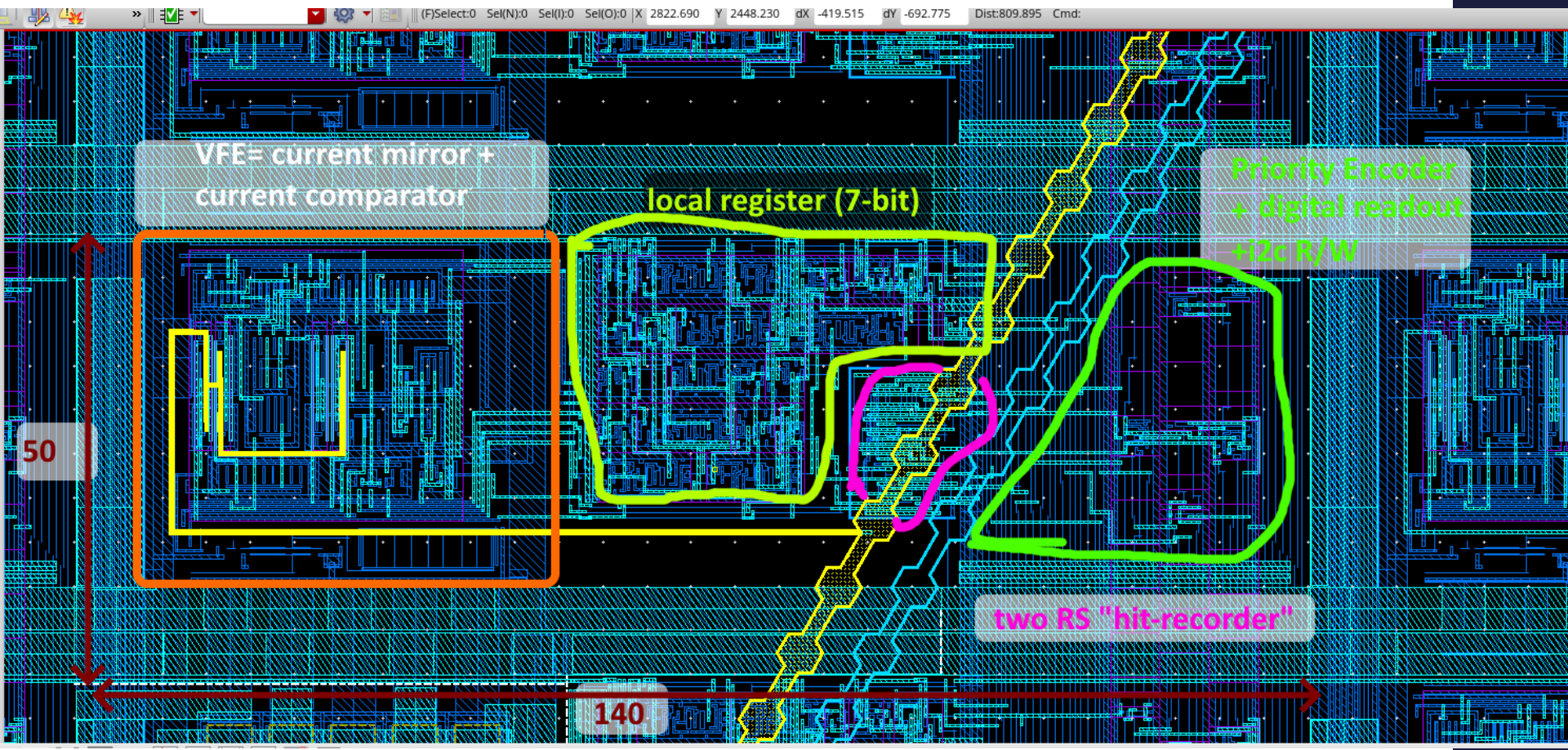
Digital implementation Diagram



In-pixel Logic Diagram

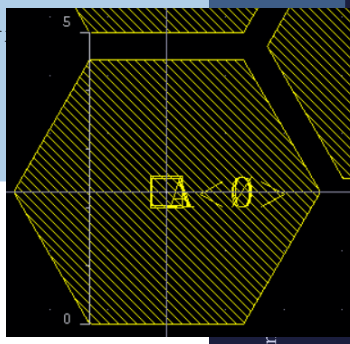


M1 M2 Layout readout-cell detail 140u x 50u



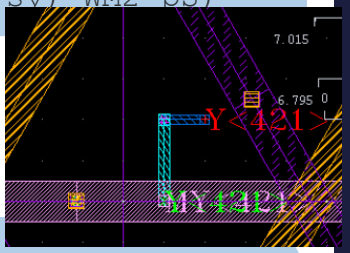
- Every TOP to M3 layer inside the active matrice had been written in **skill code**.
- Plus the additional M2 line essential to the interconnection algorithm
- More than a thousand effectives code lines

```
procedure(createHexagon( @optional (x 0) (y 0) (layer "MET1")  (cv nil) (r 8/sqrt(3)) (orient "R0")(fixGrid t))
let( (rot pi i pointList dg grid) unless(cv cv=geGetWindowCellView())
grid=0.005 pointList='() rot=0 pi=acos(-1) dg=list(layer "drawing") pointList=tconc(pointList list(x y))
for(i 0 5  rot=rot+pi/3.0 x=x+r*cos(rot) y=y+r*sin(rot) pointList=tconc(pointList list(x y)));;end i
pList=car(pointList)
(when fixGrid pList=foreach(mapcar xy pList list(round(car(xy)/grid)*grid round(cadr(xy)/grid)*g
hex=dbCreatePolygon( cv dg pList)
cBox=centerBox(hex~>bBox) dbMoveFig(hex nil list(0.5*r:-r*sqrt(3)/2 orient)) hex));
load("~/Skill/hexagone.proc.il")
```




```
dbCreateVia(cv techFindViaDefByName(tech "ML_M5_S") (0:0) "R0")
```

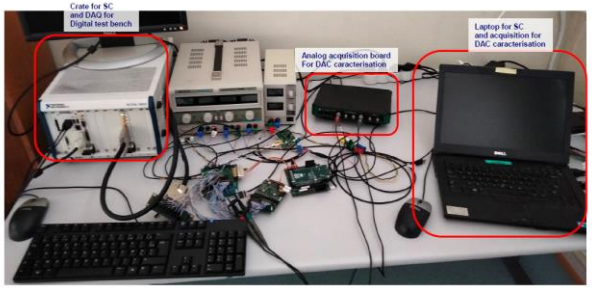
```
stackVias(viasM5_M2 Xmod:Ymod cv)
createPinLPPcv(pinName xy2bBox(Xmod:Ymod WM2/2.0) "M5" nil cv t)
dbCreatePath(cv "M2" list(Xmod:Ymod X0+sx:Ymod X0+sx:Y0+sy X0+vfe_sx:Y0+sy) WM2 pS)
```




```
pavement=dbOpenCellViewByType( "picmic0_lib" "pavement4" "layout" "maskLayout" "r
inst=(dbCreateInst cv pavement nil X0:Y0 "R0");<== insert pavage hexas_5u in cellView
dbFlattenInst(inst 1 nil t)
```

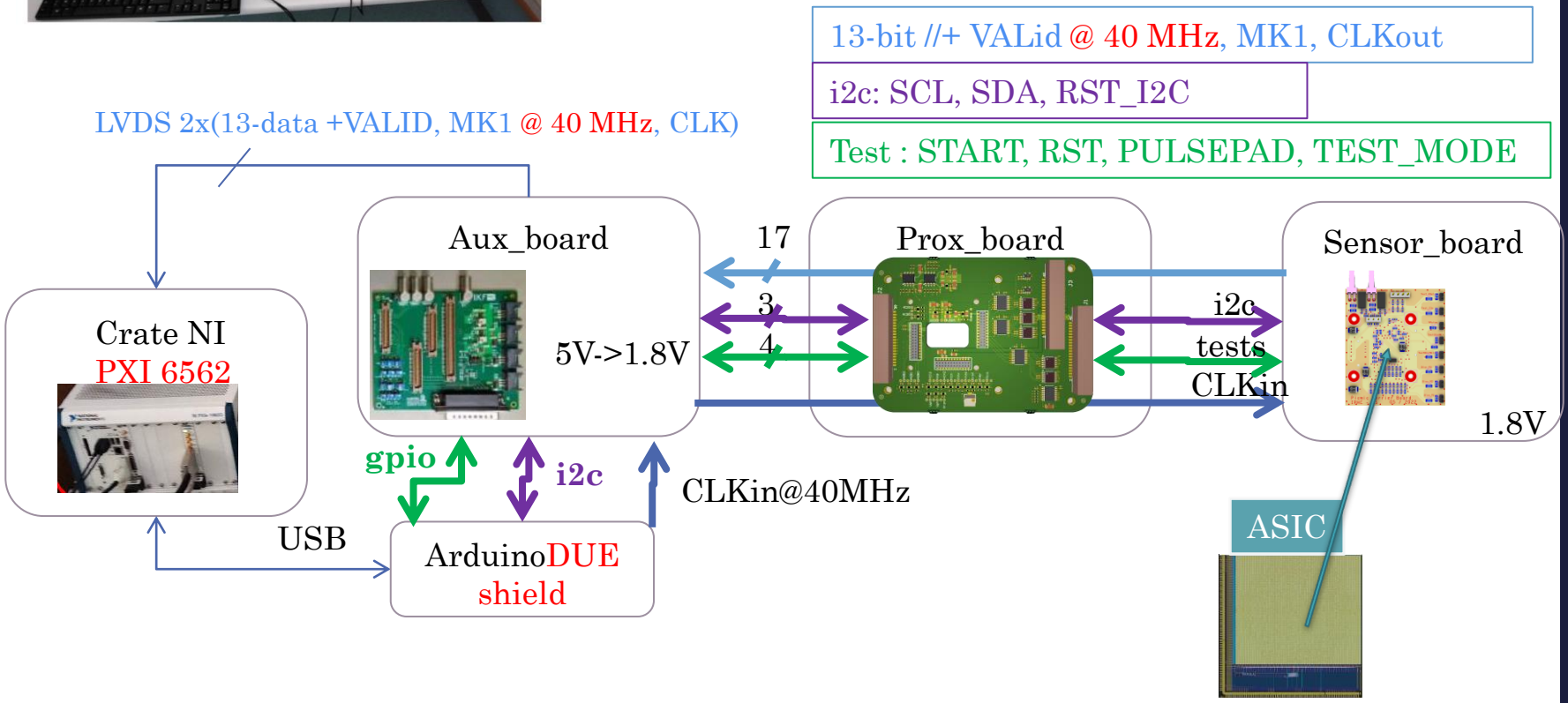
...

PICMIC0 DAQ



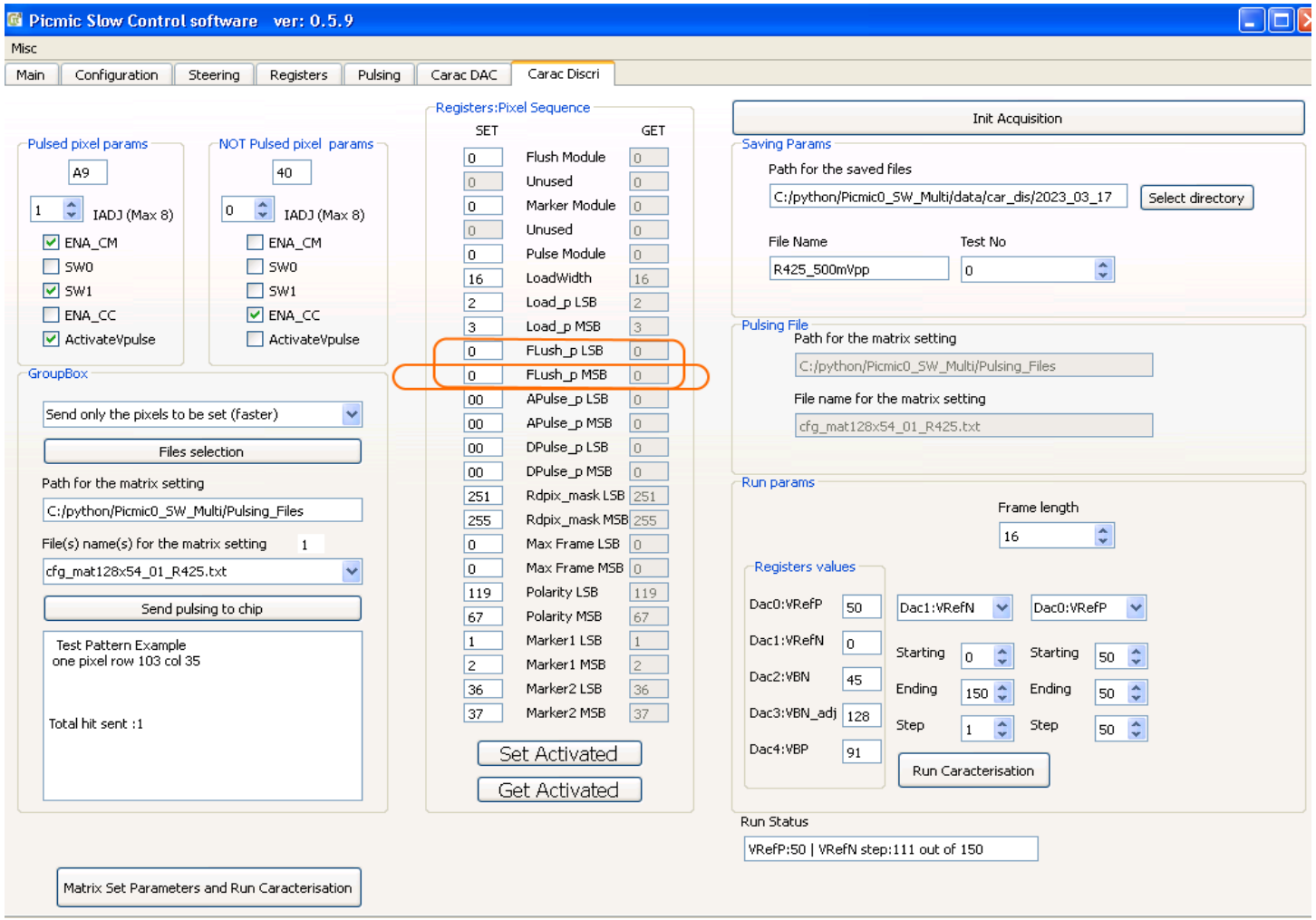
Test bench HW developped by IPHC

- New design for sensor board & Arduino shield
- Most of test bench HW **reused** from a previous project
- I2C : Arduino DUE board
- DAQ : NI PXI6562 board 16 I/O up to 200 MHz

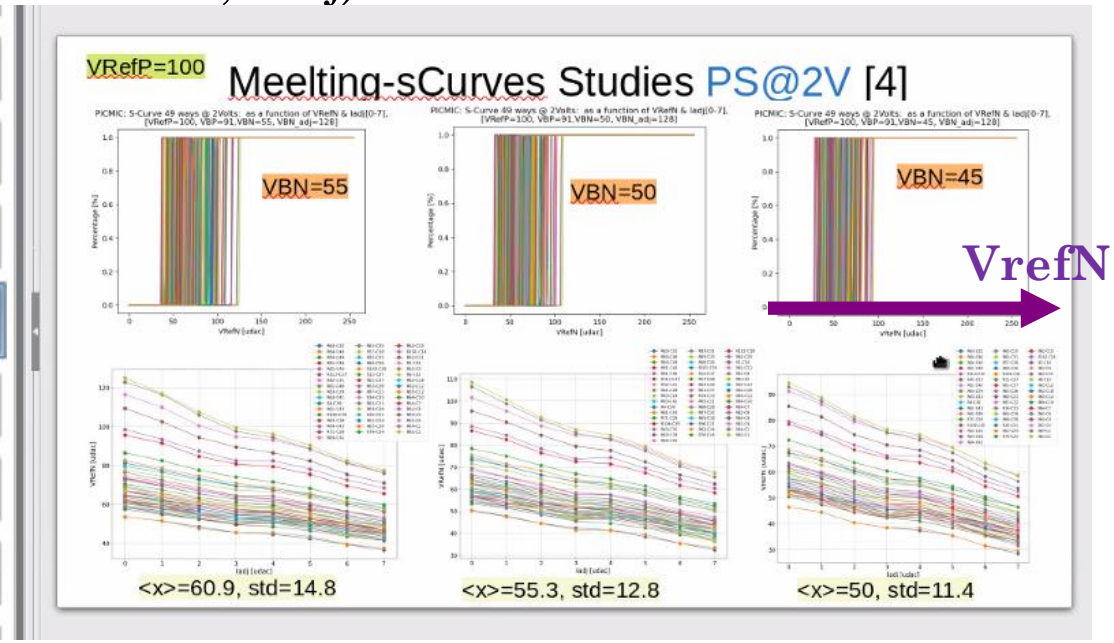


Test bench SW developped by IPHC & IP2I

- ✓ GUI application I2C + DAQ (PC : Python + DLL written in C, uC : Arduino C, C++)
- ✓ System validation
- ✓ PICMIC functionnal test
- ✓ Integration of automatic discriminators characterization in GUI application



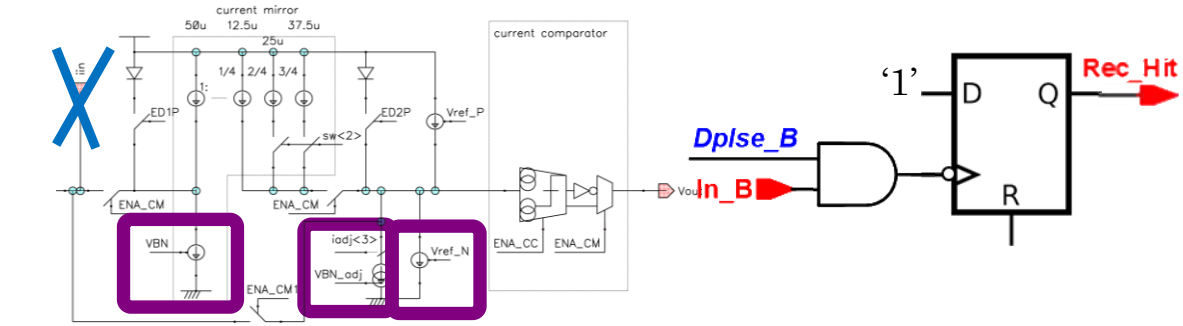
3 dimension scan feature depending on various tuning parameters (VBN, VrefN, iadj)

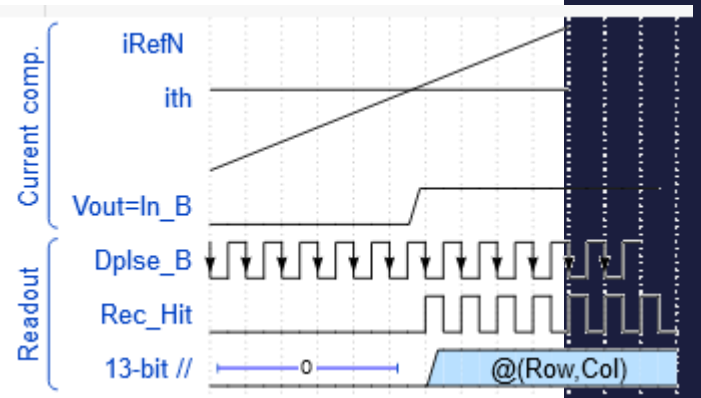


1st results :

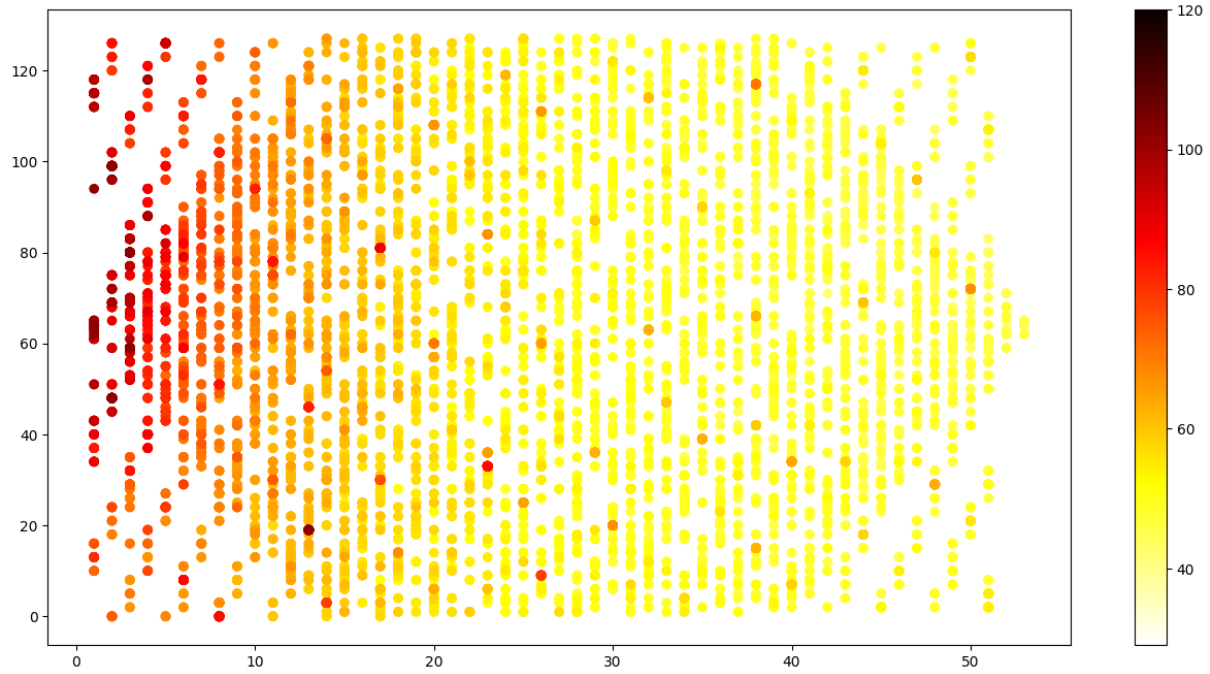
- Piedestal S-curve of the threshold trigger measure up to 5 dimensions
- Here we measure threshold spread as function of the global current parameter (**VrefN**)
- The adjustment of the 3-bit inner readout-cell local current source permits to adjust the 50% crossing of the S-curve.
- Digital readout (without input signal)

Without input signal



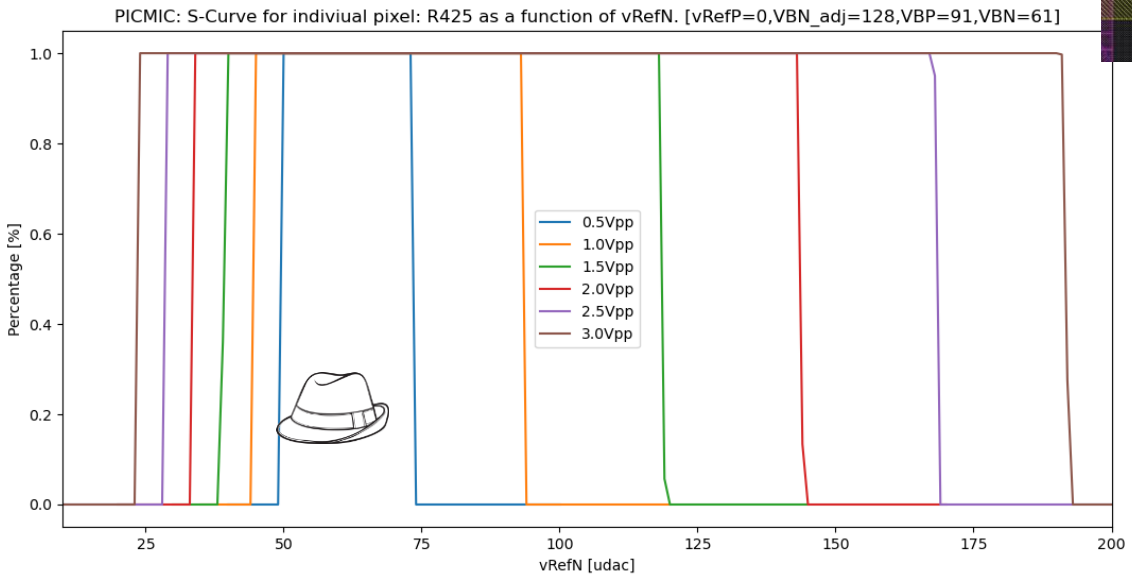


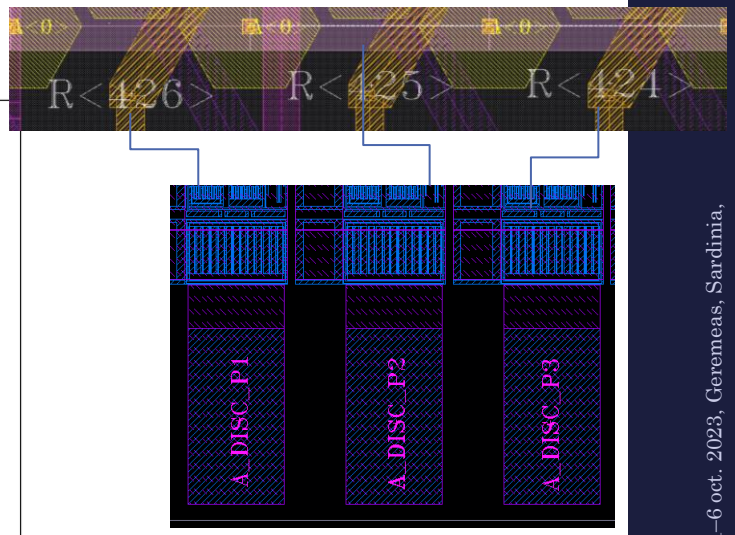
Unwanted effect observed : threshold spread in the current courant comparator reference, due to IR-drop in the matrice, **before** calibration. Threshold vary from 40µA to 120µA



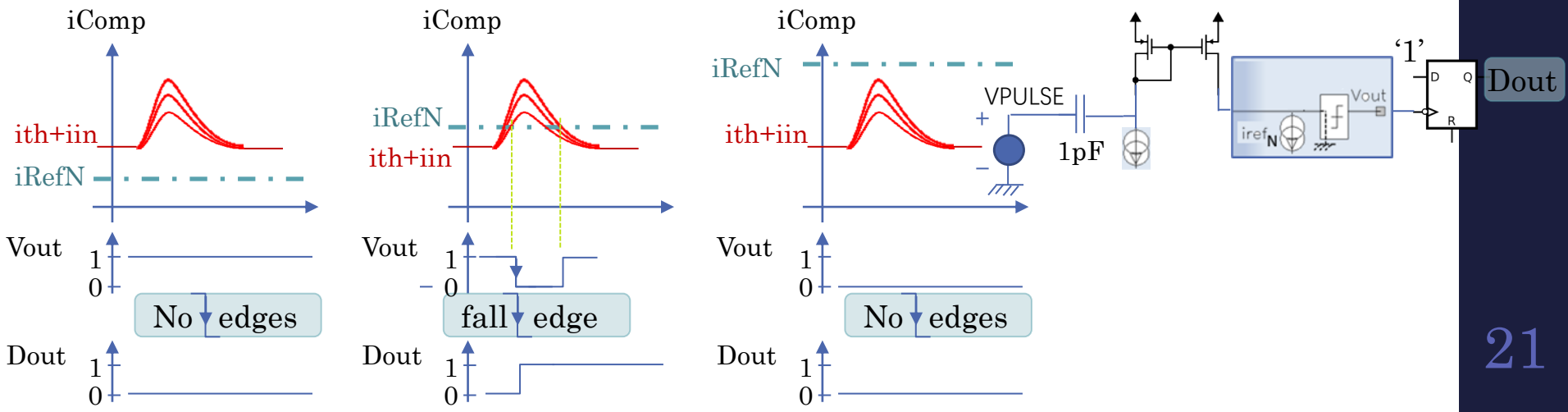
Also, 38 channel are noisy due to coupling between the memory Flush signal in Metal 2 with a few horizontal Metal 3 lines

2nd result :



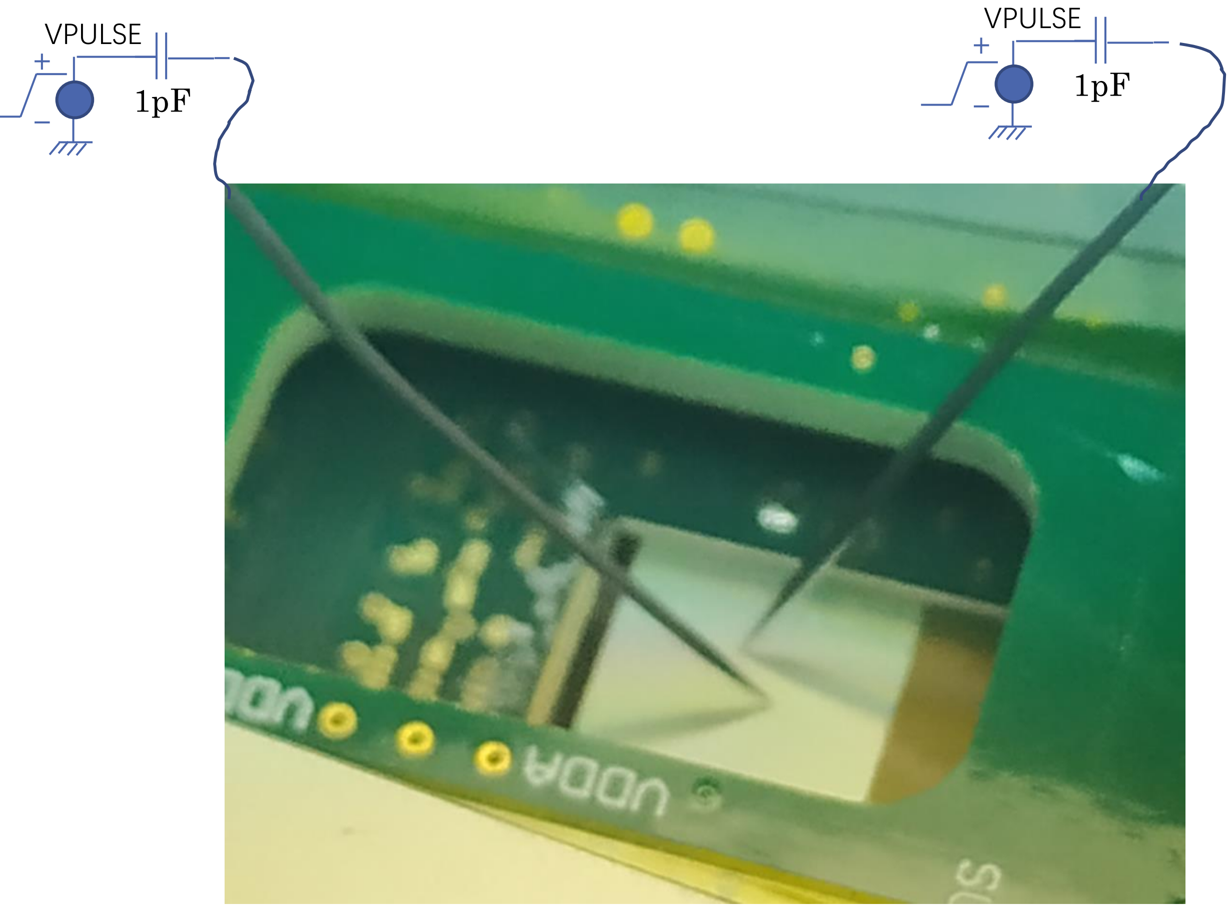


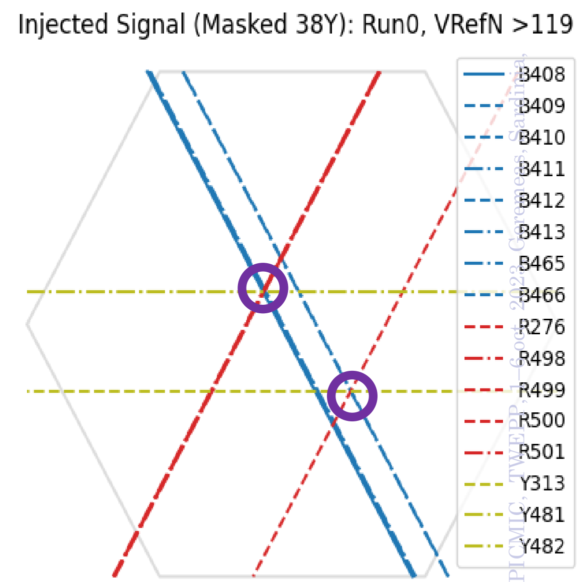
3 channels can be injected from outside through a PAD
Voltage injection through external 1pF, then scan of the threshold of the current comparator
Below the threshold current, the falling edge sensitive comparator does not trig.
The signal is recorded on falling edge, so above a certain value of iRefN added to the threshold current, the comparator does not trig anymore (_|‾|_ *'hat_chapeau_capello effect'*)



Main experimental Result :

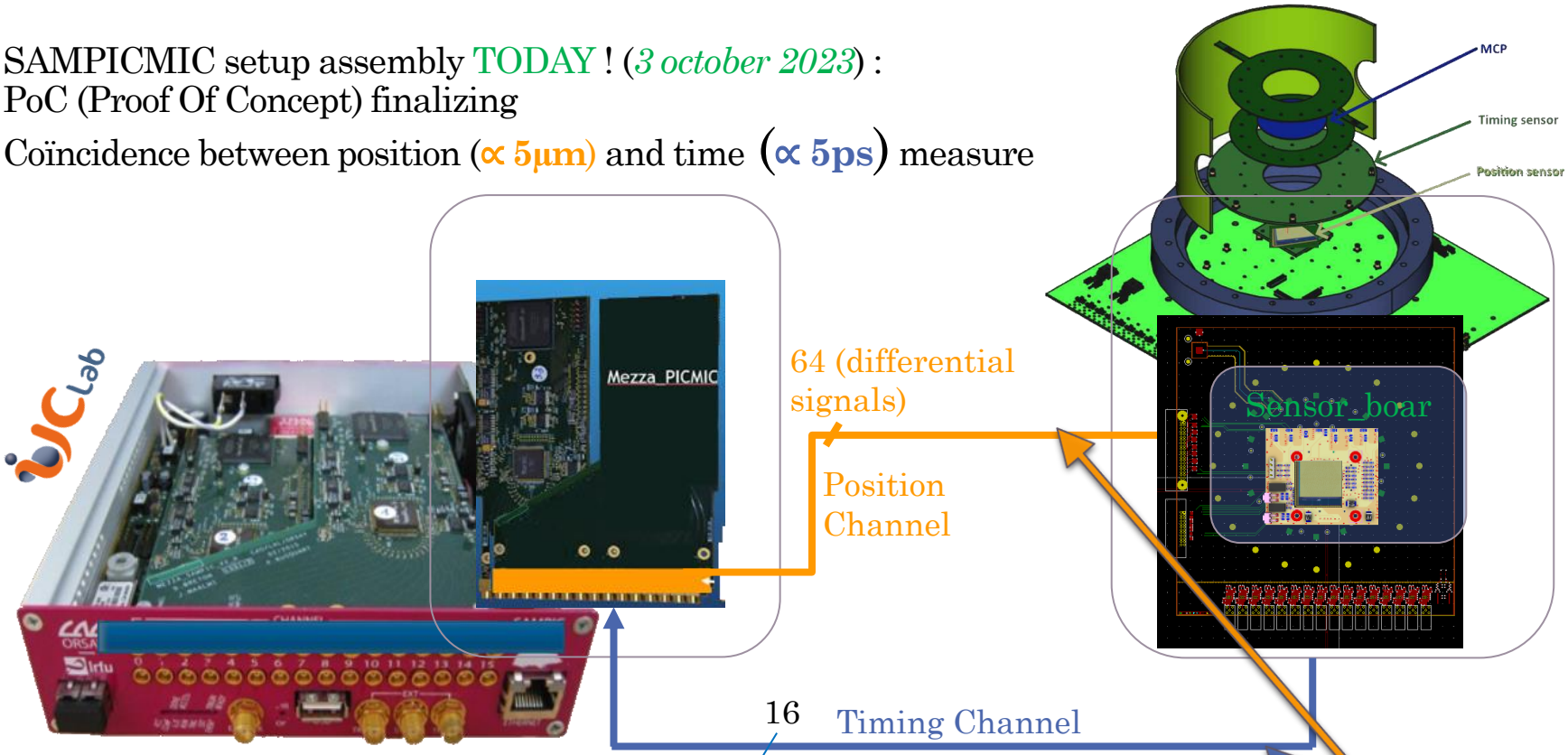
5µm resolution probes position detection of two simultaneous « Hits »





SAMPICMIC setup assembly TODAY ! (*3 october 2023*) :
PoC (Proof Of Concept) finalizing

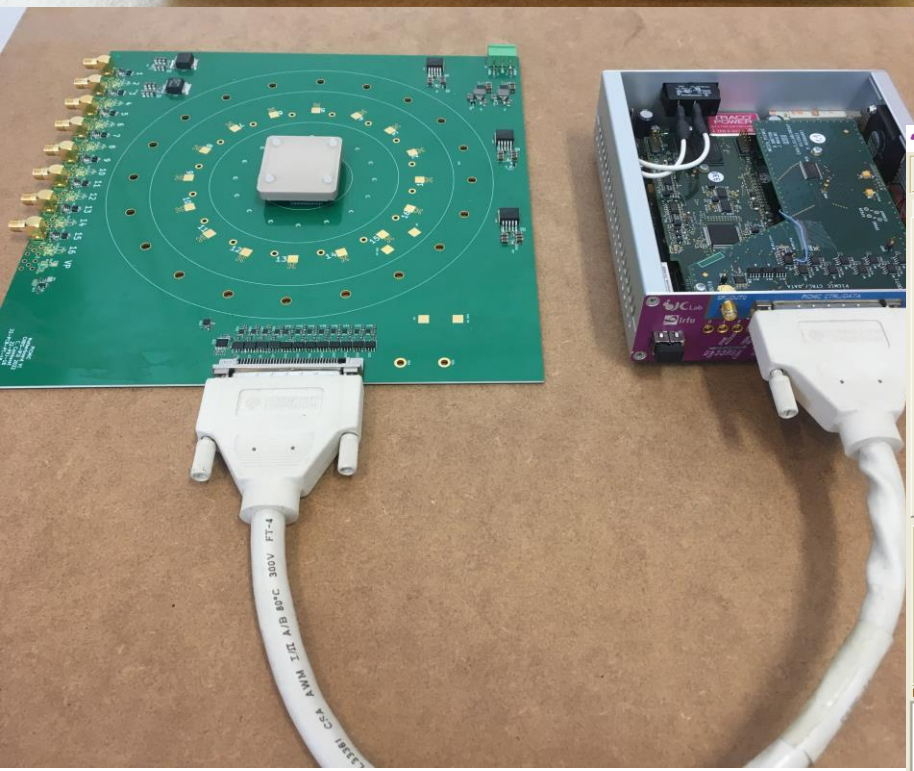
Coïncidence between position (∝ 5µm) and time (∝ 5ps) measure

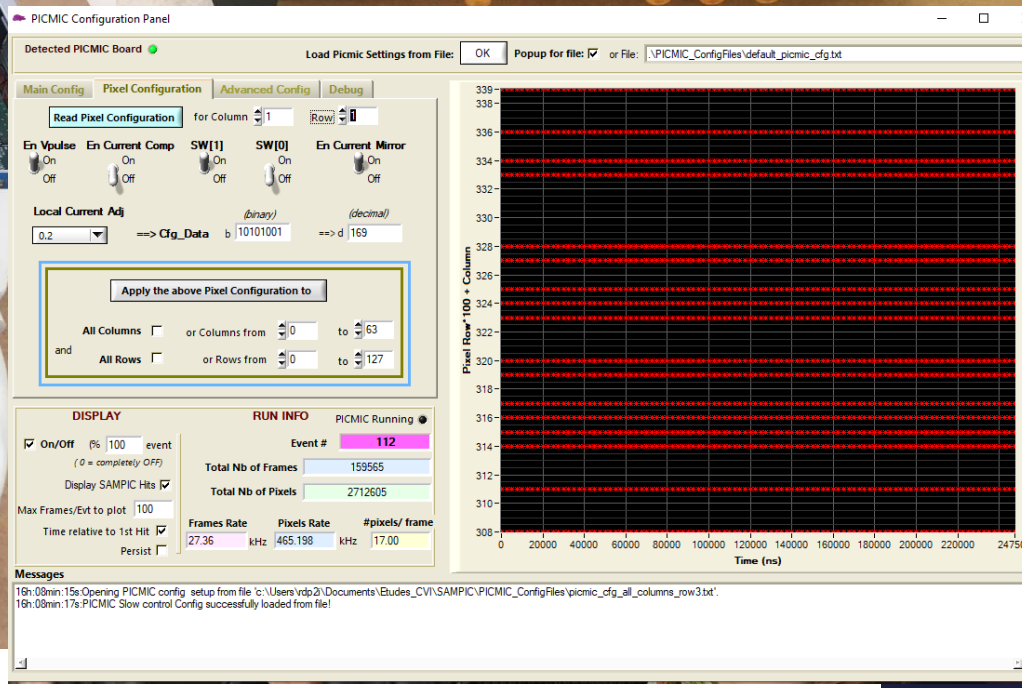


5µm measure retrieve also a rough time estimation: X,Y @ nb*400ns

16 time measure @ 3ps retrieve also a 1mm position estimation: t_1 - t_{16} @ x*mm, y*mm

SAMPICMIC HW + FW ready!





❖Summary

- ❖ FONCTIONNAL ASIC that fullfil the primary goal for the PoC
- ❖ First time such current mirror and current comparator used in our Asics.
- ❖ I2C is fonctionnal, testbench is fonctionnal
- ❖ (minor) default on one 8-bit 250µA global DAC (not described here)
- ❖ Threshold spread due to IR-drop
- ❖ Cross-talk identified on 38 channels
- ❖ layout out our own risk : 3 DRC Waiver (>max PAD coverage, > PAD opening, 60° path)

❖ Ongoing ASIC further improvement

❖New foundry in 180nm Q1 2024:

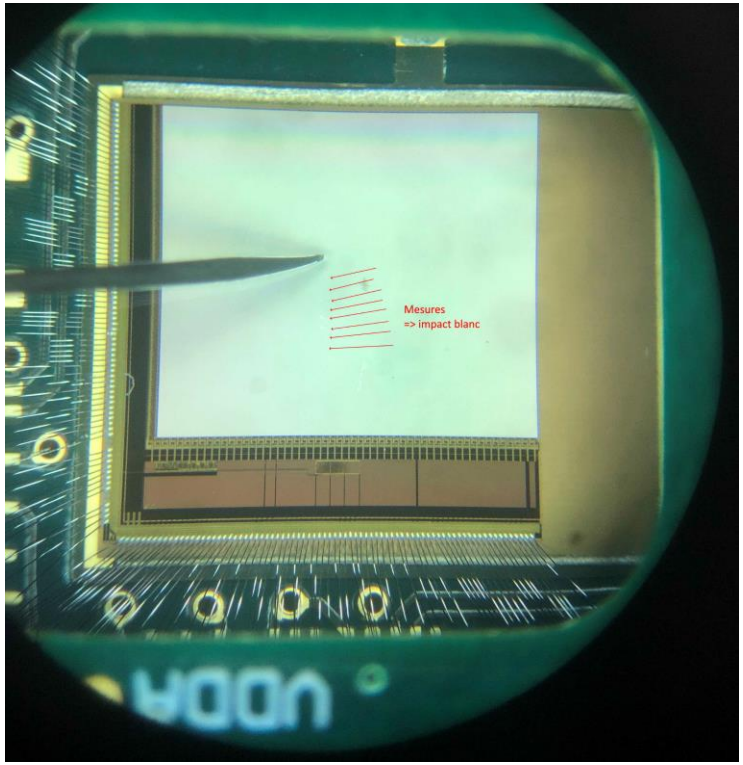
- ❖ Hit rate improvement from 2,5MHz to 10MHz or 100MHz expected
- ❖ Digital data timestamp for firwmare easyness
- ❖ Sensor size to increase to 1.48 cm x 1.28 cm
- ❖ ~~Hexagon pitch shrink from 5µm to 1.25µm~~

❖ Migrate in a smaller tech node like 65nm or less (with more metal layers)

❖detector level ongoing improvement feature

❖PoC
❖Idrogen (high rate readout)
❖…

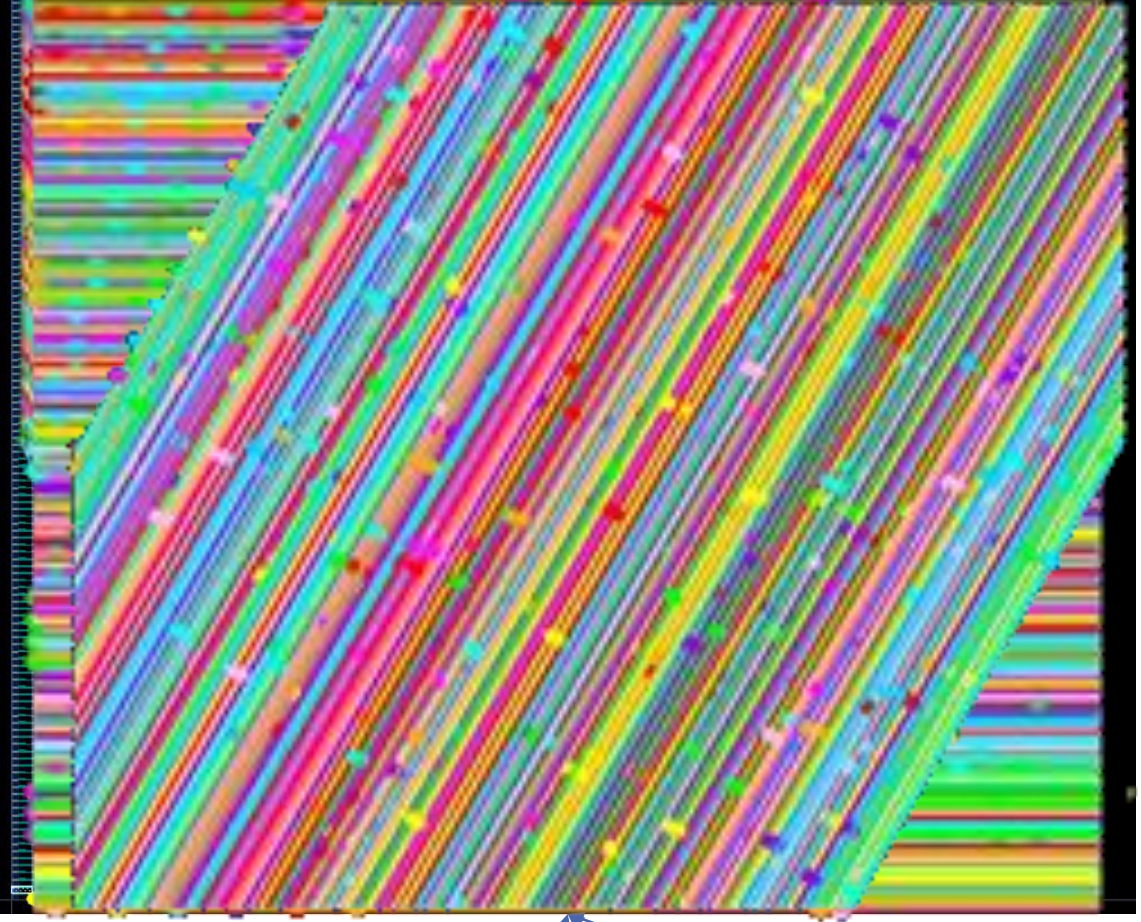
Thanks to the whole team involved : ASIC team, PCB team, DAQ/Firmware team, Scientist team, bounding team …



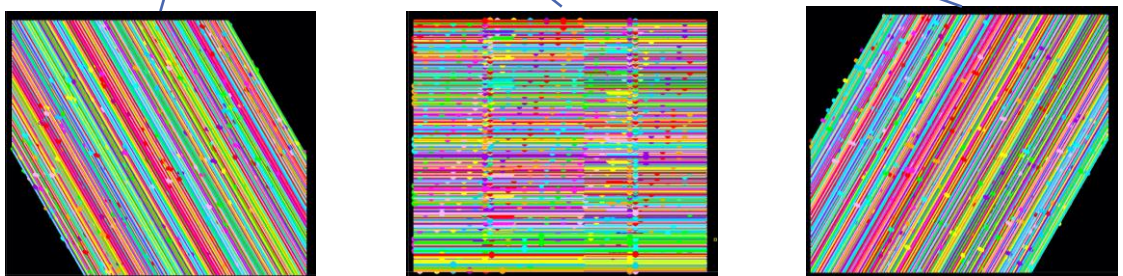
BACKUP

13-bit in parallel representing X,Y address of the touched readout-cell
X [0-53] on 6 bits, Y [0-127] on 7 bits : 0b**xxxxxx**yyyyyyy

grep "R<426" picmic_adress_table.csv 15;6;1926 ; 0x786 ; **0011110000110**; R<426>
grep "R<425" picmic_adress_table.csv 35;103;4583, 0x11E7; **1000111100111**;R<425>
grep "R<424" picmic_adress_table.csv **28**;**68**;3652, 0xE44 ; **0111001000100**; R<424>

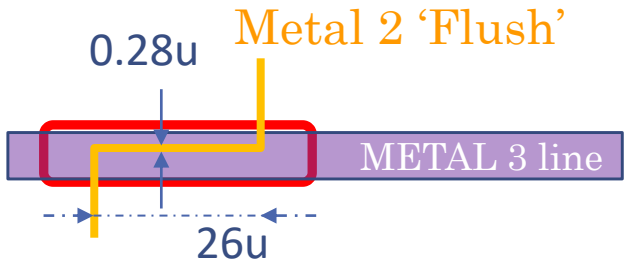


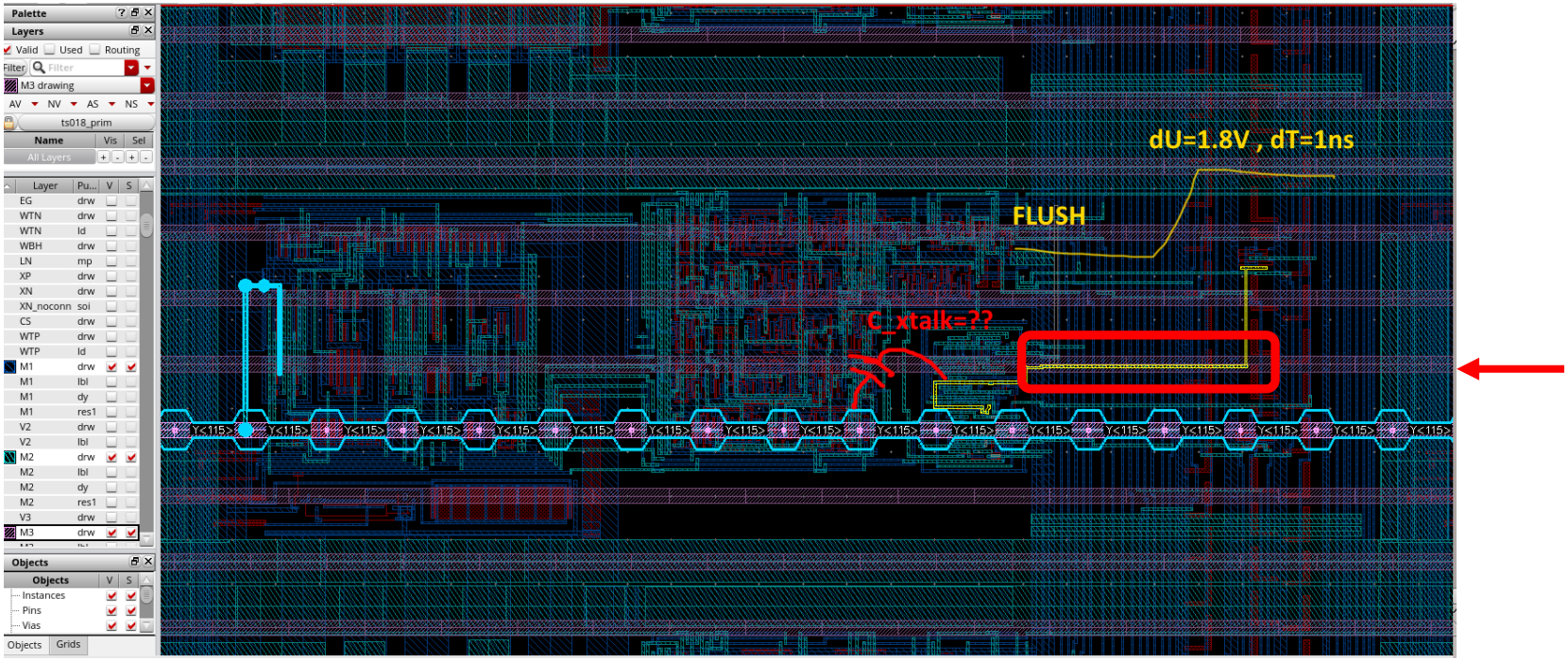
Interleaved superposed 852 lines in 3 direction, of the hexagonal pavement in a staggered way



Unwanted effect observed explained :

Dff Flush signal in M2 overlap M3 possibly every 6 cells depending on the digital placement





- Flushing one memory inject to another cell a signal.
- Parasitic extraction gives a value of **4fF**. The colormap default of 20uA permits to estimate the rising edge (dT) of the signal dT=C*dU/di = 4f * 1.8/20u = 360ps edge.
- When the FLUSH signal totally superposed to a Metal 3, le coupling is maximum

imad_rand_vias9.il - emacs@sbgue13.in2p3.fr

File Edit Options Buffers Tools Help

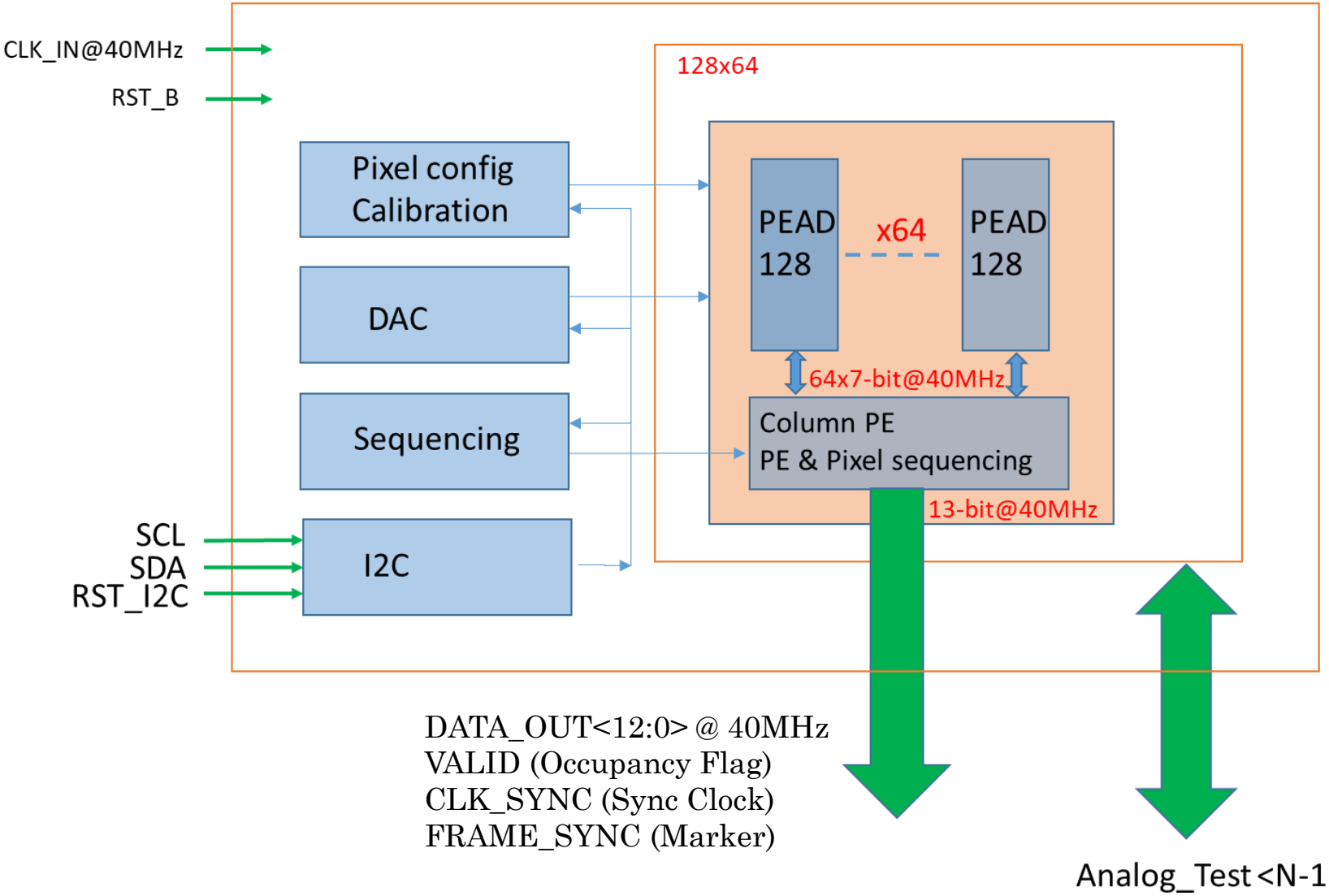
```
;printf("I_%d_%d %L\n" xgridnb ygridnb mapcar('plus X0:Y0 27/2*dx:64/2*dy))

case(color
("Yellow" (dbCreateInst cvi roy sprintf(nil "I_%d_%d" X0/dx+27 Y0/dy+64 ) X0:Y0  "R0") ;putCell(roy X0:Y0) ;
          X=X-9           ;label(cv "X:Y") label(cv "X0:Y0")
          ; check diamond intersection grid
          xgrid=floor(X*1.0/(sqrt(3)*pitch))*sqrt(3)*pitch;=> 3637.307
          ygrid=floor(Y*1.0/(1.5*pitch))*1.5*pitch;=> 60.0
          ygridnb=floor(Y*1.0/(1.5*pitch));*1.5*pitch;=> 8
          xifodd=if(oddp(ygridnb) sqrt(3)*pitch/2 0);<= should i use that too for Bleu ?
          Xmod=xgrid+Hxadj+xifodd;+Ryadj/sqrt(3)
          Ymod=ygrid+Hyadj;<= abs or not abs TOCHECK
          createPinLPPcv(pinName xy2bBox(X:Y WM2/2.0) "M3" nil cv t)
          if(Y>=Y0+dy-pwrWidth then nytop+=1
          sxc=Xmod+nbsxa*sqrt(3)*pitch syc=Y
          sxa=(sxc-(X0+sx)) sya=sxa*sqrt(3)
          sxb=X0+sx syb=Y-sya
          sxd=Xmod-1.0/4*sqrt(3)*pitch
          syd=Y-sqrt(3)*(sxc-Xmod)-1.5/2*pitch sxe=X0+sx sye=syd
;         label(cv "sxe:sye") label(cv "sxb:syb") label(cv "sxd:syd") label(cv "sxc:syc") label(cv "Xmod:Ymod
          stackVias(viasM4_M3 sxc:syc cv);<= remonte en M4
;         p=dbCreatePath(cv "M4" list(sxc:syc sxd:syd ) Wfix );<= gives DRC error => triangle at end
          create60poly(cv "M4" list(sxc:syc sxd:syd) sqrt(3)/4*Wfix)
;         create60poly(cv "M4" list(sxc:syc sxd:syd ) Wfix)
;         pathTriangle(cv p)
          stackVias(viasM4_M3 sxd:syd cv);<= redescend en M3
;         dbCreateRect(cv "M4" xy2bBox(sxc:syc WM4/2));<= check /2 later with DRC
          dbCreatePath(cv "M3" list(fixx(sxd):syd fixx(sxe):sye) WM3b  pS)
          stackVias(viasM3_M2 sxe:sye cv);<= descend en M2
          dbCreatePath(cv "M2" list(sxe:sye X0+sx:Y0+sy X0+vfe_sx:Y0+sy) WM2 pS)
          else ;=> creer le net vertical 'normal' quand pas de bypass
          stackVias(viasM3_M2 X:Y cv)
          dbCreateRect(cv "M2" xy2bBox(X:Y WM2));<= check /2 later with DRC
          dbCreatePath(cv "M2" list(X0+sx:Y X0+sx:Y0+sy X0+vfe_sx:Y0+sy) WM2 pS)
          ); fin cas Y ~~/  en ZigZag
          dbCreatePath(cv "M2" list(X:Y0+sy X0+sx:Y0+sy) WM2 )
          createPinLPPcv(pinName xy2bBox(X0+sx:Y0+sy WM2/2.0) "M2" nil cv t);<= Front-End Input PIN
)
("Red" (dbCreateInst cvi roy sprintf(nil "I_%d_%d" X0/dx+27 Y0/dy+64 ) X0:Y0  "R0") ;putCell(ror X0:Y0)
;       dbCreateRect(cv "GC" list(X0:Y0 X0+dx:Y0+dy))  ;<= GC only for coloring visual
        Y=Y-9 Y=Y+3.67
        xgrid=floor(X*1.0/(sqrt(3)*pitch))*sqrt(3)*pitch
        ygrid=floor((Y-2.5)*1.0/(1.5*pitch))*1.5*pitch+2.5;<= MERCI HERVE :)
        Xmod=X+sqrt(3)/4*pitch-(Y-ygrid)/sqrt(3)
        Ymod=ygrid+1.5*pitch/2
        while(Ymod-Y0 >= 1.5*pitch-minM2w/2 Xmod=Xmod-sqrt(3)/2*pitch Ymod=Ymod-1.5*pitch)
        when(Xmod-X0<= 0.0 Xmod=Xmod+sqrt(3)/2*pitch Ymod=Ymod+1.5*pitch)
        when(Xmod-X0<= 0.0 printf("** WARNING ** %L %s\n" Xmod:X0 pinName))
;        label(cv "X:Y") label(cv "Xmod:Ymod") label(cv "X0:Y0");       label(cv "xgrid:ygrid")
        Xmod=fixx(Xmod) Ymod=fixx(Ymod) ;<= fix grid issues
```

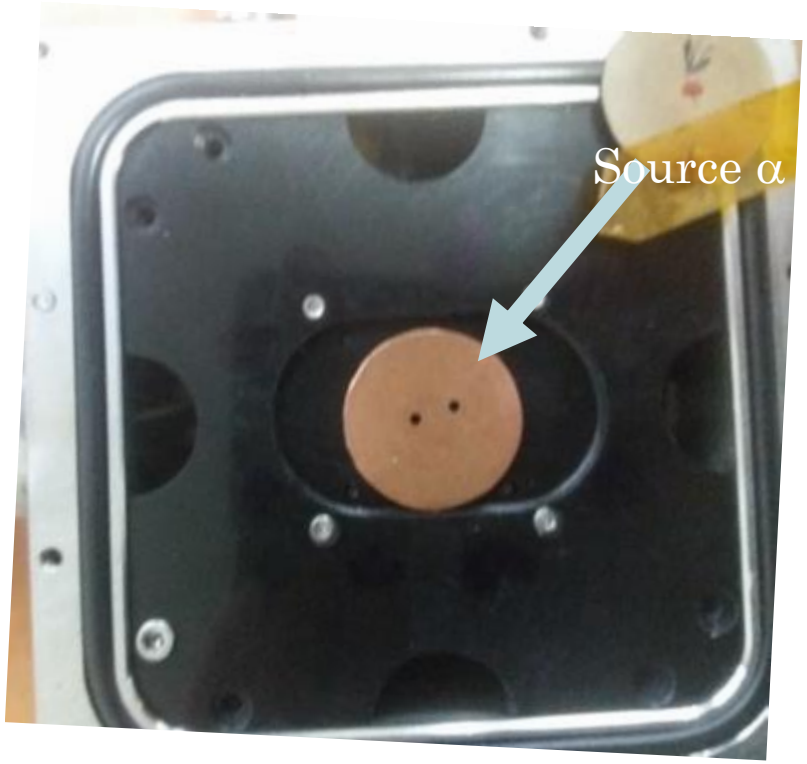
0:--- imad_rand_vias9.il 51% L121 (SKILL)

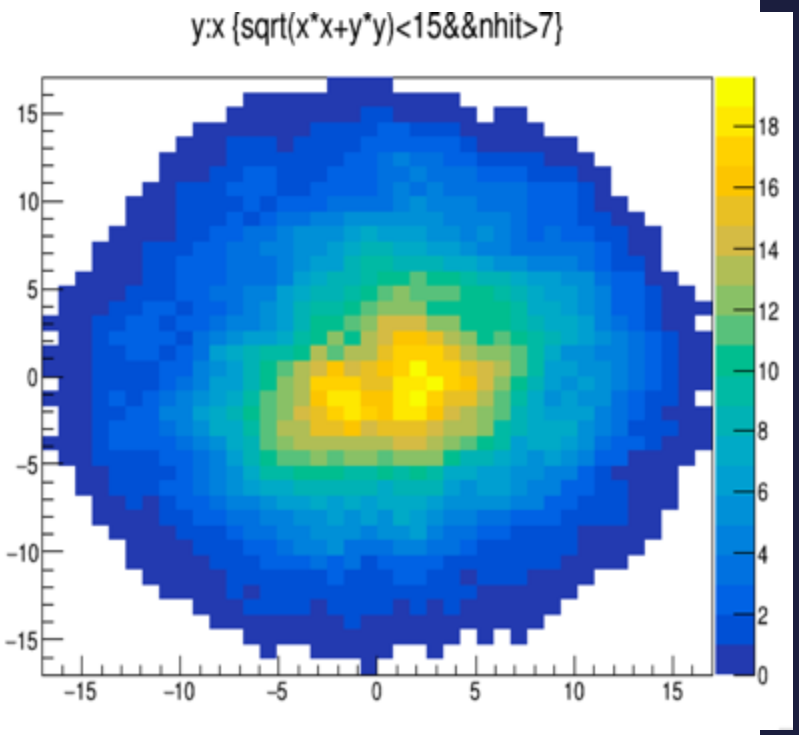
- Every TOP to M3 layer inside the active matrice had been written in skill code.
- Plus the additional M2 line essential to the interconnection algorithm

Digital Picmic Architecture



Temporal Measure





Preliminary routing example of the Front-end + Digital part