Inner Tracker Excercise

Antonio Cassese, Mauro Dinardo

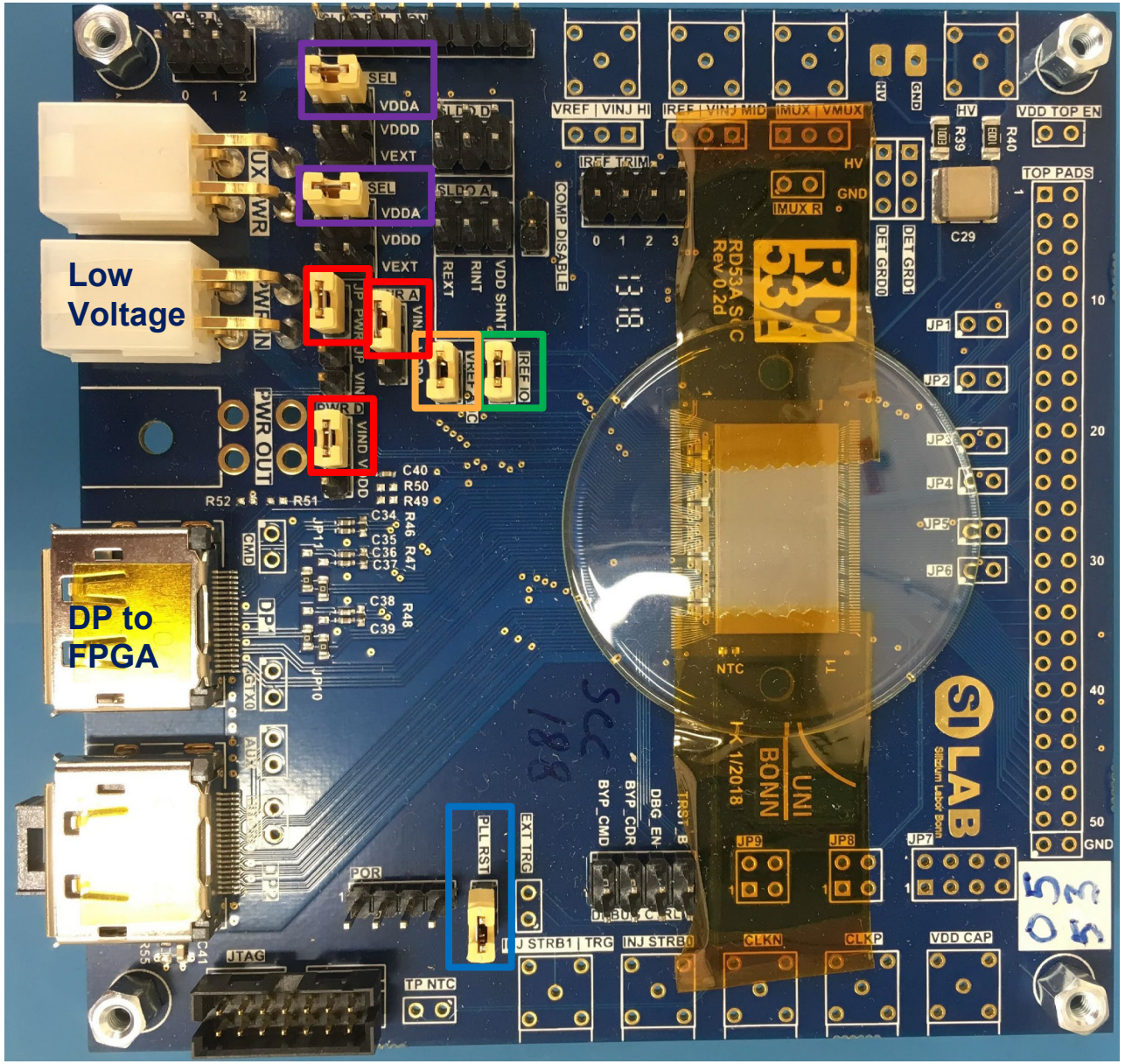
Part of this work is based on Natalia's instructions prepared for 2019 Tracker DAQ school

Documentation

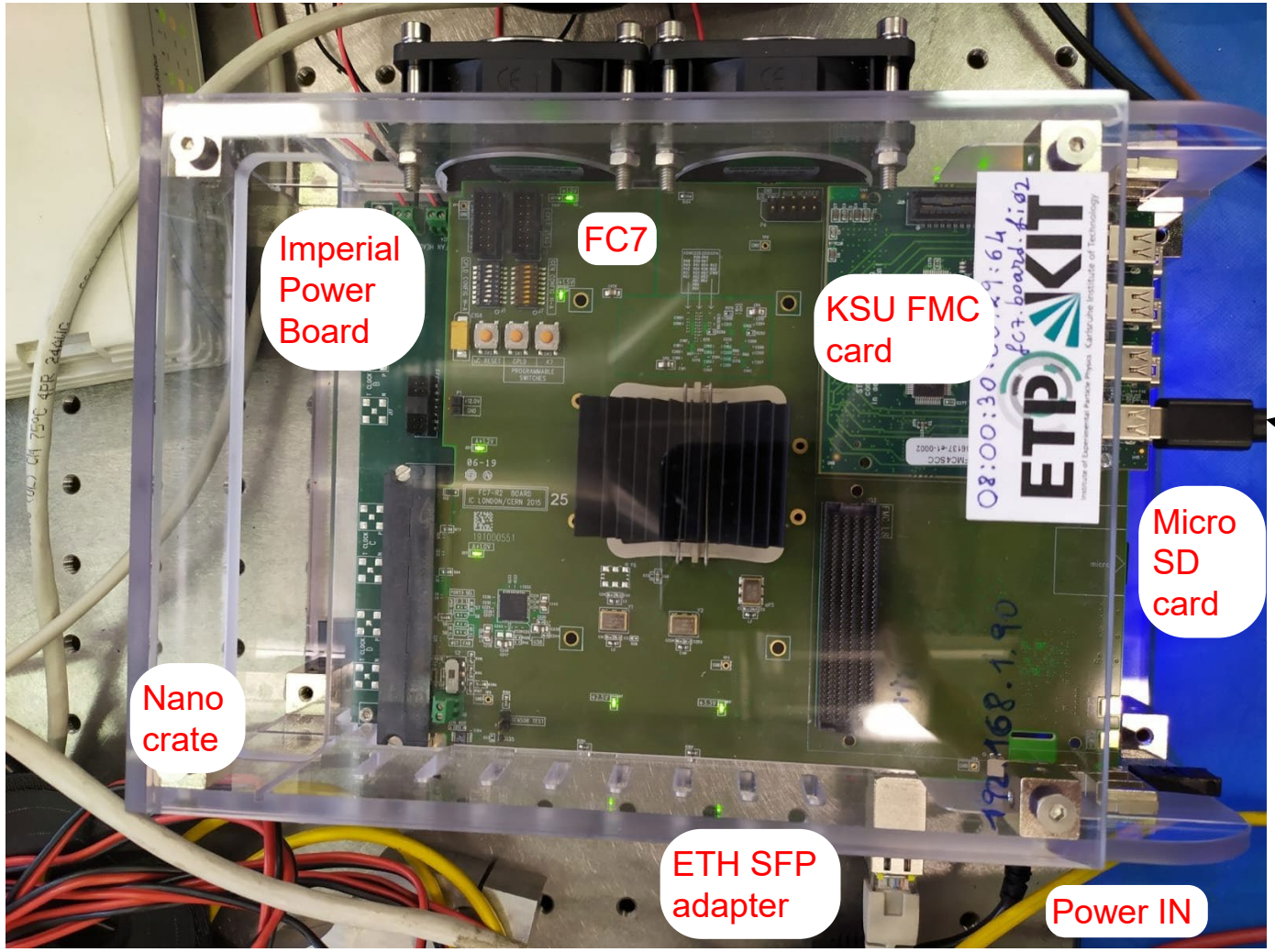
- RD53 manual: https://cds.cern.ch/record/2287593/
- RD53 testing twiki: https://twiki.cern.ch/twiki/bin/viewauth/RD53/RD53ATesting
- Middleware repository: https://gitlab.cern.ch/cms_tk_ph2/Ph2_ACF/-/tree/daqSchool2021
- Firmware download: https://gitlab.cern.ch/cmstkph2-IT/d19c-firmware/-/releases

SCC jumpers

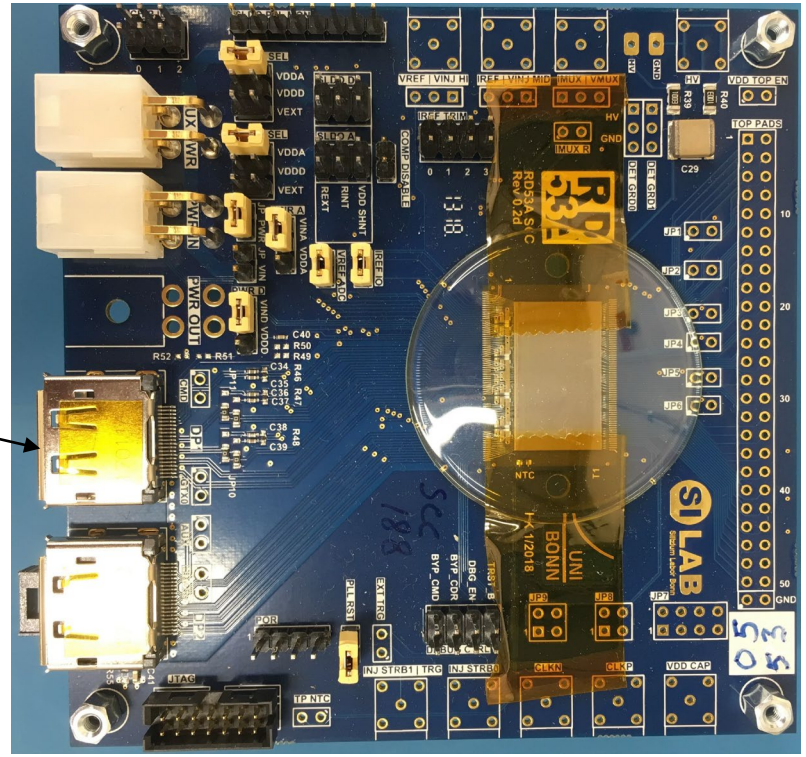
- Necessary jumpers on SCC
  - ✓ IREF IO
  - ✓ VREF ADC
  - ✓ VDD CML SEL ⟶ VDDA
  - ✓ VDD PLL SEL ⟶ VDDA
  - ✓ PLL RST
  - ✓ JP PWR
- LDO power mode:
  - ✓ PWR A to VINA (to the left)
  - ✓ PWR D to VIND (to the left)



SCC to FC7 connection



DP to MiniDP Cable



Middleware preparation and configuration

- In Ph2_ACF create a working directory (create as much working folder as you need, for example one for each sensor you are testing or whatever)

```
>mkdir workingFolderSensorName
```

Folder name example

- Copy from settings folder in Ph2_ACF the configuration file CMSIT.xml and the ASCII file in settings/RD53Files/CMSIT_RD53.txt

```
>cd workingFolderSensorName
>cp ../settings/CMSIT.xml .
>cp ../settings/RD53Files/CMSIT_RD53.txt .
>chmod -x CMSIT.*
```

This is not mandatory, but I prefer not to have execution permises for configuration file only

- You need to properly configure the CMSIT.xml wrt the scans/calibrations you need
- Remember to properly set the IP address has explained in FC7 setup
- After firmware load, please reset the board:

```
>CMSITminiDAQ -f CMSIT.xml -r
```

- As an example see the monitoring in the next slide

Monitoring

- For the monitoring of some parameters such as V_{DDD}, V_{DDA}, V_{REF} (see next slides), you need to switch on the monitoring by putting to one the RD53 monitoring enable flag in CMSIT.xml, as shown in figure

```
227   <!-- === Monitoring parameters ===
228        MonitoringSleepTime: sleep for monitoring thread in milliseconds
229 -->
230   <MonitoringSettings>
231     <Monitoring type="RD53" enable="1">
232       <MonitoringSleepTime>          1000 </MonitoringSleepTime>
233       <Enable name="AllVariables">      1 </Enable>
234     </Monitoring>
235   </MonitoringSettings>
```

Enables monitoring

Monitoring repetition rate

- Then you need to run CMSITminiDAQ, for example:

```
>CMSITminiDAQ -f CMSIT.xml -c physics
```

SCC configuration (1)

The trimming of the voltages on the board is useful, but not strictly mandatory for the school excercise

- Power supply (PS):
  - ✓ Suggested PS voltage 1.8V (? Better on SCC and control the voltage drop on the cables?)
  - ✓ $V_{in}D$ and $V_{in}A$ are shorted on SCC (JP PWR jumper):
    - Single PS channel for both analog and digital power (you need enough current, i.e. 2 A or more)
    - Two channels, one for analog and one for digital: you need to switch them on simultaneously

You should have a "both on" button or similar

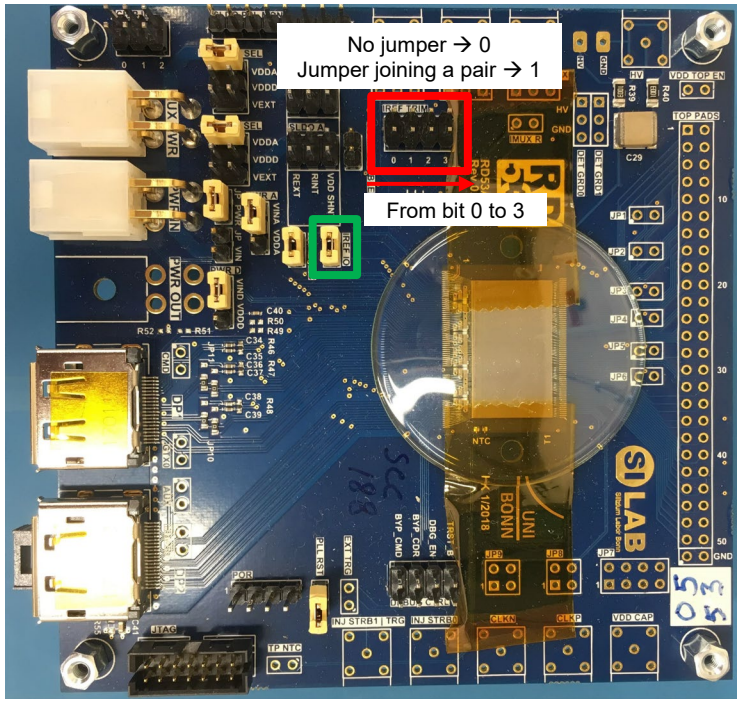
- V_{DDA}, V_{DDD} = **1.2V**:
  - ✓ In LDO mode, once the chip has the PLL locked, it is possible to set V_{DD}s via voltage trim bits
  - ✓ You need to properly set the registers named SLDO_ANALOG_TRIM & SLDO_DIGITAL_TRIM in RD53A manual
  - ✓ In Ph2_ACF, the registers are called VOLTAGE_TRIM_ANA and VOLTAGE_TRIM_DIG

5 bit registers

  - ✓ Measure the V_{DD} with a multimeter between GND and PWR A for V_{DDA} or PWR D for V_{DDD}
  - ✓ Adjust bits until both are at 1.2V

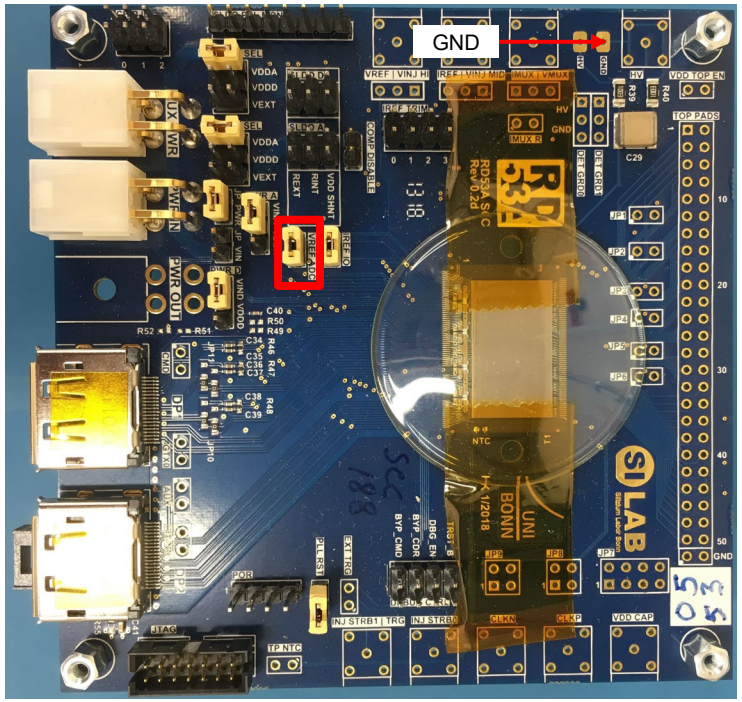
In order to write registers you need to run CMSITminiDAQ

SCC configuration (2)



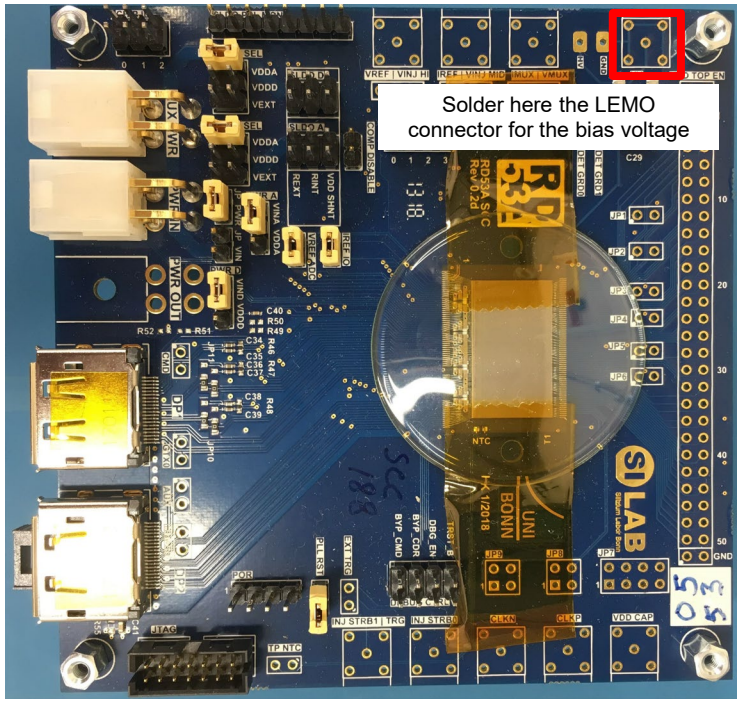
- I_{REF} = **4.0 µA**:
  - ✓ Using a multimeter, measure the current on the SCC card between the two pins IREF IO (for this measurement you need to remove the jumper and measure the current between the two pins)
  - ✓ Adjust I_{REF} properly adding or removing jumpers on the 4 IREF TRIM its on the SCC
  - ✓ The higher the bit value, the higher the I_{REF} current
  - ✓ **Put back the jumper IREF IO!**

SCC configuration (3)



- V_{REF} = **0.9 V**:
  - ✓ Using a multimeter, measure the voltage on the SCC card between the VREF ADC jumper and GND (you need a jumper on VREF ADC with a probing hole in the middle, allowing you to use it as a test point even with the jumper itself)
  - ✓ Adjust V_{REF} properly using MONITOR_CONFIG_BG register in the xml configuration file

SCC configuration (4) - Bias voltage



- For this exercise it is not mandatory to switch on the high voltage, but
- IF you want to bias your sensor, do it at your own risk, and at lest be sure that:

  - ✓ You never bias the sensor with LV off (switch LV on before biasing, switch bias off before LV off)
  - ✓ Limit the maximum current of your bias voltage supplier such that you can't break your sensors
  - ✓ Bias your voltage with the right polarity and "gently" (no fast ramps and no bumps)
  - ✓ Depending on your sensor type, choose a proper value for the bias voltage
  - ✓ Take care of all the possible sensor breaking circumstances

Latency coarse

- Copy CMSIT.xml into CMSIT_Latency.xml (for example)

```
>cp CMSIT.xml CMSIT_Latency.xml
```

- Change this file, taking care of:

```
183      <Setting name="nTRIGxEvent">          10 </Setting>
193      <Setting name="LatencyStart">          0 </Setting>
194      <Setting name="LatencyStop">         511 </Setting>
211      <Setting name="DoFast">                1 </Setting>
```

Latency will be performed in step of 10 units of latency, givin you a coarse result

Full range

Scan performed in a subset of pixels

- Run a latency scan:

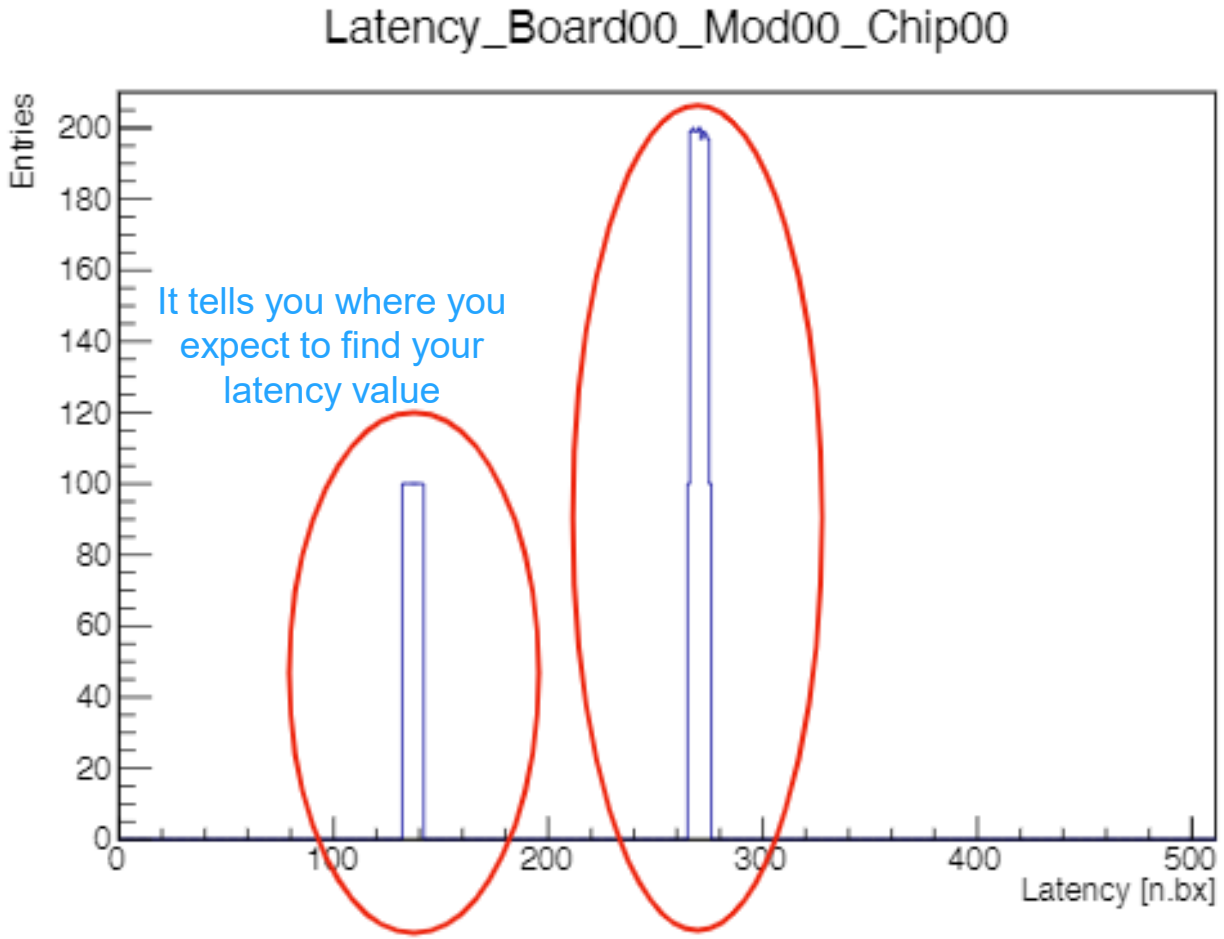
```
>CMSITminiDAQ -f CMSIT_Latency.xml -c latency
```

- Look at the generated root file:

```
>root -l FileName.root
```

WARNING! Remember to put DoFast back to zero after latency

Latency coarse



Actual peak to be used to obtain the correct latency

Spurious peak due to the arming process of the injection circuit

Latency fine

- Change CMSIT_Latency.xml file, according to the previous results:

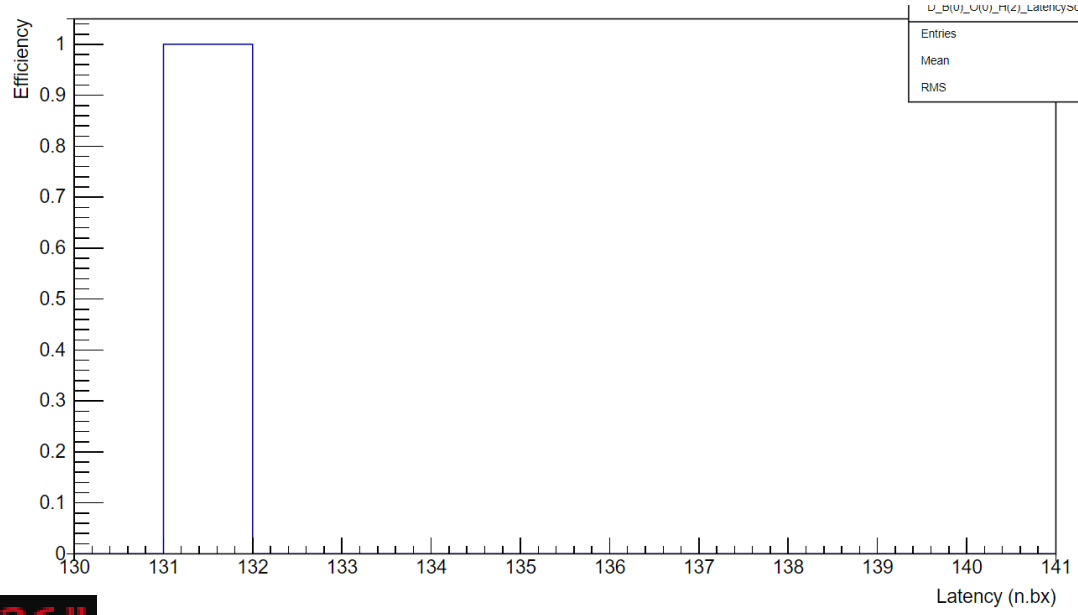
```
183     <Setting name="nTRIGxEvent">          1 </Setting>
```

```
193     <Setting name="LatencyStart">       110 </Setting>
194     <Setting name="LatencyStop">        150 </Setting>
```

- Run a latency scan:

```
>CMSITminiDAQ -f CMSIT_Latency.xml -c latency
```

- Put the found value +nTRIGxEVENT/2 (it looks from that value on, where on is determined nTRIGxEVENT) in your CMSIT.xml file and all of your config files from now on




```
59              LATENCY_CONFIG          =   "136"
```

Pixel alive

- Copy CMSIT.xml in CMSIT_Scurve.xml
- Change the file, taking care of:

```
176      <Setting name="nEvents">                100 </Setting>
177      <Setting name="nEvtsBurst">             100 </Setting>
```

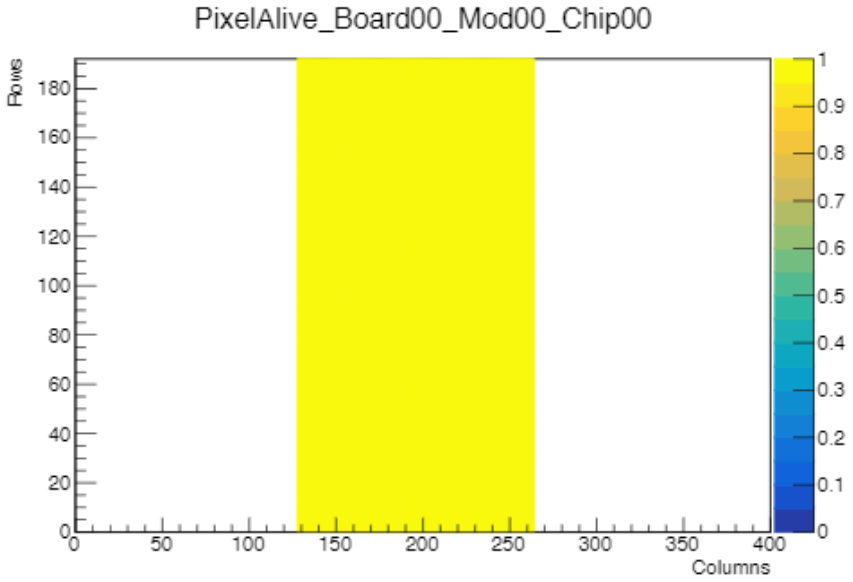
```
184      <Setting name="INJtype">                1 </Setting>
```

1 Analog injection
2 Digital injection

```
>CMSITminiDAQ -f CMSIT_Scurve.xml -c pixelalive
```

```
||I| Readout chip error report for [board/opticalGroup/hybrid/chip = 0/0/0/0]
||I| LOCKLOSS_CNT        = 0
||I| BITFLIP_WNG_CNT     = 0
||I| BITFLIP_ERR_CNT     = 0
||I| CMDERR_CNT          = 7508
||I| SKIPPED_TRIGGER_CNT = 0
||I| BCID_CNT            = 47740
||I| TRIG_CNT            = 1000
||I| Average occupancy for [board/opticalGroup/hybrid/chip = 0/0/0/0] is 0.999590
||I|    --> Number of potentially masked pixels in this iteration: 0
||I|    --> Total number of potentially masked pixels: 0
||I|    --> PixelAlive saved the configuration file for [board/opticalGroup/hybrid/chip = 0/0/0/0
]
||I|    --> PixelAlive saving histograms...
||I| Closing result file
||I| >>> Destroying interfaces <<<
||I| >>> Interfaces  destroyed <<<
||I| @@@ End of CMSIT miniDAQ @@@
```

No pixels masked



Noise

- Copy CMSIT.xml in CMSIT_Noise.xml
- Change the file, taking care of:

```
180      <Setting name="nEvents">      10000000 </Setting>
181      <Setting name="nEvtsBurst">      10000 </Setting>
```

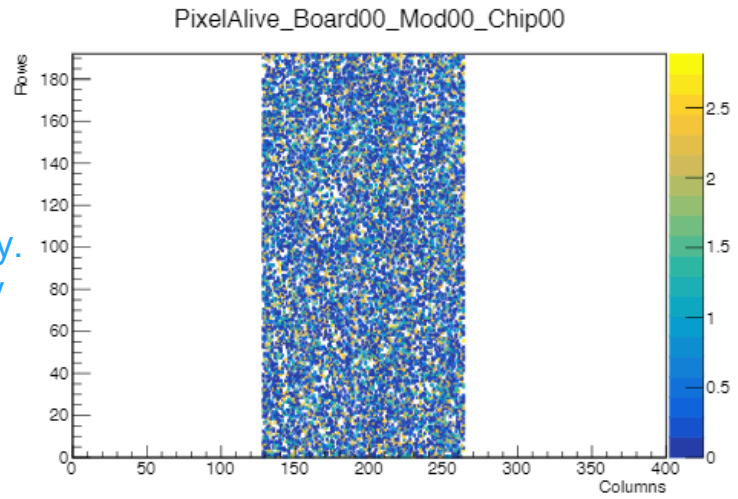
```
184      <Setting name="INJtype">              0 </Setting>
```

```
>CMSITminiDAQ -f CMSIT_Noise.xml -c noise
```

```
||I| Readout chip error report for [board/opticalGroup/hybrid/chip = 0/0/0/0]
||I| LOCKLOSS_CNT        = 0
||I| BITFLIP_WNG_CNT     = 0
||I| BITFLIP_ERR_CNT     = 0
||I| CMDERR_CNT          = 30147
||I| SKIPPED_TRIGGER_CNT = 0
||I| BCID_CNT            = 60172
||I| TRIG_CNT            = 10000
||I| Average occupancy for [board/opticalGroup/hybrid/chip = 0/0/0/0] is 0.000000
||I|    --> Number of potentially masked pixels in this iteration: 0
||I|    --> Total number of potentially masked pixels: 0
||I|    --> PixelAlive saved the configuration file for [board/opticalGroup/hybrid/chip = 0/0/0/0
]
||I|    --> PixelAlive saving histograms...
||I| Closing result file
||I| >>> Destroying interfaces <<<
||I| >>> Interfaces  destroyed <<<
||I| @@@ End of CMSIT miniDAQ @@@
```

No pixels masked

PixelAlive plot should be empty. This not empty plot is just for demonstration



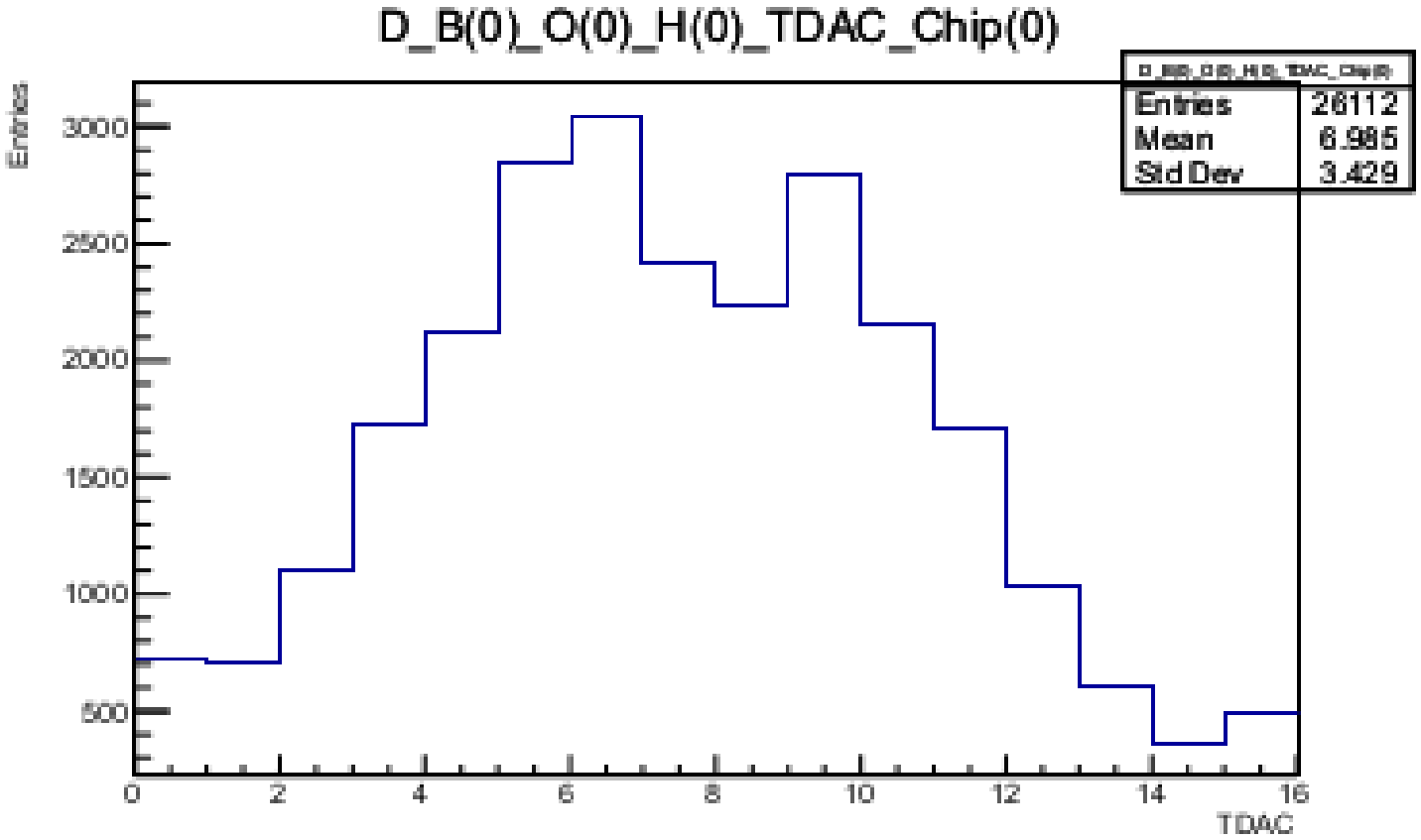
A particularly low threshold was used just for demonstration

Threshold equalization

- Search pixel by pixel TDAC values to equalize their thresholds
- The configuration file is the same of the pixel alive

```
>CMSITminiDAQ -f CMSIT_Scurve.xml -c threqu
```

TDAC distribution should look like this.
Be aware of first and last bin not to be too much populated



Threshold adjustment

- Searches for the global threshold to a target value in electrons
- The configuration file is the same of the pixel alive
- You just need to set your target threshold in electrons

```
207      <Setting name="TargetThr">          2000 </Setting>
```

```
>CMSITminiDAQ -f CMSIT_Scurve.xml -c thradj
```

- The value(s) reported has to be used to properly set the threshold in the xml files (all of them!)

```
32                    Vthreshold_LIN          =  "351"
||I| Global threshold for [board/opticalGroup/hybrid/chip = 0/0/0/0] is 351
||I|    --> ThrAdjustment saved the configuration file for [board/opticalGroup/hybrid/chip = 0/0/0/0]
||I|    --> ThrAdjustment saving histograms...
||I| Closing result file
||I| >>> Destroying interfaces <<<
||I| >>> Interfaces  destroyed <<<
||I| @@@ End of CMSIT miniDAQ @@@
```

Scurve

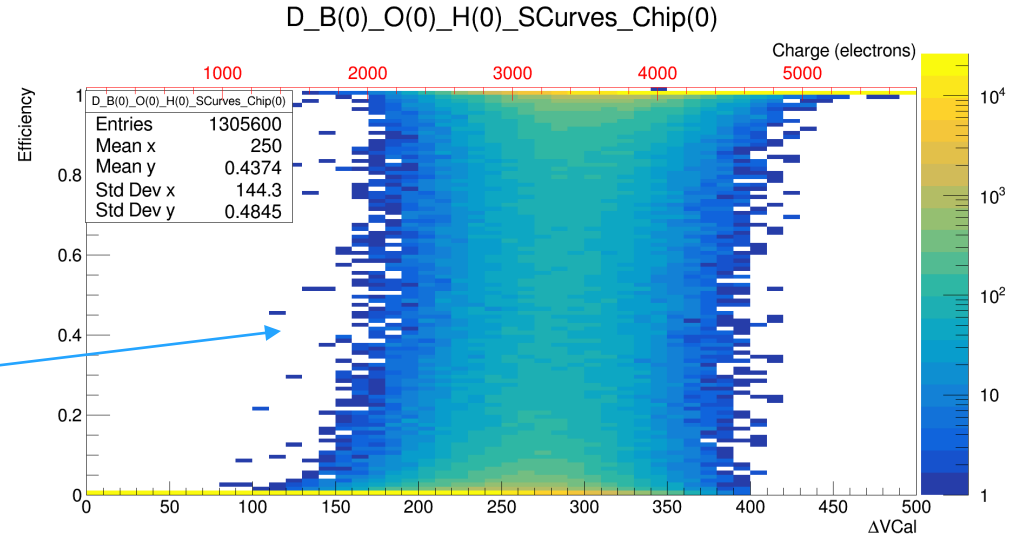
- The configuration file is the same of the pixel alive (well actually we called it scurve…)
- You just need to set the range and the number of bins you want

```
196     <Setting name="VCalHstart">      100 </Setting>
197     <Setting name="VCalHstop">       600 </Setting>
198     <Setting name="VCalHnsteps">      50 </Setting>
199     <Setting name="VCalMED">         100 </Setting>
```

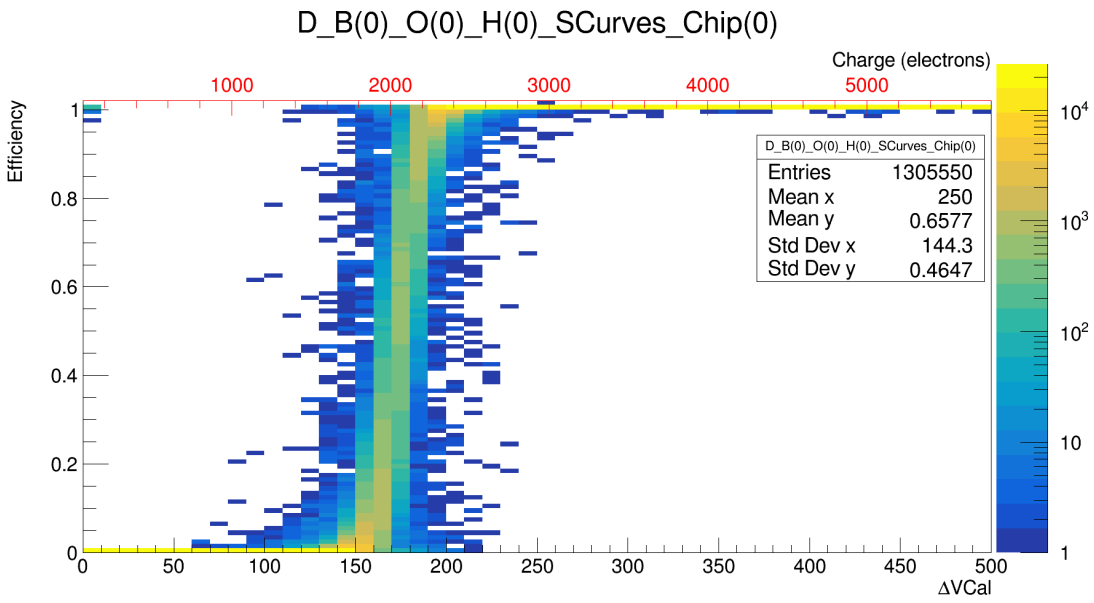
```
>CMSITminiDAQ -f CMSIT_Scurve.xml -c scurve
```

- CValHstart – VCalMED →First bin
- CValHstop – VCalMED →Last bin
- VCalHnseps →Number of bins

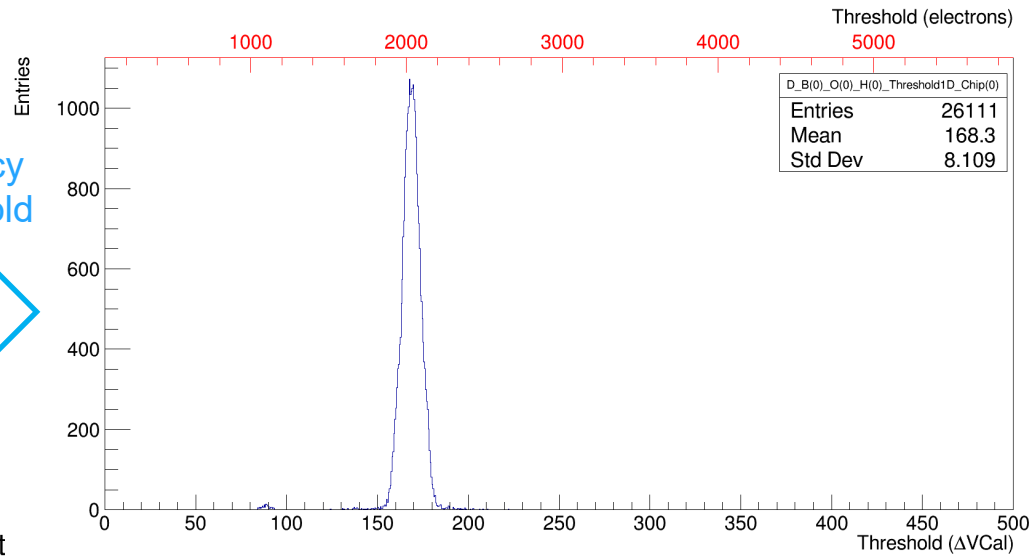
High thresholds and not equalized

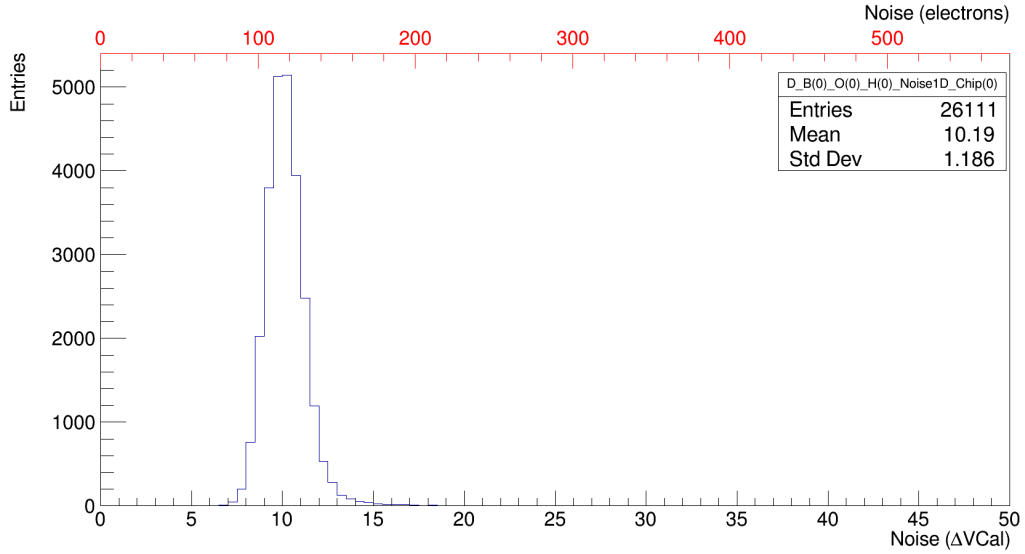


Scurve after 2000 electrons tuning



ΔVcal for 50% efficiency corresponds to Threshold value, pixel by pixel





Common suggestions

- Run the board reset with "CMSITminiDAQ -f CMSIT.xml -r" **every time** you load the firmware to the fpga with "fpgaconfig -c CMSIT.xml -i your-fw-name"
  - ✓ Without the reset it won't work.
  - ✓ Every time the crate/FC7 is power cycled you need to load the firmware and afterwards do the reset.
- In case you have communication problems, you can try using a different DisplayPort cable
  - ✓ For all the normal operation that are covered in school only one cable is needed: The one that goes from DP1 to DP1. The other connection are not needed
- If you still have problems, you can try with the 640 Mbps firmware version.

WARNING!!!
Changing the power type and voltages can be potentially dangerous.
Please do it only if you are expert enough to understand what you are doing!

- If you still have problems, you can try changing the powering:
  - ✓ Increase the PS voltage to 2.0V. This is the **maximum allowed for safe operation**.
  - ✓ If it still doesn't work, you can change the powering mode to direct powering:
    - Turn off the power supply and adjust the jumper settings: change the jumpers on the two **PWR_A and PWR_D headers to VDDA and VDDD** respectively
    - Following the exercise you should have them now set to VINA and VIND
    - In this mode you have to be careful as the **maximum allowed voltage is 1.32V**! It is suggested to power it with **1.25V**. A bit higher than 1.2 V to account for some voltage drop on the cable.
    - In this direct powering you cannot do the trimming with VOLTAGE_TRIM_ANA and VOLTAGE_TRIM _DIG registers. Just skip this part.
- If you still have problems and you have more chips available, try to use a different one