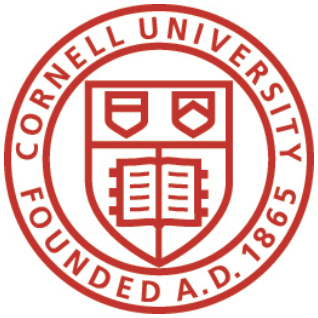
The Online Calibration, Operation and Performance of the CMS Pixel Detector

Ben Kreis (Cornell University) for the CMS Collaboration

Pixel2010, Grindelwald, Switzerland

6-10 September 2010

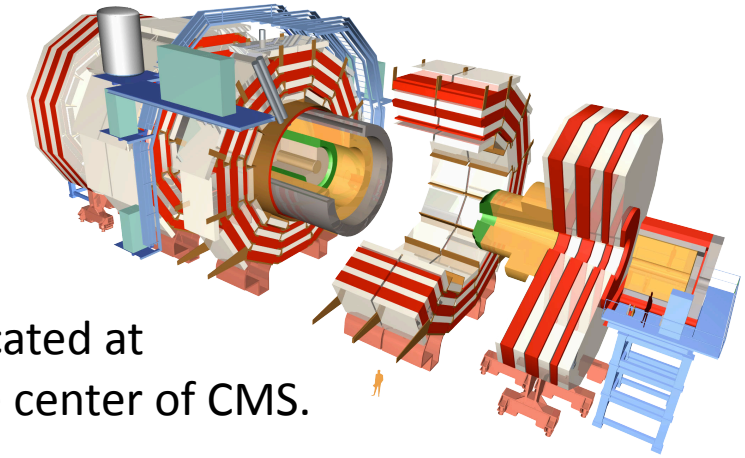


Outline

- Introduction
- Calibrations to optimize performance
  - Threshold calibrations
  - Optimizing the analog response
- Operations
  - Regular recalibrations
  - Experience with beam background
- Summary

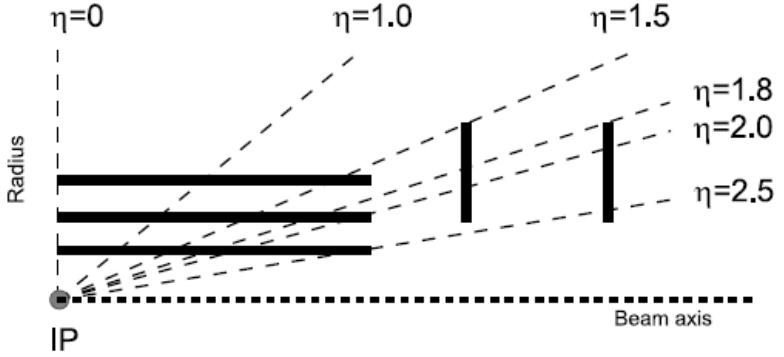
CMS Pixels at the LHC

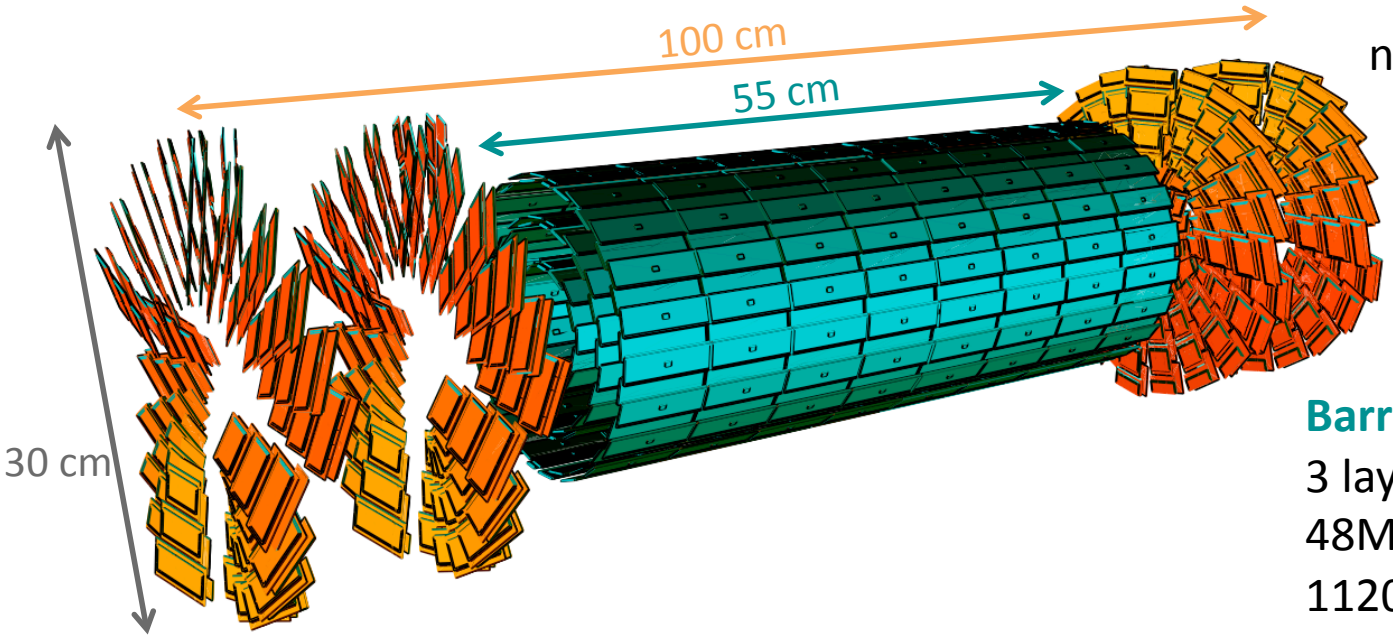
As introduced by Gino Bolla,



Located at
the center of CMS.

Provides precise vertexing and track-finding.
3-hit (2-hit) coverage for |η|<2.1 (|η|<2.5)



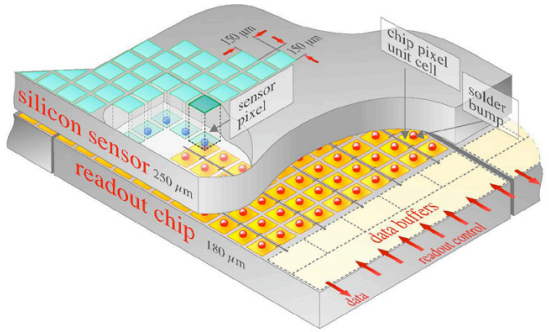


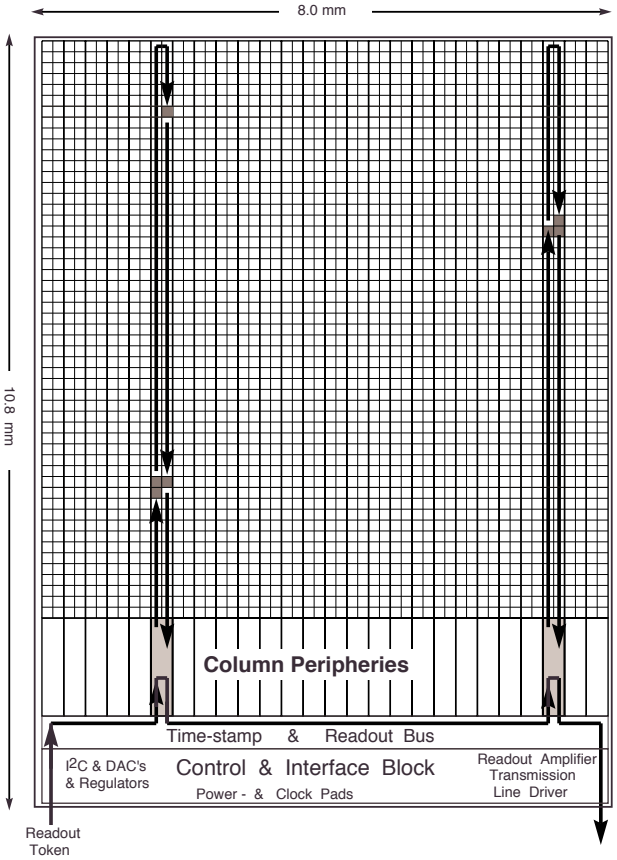
n-in-n silicon sensors

Forward Pixels (FPix)
4 disks at z=±34.5, ±46.5 cm
18M pixels, 4320 ROCs
192 Readout links

Barrel Pixels (Bpix)
3 layers at r=4.3, 7.2, 11.0 cm
48M pixels, 11520 ROCs
1120 Readout links

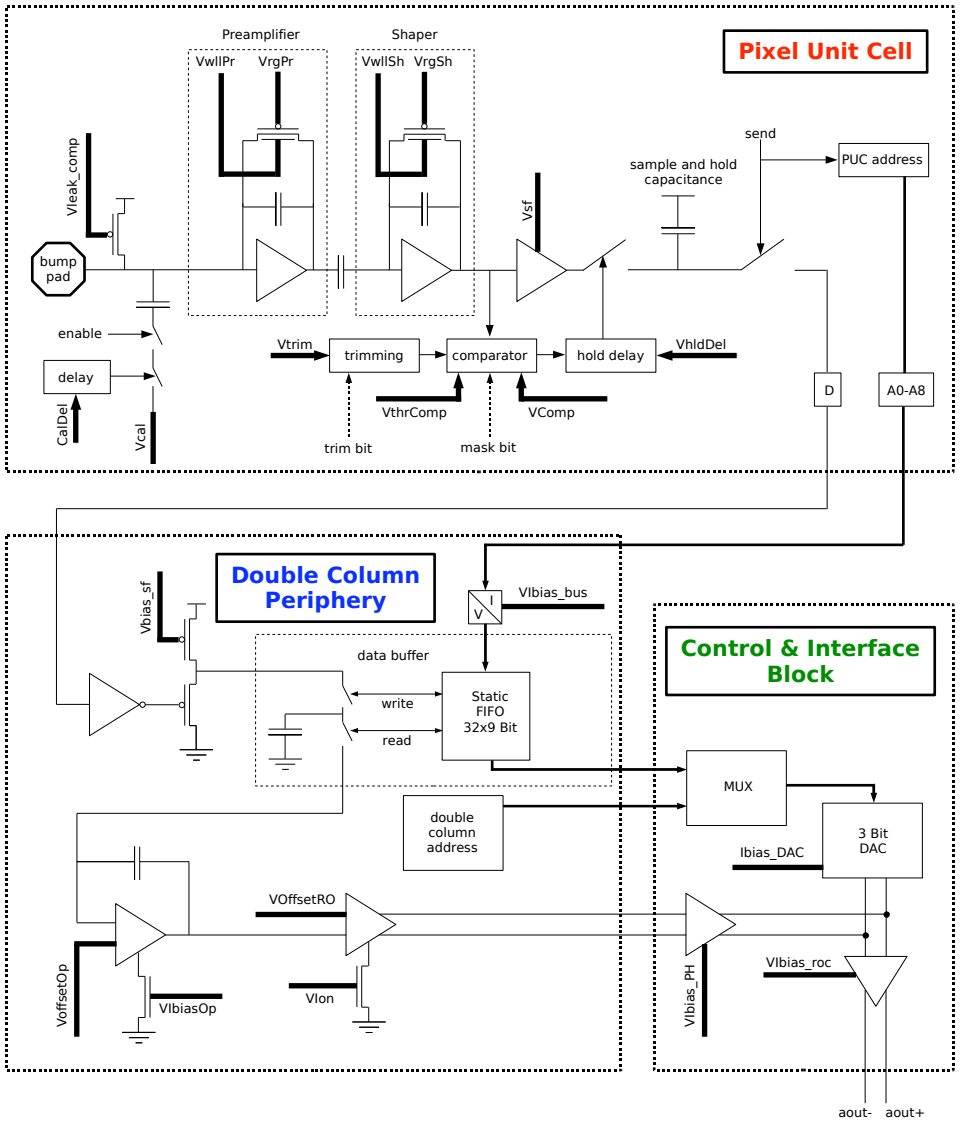
Readout Chip (ROC)





- PSI design, manufactured by IBM
  - 0.25 μm process, ~1.3 million transistors
- 15840 ROCs, 4160 pixels/ROC
- Automatic zero-suppression
- Double column drain architecture
  - Hits buffered until trigger decision arrives
  - Single 25ns-wide bunch-crossing (BX) readout
- 40 MHz analog readout
  - Analog pulse height
  - Other info encoded in analog signal
    - e.g. hit pixel address in base-6

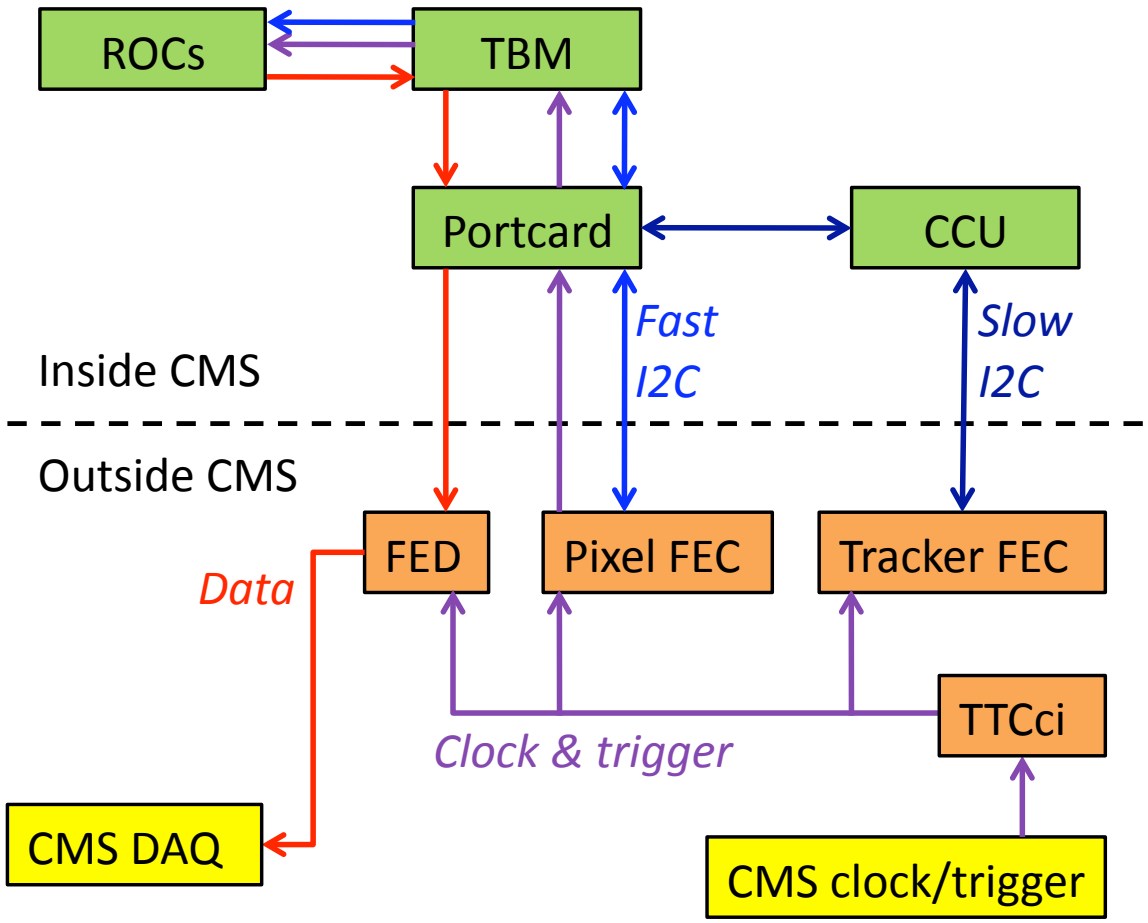
Readout Chip (ROC)



- Amplification, shaping, and zero-suppression on pixel.
- Charge injection for calibration, `VCal`
  - Average ROC response known from x-ray tests:

    electrons = 65.5*`VCal` – 414
- Additional adjustments on double column periphery and C&I block.
- 26 DACs and 3 registers per ROC. 4 trim-bits and 1 mask-bit per pixel.
  - Programmable
  - Configured before every run

Pixel DAQ



Pixel Online Software

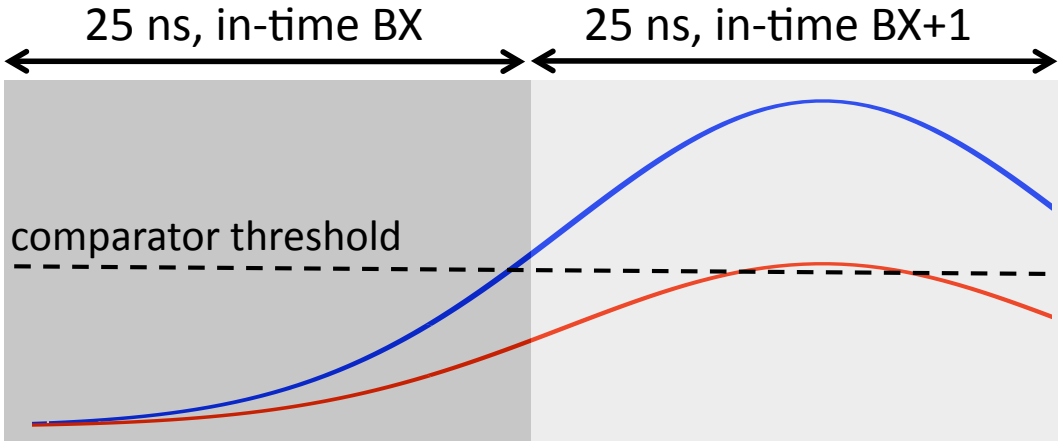
- Configures ROCs & DAQ before running
- Monitors DAQ status while running
- Implements calibrations

The Pixel Online Software controls these **VME devices**.

Front End Driver (FED):

- Receives optical data signal
- Digitizes and sends it to CMS DAQ
- Must be triggered to read & send data!

Introduction to Thresholds



Same timing, but they cross threshold in different BX!
It takes more charge to cross threshold in time!

Single bunch-crossing (BX) readout forces us to confront timewalk.

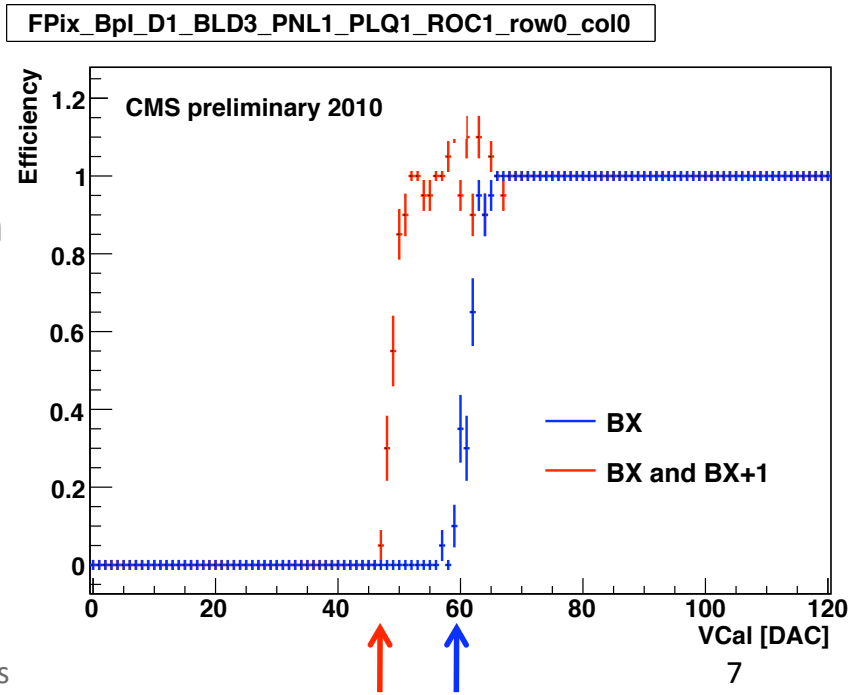
- Smaller signals cross threshold later!

Absolute threshold: Charge required to cross threshold, independent of timewalk.
In-time threshold: Charge required to cross threshold in correct BX (and be read out!).

We measure thresholds using "S-Curves".

- An S-Curve is the hit efficiency as a function of injected charge (`VCal`).
- Absolute threshold = Turn-on of sum of S-Curves from BX and BX+1.
- Calibration in-time threshold = Turn-on of S-Curve from BX only. (The true in-time threshold depends on our timing w.r.t. collisions.)

To be precise, the (summed) S-Curve is fit to an error function and the `VCal` corresponding to 50% hit efficiency is called the turn-on (a.k.a. threshold).



Threshold Trimming

2 DACs per ROC and 4 trim-bits per pixel influence comparator's threshold.

$$\text{Threshold}_i \sim -\underbrace{|C_o|\text{VcThr}}_{\text{offset of ROC}} - \underbrace{|C_1|\text{Vtrim}}_{\text{power of trim-bits on ROC}} \underbrace{(15 - \text{trimbits}_i)}_{\text{pixel trim-bits}}$$

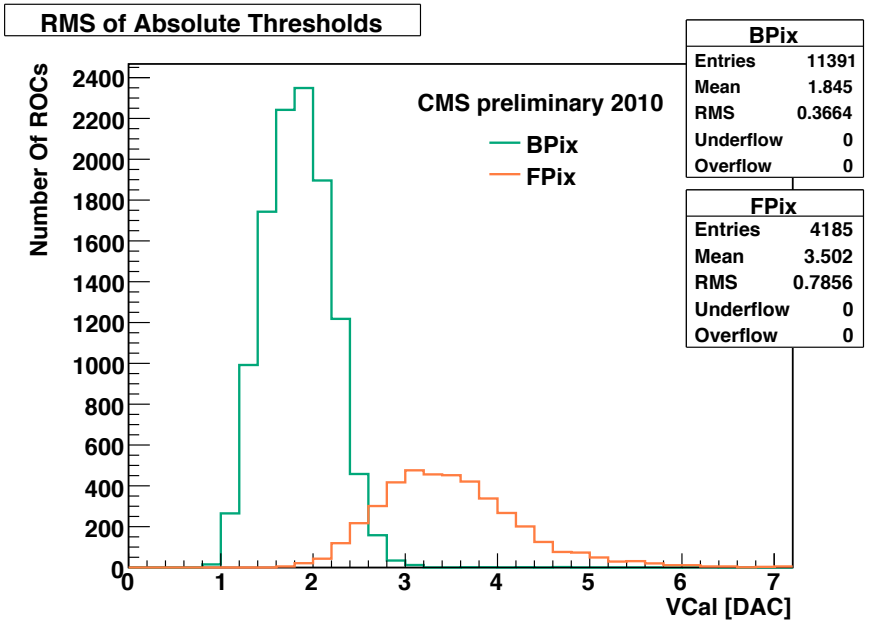
Thresholds are "trimmed" to the same value.

BPix algorithm (absolute):

- Nucl. Instr. and Meth. A 565 (2006) 67.

FPix algorithm (in-time):

1. Determine `VcThr` and `Vtrim` for each ROC.
   - About 4 iterations using ~2% of pixels
   - In each iteration,
     - Measure the influence of `VcThr`, `Vtrim`, and trim-bits on the in-time threshold.
     - Apply additional constraints to "solve" for the next values of `VcThr`, `Vtrim`, and the trim-bits.
2. Determine the trim-bits for every pixel based on the average response on the ROC.



Histogram of absolute threshold RMS .
All RMSs are well below the variation in `VCal`.

Absolute Threshold Minimization

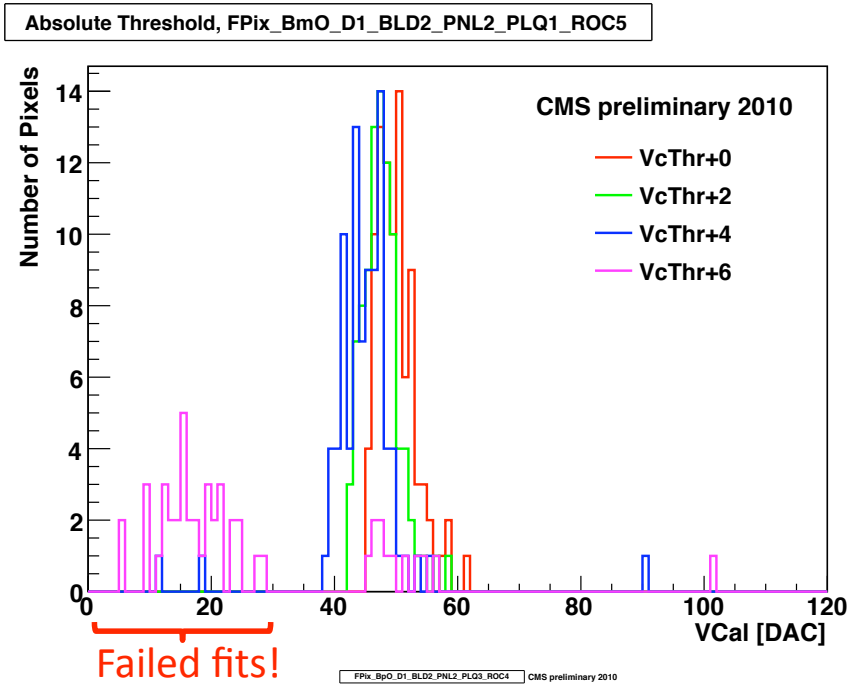
- Minimizing the thresholds increases the hit-position resolution by allowing us to make better use of charge-sharing.
  - **First step:** set the absolute threshold just above the level of cross-talk.
  - **Second step:** set the in-time threshold by adjusting speed of the ROC

Absolute threshold minimization:

Lower comparators' threshold on each ROC until it fails. Then, back away by a fixed amount.

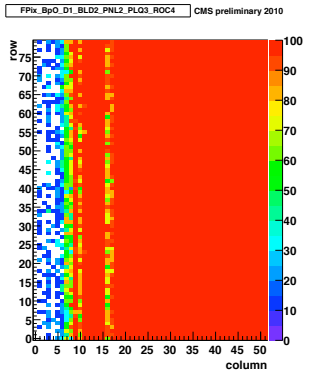
- ROC-based approach using ~2% of pixels
- We lower threshold by raising `VcThr`

Failure is identified by an artifact of the thresholds being too low: failed fits in the S-Curve calibration, which tend to report very low thresholds.



Afterwards, some additional manual tuning is done:

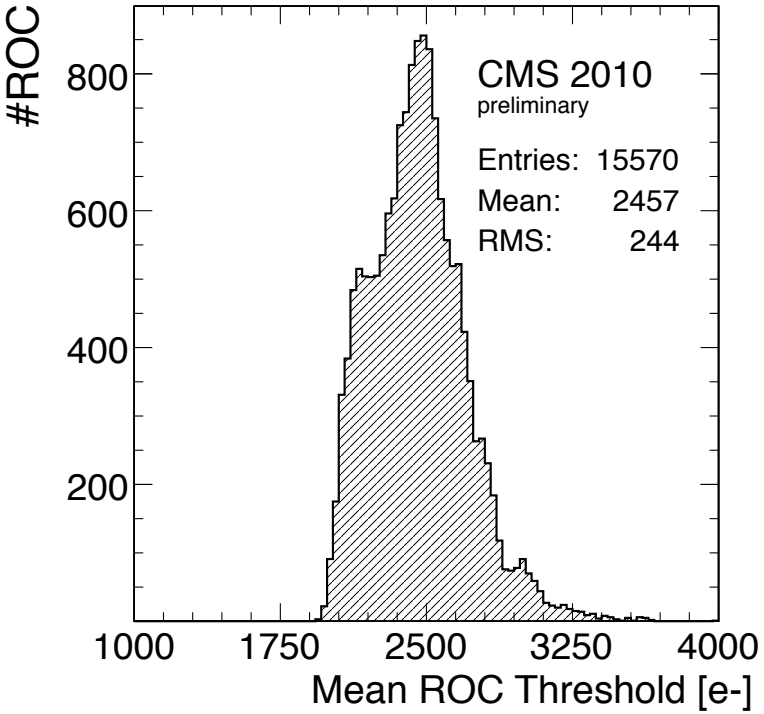
1. Check for S-Curve failures again
2. PixelAlive test on every pixel: Measure efficiency for charge well over threshold.
   - Most common failure is inefficient double columns due to full buffers

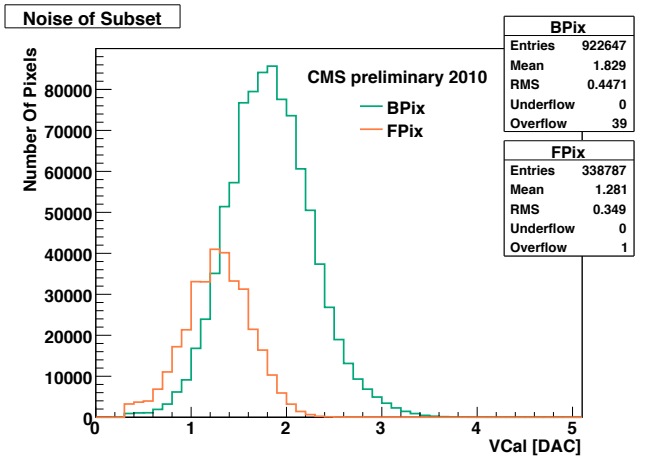


Absolute Threshold Minimization Result

- **Mean threshold = 2457 electrons**
  - ~1/10th the charge collected from a MIP

- From the active modules,
  - ~6k inefficient pixels in PixelAlive
    - At the level of one bad ROC. ~10^{-4} of pixels.
    - Does not include bad sensors or bump bonds
  - 646 "noisy" pixels masked
    - ~10^{-5} of pixels
    - Identified in cosmic ray data taking
  - Total number of dead pixels from collision data:
    - ~7.5k in BPix (~10^{-4} of pixels)
    - ~4k in FPix (~10^{-4} of pixels)
  - One FPix 'slow risetime' channel still active
    - 21 ROCs (~10^{-3} of pixels).
    - Efficiency is time dependent. Can be >95% efficient.





Noise is < 150 electrons, much less than the threshold set just above cross-talk.

In-Time Threshold

The in-time thresholds depend on Vana, a DAC that regulates the voltage applied to the analog part of the ROC.

BPix Vana Calibration:

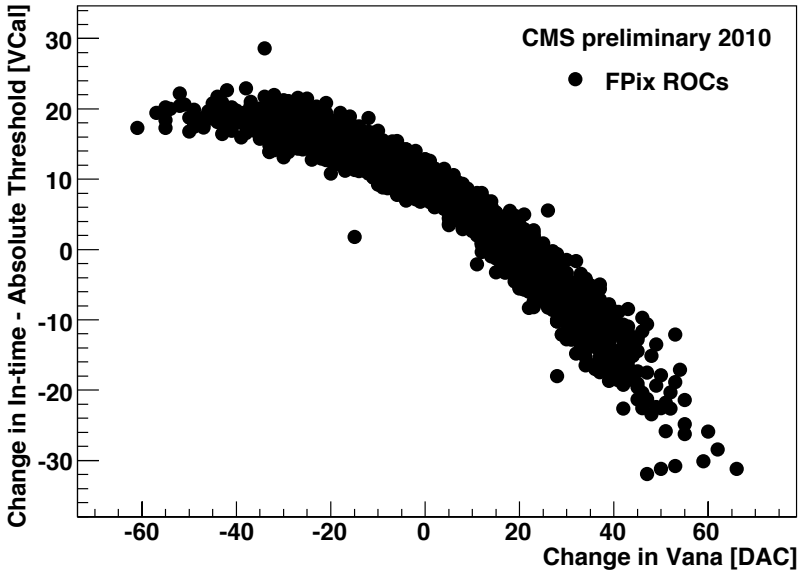
- Measured analog current directly during module testing. Targeted 24 mA.

FPix Vana Calibration:

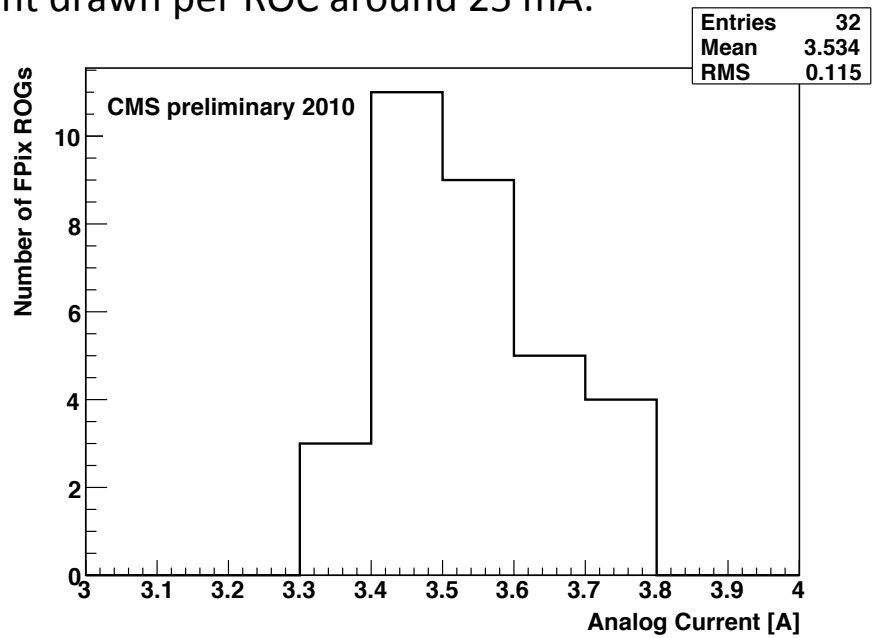
- Set Vana so that:

  average calibration in-time threshold – average absolute threshold = 12 VCal = 786 electrons

- 12 VCal was found to make the average analog current drawn per ROC around 25 mA.



Decrease Vana, difference increases.
Increase Vana, difference decreases.



24 mA/ROC *28 mA/ROC*

Offline methods using collision data find a true in-time threshold 700-1000 electrons larger than the absolute threshold. This is consistent with our expectations from charge injection. May reduce in 2012.

Optimizing Analog Response

We maximize the linearity and range of the gain (pulse height vs `VCal`) to optimize the charge calibration.

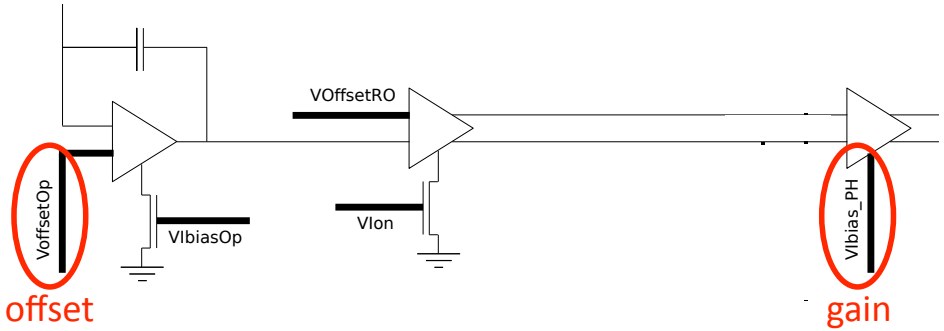
Linearity calibrations:

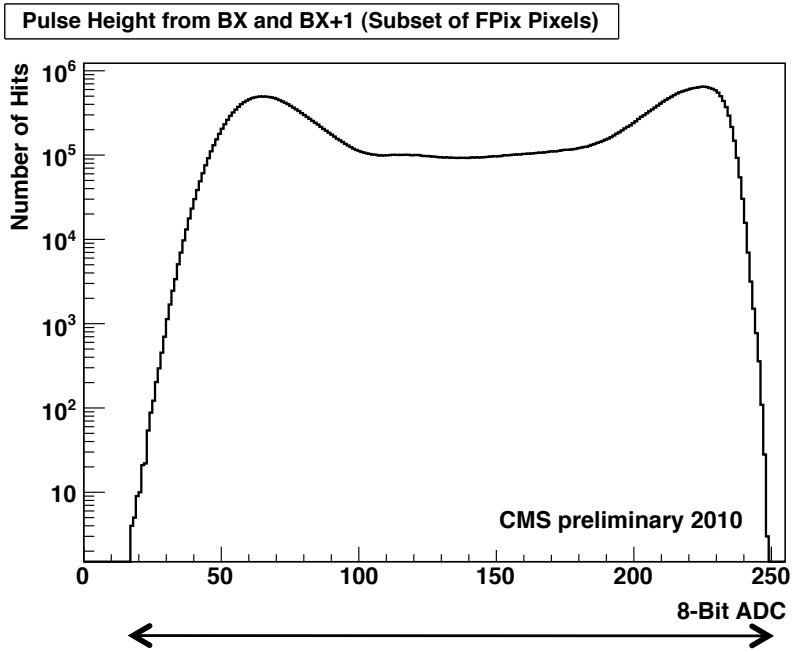
1. `Vsf`, regulates voltage of sample and hold circuit.
   - BPix: increased `Vsf` until average linearity reached target or digital current limit was reached.
   - FPix: algorithm involving delay in sampling.

2. `VHldDel`, delay set to sample maximum of pulse.

Pulse height range calibrations:

- PHRange = PH(large `VCal`) – PH(small `VCal`)
  - We maximize this within the range of the FED's ADC
- Perform a 2D scan: PHRange versus one gain DAC and one offset DAC.





Range is maximized!

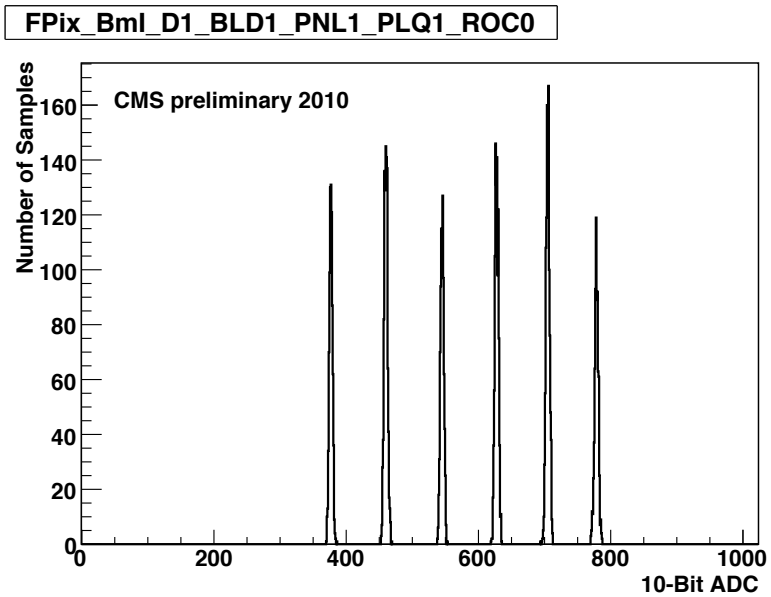
Final results will be presented by Urs Langenegger in an upcoming talk.

Operations – Regular Recalibration

The vast majority of the settings we obtained will be used for the entire 2009 – 2011 LHC run. They will not need to be changed until we change the operating temperature or accumulate significant radiation damage.

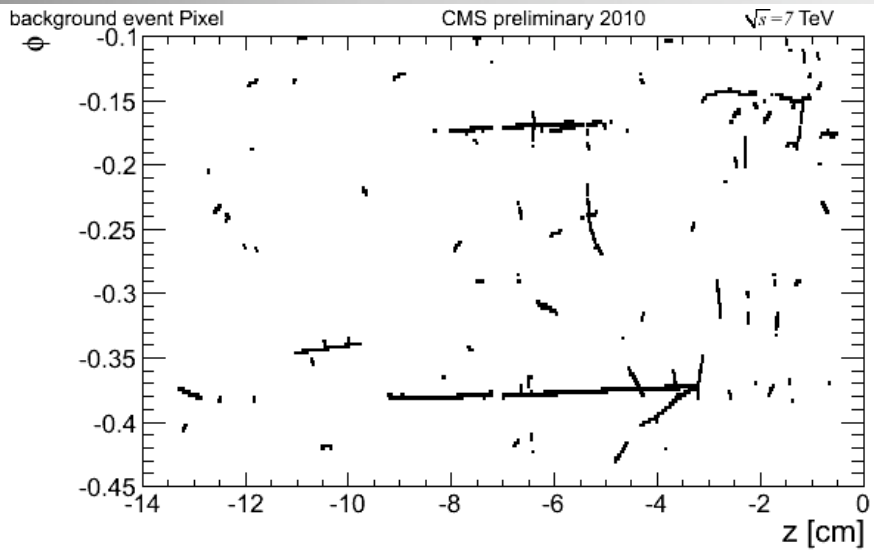
We do, however, regularly recalibrate a few parameters in the FEDs, the devices that receive and digitize the optical signal.

1. **FED Baseline**
   - Adjusts offsets in the FEDs' optical receivers to keep the signal within the ADC's range.
   - Must be done when the signal shifts up or down due to temperature fluctuations on laser.
   - Performed ~weekly. Temperature changes of 2-3 °C are handled without recalibration by an automatic baseline correction.
2. **FED Address Levels**
   - ADC levels necessary to decode pixel addresses
   - Settings are stable. Performed every few weeks, but mostly just as a check.
3. **FED Clock Phase**
   - Phase of ADC. Performed only as a check ~monthly.



Histogram of address levels from one ROC. Shows excellent separation.

Operations - Beam Background



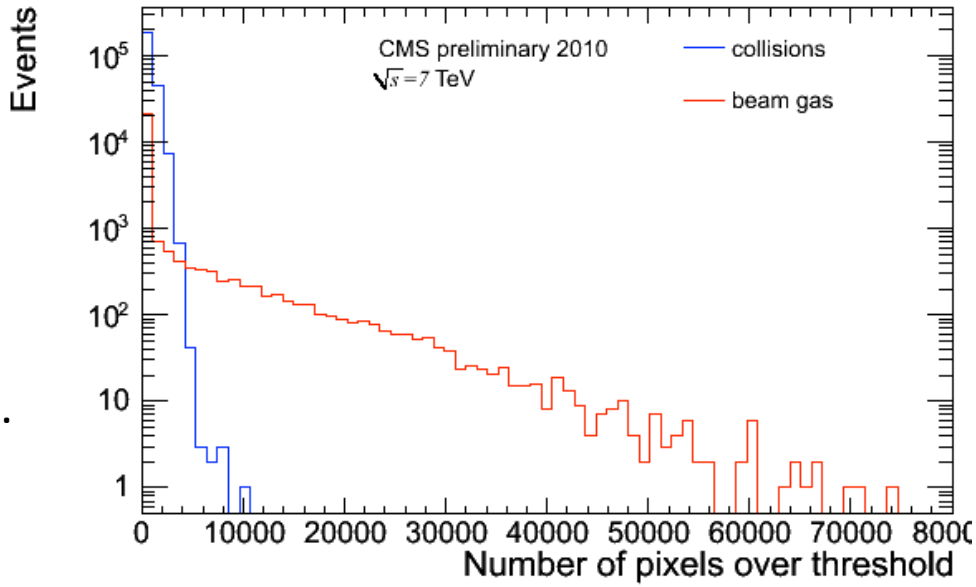
occupancy map from a single event

Beam-background events:

- We observe showers of particles that graze the detector along the beam axis (z).
- They occur coincident with bunch crossings.
- These events are consistent with beam interactions with gas in the beam pipe.

These events:

- lead to a *huge* occupancy, especially in the barrel layers.
- impose challenges to maintaining event synchronization, especially at high trigger rates.

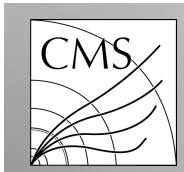


Operations - Beam Background

- The problem: Reading in a large beam background event can block one or more FED inputs for a long time. All of the hits must be read in! Meanwhile, the FED is triggered for the events that follow the beam background event.
- The effect: The events that follow the beam background event eventually come, but not when they are expected. If nothing is done, the FED inputs are out of synchronization forever.
- The solution:
  - Part 1: Drop the events that did not arrive when expected.
    - Synchronization is regained once all of these events have been dropped.
    - Works well at lower trigger rates. However, at higher trigger rates, more and more events must be dropped.
  - Part 2: If many events must be dropped, pause the CMS trigger so that no more events must be dropped and the FED can take the time required to resynchronize itself.
    - Works well at high trigger rates.
    - Causes only ~ 0.5% deadtime.

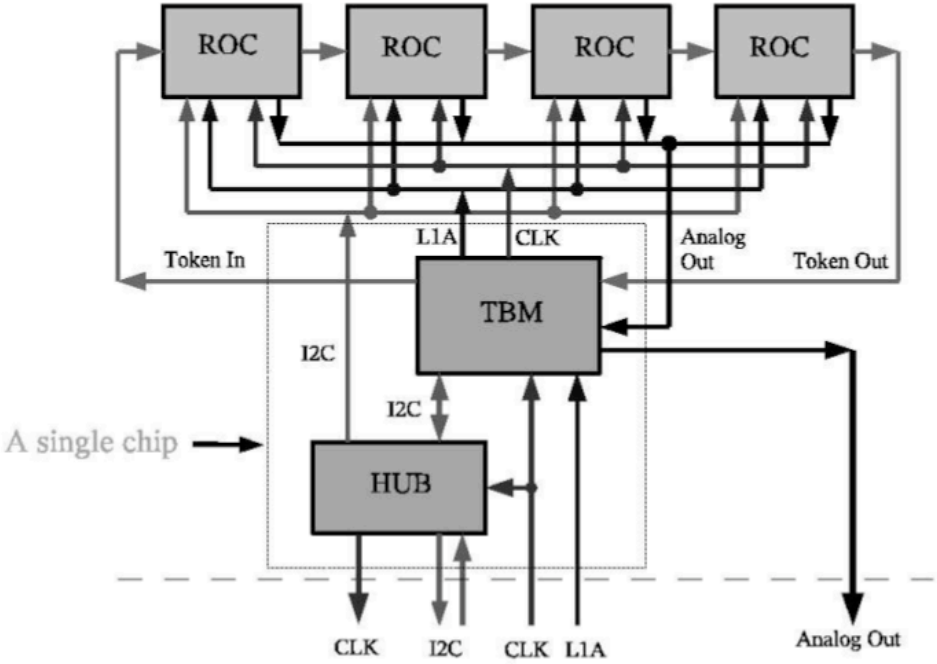
Summary

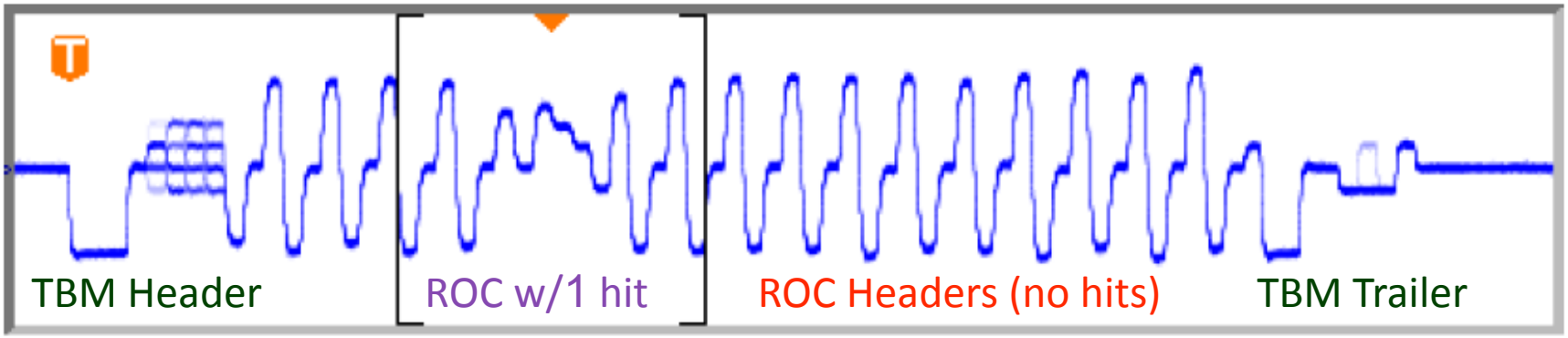
- We performed calibrations to optimize the detector's performance.
  - Threshold trimming
  - Absolute threshold minimization
  - In-time threshold calibration/`Vana`
  - Maximized gain linearity and range
- All ROC settings and most other settings will not change before the 2012 break in LHC running.
- We have implemented a solution to maintain efficiency despite exposure to large beam background events.



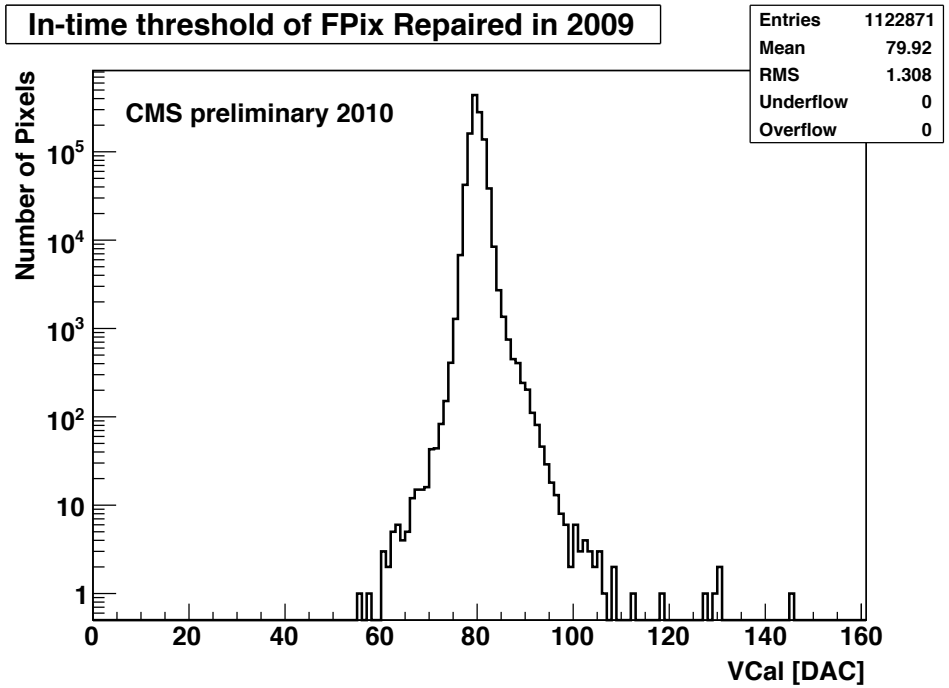
Backup Slides

Analog Readout of ROC

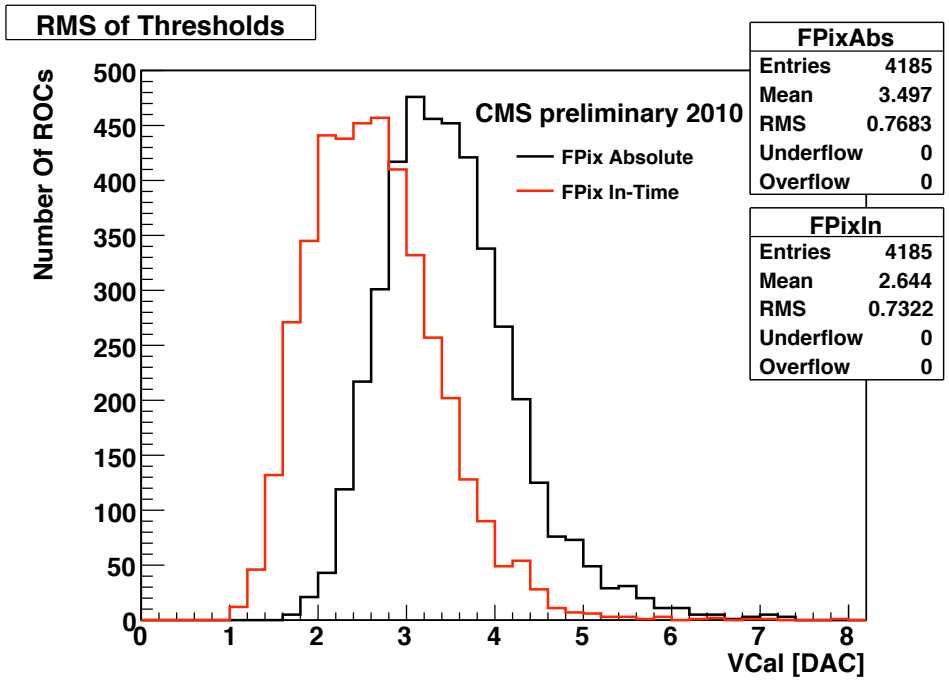




Threshold Trimming

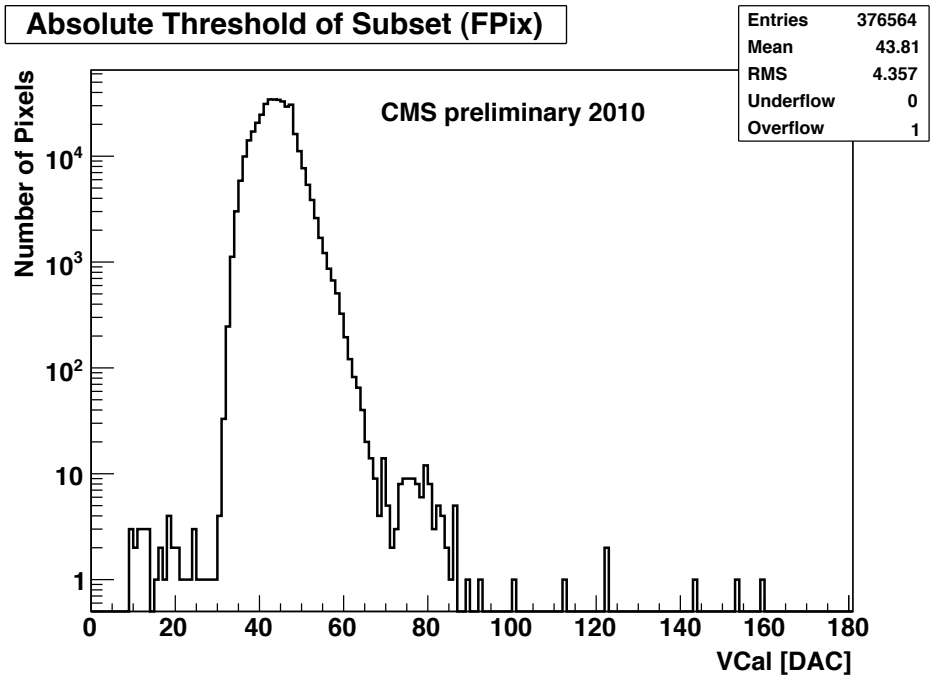


Threshold distribution from ~1M pixels after trimming. The RMS is very small (1.3 VCal) and increases due to further calibrations.

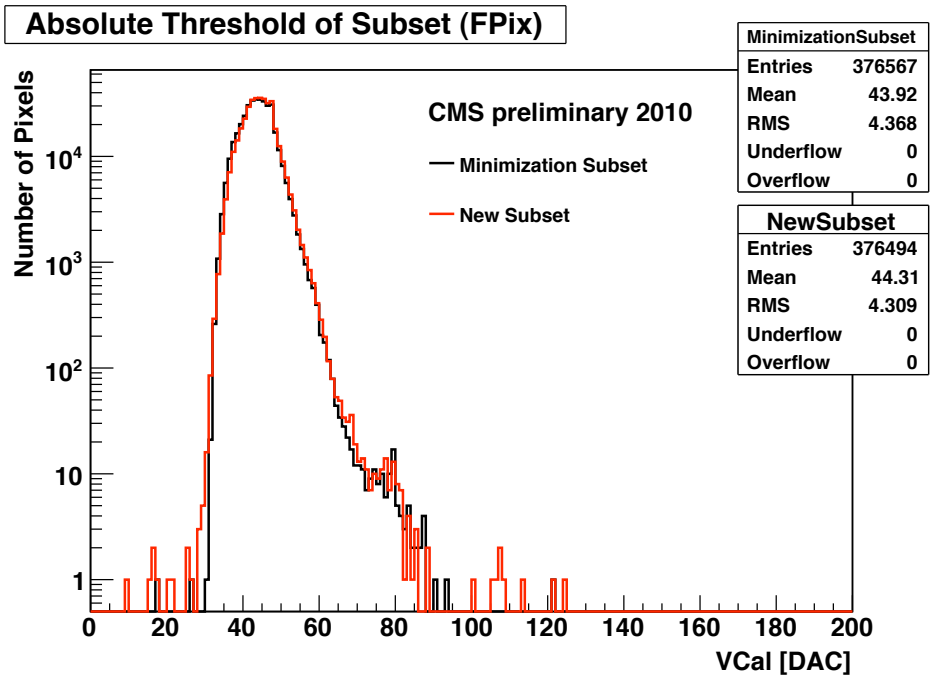


Distributions of absolute and in-time threshold RMSs. Each entry is one ROC. The absolute threshold RMSs are larger because the FPix trimmed the in-time thresholds.

Absolute Threshold Minimization

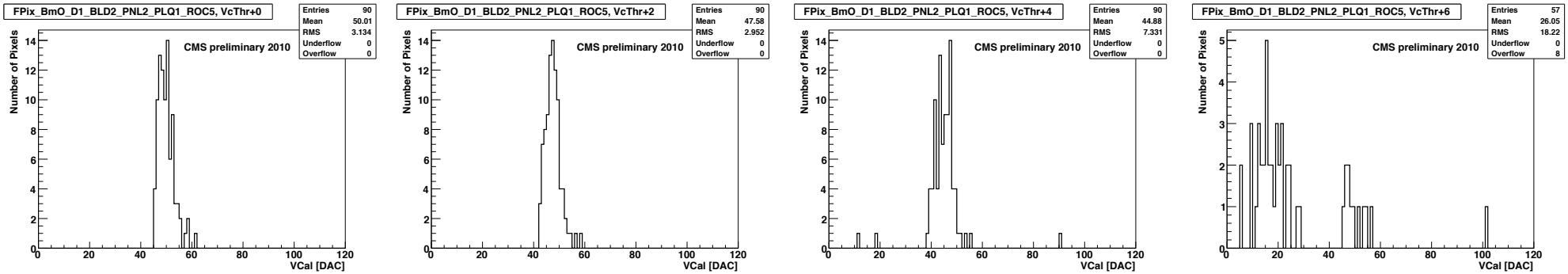


The FPix absolute threshold distribution from a subset of pixels after the automatic procedure and before manual tuning. Only a few failures are present.



The FPix absolute threshold distribution from two different subsets of pixels towards the end of the minimization. The red distribution is a new subset not used in the minimization. Its similarity to the black shows that using only 2% of the pixels is enough.

Absolute Threshold Minimization

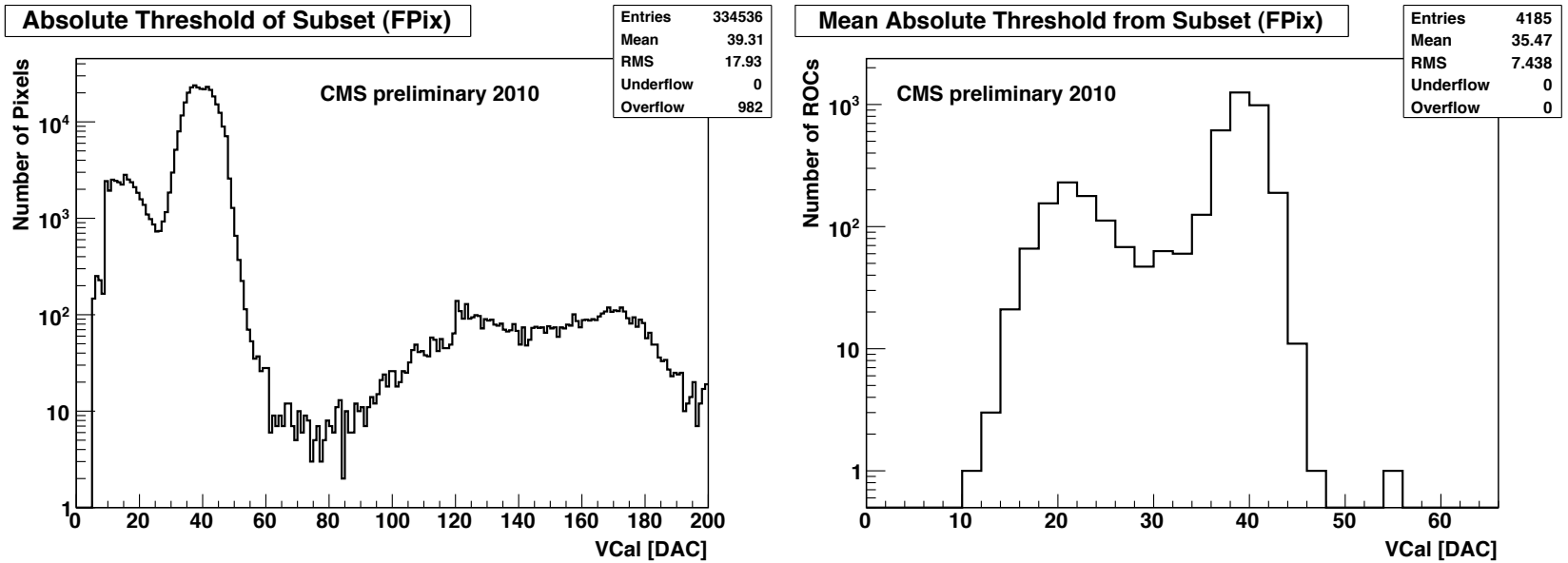


lower threshold →

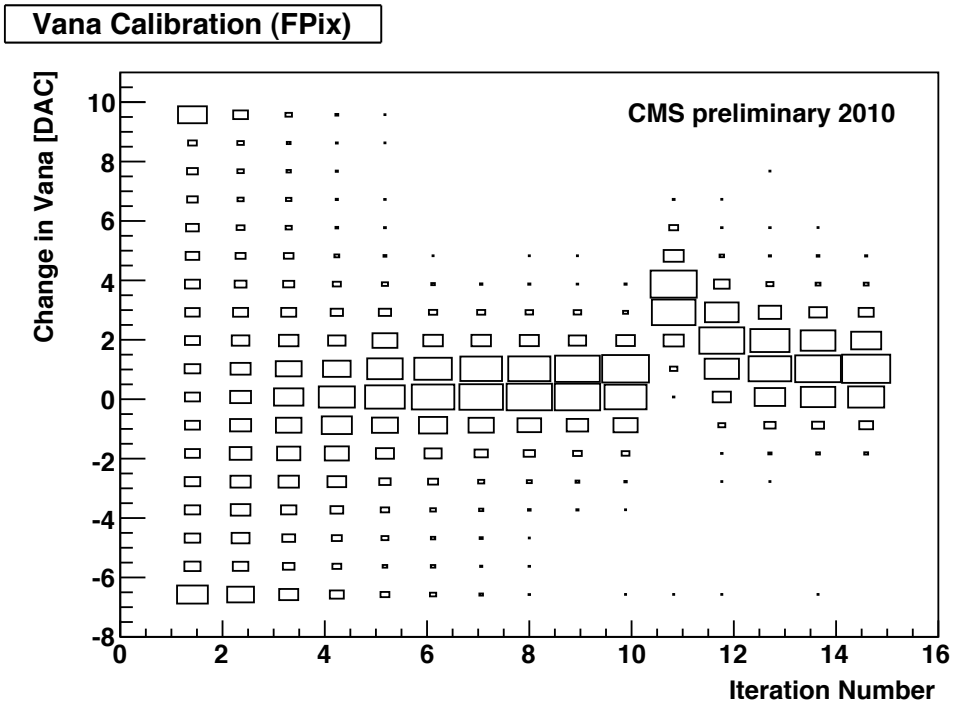
Precise definition of failure:

- ROC's mean threshold < 35 VCal
- Any pixel threshold < 32 VCal
- Any pixel missing from readout

When the thresholds are set far too low (VcThr+10):



Vana Calibration

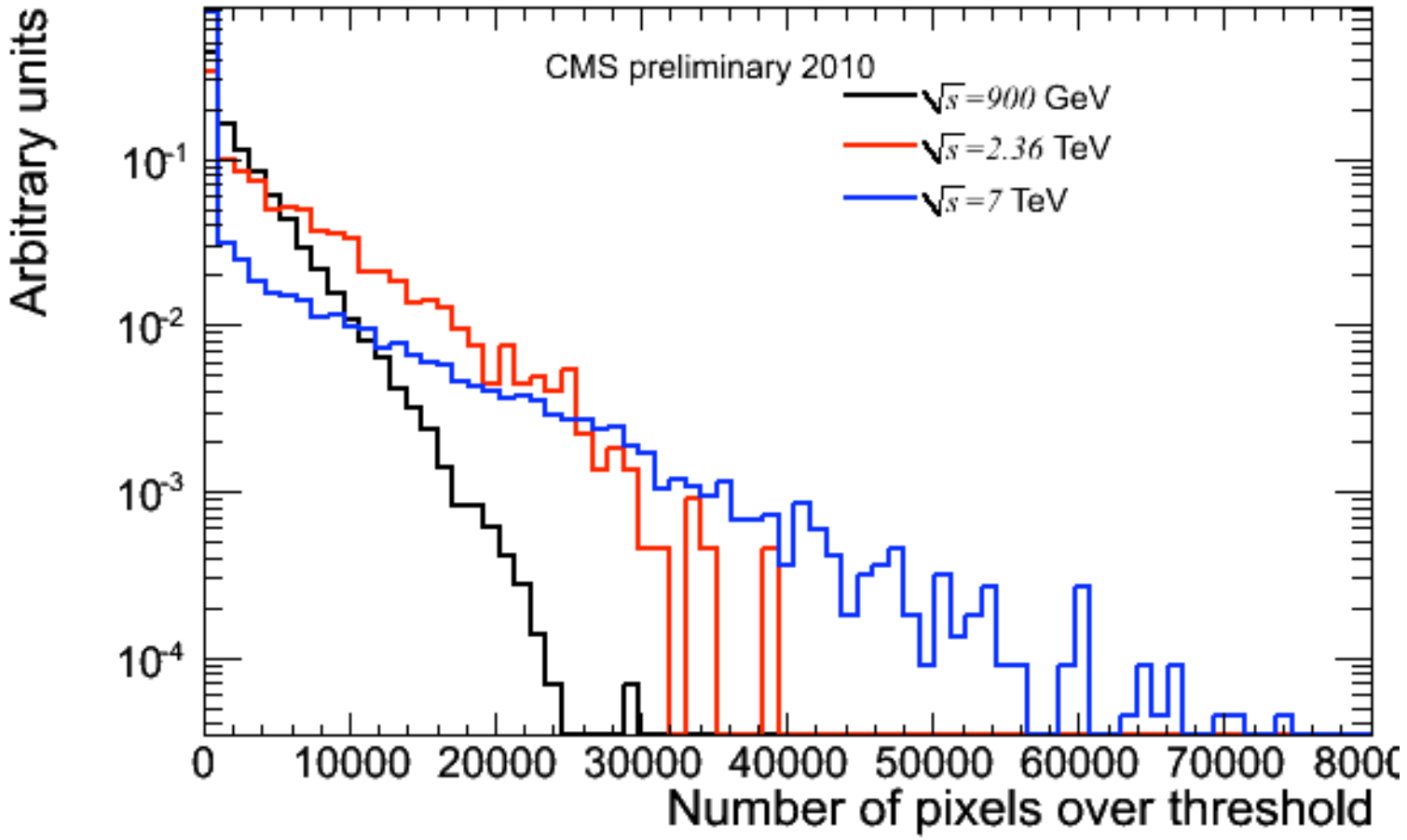


Plot showing the convergence of the Vana calibration. We originally targeted a difference of 15 VCal, then moved to 12 VCal in iteration 11.

Operating Conditions

- Power
  - Sensor bias voltage is only ON during "LHC Stable Beams" or when there is no beam.
    - CMS monitors beam conditions with dedicated detectors
    - Automatic safety mechanisms are in place
- ROCs
  - When the bias voltage is OFF, the ROCs are disabled and thresholds are set to maximum. Automatic in Pixel Online Software.
- Real-time data quality monitoring

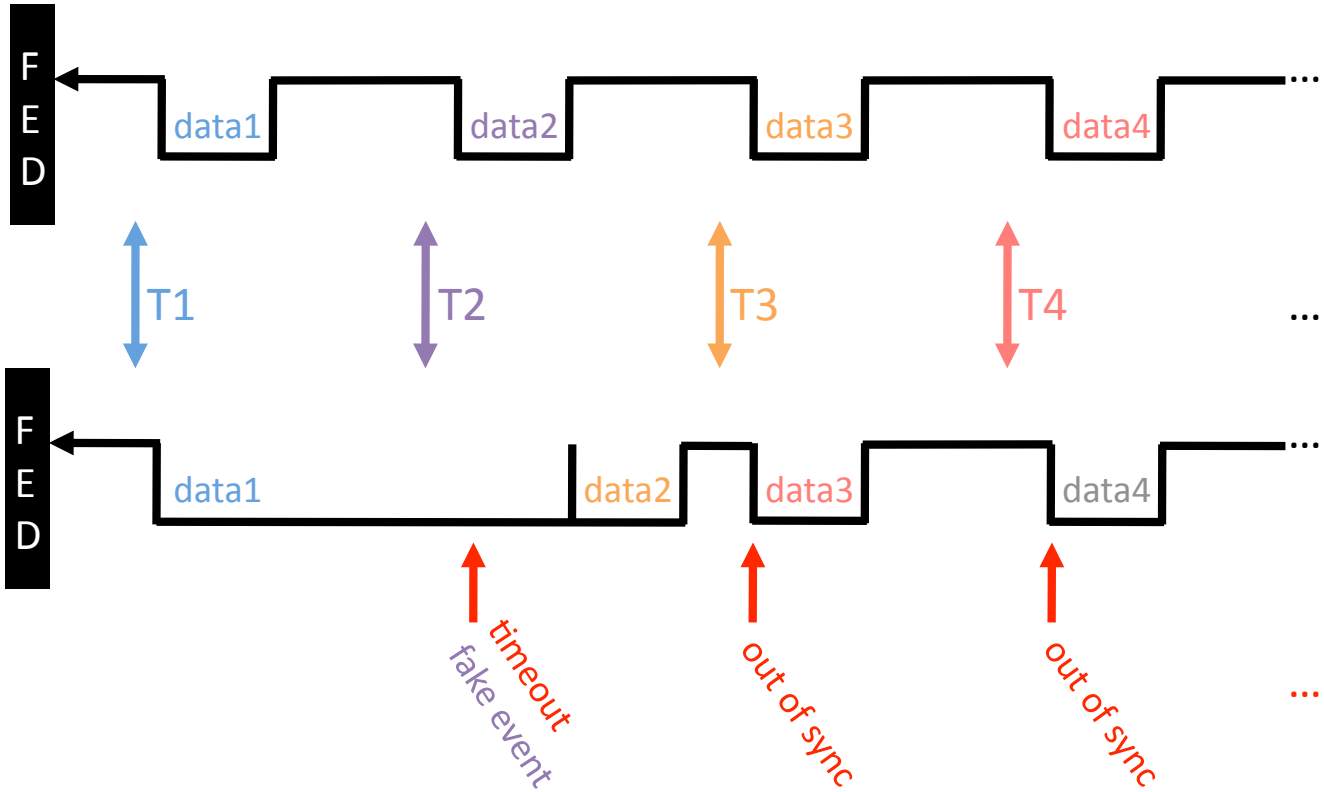
More on Beam Background



We have observed that the occupancy induced by beam background events increases with beam energy. The rate is constant.

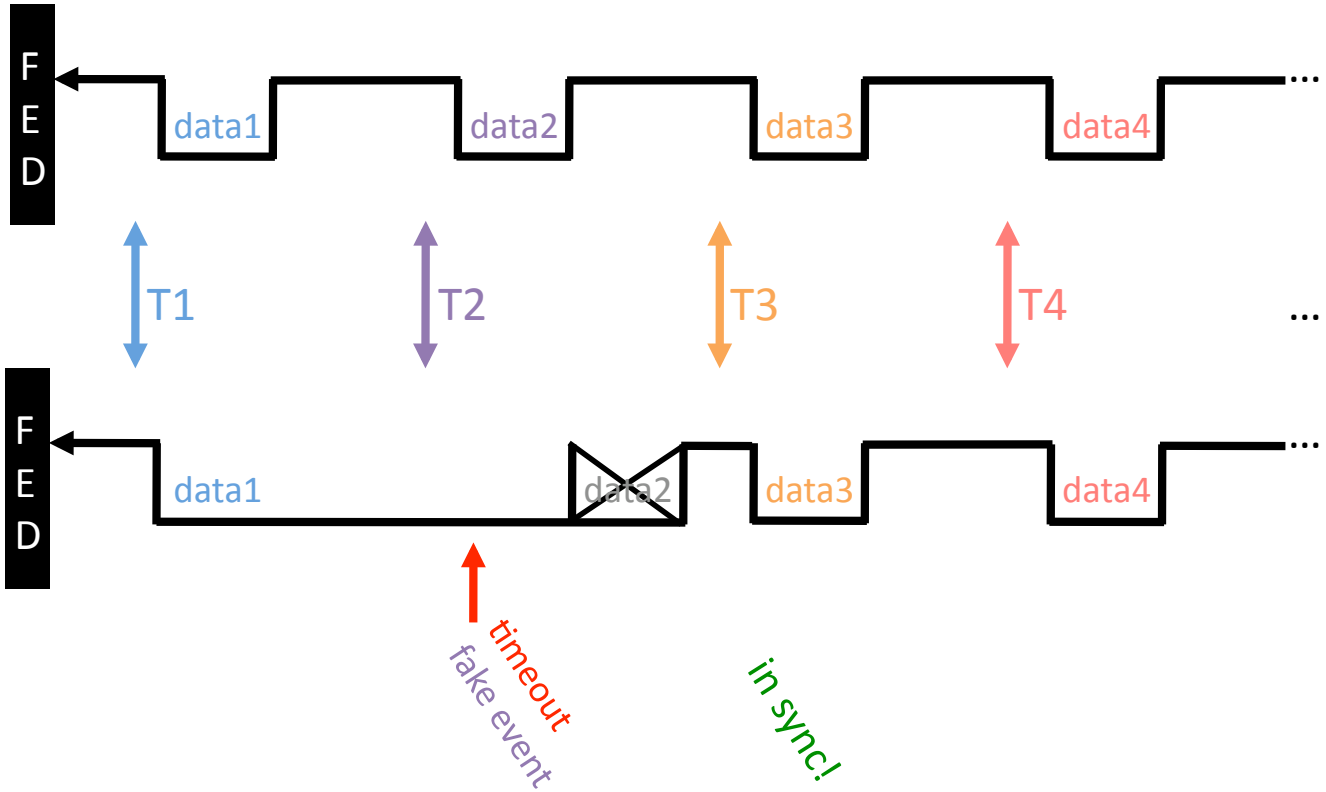
Handling Beam Background

- In collision events, the FED receives a trigger and then the data from the front end follows.
- After a large event arrives,
  1. FED receives T1 and begins reading in data1
  2. FED receives T2, but data2 doesn't arrive in time (timeout!)
  3. FED sends on fake event to satisfy CMS DAQ system
  4. On next trigger, T3, data2 is in the FED. The FED is out of sync forever.



Handling Beam Background

- **Solution, Part 1:** Add a timeout counter. Drop an incoming event for each timeout.
- After a large event arrives,
  1. FED receives T1 and begins reading in data1
  2. FED receives T2, but data2 doesn't arrive in time (timeout!)
  3. FED sends on fake event to satisfy CMS DAQ system
  4. One incoming event, data2, is dropped because there was one timeout.
  5. On next trigger, T3, the FED is in sync.



Handling Beam Background

- What happens at higher rates? (minimal case below: T3 arrives while reading in data2)
  - There are consecutive timeouts. More events are dropped.
  - If high rate continues, consecutive timeouts continue. Maybe forever!
    - **Solution, Part 2**: After N consecutive timeouts, stop triggers so that FED can resynchronize itself. We use N=4.