ITkPixV1.1 self trigger source scan

Angira Rastogi, Simone Pagan Griso, Timon Heim

Weekly Instrumentation Meeting

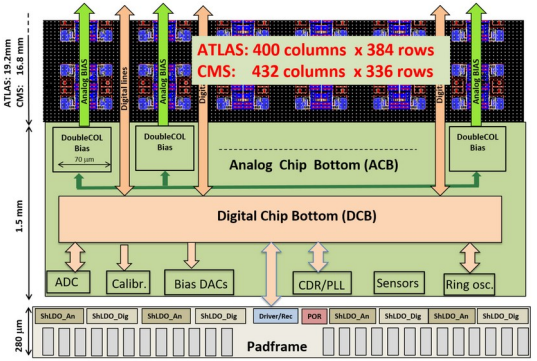
Aug 5, 2022

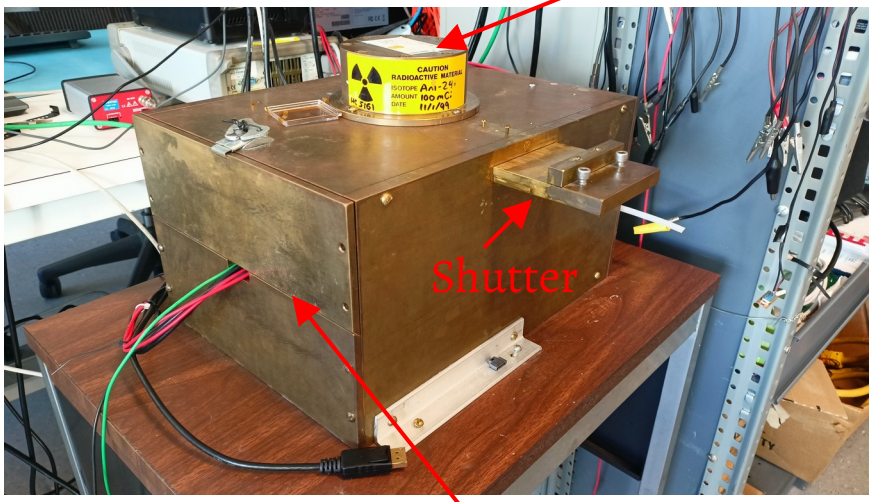
BERKELEY LAB

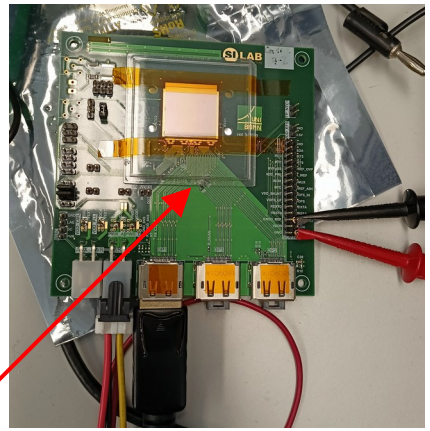
Introduction

Goal: Irradiating the ITkPixV1.1 chip with Americium-241 and performing the readout **(source scan)**.

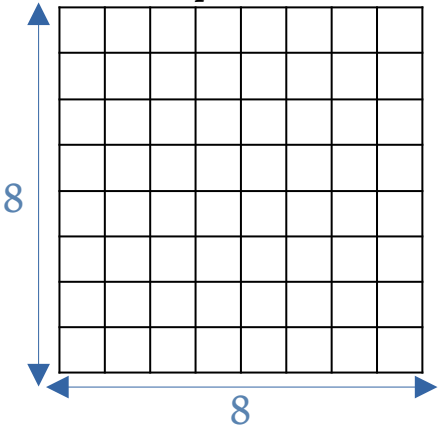
- Helps in identifying the disconnected bump bonds between sensor and chip.







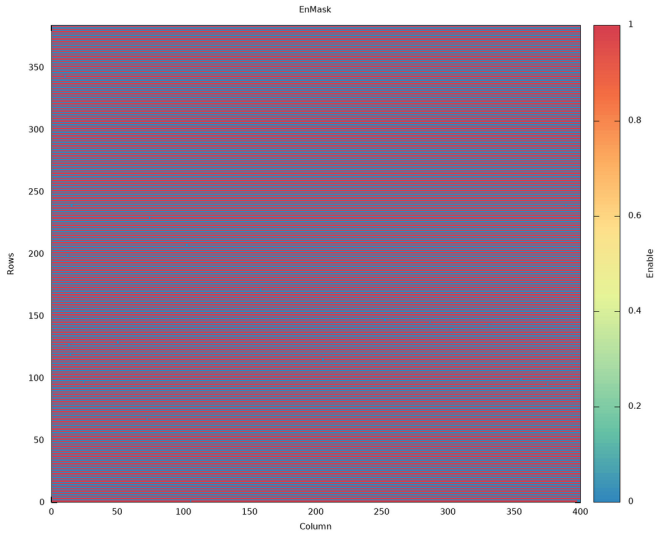
8x8 pixel core.



Quick recap

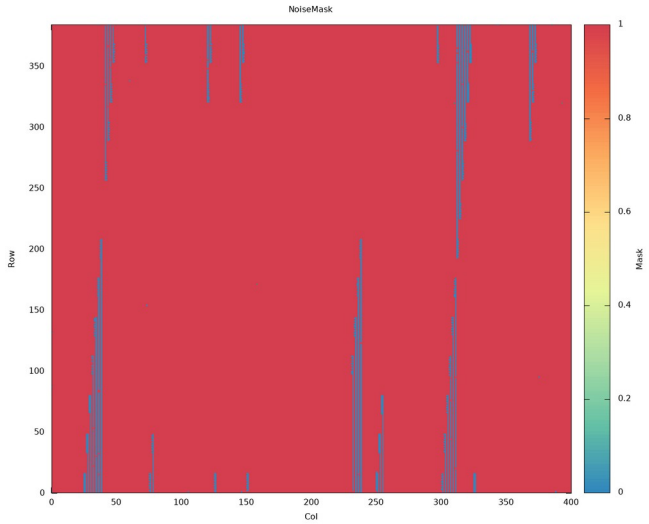
Self trigger Analog scan

MaskLoop(64,0,2)



Self trigger source scan

(Noise analysis)



- Source scan was failing for MaskLoop(64,0,1), no triggers were being generated.
  - Tested various configurations for HitORPatternLUT (2/4/8/16/65534).
  - Also changing HitORMasks 0/1/2/3 (enabling one core column only at a time).
  - Enabling one HitOR lane (HitORPatternLUT=2) with one core column at a time (HitORMasks 0/1/2/3).
- Also, inconsistency in the above two results: masking pattern from analog scan not matching the noise occupancy in source scan.
  - Problem found, older YARR setup.

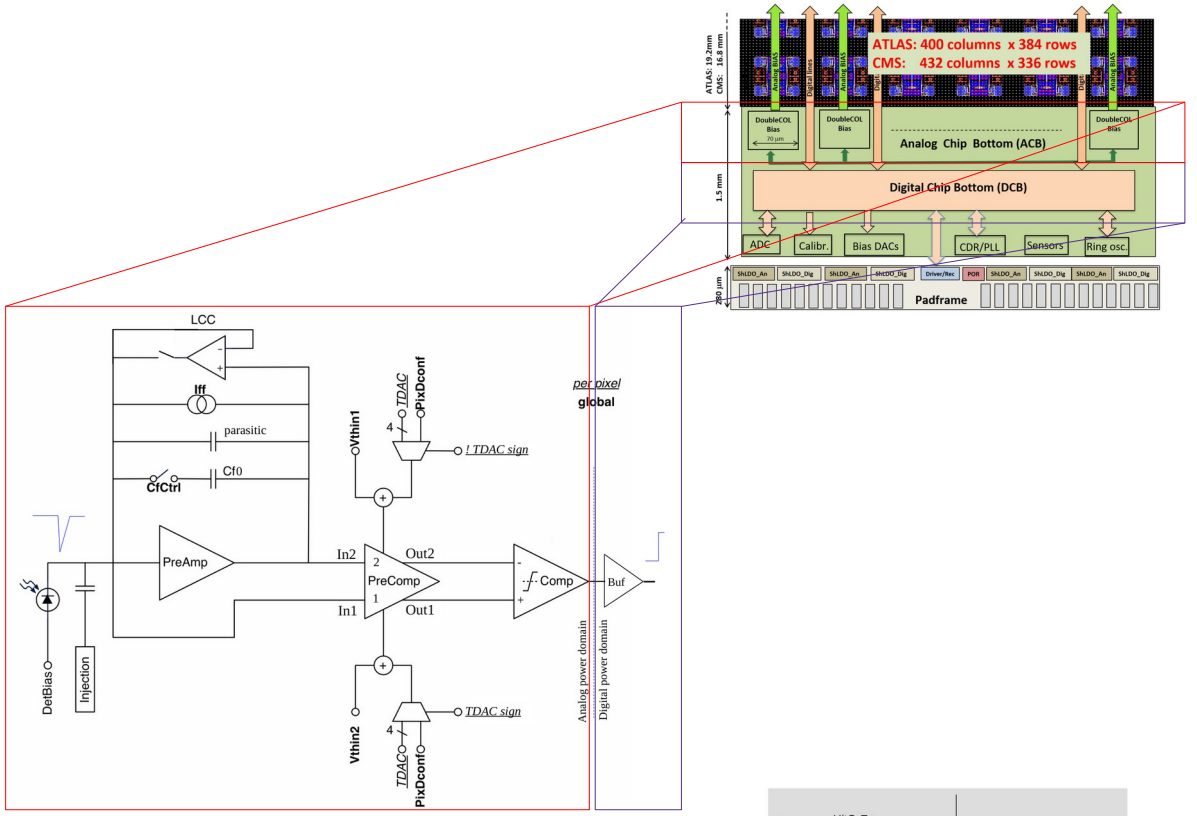
Outline

- Masking sequence
- Self triggering
- Source scan

Masking

- Standard digital scan (-m = 1)
- Standard analog scan
- Standard noise scan

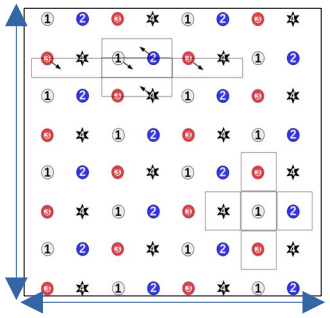
Failure Name	Scan Type
Digital Dead	Digital Scan
Digital Bad	Digital Scan
Merged Bump	Analog Scan
	Crosstalk Scan
Analog Dead	Analog Scan
Analog Bad	Analog Scan
Tuning Failed	Threshold Scan
	ToT Test
Noisy	Noise Scan
Disc. Bump	Source Scan
High Crosstalk	Crosstalk Scan

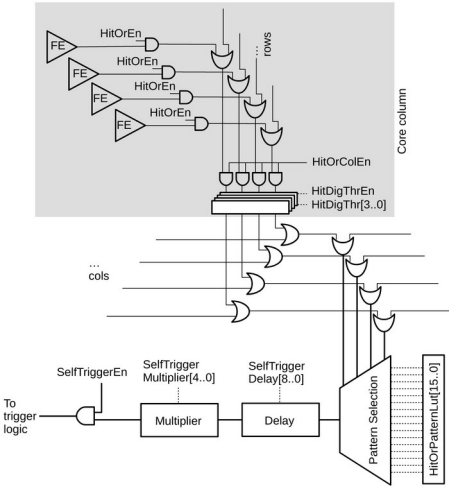


- Self-trigger digital scan
- Self-trigger analog scan

  - checks the HitOR operations

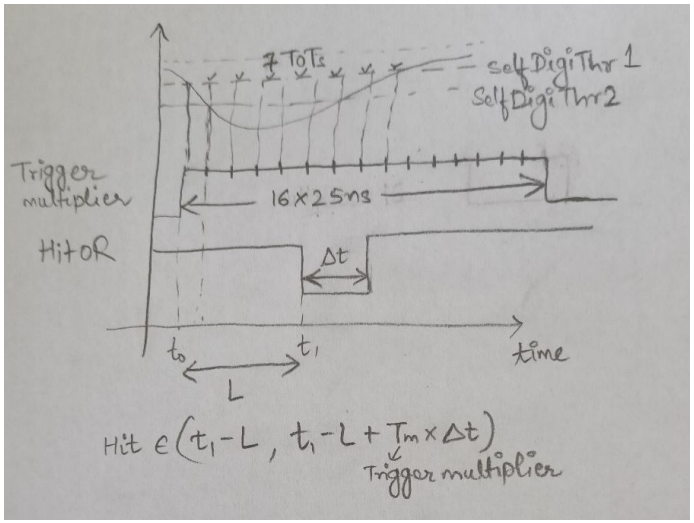
8x8 pixel core

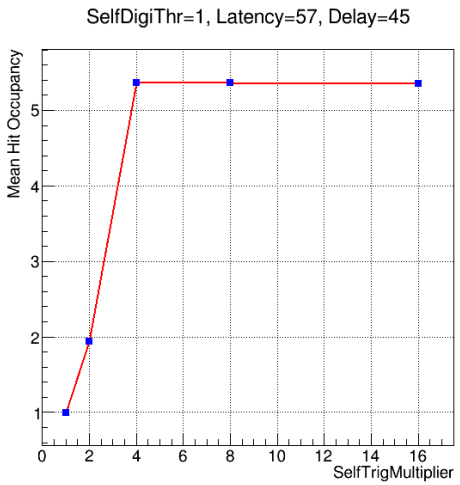


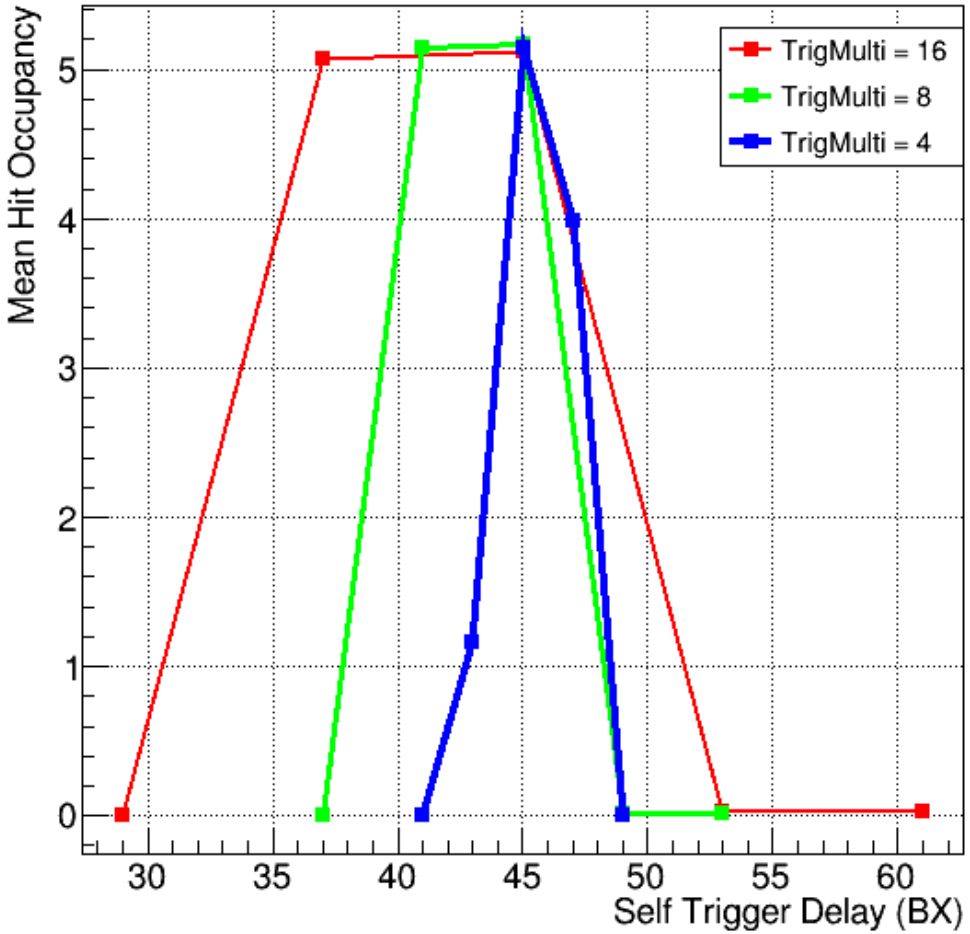


Self triggering logic

Readout from chip happens at a clock/trigger which can be applied externally or internally at a hit instance **(self-triggering)**.



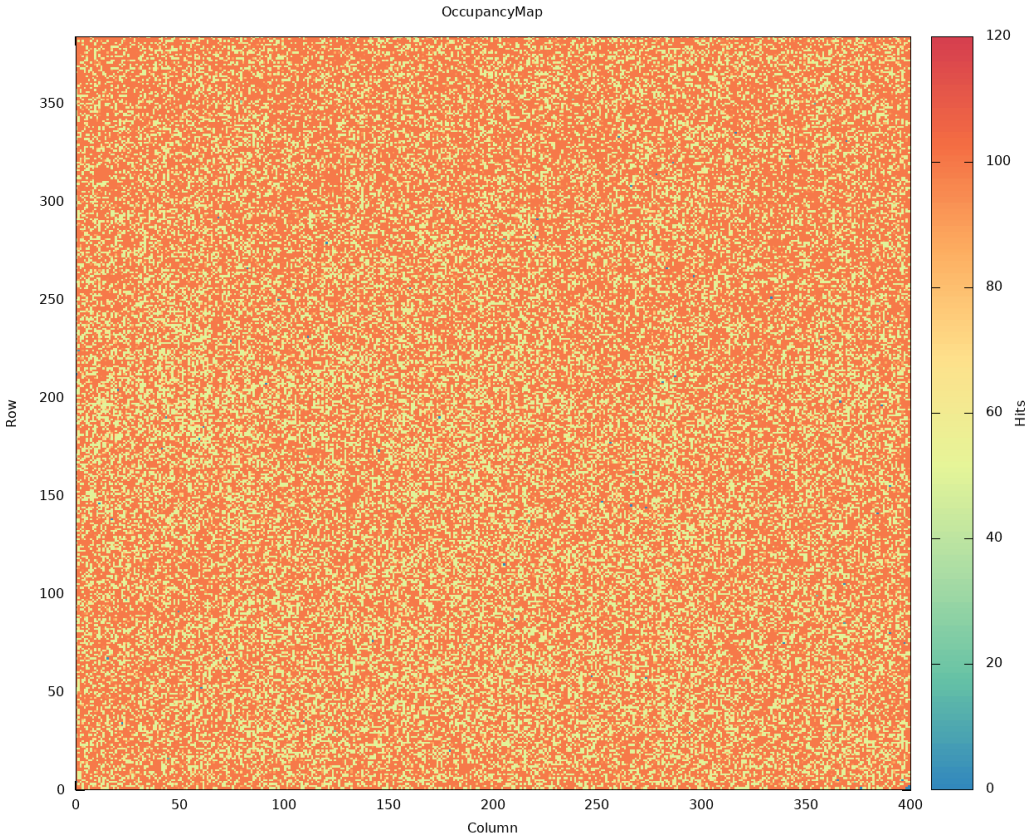




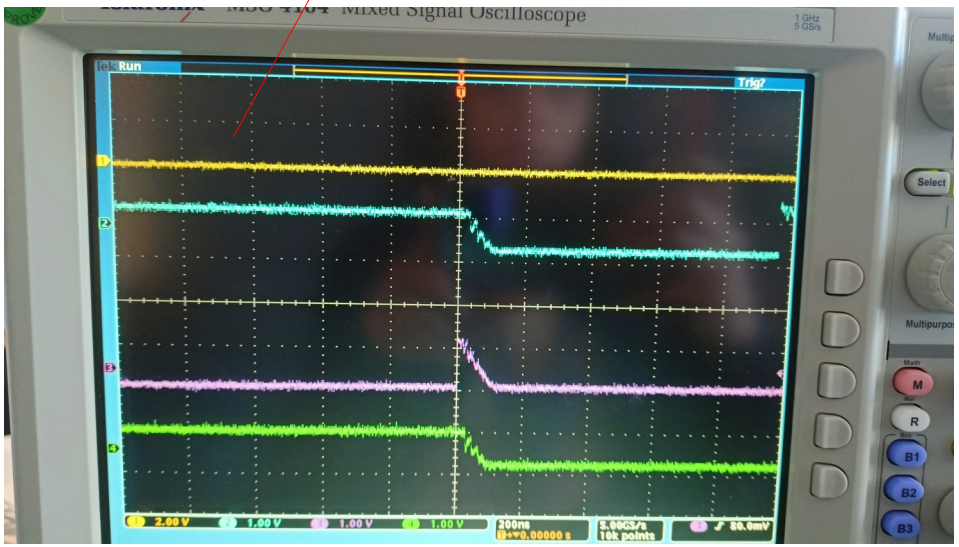
- Various self-trigger parameters: Self-trigger Delay, Self-trigger Multiplier and Self Digital Threshold.
- Should be optimized.

Results: Self-trigger analog scan

MaskLoop(64,0,1)



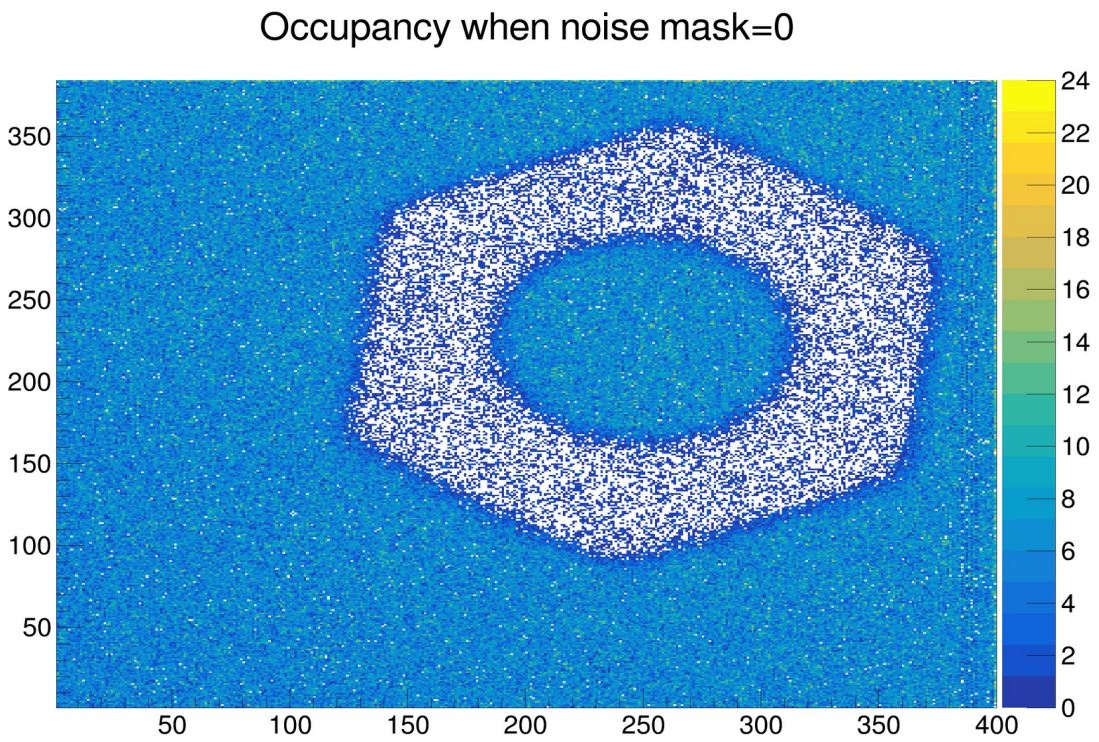
Ignore this, connected to a different oscilloscope for trigger rate measurement.



- Self-trigger hit logic had the shortcoming of counting each injection twice (discharging & charging as two edges).
- This resulted in almost all the pixels being masked, since charge injected ≠ hit occupancy.
- **Fix:** Modified the logic to allow hit occupancy per pixel between (injection-10,2*injection+10) for qualifying the pixels as 'working'. Merge request

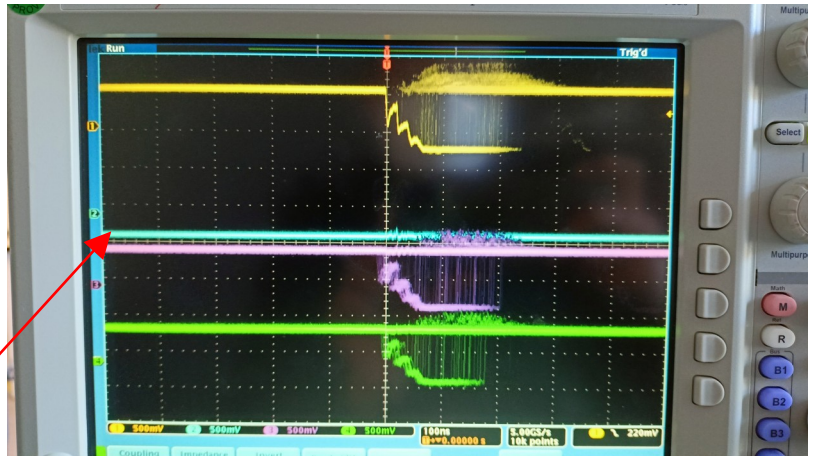
All the HitOR lanes seems to be receiving the triggers.

Source scan

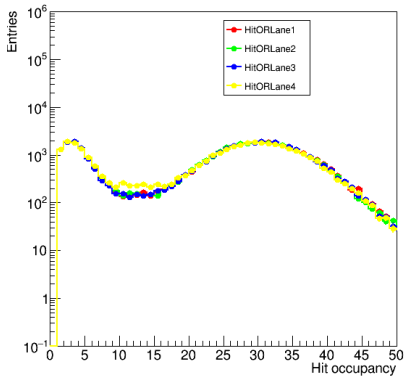


- Latency = 57
- SelfTrigDelay = 45
- SelfTrigDigiThr = 1
- SelfTrigMulti = 4
- Time = 60s

- Self-trigger source scan (Noise analysis, Noise Thr ~ 5*mean occupancy)
- HitOrPatternLUT = 65534 (OR of all lanes)
- HitORMasks_0/1/2/3 = 0/0/0/0 (all enabled)



Issue #1: One HitOR lane stuck?



Issue #2: Scan 'KILLED' after t=120s due to excessive RAM usage.

Source scan: Memory issue

```
369     ScanHelper::banner(logger,"Scan");
370
371     logger->info("Starting scan!");
372     std::chrono::steady_clock::time_point scan_start = std::chrono::steady_clock::now();
373     scanBase->run();
374     scanBase->postScan();
375     logger->info("Scan done!");
376
377     // Join from upstream to downstream.
378     for (unsigned id=0; id<bookie->getNumOfEntries(); id++) {
379         FrontEnd *fe = bookie->getEntry(id).fe;
380         if (fe->isActive()) {
381           fe->clipRawData.finish();
382         }
383     }
384
385     std::chrono::steady_clock::time_point scan_done = std::chrono::steady_clock::now();
386     logger->info("Waiting for processors to finish ...");
387     // Join DataProcessor
388     // Join histogrammers
389     for( auto& proc : procs ) {
390       proc.second->join();
391     }
392
393     std::chrono::steady_clock::time_point processor_done = std::chrono::steady_clock::now();
394     logger->info("Processor done, waiting for histogrammer ...");
395
396     for (unsigned id=0; id<bookie->getNumOfEntries(); id++) {
397         FrontEnd *fe = bookie->getEntry(id).fe;
398         if (fe->isActive()) {
399             fe->clipData.finish();
400         }
401     }
402
403     // Join histogrammers
404     for( auto& histogrammer : histogrammers ) {
405       histogrammer.second->join();
406     }
407
408     logger->info("Processor done, waiting for analysis ...");
409
410     for (unsigned id=0; id<bookie->getNumOfEntries(); id++) {
411         FrontEnd *fe = bookie->getEntry(id).fe;
412         if (fe->isActive()) {
413             fe->clipHisto.finish();
414         }
415     }
```

scanConsole.cpp

Scan gets 'KILLED' after t~120s due to excessive RAM usage.

```
// Clipboards to buffer data
ClipBoard<RawDataContainer> clipRawData;
ClipBoard<EventDataBase> clipData;
ClipBoard<HistogramBase> clipHisto;
```

selftrigger_source.json 1.19 KiB

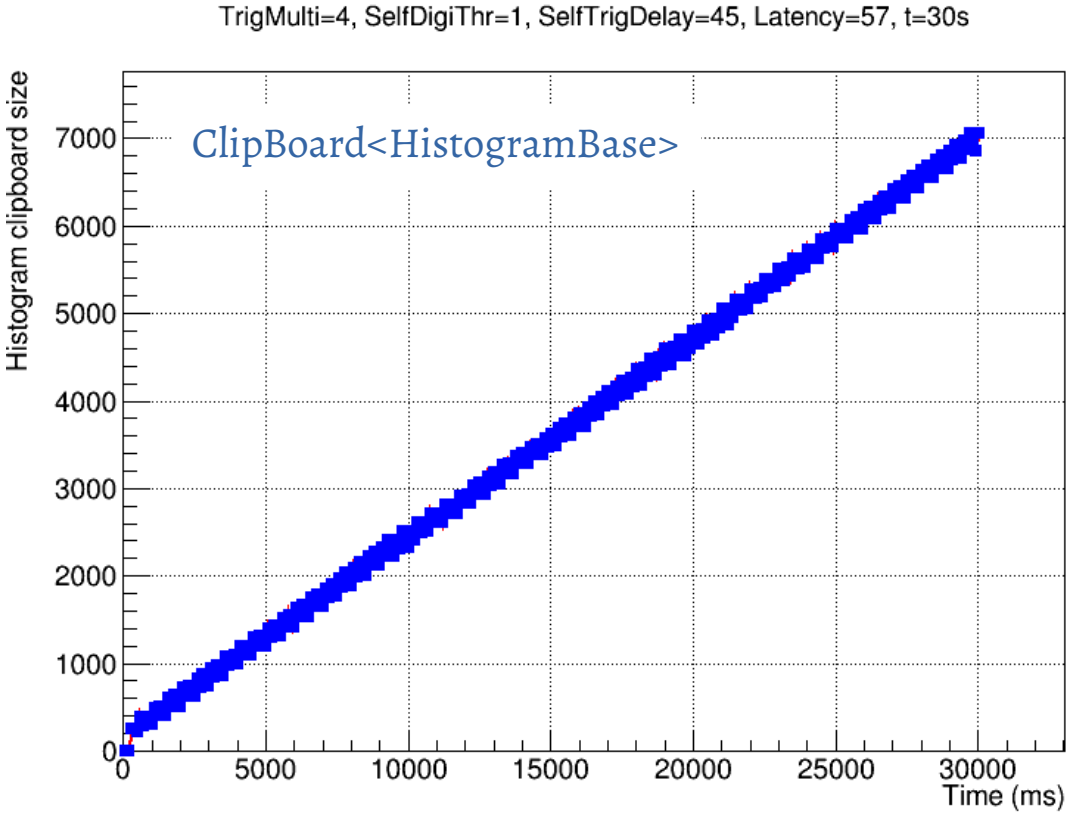
```
{
  "scan": {
    "analysis": {
      "0": {
        "algorithm": "NoiseAnalysis"
      },
      "n_count": 1
    },
    "histogrammer": {
      "0": {
        "algorithm": "OccupancyMap",
        "config": {}
      },
```

bin/scanConsole -r configs/controller/specCfg-rd53b-4x4.json -c configs/connectivity/0x131a5.json -s configs/scans/rd53b/selftrigger_source.json -p

Diagnostics

```
1448  void NoiseAnalysis::processHistogram(HistogramBase *h) {
1449      if (h->getName() == OccupancyMap::outputName()) {
1450          occ->add(*(Histo2d*)h);
1451      } else if (h->getName() == HitsPerEvent::outputName()) {
1452          n_trigger += ((Histo1d*)h)->getEntries();
1453      }
1454  }
```

```
131  void Histo2d::add(const Histo2d &h) {
132      if (this->size() != h.size())
133          return;
134      for (unsigned int i=0; i<(xbins*ybins); i++) {
135          double d = h.getBin(i);
136          data[i] += d;
137          max = std::max(d, max);
138          if (h.isFilled(i))
139                  m_isFilled[i] = true;
140      }
141      entries += h.numOfEntries();
142  }
```




```
logger->info("Starting scan!");
std::chrono::steady_clock::time_point scan_start = std::chrono::steady_clock::now();

thread_diagnostics = std::thread([&](){
    while(run_thread){
      for (unsigned id=0; id<bookie->getNumOfEntries(); id++) {
        FrontEnd *fe = bookie->getEntry(id).fe;
        std::this_thread::sleep_for(std::chrono::seconds(1));
        if (fe->isActive()) {
          std::chrono::steady_clock::time_point size_stamp = std::chrono::steady_clock::now();
          evtdata_size.push_back(fe->clipData.size());
          histodata_size.push_back(fe->clipHisto.size());
          timestamp_size.push_back(std::chrono::duration_cast<std::chrono::seconds>(size_stamp-scan_start).count());
        }
      }
    }
  });

scanBase->run();
scanBase->postScan();
logger->info("Scan done!");

//kill the thread
run_thread = false;
thread_diagnostics.join();
```

Running multithreading

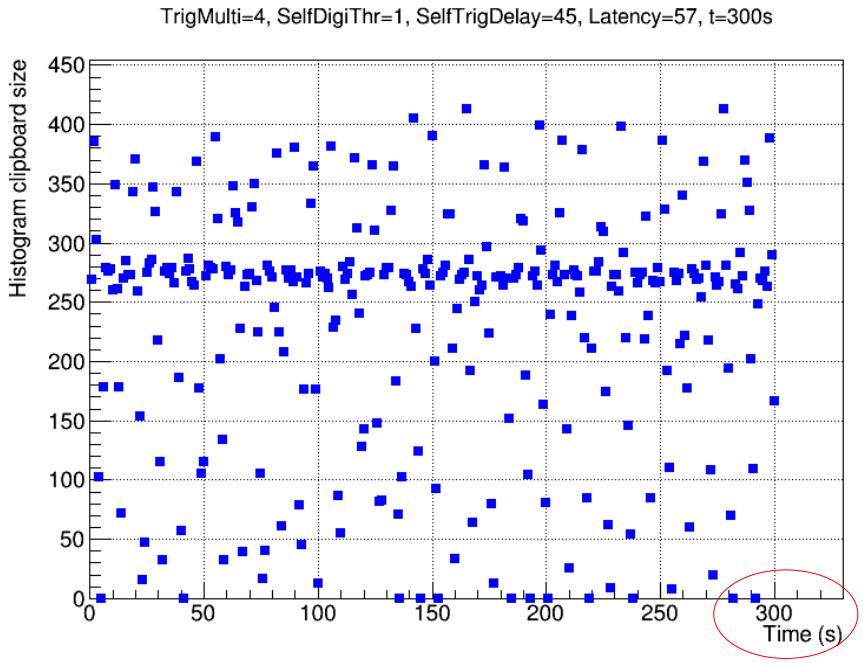
The fix

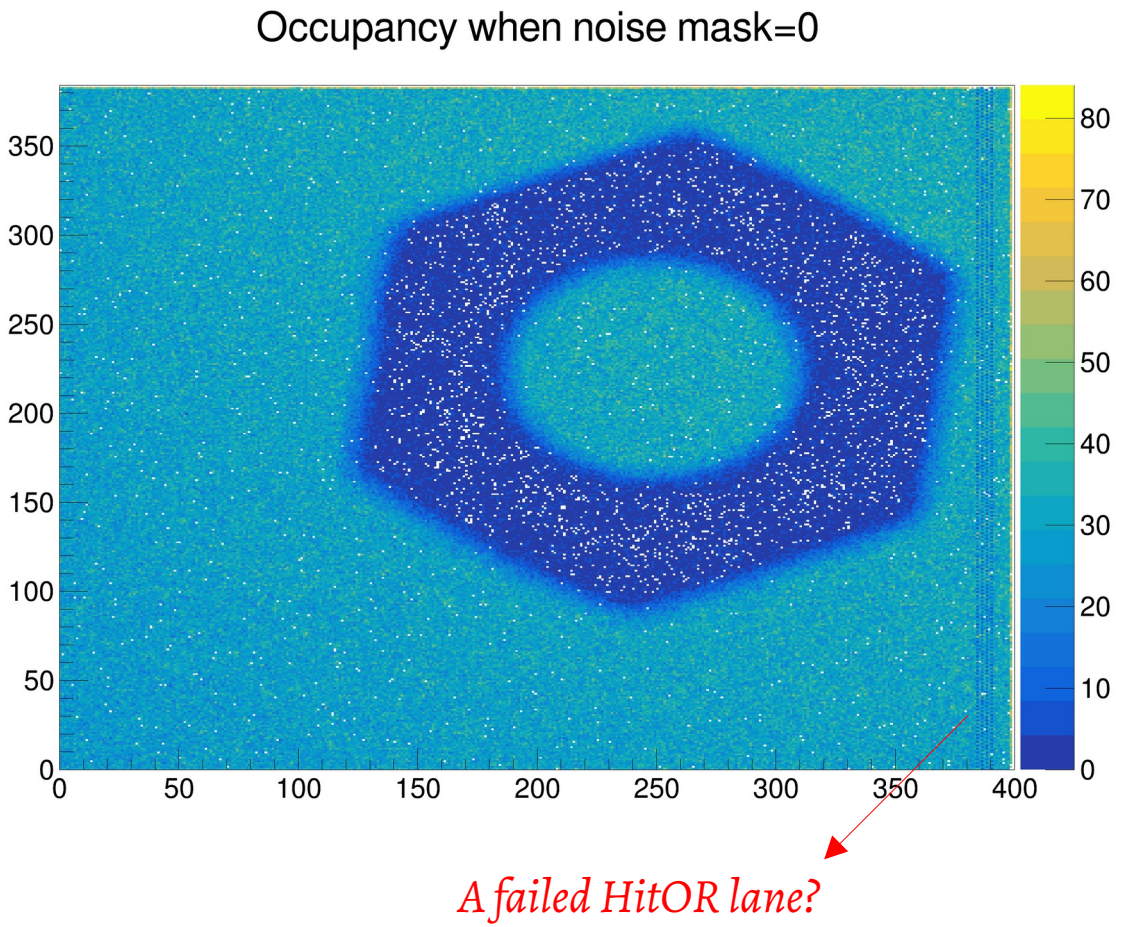
src/libUtil/Histo2d.cpp

```
@@ -132,11 +132,12 @@ void Histo2d::add(const Histo2d
132 132     if (this->size() != h.size())
133 133         return;
134 134     for (unsigned int i=0; i<(xbins*ybins); i++) {
    135 +       if (h.isFilled(i)){
135 136         double d = h.getBin(i);
136 137         data[i] += d;
137 138         max = std::max(d, max);
138     -       if (h.isFilled(i))
139     -           m_isFilled[i] = true;
    139 +       m_isFilled[i] = true;
    140 +     }
140 141     }
```

Merge request

- SelfTrigDelay = 45, Latency = 57
- SelfTrigDigiThr = 1, SelfTrigMulti = 4
- Time = 300s

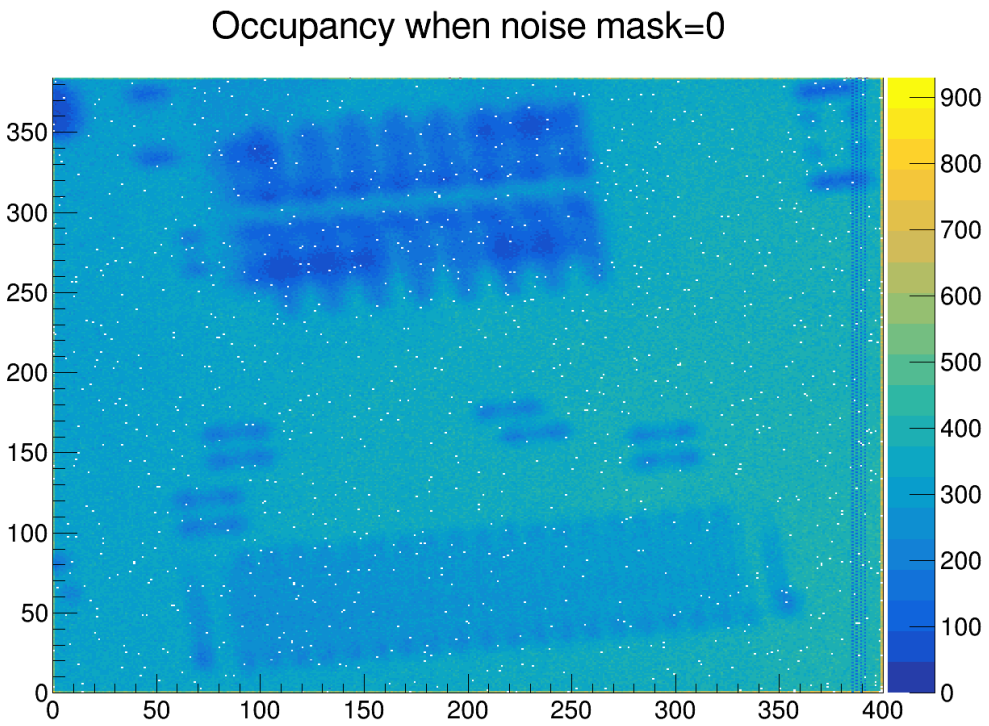




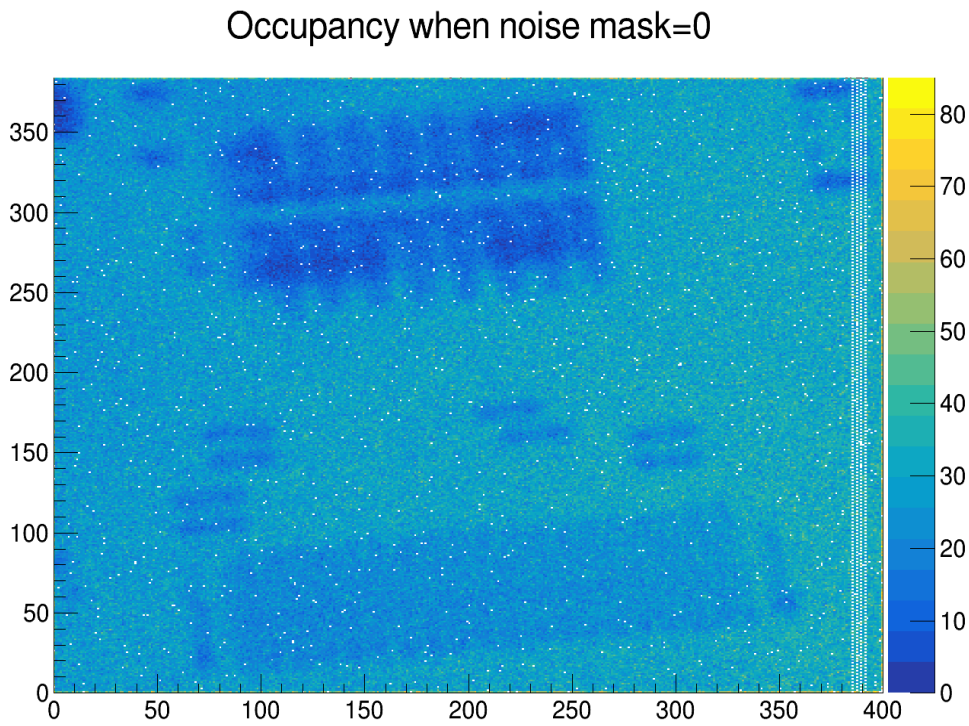
A failed HitOR lane?

Source scan runs without excessive memory consumption now!

Source Scan: Stuck HitOR lane



- Now with a flex PCB on top of chip
- Scan time = 1 hr
- HitOR Lane4 in one core column is not working.



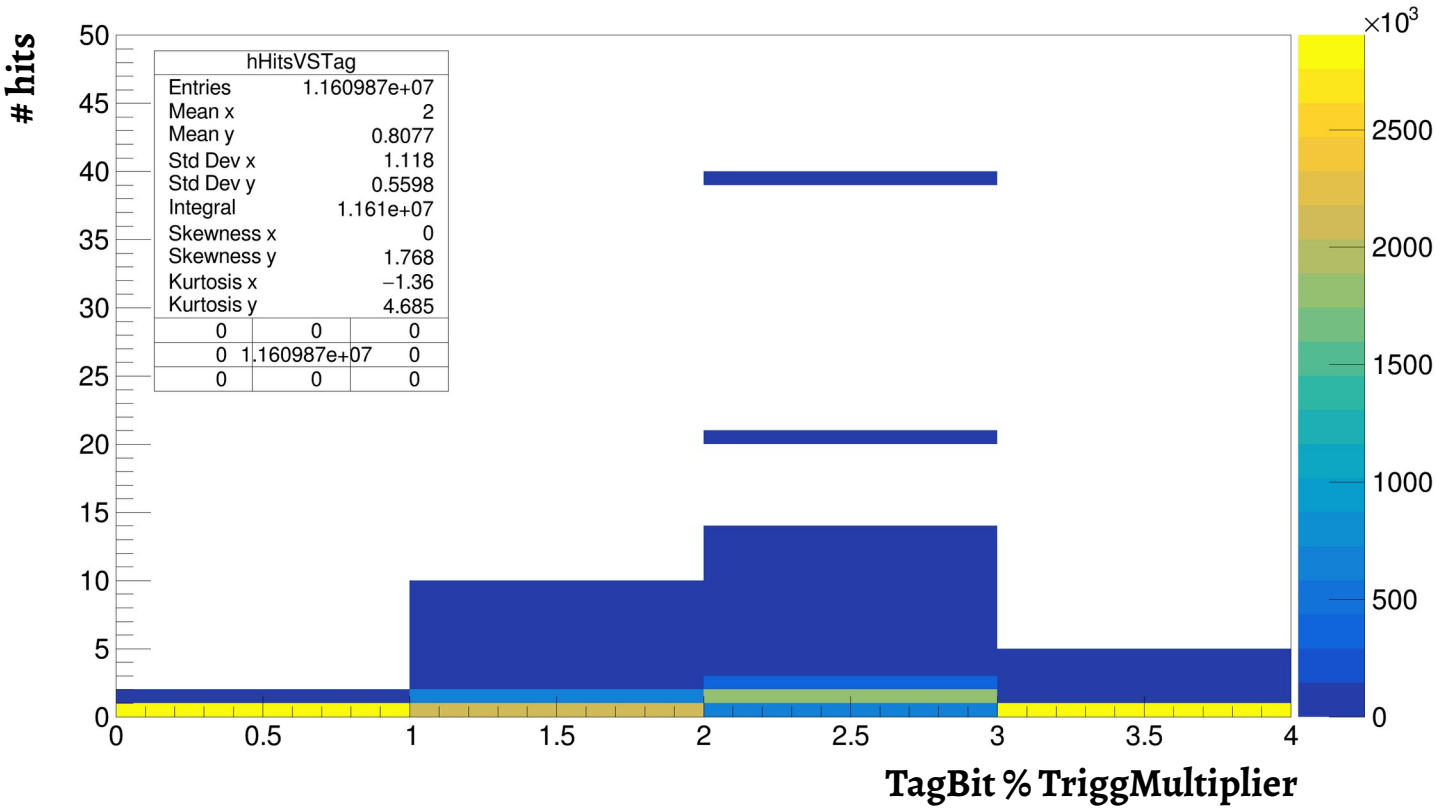
- With pToT analog scan masking – entire column is masked.
- ToT = 0 for all the pixels connected to Lane 4 in that core column. Test a different chip!
- Scan time = 300s

Conclusion

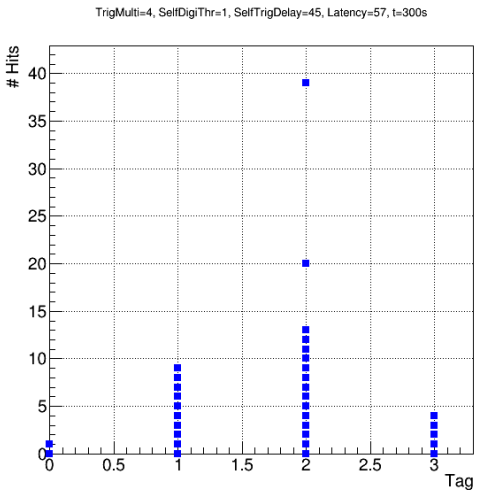
- Self-triggering logic with optimized value of parameters in place.
- Self-trigger analog scan also works for all the pixel masks, after the exception for the allowed occupancy range.
- Source scan implementation had two major issues:
  - Memory consumption: fixed.
  - Failing HitOR lane: ongoing *(perhaps a problem with the chip itself)*.
- Also studying the attenuation from flex PCB (material on top of chip).

Extra

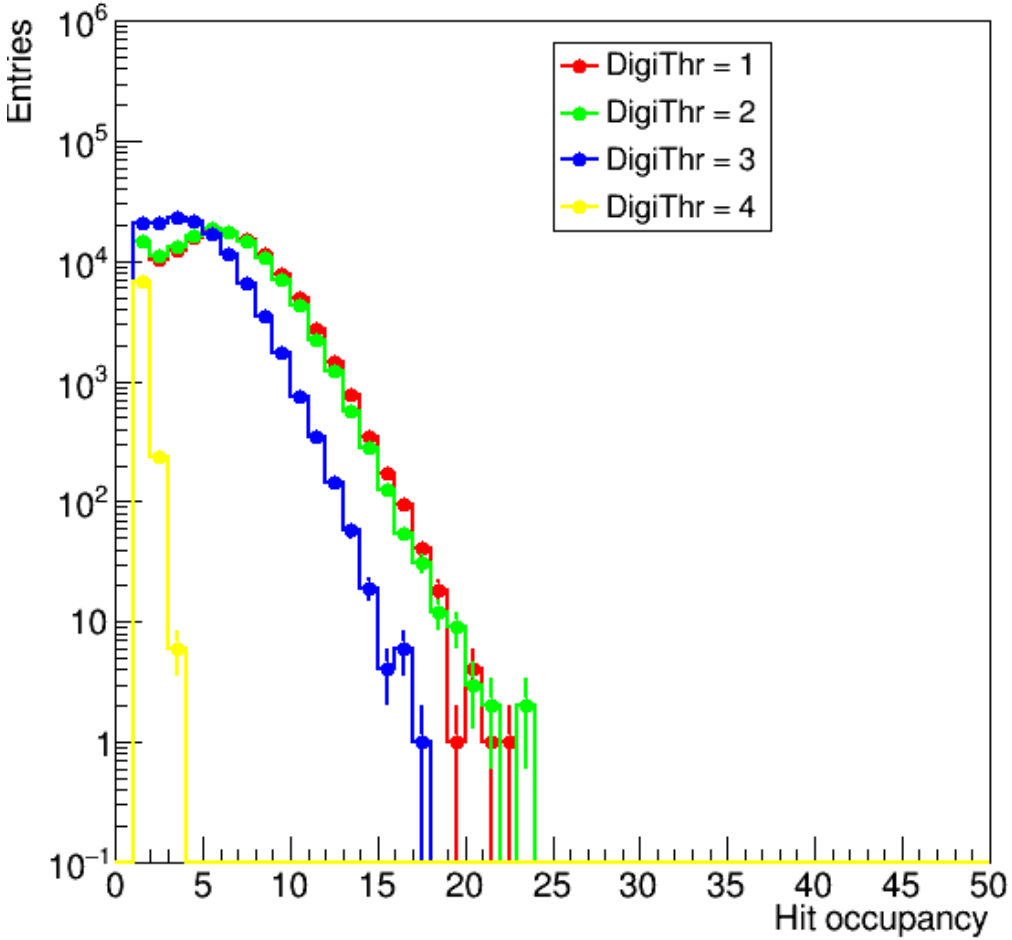
Hit occupancy vs Tags

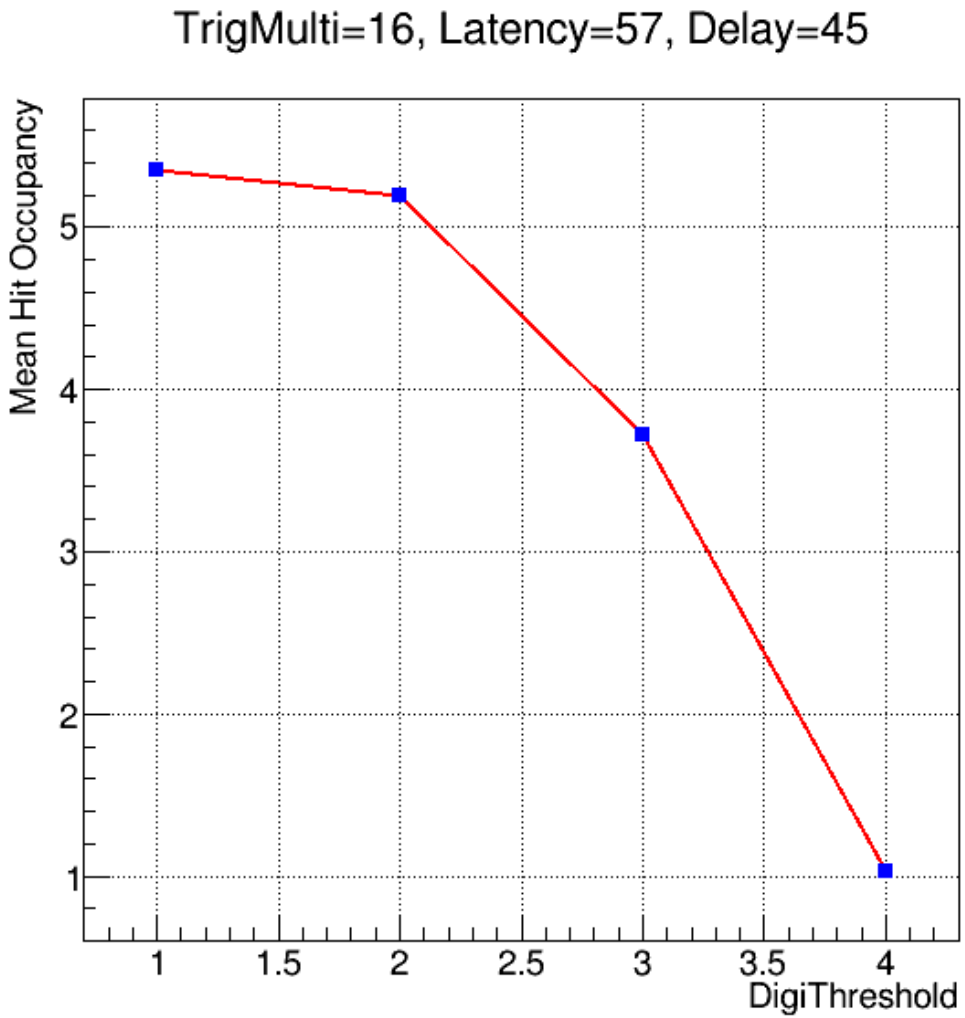


- 16 tag bits (216-231), TriggMultiplier = 4, SelfDigiThr=1
- Hit arrival at one of the triggers, corresponds to (Tag Bit % TriggMultiplier).
- Most of the hits arrive at the second or third tag.
- Also checked with higher trigger multiplier, hits arrive at the second or third last tag.



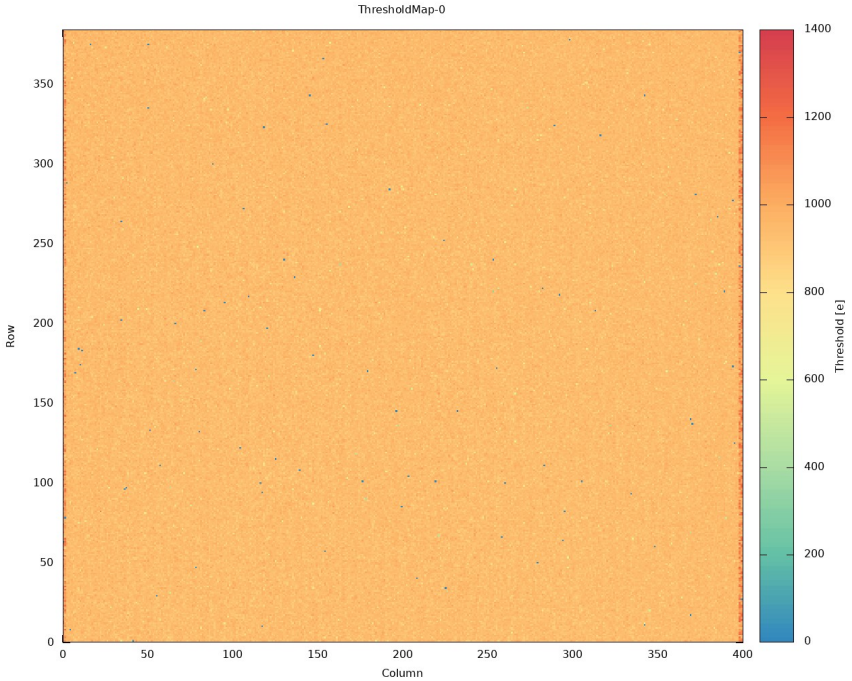
Hit occupancy vs SelfDigiThr

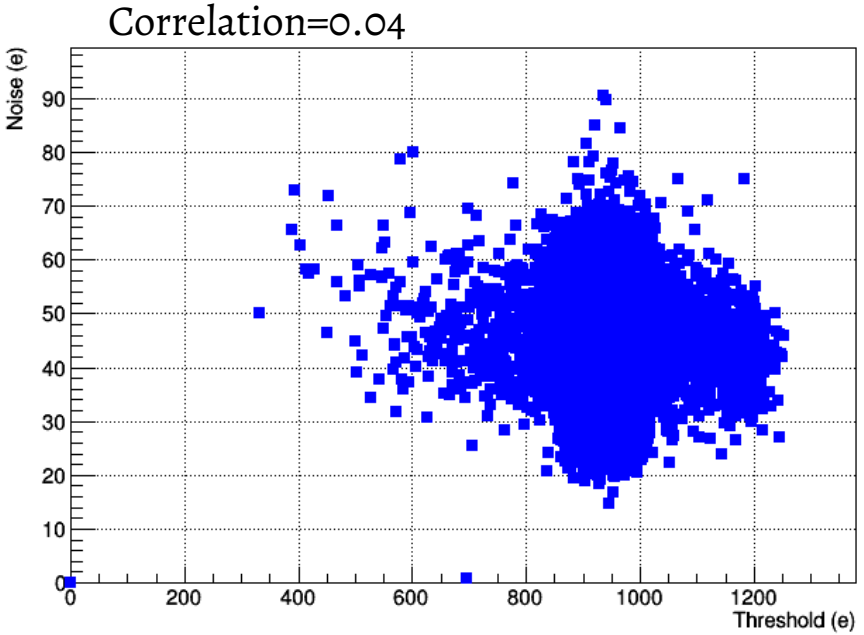


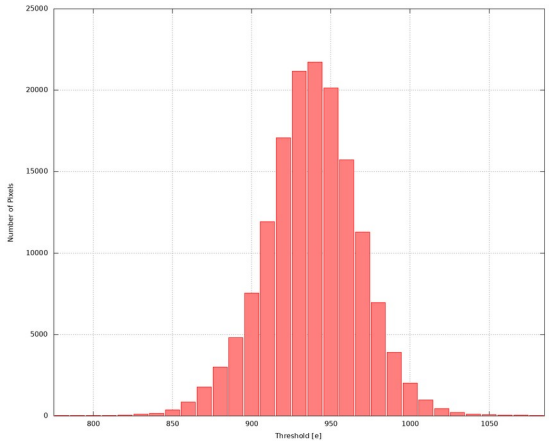


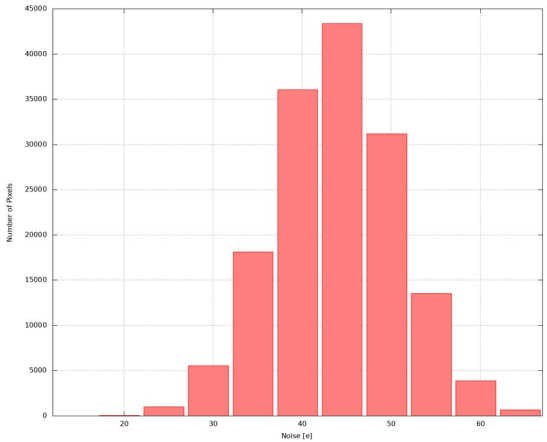
Mean of hit occupancy decreases with increasing DigiThr (as expected).

Threshold scan (bare chip)









Threshold scan (sensor+chip)