

Single Event Upset Monitoring

Tamas Almos VAMI

Wigner RCP IPNP, Hungarian Academy of Sciences, Budapest

Pixel offline meeting, 12.06.2014.



Single Event Upsets (SEUs)

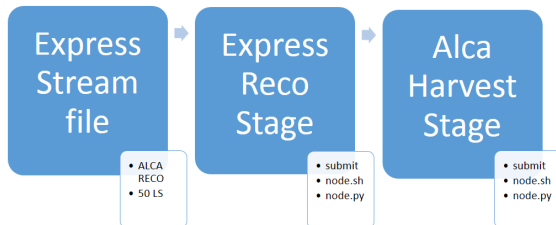
- Induced due to ionizing radiation
- Radiation can flip the state of a logic cell
 - Temporary malfunction in a ROC
 - Decrease the performance of the Pixel Detector
- Solution: Reprogramming

SiPixelMonitoringTools

- Developed by Luigi Calligaris
- Recognize the temporary or permanent bad ROCs
- Pass this information to a text file



- ALCARECO (Alignment Calibration RECO)
- Clusters in Pixel Detector
- Longer than 50 LumiSections



Structure of the software

ExpressRecoStage

- `expressRecoStage_lxbatch_submit_oneRunPerJob.py`
- `expressRecoStage_lxbatch_node.sh`
- `expressRecoStage_node.py`

AlcaHarvestStage

- `alcaHarvestStage_naf_submit.py`
- `alcaHarvestStage_naf_node.sh`
- `alcaHarvestStage_node.py`



The main program

- ① SEUDetectorClusterCounter
Takes a collection of pixel clusters from the events.
- ② SEUDetectorSEUFlaggerForSingleLumis
Checks if the average of the occupancies is big enough. Flags those ROCs that are under the average occupancy in that region.
- ③ SEUDetectorTimeFilter
Checks the length in time of the SEU candidates.
- ④ SEUDetectorLumiBlockIntervalsMerger
Merges time intervals when they are "close enough".
- ⑤ SEUDetectorMonitor
Fetches for the corresponding ROCs their counts and makes plots.



Implementation

```
git cms-addpkg Alignment/CommonAlignmentAlgorithm
git cms-addpkg Alignment/CommonAlignmentProducer
svn co https://svn.cern.ch/repos/cmsspxmon/trunk ./
```

Changes to be done:

- in: `expressRecoStage_lxbatch_submit_oneRunPerJob.py`
"`{: 0 > 9}`" $\rightarrow$ `{0}`
- in: `alcaHarvestStage_naf_submit.py`
`qsub` was used $\rightarrow$ `bsub`
- everywhere where `SEUDetectorCounterForAllClusters` $\rightarrow$
`SEUDetectorClusterCounterForAllClusters`



Current Status

Run on the followings:

- StreamExpress/Run2012A-TkAlMinBias-v1/ALCARECO
- StreamExpress/Run2012B-TkAlMinBias-v1/ALCARECO
- StreamExpress/Run2012C-TkAlMinBias-v1/ALCARECO
- StreamExpress/Run2012D-TkAlMinBias-v1/ALCARECO

Plots

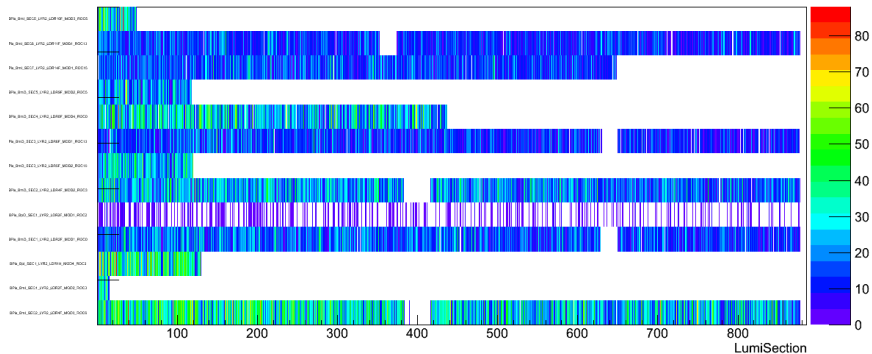
- 1 Module vs. Lumisection
- 2 ROC map
- 3 At least one SEU map



Plots

Module vs. Lumisection

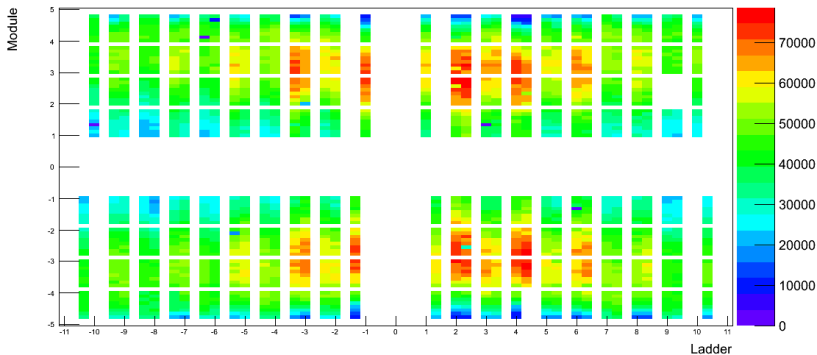
Occupancy for ROCs undergoing single event upsets in BPix Layer2 for run 207273



Plots

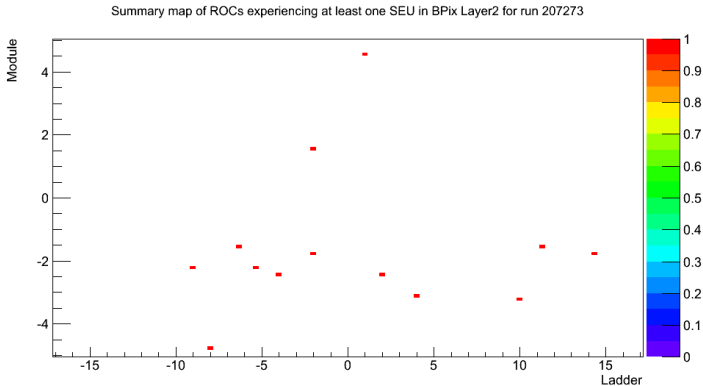
ROC map

Summary map for ROC cluster occupancy in BPix Layer1 for run 207273



Plots

At least one SEU map



```
# [run] [detId] [ROCNum] [RocNameString] Type(=dead or SEU)
detId302188824 ROCNum00 BPix_BpI_SEC1_LYR3_LDR3F_MOD2_ROC0
dead
```

```
detId302188824 ROCNum01 BPix_BpI_SEC1_LYR3_LDR3F_MOD2_ROC1
dead
```

```
...
```

```
detId302058272 ROCNum09 BPix_Bp0_SEC5_LYR1_LDR6F_MOD4_ROC9
seu lumiBlockBegin00043 lumiBlockEnd00878
```

```
detId302196512 ROCNum15 BPix_BpI_SEC7_LYR3_LDR17F_MOD4_ROC15
seu lumiBlockBegin00750 lumiBlockEnd00877
```



Comperission to SEUs from efficiency studies

SEUs from efficiency studies

- Precise method
- Takes longer time
- Janos created a SEU list (SEUs per fill)

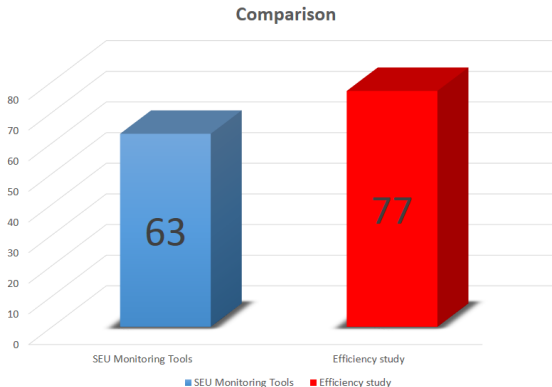
Comperission

- Script which makes ROC strings from Janos's list (where were only vector indexes)
- Select those runs which are in the given fill
- Compare the number of them



Results

- SiPixel Monitoring Tools find 63 SEUs
- Efficiency Studies finds 77 SEUs



Conclusions

- SiPixel Monitoring Tools is a general method to find SEUs and dead ROCs
- It finds less SEU then Efficiency Studies, but it is faster

Plans

- Improve the efficiency of the SEU finding process
- Implement the automatic recognition of the SEUs in the Prompt Calibration Loop

