SiPixelDigitizer Changes (module_killing_conf) 2nd Iteration

Jamila Bashir Butt
University of Nebraska Lincoln

SimTracker/SiPixelDigitizer/

- Problem:
  - To kill same modules using module_killing_conf as done by module_killing_DB.
- module_killing_conf: Current
  - Kills module
    - Whole module if parameter module = "whole"
    - Half module with rows > 80 if parameter module = "tbmA"
    - Half module with rows < 80 if parameter module = "tbmB"
  - Note:
    - For "tbmA" rows < 80 if along -Z and for "tbmB" rows > 80 if along –Z
    - For "tbmA" rows > 80 if along +Z and for "tbmB" rows < 80 if along +Z
  - The "tbmA" and "tbmB" module rule need to be modified for –Z and +Z
  - The module_killing_confcan not kill individual bad ROCs
  - Need to extend the rules for endcap
- Last Update
  - https://indico.cern.ch/conferenceDisplay.py?confId=148989

Reminder of Changes

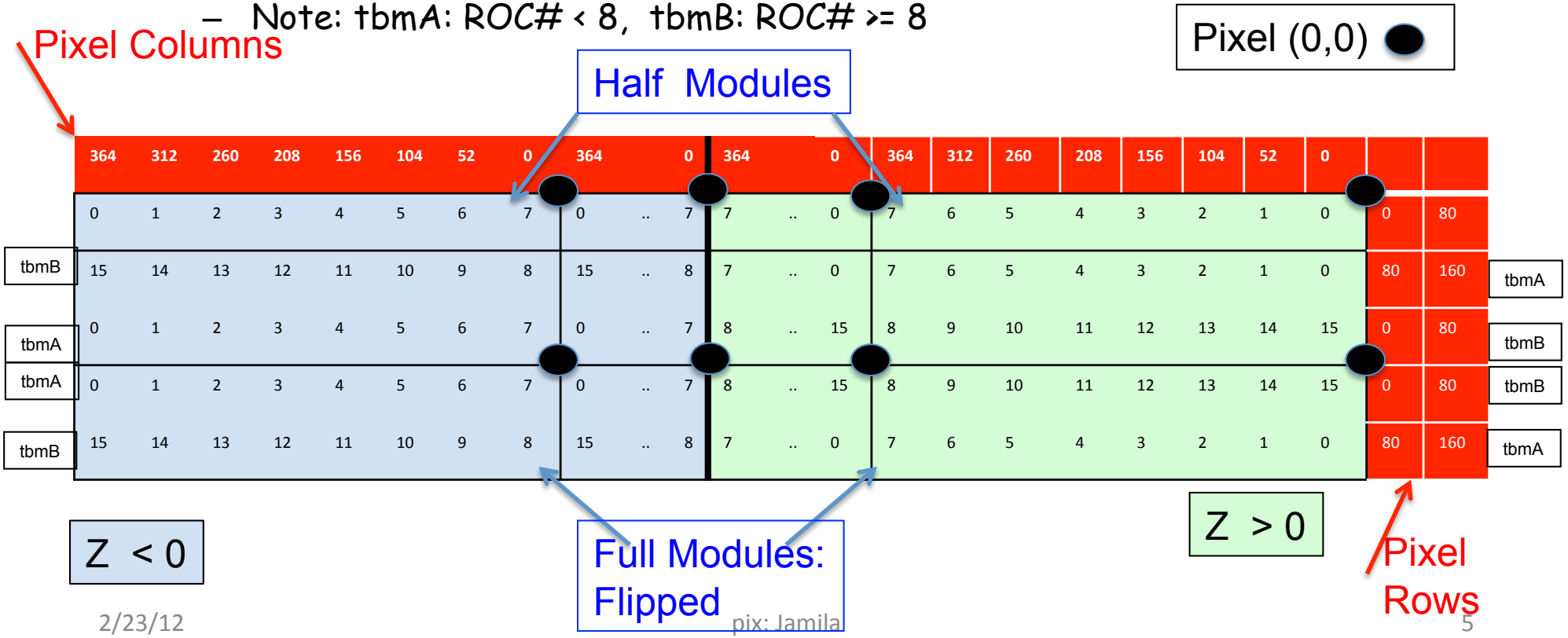
- Introduced a new parameter in SimTracker/SiPixelDigitizer/python/PixelDigi_cfi.py
  - Dead_RocID:
    - Value is "-1" when all ROCs are bad in a module/tbmA/tbmB
    - Value is the bad roc number in other case
  - For now code considers Dead_RocID as a single integer i.e cms.int32
    - We can use only one bad ROC in a module
  - It should be easy to change cms.int32 to vector (cms.vint32) in case we start to see more than one bad ROC in a module etc.
- Change the function module_killing_conf in code SimTracker/SiPixelDigitizer/src/SiPixelDigitizerAlgorithm.cc
  - Incorporate the new parameter Dead_RocID.
  - Modify the rule for tbmA/tbmB if the are on +Z or –Z side
  - Kill module differently depending if the are in Barrel or endcap
  - The killing follows the geometry of Pixel Barrel and Pixel Endcap
  - Details on next slide
  - Code Snapshot in BACKUP

Changes Continued

- Additional variables introduced to access detector Geometry:
  - DetId detId = DetId(detid); // Get the Detid object
  - //unsigned int detType=detId.det(); // det type, pixel=1
  - unsigned int subid=detId.subdetId(); //subdetector type, barrel=1, forward=2
  - PXFDetId pdetIdf = PXFDetId(detid);
  - unsigned int disk=pdetIdf.disk(); //1,2,3
  - unsigned int blade=pdetIdf.blade(); //1-24
  - int zindexF=pdetIdf.module(); //
  - unsigned int side=pdetIdf.side(); //size=1 for -z, 2 for +z
  - unsigned int panel=pdetIdf.panel(); //panel=1
  - PXBDetId pdetId = PXBDetId(detid); //
  - //unsigned int detTypeP=pdetId.det(); // pix det identification
  - //unsigned int subidP=pdetId.subdetId(); // sub det id
  - // Barell layer = 1,2,3
  - int layerC=pdetId.layer();
  - // Barrel ladder id 1-20,32,44.
  - int ladderC=pdetId.ladder();
  - // Barrel Z-index=1,8
  - int zindex=pdetId.module();

Barrel

- Start with Barrel mask subid=detId.subdetId(); //subid=1
  - All output -> 0 if module="whole"
  - For other modules where a part of module is dead use the geometry to kill that part
  - Part of geometry for Barrel Pixel is shown in the figure
  - Code follows the geometry as explained on next slide
  - Note: tbmA: ROC# < 8, tbmB: ROC# >= 8

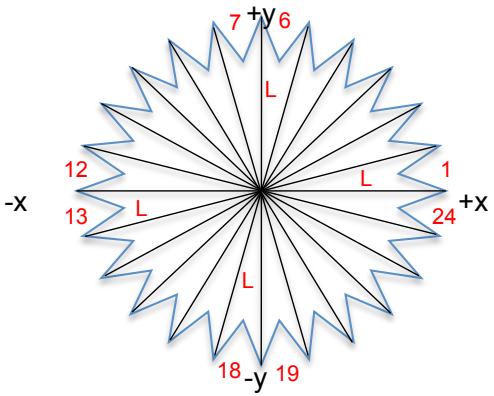


Barrel Continued

- Divide in two cases -Z zindex <= 4 and +Z zindex > 4
- For each case have
  - Half Modules (Z < 0)
  - Taking quartLdr=5,8,11 for layer 1,2,3 respectively)
  - Half Modules located at quartLdr, quartLdr+1, 3*quartLdr, 3*quartLdr+1
    - Module = "none"
    - if Dead_RocID=-1 all output -> 0; Otherwise loop on all ROC# (0-7);
    - for the case when loop counting match bad ROC# set the output->0
    - Correct column range is obtained by 416 -52*ROC#
  - Full Modules (Z < 0)
    - Module ="tbmA", "tbmB" or "none"
    - If Dead_RocID=-1 all output -> 0; Otherwise loop on all ROC# (0-7) for Rows< 80;
    - for the case when loop counting match bad ROC# (column # is in correct range) set the output->0
    - If Rows >= 80 ROC# >=8 so replace ROC# by (ROC#%8) to get counting from 0-7 and loop on all Rocs.
    - Get correct column range by 416 – 52*ROC#
  - Z > 0:
    - Half modules: Correct column range obtained by 52* ROC#
    - Full modules: Logic of Z < 0, Rows < 80 is replaced by that of Rows >=80 and vice versa

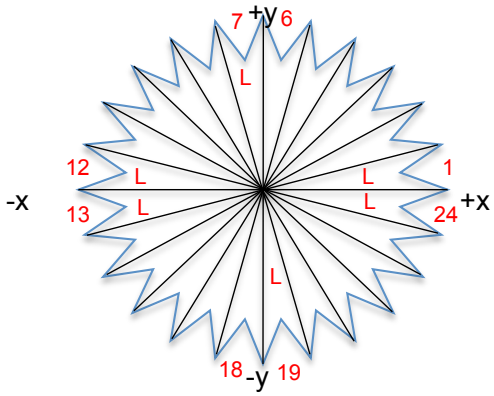
Endcap Geometry

- Start with Endcap mask subid=detId.subdetId(); //subid=2
  - All output -> 0 if module="whole"
    - Position of Left/Right panels are shown for Disk1, Disk2 (Z < 0)
    - Positions of Left/Right panels is swapped for Z > 0 D1 w.r.t Z < 0 D1 when looking towards IP (Similar for D2)

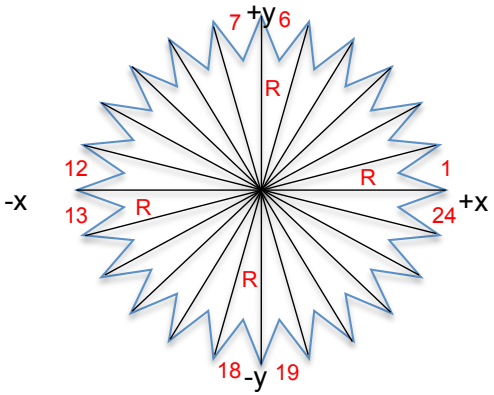


4Minus D1/D2

Looking from IP

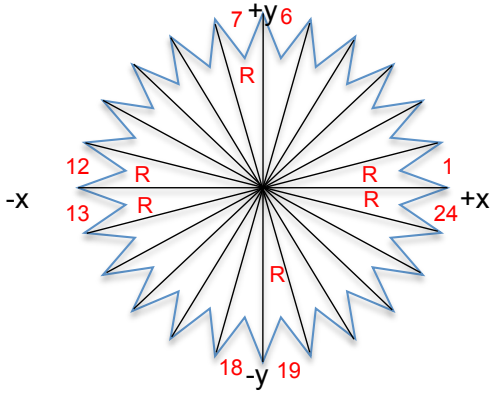


3Minus D1/D2



4Plus D1/D2

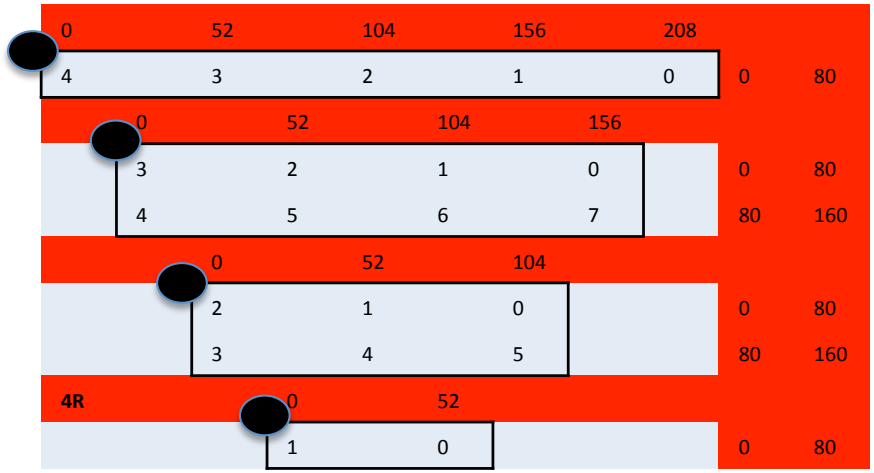
Looking towards IP

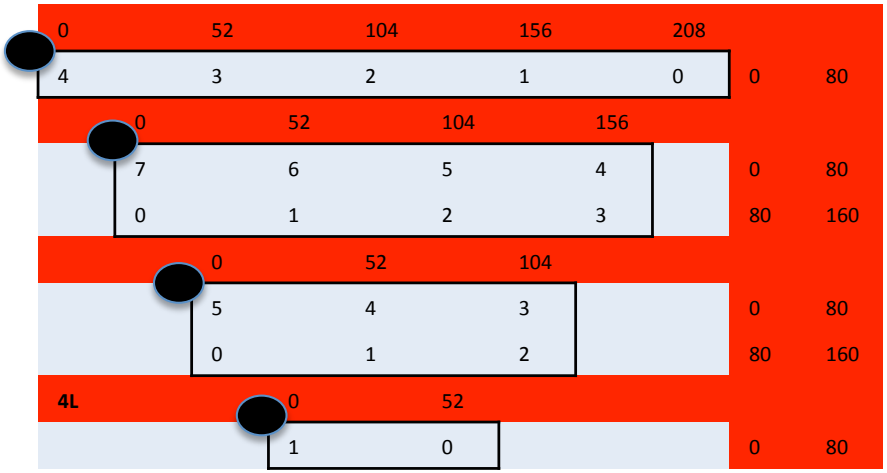


3Plus D1/D2

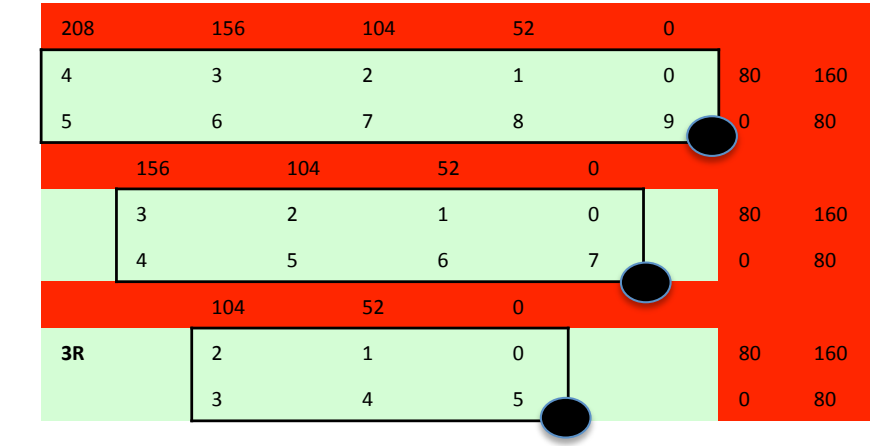
Panel 3L, 3R, 4L, 4R

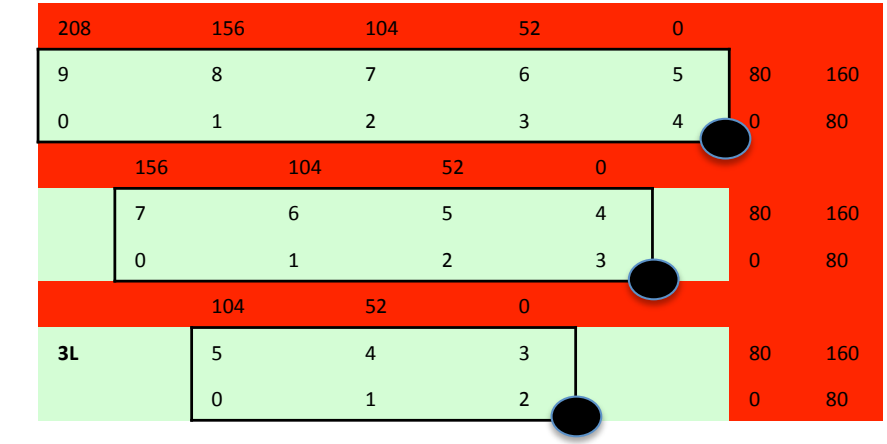
- Panels with 4 plaquettes have Pixel(0, 0) located at upper left corners and those with 3 have Pixel (0,0) at their lower right corners
- For left type panels the ROC 0 is at bottom left and for right type panels ROC 0 is at top right corner





Pixel (0,0)





Endcap

- Consider cases: Panel-1 (4: 1X2, 2X3, 2X4, 1X5) and Panel-2 (3: 2X3, 2X4, 2X5) and module ="none"
- Panel-1 (4 plaquette): skip if mod is not same as zindexF
  - Z < 0: Blade 1,6,13,18 are "Left" all other are "Right"
  - Z > 0: Blade 1,6,13,18 are "Right" all other are "Left"
  - "Left" Rows < 80: Correct column range is obtained by 52*(M-ROC#)
  - "Left" Rows >=80: Correct column range is obtained by 52*ROC#
  - "Left" Rows >= 80: ROC# < M but "Right" Rows >= 80: ROC# > M
- Panel-2(3 plaquette): skip if mod is not same as zindexF
  - Z < 0: Blade 1,7,12,13,19,24 are "Left" all other are "Right"
  - Z > 0: Blade 1,7,12,13,19,24 are "Right" all other are "Left"
  - "Left" Rows < 80: Correct column range is obtained by 52*(M-ROC#)
  - "Left" Rows >=80: Correct column range is obtained by 52*ROC#
  - "Left" Rows >= 80: ROC# > M but "Right" Rows >= 80: ROC# < M

mod: loop over modules in the panel

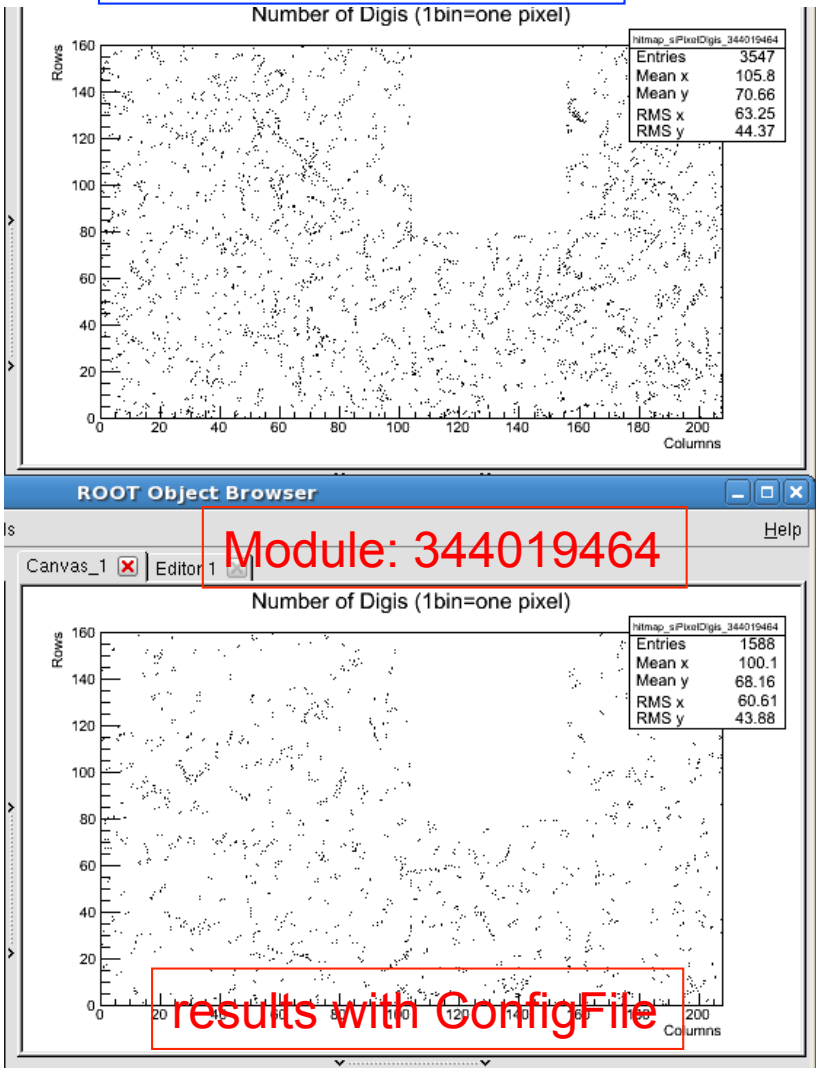
If ROC# > M; Use ROC#%M to loop on ROCS from 0-M
Loop over M where M:2,3,4,5 for Panel1 and M:3,4,5 for Panel2

Note: M= 3,4,5 in 2X3, 2X4, 2X5 etc

Comparison Plots for Bad ROCs: DB Vs Config

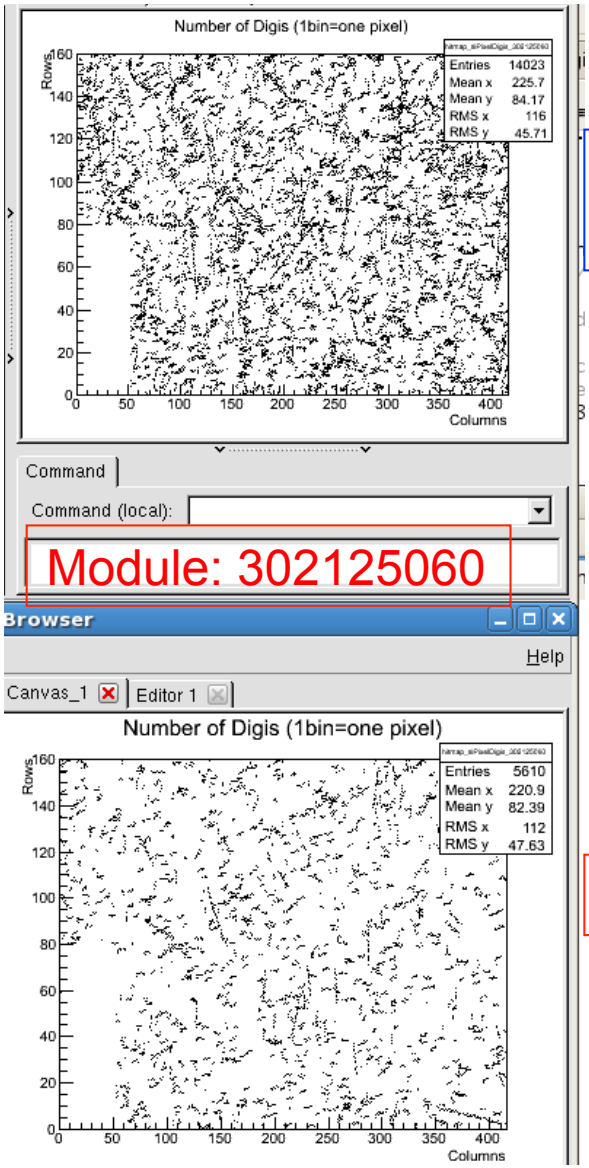
- Comparison of results From DB and conf
  - For Muons, QCD and ttbar
    - cms.PSet(Dead_detID = cms.int32(302055940), Module = cms.string("tbmB") , Dead_RocID = cms.int32(-1)) ## -Z: Rows >= 80 // BadRocs=ff00 ROC= 8 9 10 11 12 13 14 15
    - cms.PSet(Dead_detID = cms.int32(302125060), Module = cms.string("tbmA") , Dead_RocID = cms.int32(7)) ## -Z: Rows < 80 //BadRocs=0080 (col 0-51) ROC= 7
    - ,cms.PSet(Dead_detID = cms.int32(302125076), Module = cms.string("tbmA") , Dead_RocID = cms.int32(3)) ## +Z: Rows >= 80 //BadRocs=0008 (col 156-207) ROC= 3
    - ,cms.PSet(Dead_detID = cms.int32(302126364), Module = cms.string("tbmB") , Dead_RocID = cms.int32(-1)) ## +Z: Rows < 80 //BadRocs=ff00 ROC= 8 9 10 11 12 13 14 15
    - ,cms.PSet(Dead_detID = cms.int32(344019464), Module = cms.string("none") , Dead_RocID = cms.int32(6)) ## -Z: Rows >= 80 //BadRocs=40 (col 104-155) ROC= 6 Blade 11, panel 2, Module 2

Reference plots: results with DataBase



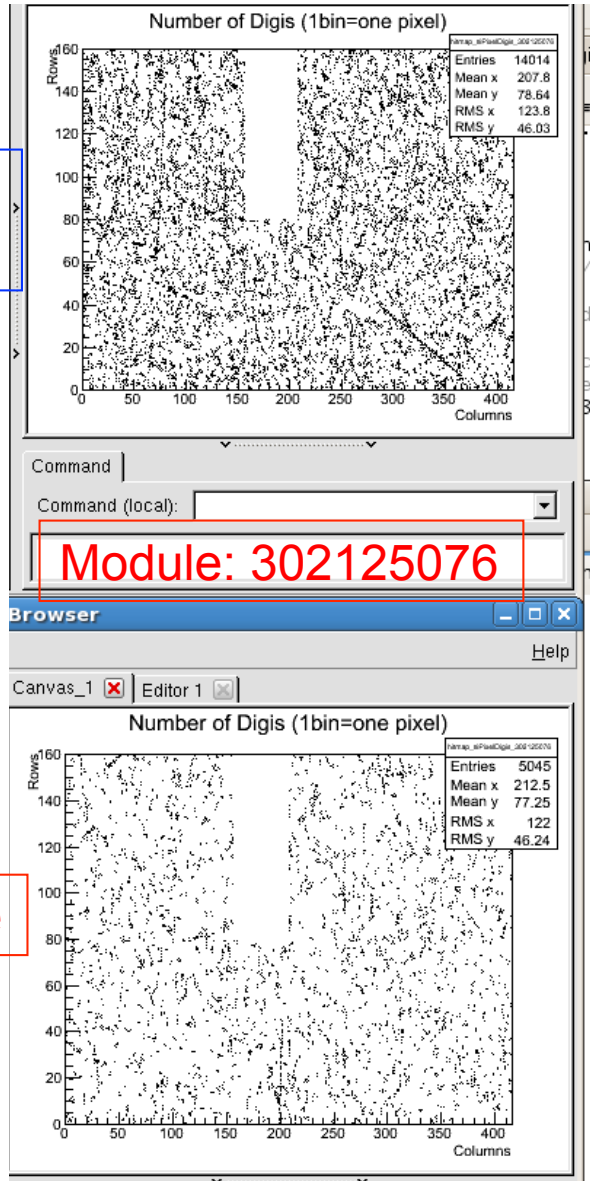
More Comparison plots

- Comparison of results From DB and conf



Reference plots:
results with DataBase

results with ConfigFile



Conclusion/Summary

- Config file reproduces the single dead modules as well as whole dead modules.
- Test done for modules also bad in DB
- The code can be changed to accommodate more than one bad coming from a single module (Not done yet).

- TODO:
- Need to test with more ROCs.
- Any way to add bad ROCs in DB and then use killing_module_DB and killing_module_conf to cross check

BackUP

- Source for Geometry information:
- Barrel+endcap:
  - https://hypernews.cern.ch/HyperNews/CMS/get/pixelOfflineSW/914/1.html

- Endcap
  - **CMS IN -2009/033**

Barr el Code

```
if(zindex <= 4){ //cout << " -Z ===="<<detid<<" "<<endl;
  if (ladderC==quartLdr || ladderC==quartLdr+1 || ladderC==3*quartLdr || ladderC==3*quartLdr+1){// half modules
    if (Module=="none" && ip.first <   80){
      if (Dead_RocID==-1)  {i->second.set(0.);}
      else {
        for(drId =0; drId < 8; drId++){
          if(Dead_RocID==drId && ip.second <= (416-52*drId) && ip.second > (416-52*(drId+1))){
            //cout <<detid<<" -123 "<<drId<<" (row, col)= ("<<ip.first<<","<<ip.second<<") "<<52*drId<<" (L, Ld)=
            i->second.set(0.);//cout<<detid<<"  "<<drId<<" (row,col)= ("<<ip.first<<","<<ip.second<<") "<<52*drId<
          }
        }
      }
    }
  }//end half modules Z<0
  else {// full modules
    if((Module=="tbmA"||Module=="none") && ip.first <   80){
      if (Dead_RocID==-1)  {i->second.set(0.);}
      else {
        for(drId =0; drId < 8; drId++){
          if(Dead_RocID==drId && ip.second <= (416-52*drId) && ip.second > (416-52*(drId+1))){
            i->second.set(0.);
          }
        }
      }
    }//end < 80
    else if((Module=="tbmB"||Module=="none") && ip.first >= 80){
      if (Dead_RocID==-1)  {i->second.set(0.);}
      else {Dead_RocID2=Dead_RocID%8;
        for(drId =0; drId < 8; drId++){
          if(Dead_RocID2==drId && ip.second >= 52*drId && ip.second < 52*(drId+1)){
            i->second.set(0.);
          }
        }
      }
    }//end rows>=80 Z<0
  }//end full module Z<0
}// end -Z
if(zindex > 4){ //cout << " +Z ===="<<detid<<" "<<endl;
  if (ladderC==quartLdr || ladderC==quartLdr+1 || ladderC==3*quartLdr || ladderC==3*quartLdr+1){// half modules
    if (Module=="none" && ip.first <   80){
      if (Dead_RocID==-1)  {i->second.set(0.);}
      else {
        for(drId =0; drId < 8; drId++){
          if(Dead_RocID==drId && ip.second >= 52*drId && ip.second < 52*(drId+1)){
            i->second.set(0.);
          }
        }
      }
    }
  }//end half modules Z>0
  else {// full modules
    if((Module=="tbmB"||Module=="none") && ip.first <   80){
      if (Dead_RocID==-1)  {i->second.set(0.);}
      else { Dead_RocID2=Dead_RocID%8;
        for(drId =0; drId < 8; drId++){
          if(Dead_RocID2==drId && ip.second <= (416-52*drId) && ip.second > (416-52*(drId+1))){
            i->second.set(0.);
          }
        }
      }
    }//end rows<80 Z>0
    else if((Module=="tbmA"||Module=="none") && ip.first >= 80){
      if (Dead_RocID==-1)  {i->second.set(0.);}
      else {
        for(drId =0; drId < 8; drId++){
          if(Dead_RocID==drId && ip.second >= 52*drId && ip.second < 52*(drId+1)){
            i->second.set(0.);
```

half

full

half

full

Z<0

Z>0

Endcap: Z < 0, Panel1

```
else if (subid==2){   //Endcap Mask
  if(side==1 && (disk==1 || disk==2)) {//-zD1 -zD2
    if(panel==1){//Panel 1 with 4 plaquettes
      if (blade==1 || blade==6 || blade==13 || blade==18){//Left Panels
        for (int modRocs=2; modRocs <6; modRocs++){//modRocs=1X(2,3,4,5)
          if(zindexF != modRocs-1) continue;
          if((Module=="none") && ip.first < 80){
            if (Dead_RocID==-1)  {i->second.set(0.);}
            else{Dead_RocID2=Dead_RocID%modRocs;
              for(drId =0; drId < modRocs; drId++){
                if(Dead_RocID2==drId && ip.second <= 52*(modRocs-drId) && ip.second > 52*(modRocs-(drId+1))){
                  i->second.set(0.);
                }
              }
            }
          }// <80
          else if((Module=="none") && ip.first >= 80 && (modRocs==3 || modRocs==4)){//modRocs=2X(3,4)
            if (Dead_RocID==-1)  {i->second.set(0.);}
            else{
              for(drId =0; drId < modRocs; drId++){
                if(Dead_RocID==drId && ip.second >= 52*drId && ip.second < 52*(drId+1)){
                  i->second.set(0.);
                }
              }
            }
          } // >=80
        }//loop on mods
      }// PL4 ends
      else{//Right panels
        for (int modRocs=2; modRocs <6; modRocs++){//modRocs=1X(2,3,4,5)
          if(zindexF != modRocs-1) continue;
          if((Module=="none") && ip.first < 80){
            if (Dead_RocID==-1)  {i->second.set(0.);}
            else{Dead_RocID2=Dead_RocID%modRocs;
              for(drId =0; drId < modRocs; drId++){
                if(Dead_RocID2==drId && ip.second <= 52*(modRocs-drId) && ip.second > 52*(modRocs-(drId+1))){
                  i->second.set(0.);
                }
              }
            }
          }// <80
          else if((Module=="none") && ip.first >= 80 && (modRocs==3 || modRocs==4)){//modRocs=2X(3,4)
            if (Dead_RocID==-1)  {i->second.set(0.);}
            else{
              for(drId =0; drId < modRocs; drId++){
                if(Dead_RocID==drId && ip.second >= 52*drId && ip.second < 52*(drId+1)){
                  i->second.set(0.);
                }
              }
            }
          } // >=80
        }//loop on mods
      }// PR4 ends
    }// panel 1 ends
    if(panel==2){//Panel 1 with 3 plaquettes
```

Left

Right

```
if(panel==2){//Panel 1 with 3 plaquettes
  if (blade==1 || blade==7 || blade==12 || blade==13 || blade==19 || blade==24){//Left Panels
    for (int modRocs=3; modRocs <6; modRocs++){//modRocs=2X(3,4,5)
      if(zindexF != modRocs-2) continue;
      if((Module=="none") && ip.first < 80){
        if (Dead_RocID==-1)  {i->second.set(0.);}
        else{Dead_RocID2=Dead_RocID%modRocs;
          for(drId =0; drId < modRocs; drId++){
            if(Dead_RocID2==drId && ip.second <= 52*(modRocs-drId) && ip.second > 52*(modRocs-(drId+1))){
              i->second.set(0.);
            }
          }
        }
      }// <80
      else if((Module=="none") && ip.first >= 80){
        if (Dead_RocID==-1)  {i->second.set(0.);}
        else{
          for(drId =0; drId < modRocs; drId++){
            if(Dead_RocID==drId && ip.second >= 52*drId && ip.second < 52*(drId+1)){
              i->second.set(0.);
            }
          }
        }
      } // >=80
    }//loop on mods
  }// PL3 ends
  else{//Right panels
    for (int modRocs=3; modRocs <6; modRocs++){//modRocs=2X(3,4,5)
      if(zindexF != modRocs-2) continue;
      if((Module=="none") && ip.first < 80){
        if (Dead_RocID==-1)  {i->second.set(0.);}
        else{
          for(drId =0; drId < modRocs; drId++){
            if(Dead_RocID==drId && ip.second <= 52*(modRocs-drId) && ip.second > 52*(modRocs-(drId+1))){
              i->second.set(0.);
            }
          }
        }
      } // <80
      else if((Module=="none") && ip.first >= 80){
        if (Dead_RocID==-1)  {i->second.set(0.);}
        else{Dead_RocID2=Dead_RocID%modRocs;
          for(drId =0; drId < modRocs; drId++){
            if(Dead_RocID2==drId && ip.second >= 52*drId && ip.second < 52*(drId+1)){
              i->second.set(0.);
            }
          }
        }
      } // >=80
    }//loop on mods
  }// PR3 ends
}// panel 2 ends
}//Side (1,1) and (1,2) ends
else if(side==2 && (disk==1 || disk==2)) {//+zD1 +zD2
```

Left

Right

Endcap: Z < 0, Panel2

Endcap: Z > 0, Panel1

```
else if(side==2 && (disk==1 || disk==2)) {//+zD1 +zD2
  if(panel==1){//Panel 1 with 4 plaquettes
    if (blade==1 || blade==6 || blade==13 || blade==18){//Right Panels
      for (int modRocs=2; modRocs <6; modRocs++){//modRocs=1X(2,3,4,5)
        if(zindexF != modRocs-1) continue;
        if((Module=="none") && ip.first < 80){
          if (Dead_RocID==-1)  {i->second.set(0.);}
          else{Dead_RocID2=Dead_RocID%modRocs;
            for(drId =0; drId < modRocs; drId++){
              if(Dead_RocID2==drId && ip.second <= 52*(modRocs-drId) && ip.second > 52*(modRocs-(drId+1))){
                i->second.set(0.);
              }
            }
          }
        }// <80
        else if((Module=="none") && ip.first >= 80 && (modRocs==3 || modRocs==4)){//modRocs=2X(3,4)
          if (Dead_RocID==-1)  {i->second.set(0.);}
          else{
            for(drId =0; drId < modRocs; drId++){
              if(Dead_RocID==drId && ip.second >= 52*drId && ip.second < 52*(drId+1)){
                i->second.set(0.);
              }
            }
          }
        } // >=80
      }//loop on mods
    }// PR4 ends
    else{//Left panels
      for (int modRocs=2; modRocs <6; modRocs++){//modRocs=1X(2,3,4,5)
        if(zindexF != modRocs-1) continue;
        if((Module=="none") && ip.first < 80){
          if (Dead_RocID==-1)  {i->second.set(0.);}
          else{Dead_RocID2=Dead_RocID%modRocs;
            for(drId =0; drId < modRocs; drId++){
              if(Dead_RocID2==drId && ip.second <= 52*(modRocs-drId) && ip.second > 52*(modRocs-(drId+1))){
                i->second.set(0.);
              }
            }
          } // <80
        }
        else if((Module=="none") && ip.first >= 80 && (modRocs==3 || modRocs==4)){//modRocs=2X(3,4)
          if (Dead_RocID==-1)  {i->second.set(0.);}
          else{
            for(drId =0; drId < modRocs; drId++){
              if(Dead_RocID==drId && ip.second >= 52*drId && ip.second < 52*(drId+1)){
                i->second.set(0.);
              }
            }
          }
        } // >=80
      }//loop on mods
    }// PL4 ends
  }// panel 1 ends
  if(panel==2){//Panel 2 with 3 plaquettes
```

Right

Left

```
if(panel==2){//Panel 2 with 3 plaquettes
  if (blade==1 || blade==7 || blade==12 || blade==13 || blade==19 || blade==24){//Right Panels
    for (int modRocs=3; modRocs <6; modRocs++){//modRocs=2X(3,4,5)
      if(zindexF != modRocs-2) continue;
      if((Module=="none") && ip.first < 80){
        if (Dead_RocID==-1)  {i->second.set(0.);}
        else{Dead_RocID2=Dead_RocID%modRocs;
          for(drId =0; drId < modRocs; drId++){
            if(Dead_RocID2==drId && ip.second <= 52*(modRocs-drId) && ip.second > 52*(modRocs-(drId+1))){
              i->second.set(0.);
            }
          }
        }
      }// <80
      else if((Module=="none") && ip.first >= 80){
        if (Dead_RocID==-1)  {i->second.set(0.);}
        else{
          for(drId =0; drId < modRocs; drId++){
            if(Dead_RocID==drId && ip.second >= 52*drId && ip.second < 52*(drId+1)){
              i->second.set(0.);
            }
          }
        }
      } // >=80
    }//loop on mods
  }// PR3 ends
  else{//Left panels
    for (int modRocs=3; modRocs <6; modRocs++){//modRocs=2X(3,4,5)
      if(zindexF != modRocs-2) continue;
      if((Module=="none") && ip.first < 80){
        if (Dead_RocID==-1)  {i->second.set(0.);}
        else{
          for(drId =0; drId < modRocs; drId++){
            if(Dead_RocID==drId && ip.second >= 52*drId && ip.second < 52*(drId+1)){
              i->second.set(0.);
            }
          }
        } // <80
      }
      else if((Module=="none") && ip.first >= 80){
        if (Dead_RocID==-1)  {i->second.set(0.);}
        else{Dead_RocID2=Dead_RocID%modRocs;
          for(drId =0; drId < modRocs; drId++){
            if(Dead_RocID2==drId && ip.second <= 52*(modRocs-drId) && ip.second > 52*(modRocs-(drId+1))){
              i->second.set(0.);
            }
          }
        }
      } // >=80
    }//loop on mods
  }// PL3 ends
}// panel 2 ends
}
```

Right

Left

Endcap: Z > 0, Panel2

Testing of Code

- Use CMSSW5_0_0_pre5
- Use Validation sample
  - CMSSW_5_0_0_pre4/RelValSingleMuPt100/GEN-SIM-RECO/START50_V3-v1
  - CMSSW_5_0_0_pre4/RelValSingleMuPt100/GEN-SIM-DIGI-RAW-HLTDEBUG/START50_V3-v1
  - CMSSW_5_0_0_pre4/RelValQCD_Pt_80_120/GEN-SIM-RECO/START50_V3-v1
  - CMSSW_5_0_0_pre4/RelValQCD_Pt_80_120/GEN-SIM-DIGI-RAW-HLTDEBUG/START50_V3-v1
  - CMSSW_5_0_0_pre4/RelValTTbar/GEN-SIM-RECO/START50_V3-v1
  - CMSSW_5_0_0_pre4/RelValTTbar/GEN-SIM-DIGI-RAW-HLTDEBUG/START50_V3-v1

- GlobalTag.globaltag = 'MC_50_V0::All'
- DeadModules_DB = cms.bool(True) → DeadModules_DB = cms.bool(False)
  - Use config file instead of DB