SiW ECAL 2017 Beam Test: raw2root + data quality checks

- Data Integrity:
  - Data structure
  - Data integrity information
- Tagging “bad” events:
  - Badbcid definition
  - Improvements?

LAL

A. Irles, Orsay 7th July 2017

RAW2ROOT.C : Data Integrity Checks

- The root macro checks the data bytes and headers.
  - If additional word is found: warning message and shift of the counter
  - If no additional word is found but spill candidate packet has wrong length (in words) → error message and reject spill candidate (count)
  - Else: if bad number of columns → error message and reject spill candidate (count)
  - Else: if bad chip id → error message and reject spill candidate (count)
  - Else: if error in bits → error message and reject spill candidate (count)

    In this case, we also save (if possible) the bcid andsca were the error happened

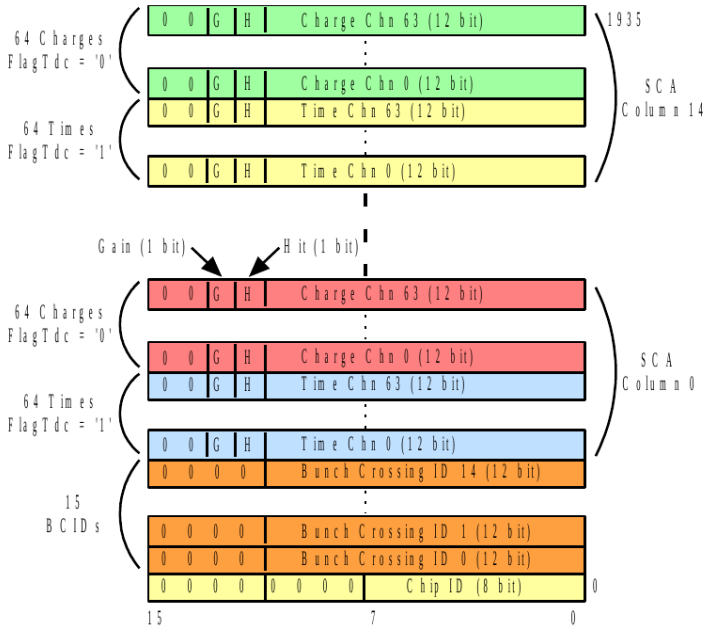
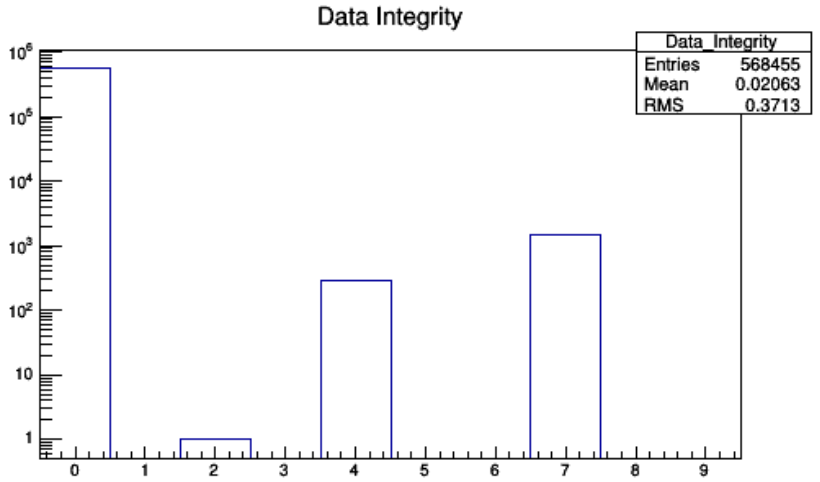


Figure 15: SKIROC2 chip memory mapping

RAW2ROOT.C : Data Integrity Checks

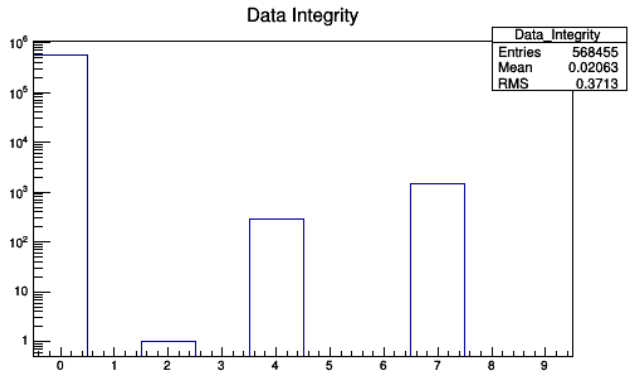
- Data Integrity histogram (spill based):
  - value = 0 --> OK
  - value = 1 --> bad data size
  - value = 2 --> more than 15 memory columns
  - value = 3 --> bad chip number
  - value = 4 --> extra bits in BCID (>12)
  - value = 5 --> extra bits in LOW GAIN charge/hbits --> expected 13 bits, no more. The 14th is for autogain mode --> not used
  - value = 6 --> extra bits in HIGH GAIN charge/hbits --> expected 13 bits, no more. The 14th is for autogain mode --> not used
  - value = 7 --> hit bit from high gain != hit bit from low gain
  - value = 8 --> Bad number of columns or bad number of channels
- Error tagging is not cumulative: if bcid is wrong but hit bit s also wrong, the event is tagged as hit bit wrong.
- Saved in the .raw.root file

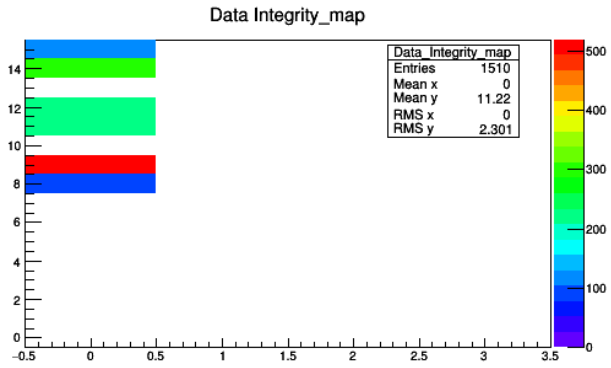


Example for layer 1 (slab 21):

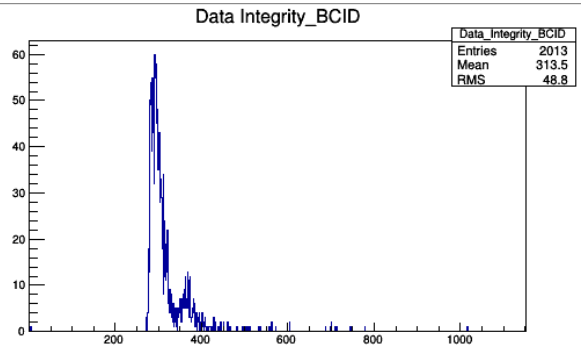
Full mip scan

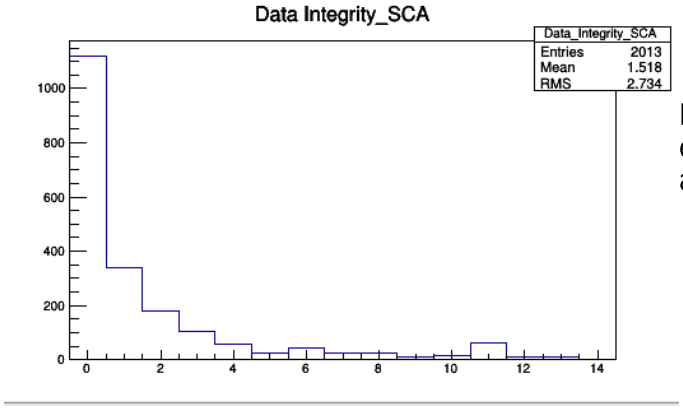
RAW2ROOT.C : Data Integrity Checks





Chip channel map (but if error appears in the first channel, then we stop counting)

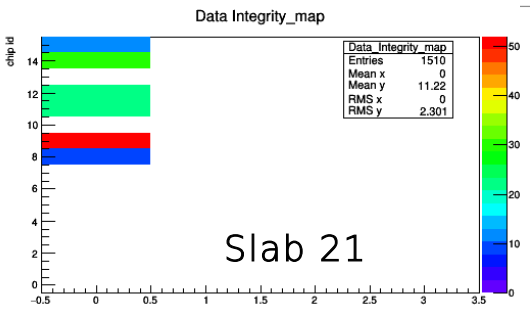


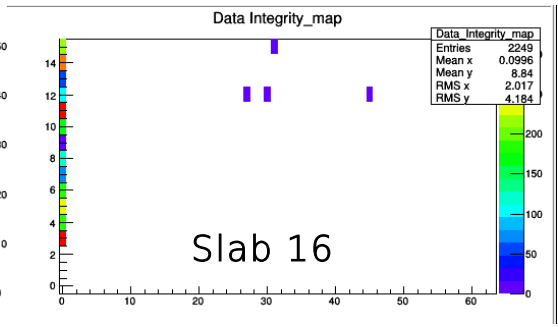


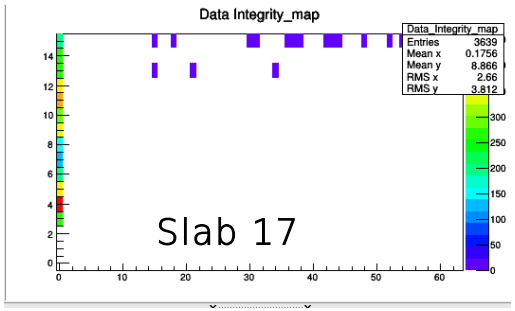
BCID spectra for rejected events looks very similar for all slabs → shifting bit ?

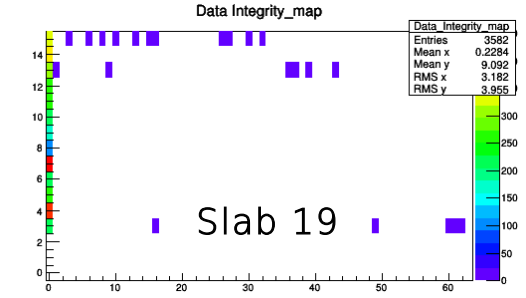
Example for layer 1 (slab 21):Full mip scan

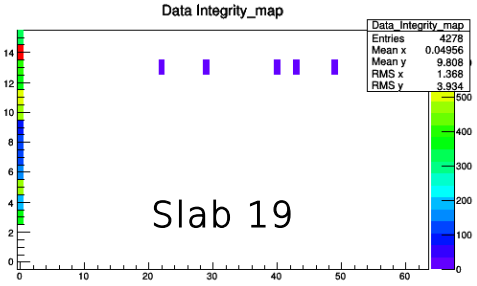
RAW2ROOT.C : Data Integrity Checks

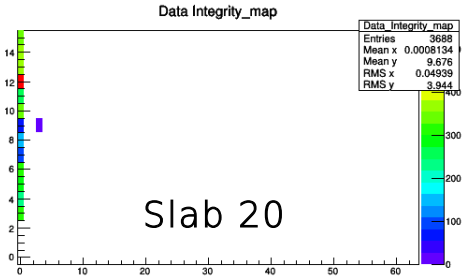


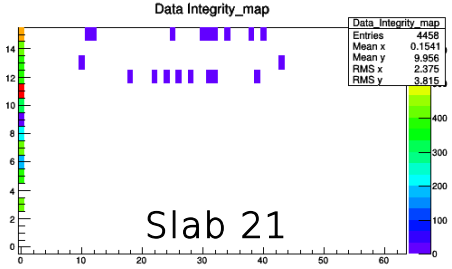








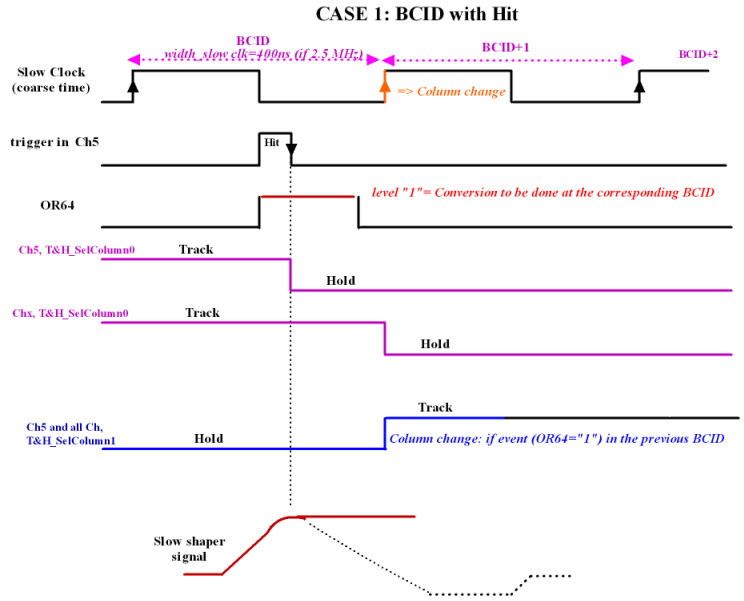




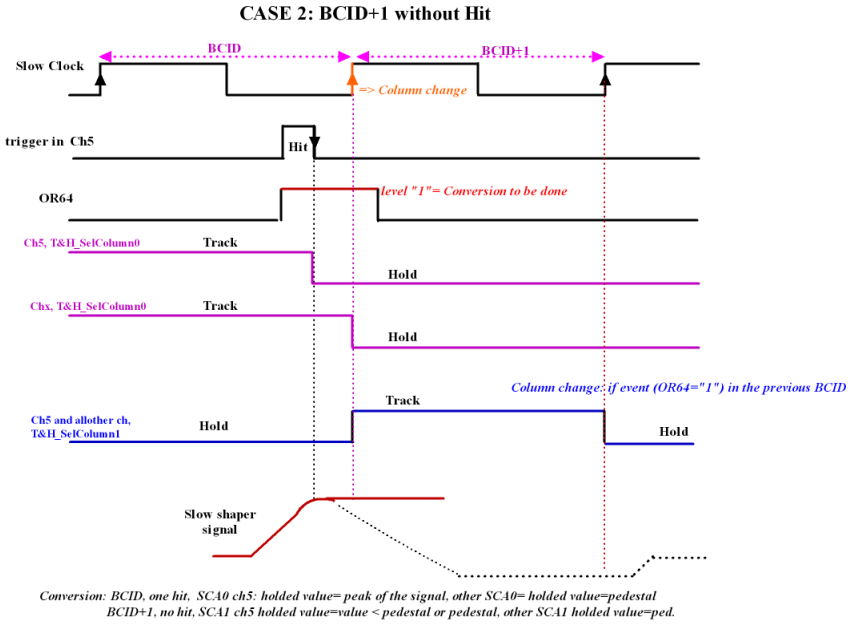
Why there are no errors in chips 0,1,2 ?

Are these chips superseeded by other sources that are not chip-taggable (i.e. bad number of columns, bad chip id, bad number of words)

BCID+1 events



Conversion: BCID with one hit, SCA0 ch5: holded value= peak of the signal, other SCA0= holded value=pedestal, other SCAi= ped
BCID+1: No conversion because OR64 level=0 during this BCID+1



Conversion: BCID, one hit, SCA0 ch5: holded value= peak of the signal, other SCA0= holded value=pedestal
BCID+1, no hit, SCA1 ch5 holded value=value < pedestal or pedestal, other SCA1 holded value=ped.

- This are not bad events !!
- Next SCA (NSCA+1) is filled with a zero, but SCA=NSCA is usable → not remove from analysis !
- ~ 15% of chances of happening (reduced to ~0 in skiroc2a)

Retriggers, plane events and underflow data

- Retrigger: different issue than BCID+1 but similar “signature”
  - Retrigger creates lots of entries in consecutive bcids.
  - Are usually associated to a pedestal shift – double pedestal distributions

    a change in the reference analog power supply level ? → fake signals
  - Identified by checking the previous bcid, if small difference → tagged.
- Plane event: similar cause ? Identified easily by selecting events with less than X channels triggered (X depends on the beam composition, usually 32 is a right number)
- Underflow data (ADC=4):
  - a) by sampling in the undershot (in principle solved/reduced in sk2a)
  - b) only at SCA, due to some noise (and timing) extra sensitivity on analog part intrinsic to skiroc2 (in principle solved in 2a) → very located channels, probably near some power/other lines that introduce some noise in the analog part of the chip.

Tagging these events: badbcid

- How are all these events identified ?
- Using the nhits[16][15] variable (counts all hits in one SCA with ADC>10)
- Using the badbcid[16][15] variable. It is filled doing the following checks sequentially:
  - ==0 if SCA=0
  - ==1-16 if bcid[chip][sca]-bcid[chip][sca-1]=1-16
  - ==+32 if a negative entry (ADC<10) is found in the chip for this sca
  - ==0 if none of the previous.

Tagging these events: badbcid

- Issues observed:
  - We tag the bcid+1 events as bad events
  - SCA 0 is always tagged as good but we see some double pedestals (see Yu's slides)
- Solution ?
  - ==1-16 if bcid[chip][sca+1]-bcid[chip][sca]=1-16
  - ==+32 if a negative entry (ADC<10) is found in the chip for this sca
  - ==0 if none of the previous.
- New RAW2ROOT version to be updated (and checked) in the repository