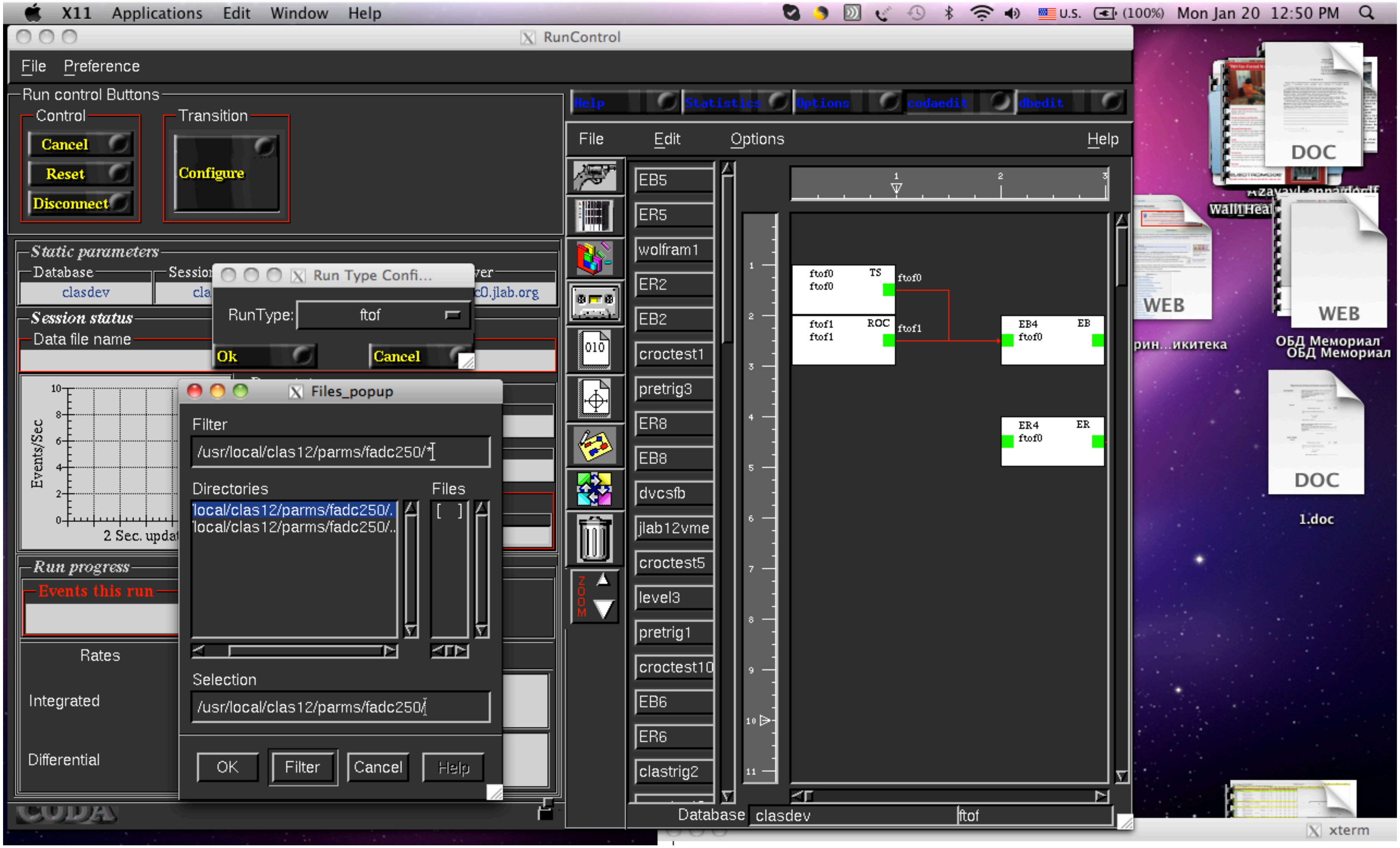
HPS TDAQ Review: Configuration, Monitoring And Testing

Sergey Boyarinov
JLAB

Feb 25, 2014

DAQ/Trigger: configuration procedures



Database: clasdev

RunType: ftof

Filter: /usr/local/clas12/parms/fadc250/*

Directories:
local/clas12/parms/fadc250/.
local/clas12/parms/fadc250/..

Selection: /usr/local/clas12/parms/fadc250/

Database: clasdev ftof

Sergey Boyarinov

DAQ/Trigger: configuration procedures (cont.)

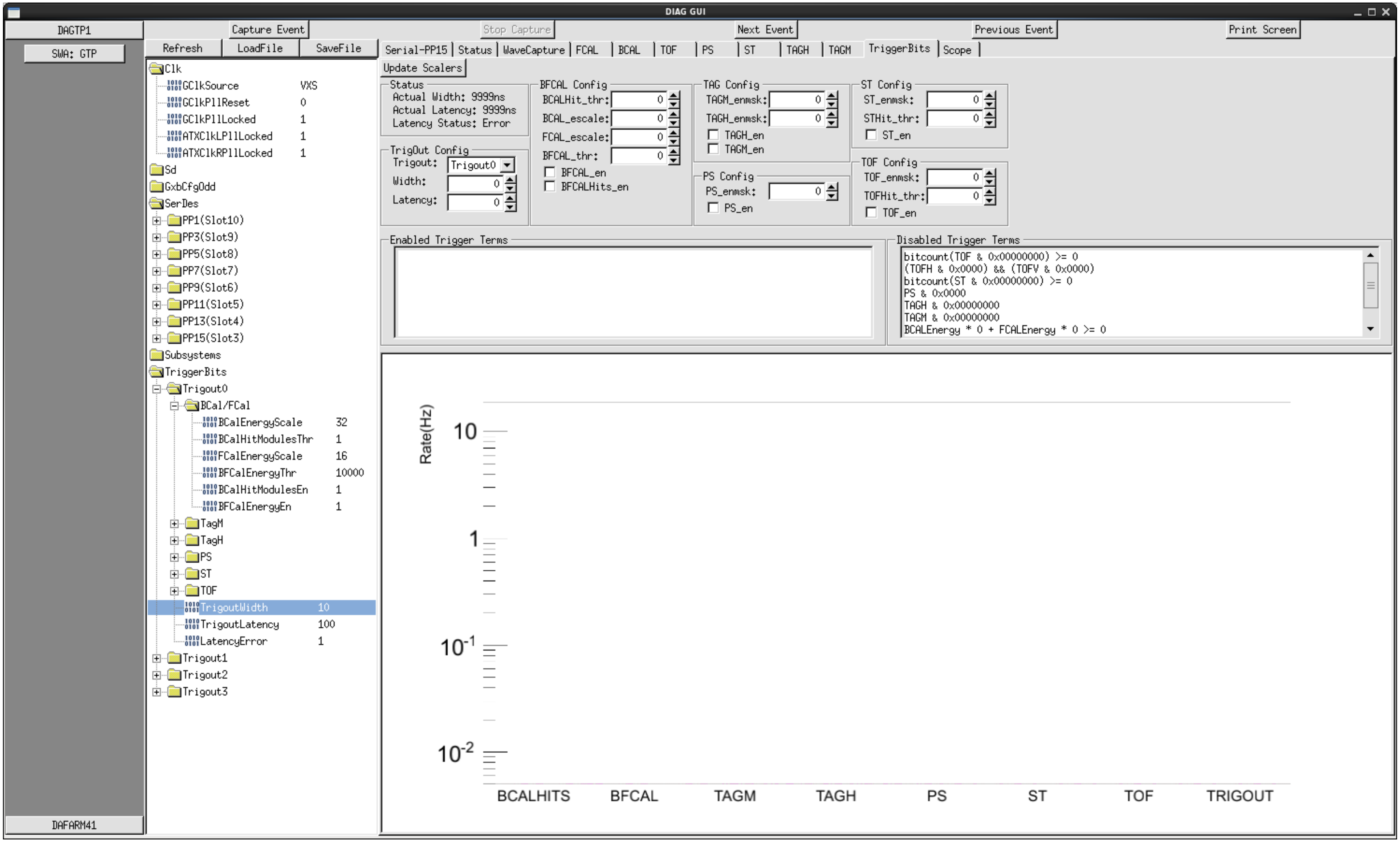
Hps_prod1.cnf:

```
   CRATE hps1
   TI_ADD_SLAVE 1
   TI_ADD_SLAVE 2
   TI_FIBER_DELAY_OFFSET 100 200
   TI_BLOCK_LEVEL 40
   TI_BUFFER_LEVEL 0
   #include “fadc250_hps.cnf”
   ……………………………
```

fadc250_hps.cnf:

```
   CRATE all
# default for all slots
   FADC250_SLOT all
   FADC250_W_OFFSET 1200
   FADC250_W_WIDTH 400
   FADC250_ADC_MASK 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
# customization for some slots/channels
   CRATE hps2
   FADC250_SLOT 9
   FADC250_TRG_MASK 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0
```

DAQ/Trigger: configuration procedures (cont.)
DiagGUI (Ben Raydo)



```
Clk
  GClkSource          VXS
  GClkPllReset        0
  GClkPllLocked       1
  ATXClkLPllLocked    1
  ATXClkRPllLocked    1
Sd
GxbCfgOdd
SerDes
  PP1(Slot10)
  PP3(Slot9)
  PP5(Slot8)
  PP7(Slot7)
  PP9(Slot6)
  PP11(Slot5)
  PP13(Slot4)
  PP15(Slot3)
Subsystems
TriggerBits
  Trigout0
    BCal/FCal
      BCalEnergyScale       32
      BCalHitModulesThr     1
      FCalEnergyScale       16
      BFCalEnergyThr        10000
      BCalHitModulesEn      1
      BFCalEnergyEn         1
    TagM
    TagH
    PS
    ST
    TOF
    TrigoutWidth        10
    TrigoutLatency      100
    LatencyError        1
  Trigout1
  Trigout2
  Trigout3
```

```
Status
 Actual Width: 9999ns
 Actual Latency: 9999ns
 Latency Status: Error

TrigOut Config
 Trigout: Trigout0
 Width:   0
 Latency: 0

BFCAL Config
 BCALHit_thr: 0
 BCAL_escale: 0
 FCAL_escale: 0
 BFCAL_thr:   0
 □ BFCAL_en
 □ BFCALHits_en

TAG Config
 TAGM_enmsk: 0
 TAGH_enmsk: 0
 □ TAGH_en
 □ TAGM_en

PS Config
 PS_enmsk: 0
 □ PS_en

ST Config
 ST_enmsk: 0
 STHit_thr: 0
 □ ST_en

TOF Config
 TOF_enmsk: 0
 TOFHit_thr: 0
 □ TOF_en
```

Enabled Trigger Terms

Disabled Trigger Terms

```
bitcount(TOF & 0x00000000) >= 0
(TOFH & 0x0000) && (TOFV & 0x0000)
bitcount(ST & 0x00000000) >= 0
PS & 0x0000
TAGH & 0x00000000
TAGM & 0x00000000
BCALEnergy * 0 + FCALEnergy * 0 >= 0
```

Sergey Boyarinov

DAQ/Trigger: configuration procedures (cont.)
DiagGUI (Ben Raydo)

The LoadFile/SaveFile will read/write a text file with and key/value pair per line. The key will match the name in the tree structure prefixed by the board type and slot number. For example:

On the GTP register tree: Clk->GClkSource is set to "VXS"...So the configuration text file would read/write the line:

```
GTP11_CLK.GCLKSOURCE       VXS
```

where:
 "GTP" is the board type
 "11" is VME slot 11
 "CLK.GCLKSOURCE" is the path name in the register tree description
 "VXS" is the read/assigned value

In the case of the Gui, the parser comes for free since the parameter names are all defined in the register tree description - so none of this has to be re-written and the parses can match the key name against the names in the register tree.

DAQ/Trigger: configuration procedures (cont.)

- CODA Configuration and Hardware Control/ Monitoring procedures will be integrated, so hardware settings can be saved to be used by CODA
- All settings will be recorded into data files, and into database if available

Monitoring

Stage 1 (FADC)

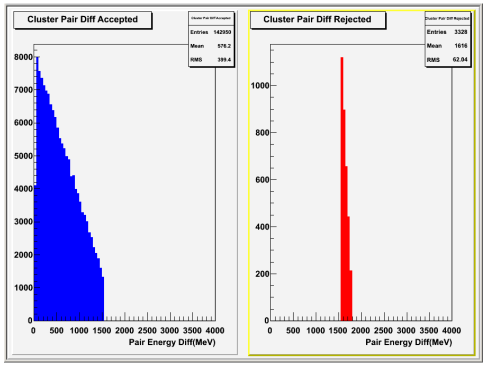
1. Scalers per channel (readout threshold based)

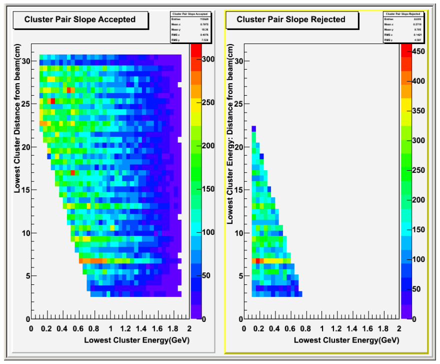
Stage 2 (CTP)

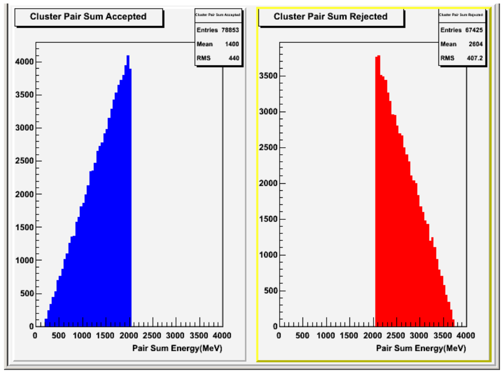
1. Individual ADC channel pulse energy histograms

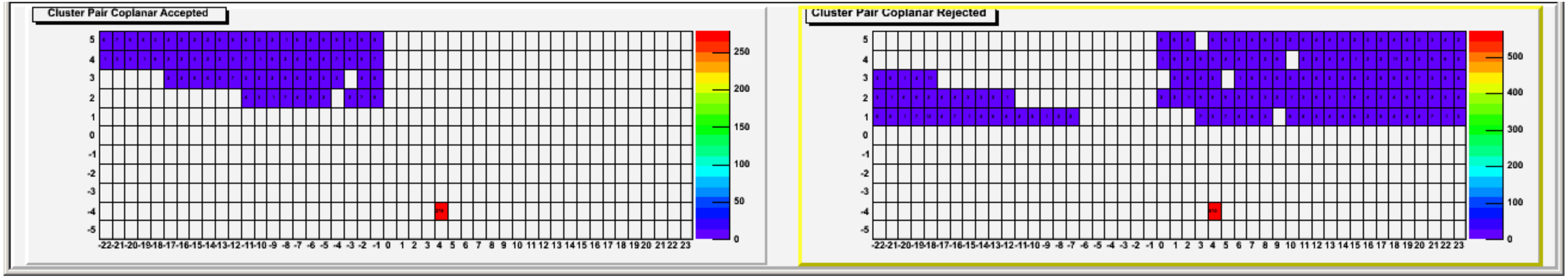
Stage 3 (SSP)

1. Cluster Hits (Position)
2. Cluster Hits (Position+Energy) - Depending on resources in SSP
3. Trigger cut accept/reject:









Monitoring (cont.)

- Hardware monitoring includes scalers (readout by DAQ and Slow Control), configuration registers, histograms etc
- Monitoring GUIs (mostly the same as configuration GUIs) can be used for some checks, but will not replace online monitoring system, which have to analyze data and provide DAQ and Trigger systems with calibration constants
- Integration with EPICS (scaler server supplying data to both DAQ and EPICS) – starts June 1

DAQ/Trigger testing procedures

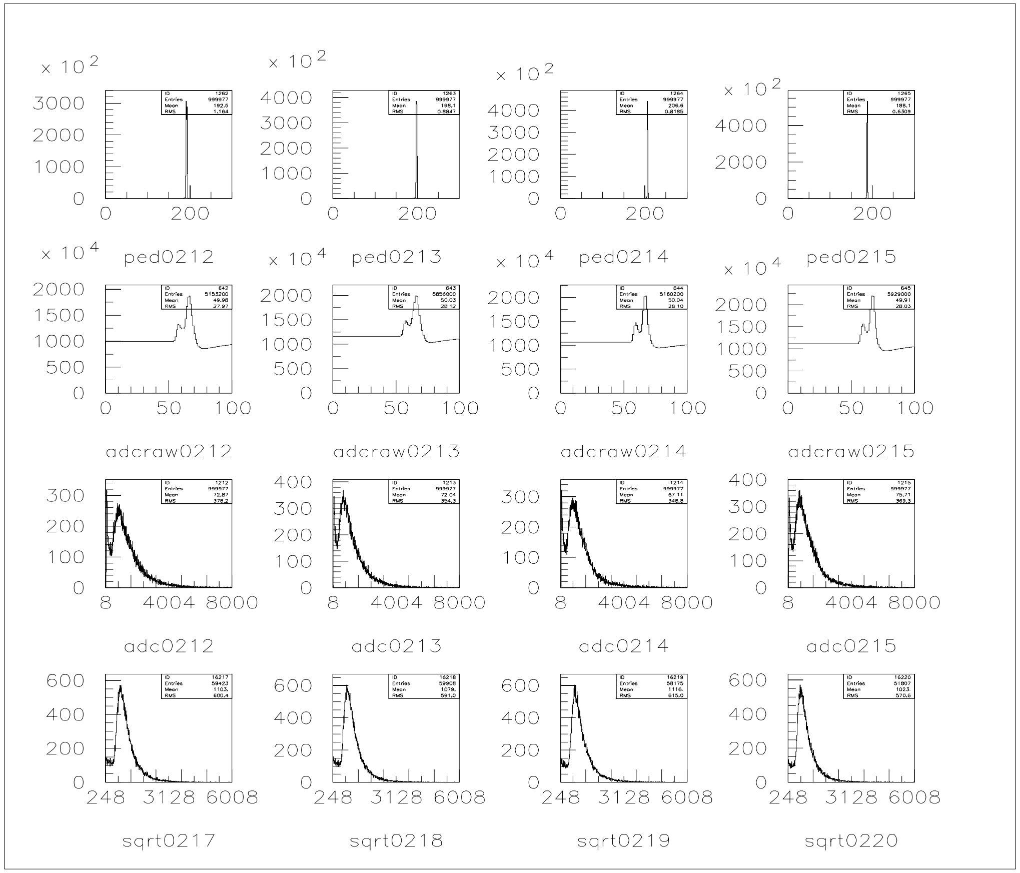
- DAQ performance test: FADC raw mode with different window width and random pulser as input trigger; measure event rate and data rate limitations
- Trigger test: playback mode test using GEANT data
- DAQ performance test will be repeated again after SVT electronics integration
- All above will be performed without ECAL and SVT detectors

DAQ/Trigger testing procedures (cont.)

- FADC pedestal measurements: special runs will determine pedestals for every channel and store them into database; pedestals will be loaded during Prestart transition into both readout and trigger units; during data taking readout chain will measure pedestal using 4 samples from non-pulse area and record it with every event so it can be compared by online monitoring with pedestal run value
- gain measurements: using MIPs in cosmic run, triggering from two external scintillating counters, or from ECAL itself using OR of single cristals in TOP, ANDing with OR of single cristals in BOTTOM
- Trigger system threshold calibration will be extracted from gain measurements

DAQ/Trigger testing procedures (cont.)

Example from CLAS12 FTOF Detector



Conclusion

- DAQ/Trigger configuration, monitoring and testing procedures are defined, partially exists and will be finalized during DAQ and Trigger commissioning
- Trigger functionality will be tested and DAQ performance measured during commissioning
- Close coordination with SVT and online data processing groups is required