SSD documentation

Online monitoring

Version Number 1
28-june-02

Authors : J. Baudot

1 Goals and scope of the online monitoring system

The online monitoring system is a set of programs and graphical user interfaces (GUIs) which goals are the following.

> In a data-taking configuration, it monitors the raw data to check the SSD runs as expected, especially checking that individual strip is alive and that no important part of the detector turns black.
> In debugging mode, on top of the previous features, it computes the pedestal, noise and signal of each strip (perform a calibration) to assess the performance of the detector.

In this document, we describe the inputs to the system, the software architecture, the algorithms used to obtain the different results and finally a kind of user's guide to the system with a list of histograms. The last section is dedicated to the status of the system.

2 Inputs to the system

Considering the first phase of the SSD installation - one ladder operational – the data acquisition will run in a specific stand-alone mode which will provide the data directly to the monitoring system without the use of the general STAR event pool frame.

It is nevertheless useful to define the different types of data that the online monitoring system will have to deal with.

- Raw-data with no zero-suppression.
- Raw-data with zero-suppression.
- Dst-data: results of the SSD reconstruction algorithm (see B.Hippolyte's Star note 420) to transform hits into space-points passed over raw-data.

3 Software architecture

The online monitoring system is based on a set of stand-alone C++ classes compiled in the ROOT framework which provides both the histograms and GUIs features.

Each software class match a facility:

- Readout of the data from file. (*If this cannot be done without the root4star framework, then the whole system will be root4star based. This is probably true to read the Dst-data.)*

- Algorithms to compute pedestal, noise, common mode shift and signal on a given strip for a number of events.
- Storage of values for pedestal, noise, common mode shift and signal for each strip.
- Histogramming, booking, filling and eventually storing to file.
- GUI.

4 Algorithms

Can't say much here except refer to Christophe Suire's PhD thesis and ALICE notes where all is explained and results for prototypes are given.

5 User's guide

The GUI does not exist yet, so nothing to say, but it will be smart!

Nevertheless, some histograms have already been produced using the output of the SSD reconstruction chain to foresee what the monitoring task could be.
A list is provided below with some comments with the necessary inputs to create each histogram and its usefulness.

Description	**#histograms**	**#events needed**	**Input**	**goals**
Pedestal, Noise, Common mode shift	One per wafer side = 640 Could be average for module to summarize on 1	500-1000 depending on the accuracy requested	Raw-data with no signal	Check the response of the SSD is normal
Gain	One per wafer side = 640 Could be average for module to summarize on 1	500-1000 depending on the accuracy requested (need calibration done)	Raw-data with calibrated signal	Check the response of the SSD is normal

Hit frequency	One per wafer side = 640 Could be average for module to summarize on 1	Hit occupancy is a few %, so >1000	Raw-data in data-taking mode	Check inefficiency or black zones
Mean number of strips per cluster	1 for the whole SSD	Hit occupancy is a few %, so >1000	Dst-data after reconstruction	Check the reconstruction is OK
Landau maxima	1 for the whole SSD	Hit occupancy is a few %, so >1000	Dst-data after reconstruction	Check the reconstruction is OK
Charge matching between sides	1 for the whole SSD	Hit occupancy is a few %, so >1000	Dst-data after reconstruction	Check the reconstruction is OK

6 Status

The system is not set up as a whole so far. But many pieces are running independently. We give a very brief overview of where bits of code (already C++ in ROOT) could be taken .

- **Readout**: Abdel Boucham should have something to read data coming from Subatech DAQ.
- **Algorithm**: Big and very detailed code in Strasbourg, also something done by Abdel.
- **Storage**: clearly the most relevant item for performance (speed). Algorithms in Strasbourg are based on one object per strip. But we already know this model can not be used for the whole detector !
- **Histograming**:
- **GUI**: probably one has to copy Serguei Panitkin stuff for general STAR online or SVT interface. Anyway, the ROOT team advertises this as a trvial task(?).

Work to gather this different items will be done during July 2002.