OLYMPUS

DAQ

Overview & Implementation

Christian Funke
HISKP University of Bonn

Olympus Meeting Hamburg 27.-28.04.11

DAQ features

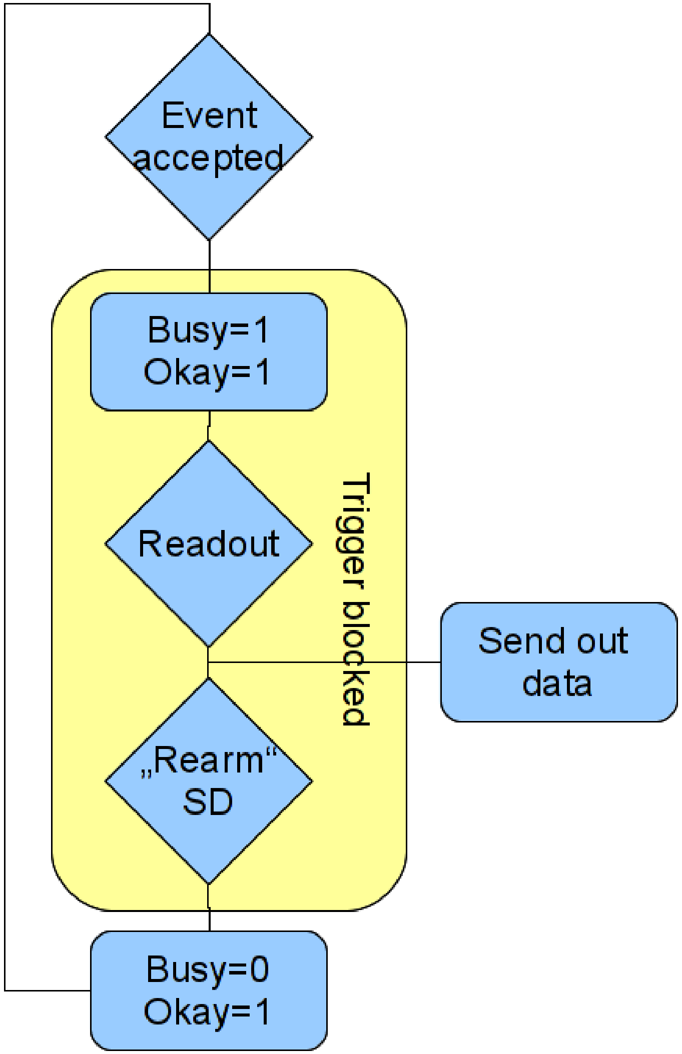
- Synchronous design
- Hardware trigger driven
- Blocking readout
- Synchronization enforced on hardware level via custom VME modules (syncsystem)
- Up to 8 different hardware event types
- Detector agnostic design

Nomenclature

- Trigger = Signal emitted by Alexanders main trigger unit – An accepted combination of subdetector triggers – readout signal for the DAQ
- Run = A single datafile taken with the DAQ system – depending on datarate about every 30 minutes a new file will be written.
- Event = aggregate information of all active subdetectors

Data cycle

1. Check if all active subdetectors have **okay** flag set and **busy** unset -> Trigger is opened
2. Hardware **event** signal sent to all subdetector readouts following an accepted trigger
3. Each subdetector sets **busy** flag while performing readout
4. Triggers are blocked until all subdetectors unset **busy**
5. Continue with point 1.



Event types

- Table Event – First event of every run where „static“ information e.g. Lookup tables and pedestal values are saved to the data file.
- Data event – regular readout of active subdetectors, up to eight different event types can be implemented (scaler events). The event type is set by the trigger.
- Modulus event – additional readout to be performed every n data events.

Table Event

Subdetector	Information Saved
Trigger	Runinformation / Slowcontrol data
Blast1	TDC LUT-files
Blast2	ADC/TDC LUT-files, pedestals?
MWPC	MWPC Config / Wiremap
Lumi	Parameters ?
Moller	QDC Mapping / Pedestals

Scaler Events (~50Hz)

Subdetector	Information saved
Trigger	Livetime/Deadtime, Triggerrates
Blast1	-
Blast2	-
Lumi	Scalers?
Moller	Flasher?

The symmetric Moller readout will need special event type !

Dataformat/Datamapping

- Data is saved in the intermediate Zebra format
- Linear binary stream with bank structure corresponding to single readout blocks – raw data

- For analysis purposes all Zebra data will be converted to „raw“ Root files
- Hardware dependent indices will be mapped to stable readout indices

Datamapping continued

- Mapping indices should be provided as LUT files which will be saved in the table events
- Unneeded information (headers etc) can be stripped at this stage
- Datadecoders have to be provided by the subdetector groups – otherwise the binary data will just be dumped in the Root tree
- No calibration will take place at this stage

Onlinemonitor

- Online spectra will be generated using the same software, which is used for datamapping
- If no decoding routines exist that detector will not show up in the online monitoring
- Input from subdetector experts is needed to determine which spectra should appear in the onlinemonitor
- Spectra should be reasonably simple

Raw ROOT format

- The used Root version will be 5.28/00
- All events will be stored in Root Tree with branches corresponding to the single subdetectors
- The data from the table event will be stored in a separate directory of the Root file
- The necessary header files and example macros to read those files will be provided at a central location

Software status

- Blast readout – done, needs testing
- MWPC readout – done
- Lumi readout – w.i.p.
- Symmetric moller readout – w.i.p
- Online monitor – framework done, need input/datamapping experts
- Root converter – see online monitor

Hardware status

- All servers delivered – 6 TB of mirrored storage available for experimental data
- New machine for central services will be setup next week
- All syncsystem hardware has been transported to Desy
- The last VME CPUs will (hopefully) be delivered in the last week of july
- As requested two additional monitors have been installed in the control room