

CMS Front-End / DAQ Interfacing

G. Antchev, E. Cano, S. Cittolin, S. Erhan, B. Faure, D. Gigi, J. Gutleber, C. Jacobs, F. Meijers, E. Meschi, A. Ninane, L. Orsini, L. Pollet, A. Racz, D. Samyn, N. Sinanis, W. Schleifer, P. Sphicas

CERN, Div. EP, Meyrin CH-1211 Geneva 23 Switzerland

Abstract

In the context of large data acquisition systems (DAQ) for LHC experiments, flexible and uniform interfaces between subsystems is a key feature to ensure best possible system integration, maintainability and a clear system upgrade policy. This approach is widely used in CMS at the border between sub-detectors front-end and the DAQ.

I. INTRODUCTION

In the case of CMS, there will be about 12 different data sources providing ~ 1 MB of data per trigger to the DAQ (see Fig. 1). Interfacing these sources with the DAQ is a critical point given the overall size and complexity of the final system (on-detector electronic, counting room electronic and DAQ). The interface is defined and elaborated within the Readout Unit Working Group where all data providers and data consumers are represented [3].

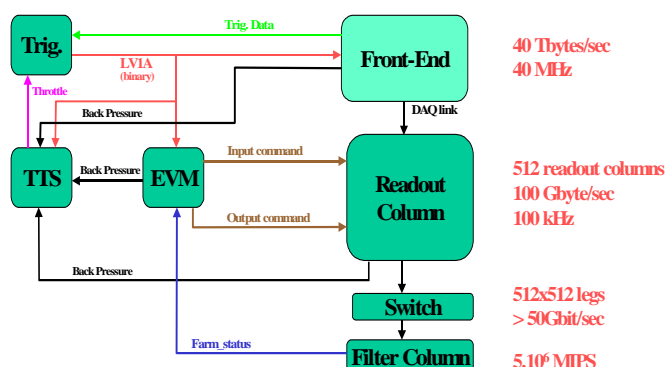


Fig. 1 CMS DAQ block diagram

II. READOUT CHAIN

The Detector Dependant Unit (formerly known as Front End Driver) can be seen as the interface between the DAQ and the sub-detector readout systems. Below the DDU, no sub-detector specific hardware is foreseen in the DAQ architecture (see Fig. 2).

A. Detector Dependent Unit

The task of the DDU is to deal with a given sub-detector with its specificities and make available the data to the DAQ link according to a unique CMS-wide specification [1]. The specification includes the minimum functionalities to be performed by the DDU (header generation, alignment checking...) and the description of the DAQ slot located on the DDU where the DAQ link is plugged.

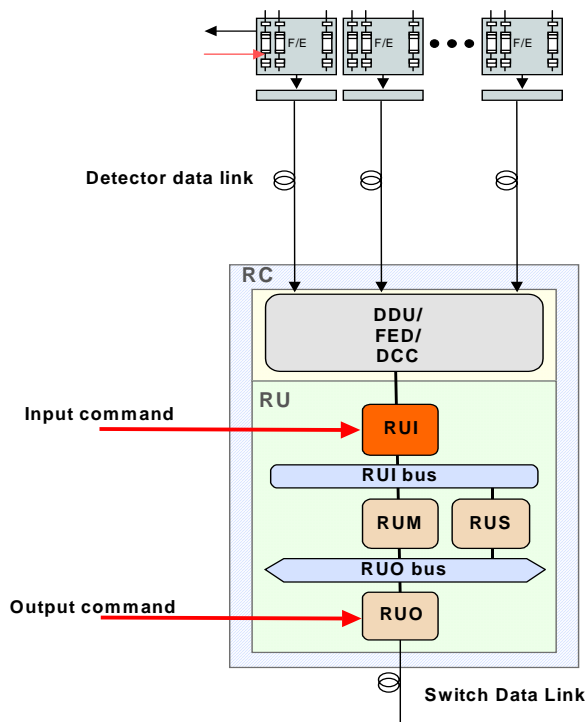


Fig. 2 Readout chain logical structure

B. DAQ slot

The FED DAQ slot is based on S-LINK¹ [2] but extended to match CMS needs (64 bits @ 100 MHz). The extension is implemented through an extra connector hence allowing the usage of standard S-LINK product until the availability of the DAQ link.

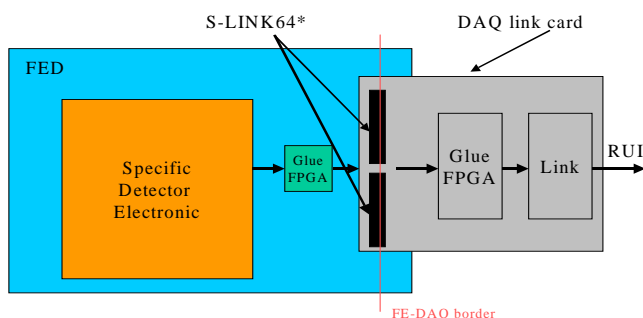


Fig. 3 DAQ port on the FED

1. Generic FIFO interface featuring 32 bits @ 40 MHz specified at CERN by R. McLaren and E. Van Der Bij

C. DAQ link requirements

The average event size at the DDU level is ~2KB with a maximum trigger rate of 100KHz. Therefore the average throughput of the DAQ link must be 200MB/sec over 200m. Due to stochastic fluctuations on the event size and large uncertainties on the LHC luminosity and the detector occupancy/noise, a safety margin of 2 is foreseen in the DAQ link design to absorb unexpected or pathologic (!) data volume.

S-LINK as well as S-LINK64 specifies a set of 2 connectors (a sending and a receiving one) but not the physical link in between. The possible link candidates are the following:

- CMS ECAL Front End link: HP G-Link based
10 ascending fibers @ 800Mbit/sec (800 MB/sec equivalent)
2 descending fibers @ 800Mbit/sec (used for flow control)
- COTS based design
i.e Gbit hardware components
- Commercial standard off-the-shelf data link
in this case, DAQ provides the glue logic between the FED-DAQ port and the commercial standard (i.e. PCI, PCI-X,...)

The design and the implementation of the S-LINK64 port on the FED is under the responsibility of the sub-detector.

The physical implementation of a typical CMS readout column is shown Fig. 4.

III. DATA TRANSFER PROTOCOL

Upon reception of a Level 1 Accept signal (LV1A), the DDU receives from the Front End the data corresponding to the selected event. After a detector specific processing, these data are encapsulated according to the common CMS data format and pushed through the DAQ port into the DAQ link card. This device moves the data in a transparent way to the other end of the link where the DDU events are queued in order of their arrival. Now the DDU events are available to the RUM as if the DDU would be located in the surface buildings.

Concurrently, the EVM, upon reception of the same LV1A, broadcasts to all RUIs the trigger_id (counter value incremented by LV1A) and an event_id. Each individual RUI can now extract the first available event in the link card, check that this event corresponds to the expected trigger_id (the data encapsulation provides a trigger_id generated at the Front End level) and store it to the RUM under the logical address

pointed by <event_id>. Later in the readout process, the event is now retrieved following this new “name”.

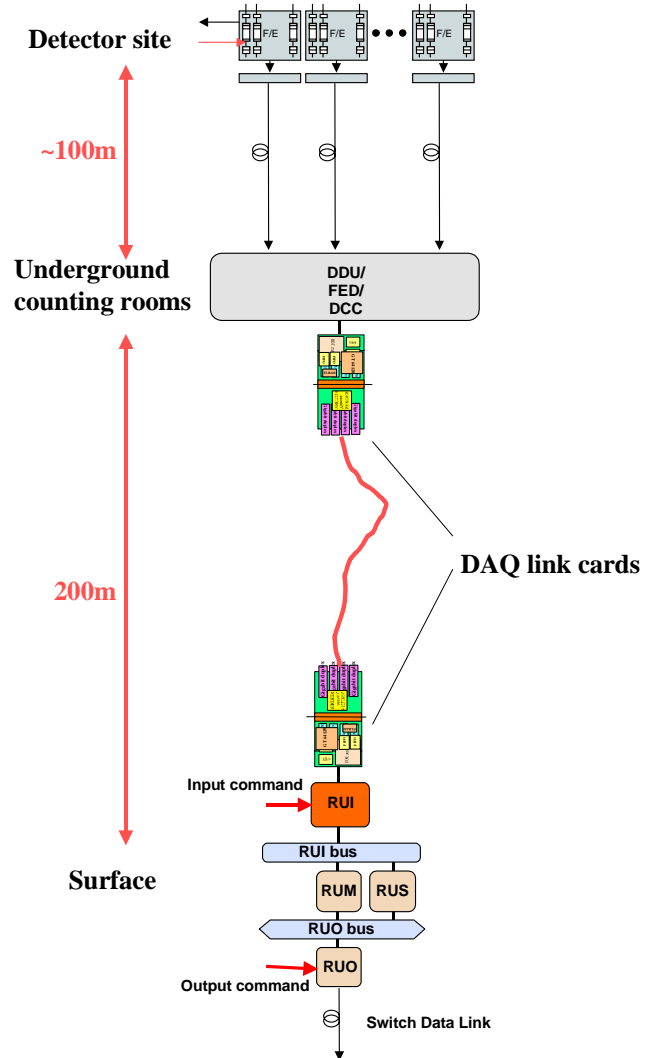


Fig. 4 Readout chain physical location

IV. TOPOLOGIES

The simplicity of S-LINK64 matches well the Front-End environment and is an easy starting point for data splitter/concentrator development. This flexibility is needed to allow the usage of either state-of-the-art high speed or low-cost components. Different possible topologies can be envisaged:

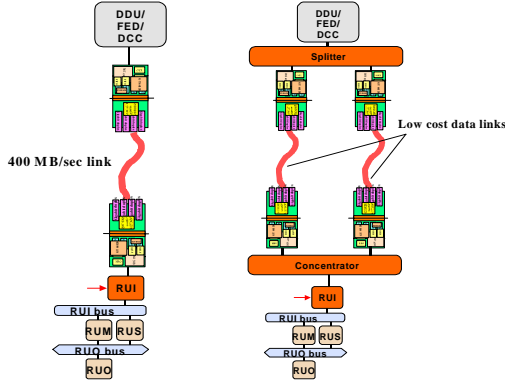


Fig. 5 one2one topology with a link at the nominal speed or low-speed

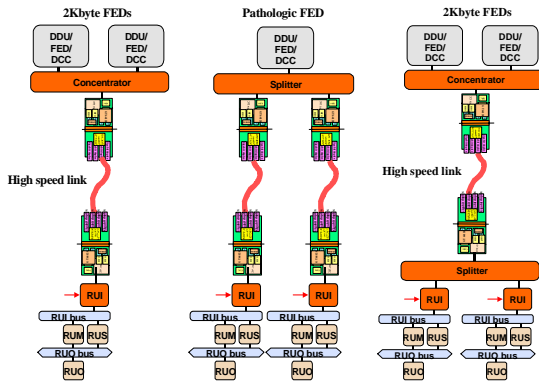


Fig. 6 n2one, one2n and n2n topologies

V. CONCLUSION

Seen from the detector side, the DAQ appears like a FIFO memory through the S-Link64 specifications. S-Link64 features a comfortable safety margin regarding expected data throughput. The different possible connection topologies allow the DAQ to adapt unexpected situations like data overload by simply adding more data links.

VI. GLOSSARY

EVM: Event manager
TTS: Trigger Throttling System
DDU: Detector Dependent Unit
FED: Front End Driver
DCC: Data Concentrator Card
RC: Readout Column
RU: Readout Unit

RUI: Readout Unit Input
RUM: Readout Unit Memory
RUS: Readout Unit Supervisor
RUO: Readout Unit Output

VII. REFERENCES

- [1] DDU design specifications A. Racz CMS note 1999-010
- [2] <http://hsi.web.cern.ch/s-link>
- [3] <http://cmsdoc.cern.ch/cms/TRIDAS/horizontal/>