Istituto Nazionale di Fisica Nucleare

Data Acquisition in Particle Physics Experiments

Ing. Giuseppe De Robertis – INFN Sez. Di Bari

Outline

- DAQ systems
  - Theory of operation
  - Case of a large experiment (CMS)
- Example of readout
  - GEM detectors for CMS upgrade
- Electronics for tests and small experiments
  - SRS
  - MOSAIC

What is the goal?

- The aim of a nuclear physics experiment is to gather data about nuclear interactions
- Nuclear particles pass through detectors which generate electrical *signals*
- These signals contain information about the particles - *type, energy, trajectory*
- The data acquisition system *digitizes, formats and stores this information in a way which can be retrieved for later analysis*

What are the problems?

- The complete set of signals which describe a single nuclear interaction is called an *Event*
- There can be thousands to millions of events occurring per second
- Detectors are large and distributed - containing many thousands of individual *channels*
- Events are of different sizes
- Only a few events are interesting

Data Acquisition Requirements

- Move the data: Detector --> Storage
- Configure and control Front-end electronics, trigger processing ...
- Monitor data flow
- Monitor detectors/hardware
- Inform operator of problems
- Experiments can run for days/weeks/months...

The Anatomy of a DAQ System

- **Triggering** (selecting events of interest)
- **Readout** (digitizing detector signals)
- **Event formatting** (standardize what we're saving)
- **Event building** (putting fragments together)
- **Event storage** (save data for analysis)
- **Run Control** (configure-start-stop experiments)
- **Monitoring** (get information about system status)
- **Slow Controls** (what is the other hardware doing?)

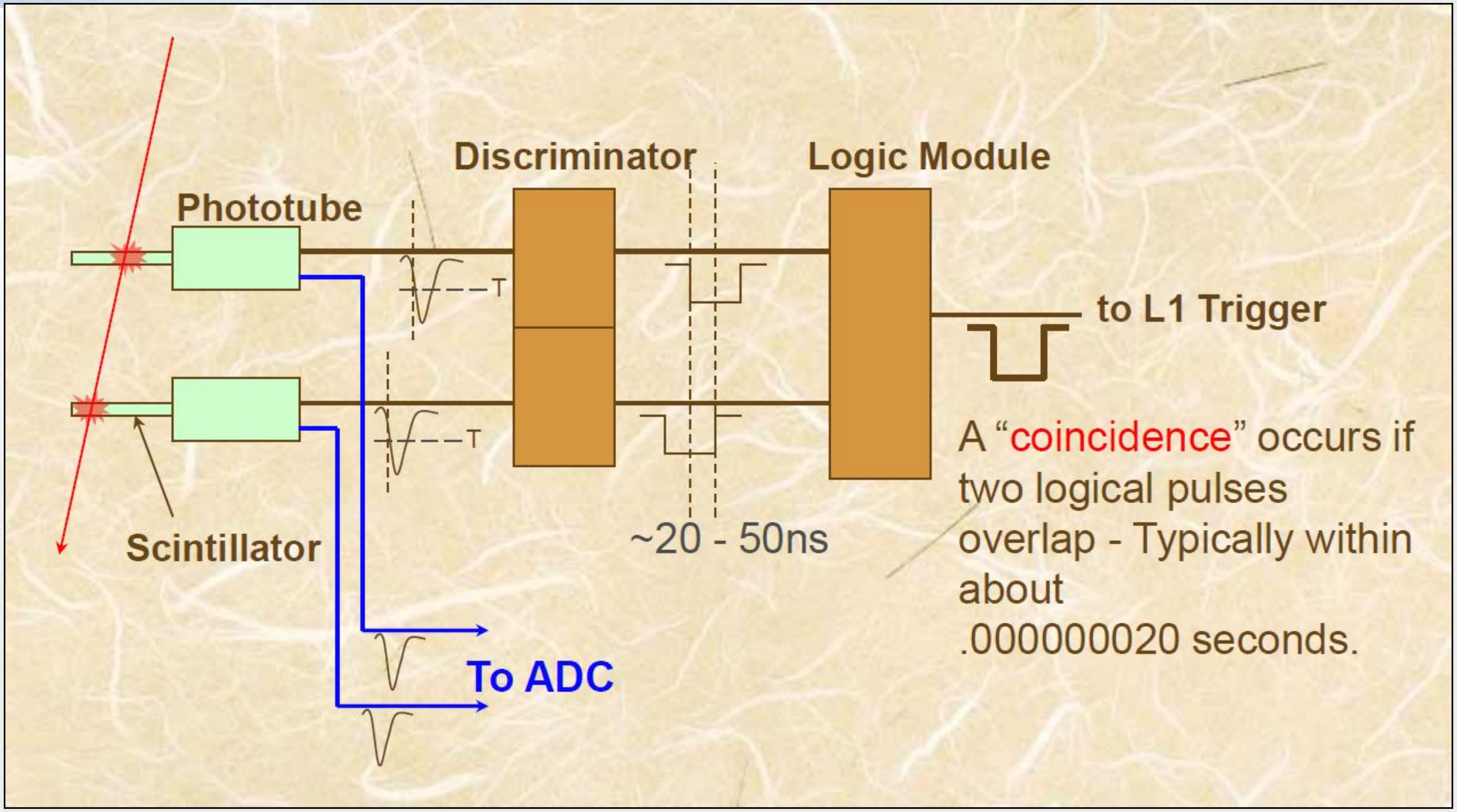
Triggering

- The data acquisition system needs to know when an interaction "*Event" has occurred in the detector*
- Some detectors are faster than others
- Signals from fast detectors are combined to make a decision on when an event has occurred. This is called a *trigger*
- Detectors have to store event data for the time needed to generate and distribute trigger. This time is called *trigger latency*

Triggering cont…

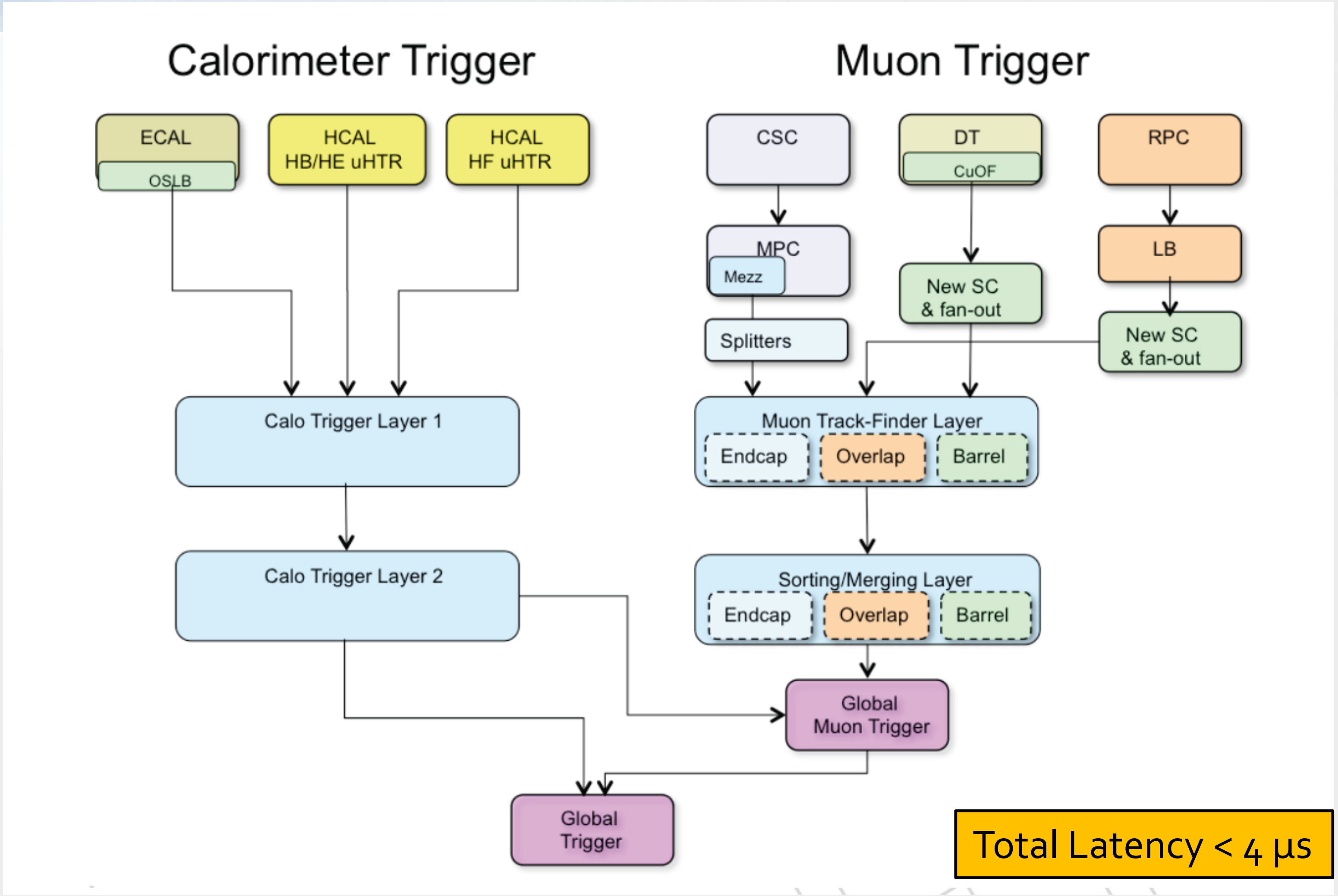
- **Triggering serves two functions:**
  1) Tells the rest of the system when to read
     - Trigger system tells DAQ to read out data
     - DAQ tells trigger system when it is busy
     - Busy time is called *dead-time* and is minimized by a well designed DAQ architecture
  2) Filters unwanted events
     - Most triggers work in levels
       - Level 1 is based on fast detectors like scintillators
       - Level 2 is based on slower detectors (drift chambers)
       - Level 3 is usually a *software filter*

A Simple Trigger



A "coincidence" occurs if two logical pulses overlap - Typically within about .000000020 seconds.

CMS Trigger subsystem



Total Latency < 4 μs

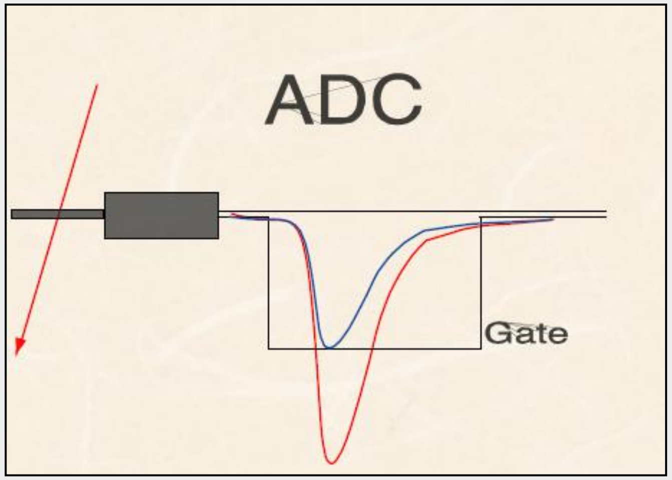
Readout

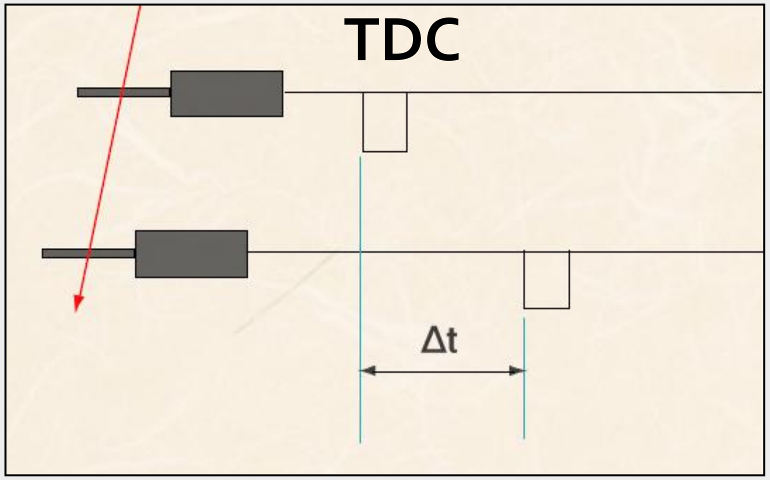
- Data takes the form of electrical signals
  - Convert Analog → Digital
  - Times -Time to Digital Converter, TDC
  - Voltages - Analog to Digital converter, ADC
  - Counts - scalars
- There are lots of signals spread over a large detector
  - Modular readout duplicated many times
  - Plug-in modules require something to plug into so that they can all be accessed together --> ***Buses***

In many modern systems A/D converters are inside the front-end electronics

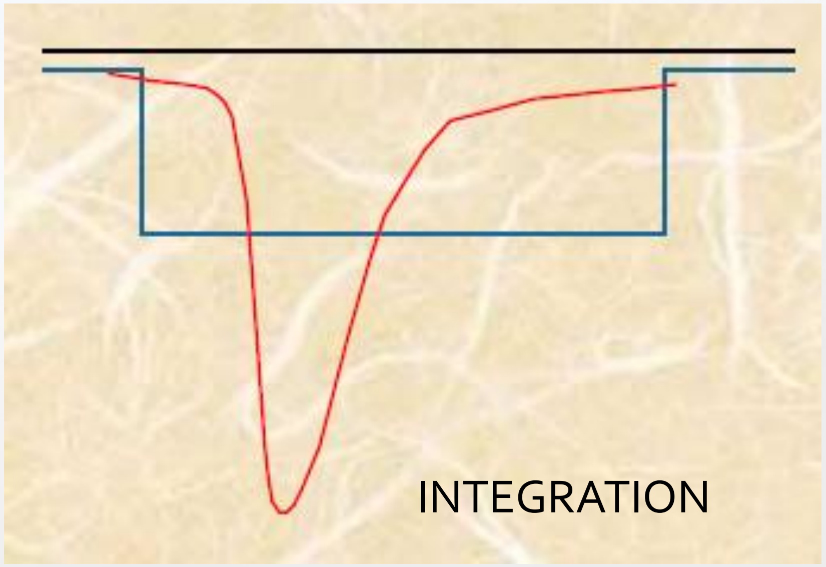
Digitizing Data

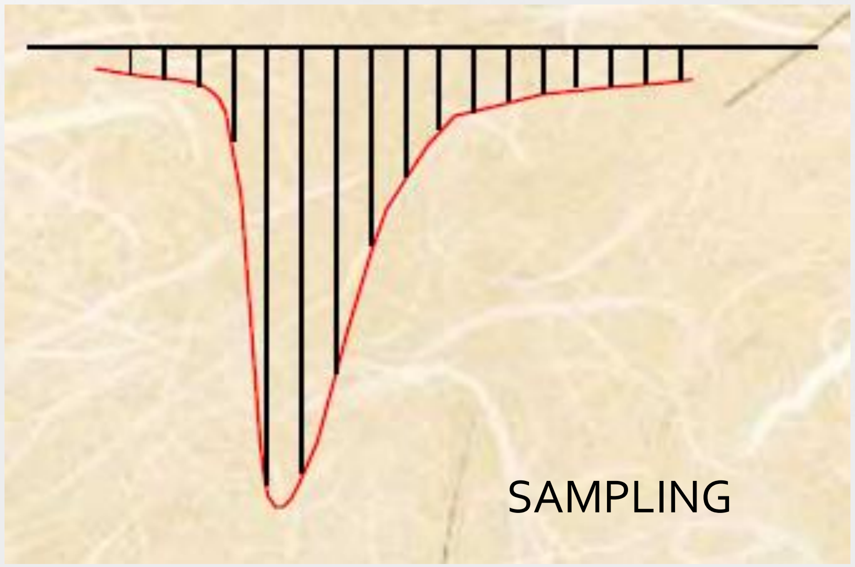
- Detector data must be digitized to be stored on a computer. When a particle passes through a detector we can measure:
  - the amount of energy (charge) it leaves (ADC)
  - the time it takes to travel between two different detectors (TDC)





Sampling vs Integration



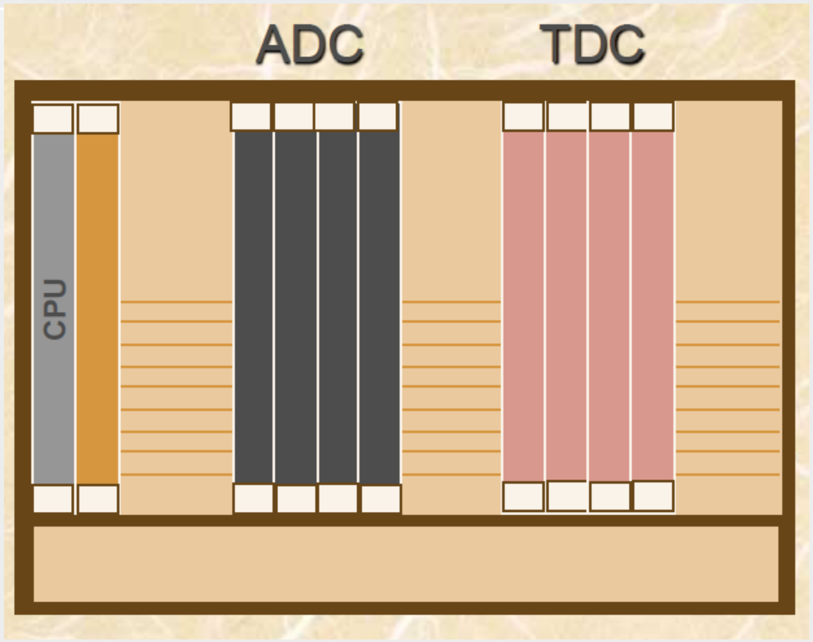


- Only one data representing the charge sum during the gate
- Slower electronics ➔ Cheaper

- Multiple data describing the waveform
- Fast electronics ➔ Expensive
- More parameters can be extracted:
  - Timing
  - Amplitude
  - Charge

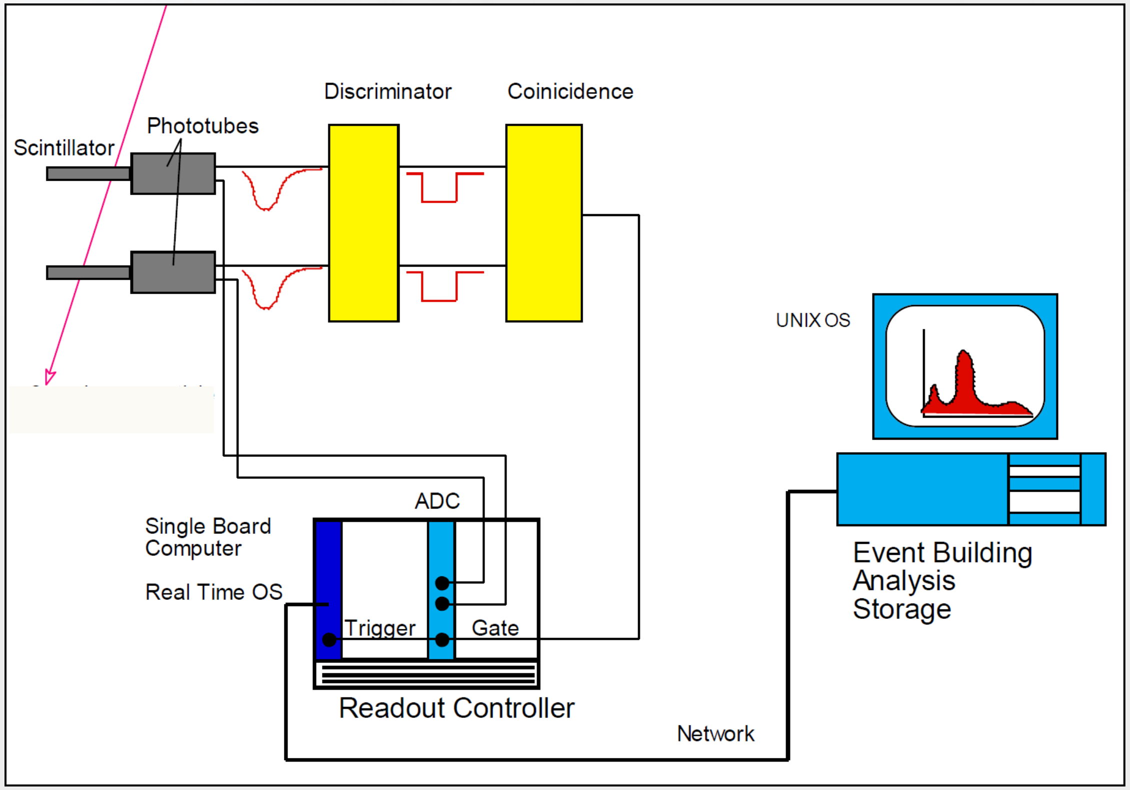
Modules/Buses

- Detectors in the Experimental Halls have many thousands of channels. Each Channel is read (digitized) by an ADC or TDC
  - Pack many circuits onto one board.
  - Pack many boards into a crate
  - Pack the crates into racks
  - Use a standard bus to link everything
- Standards: NIM, CAMAC, FASTBUS, VME, PCI, PCIe, ATCA, micro-TCA

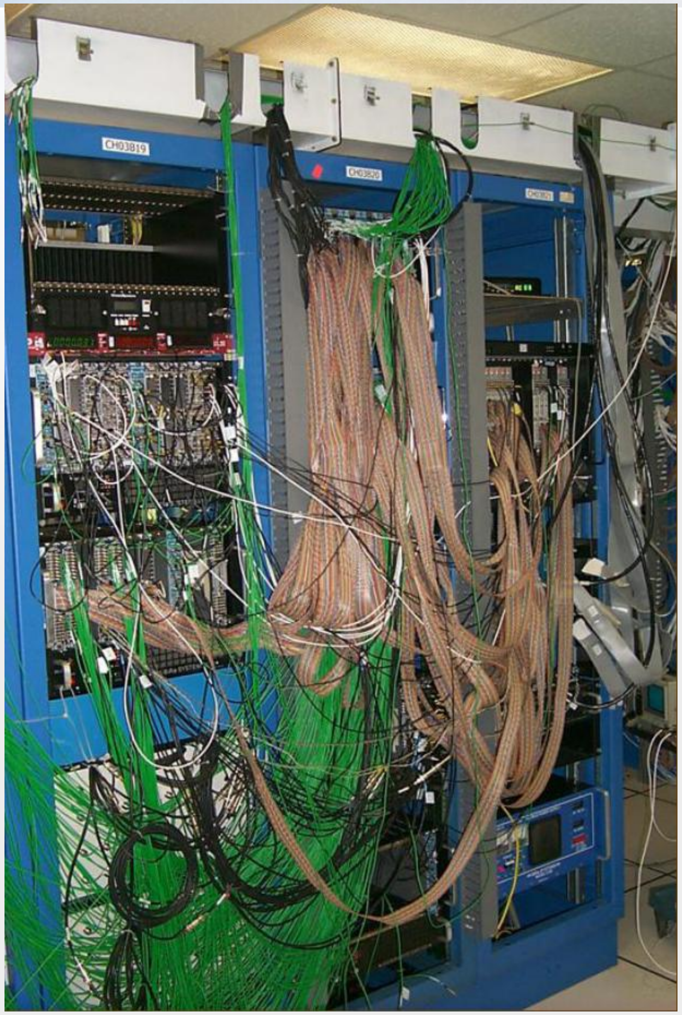


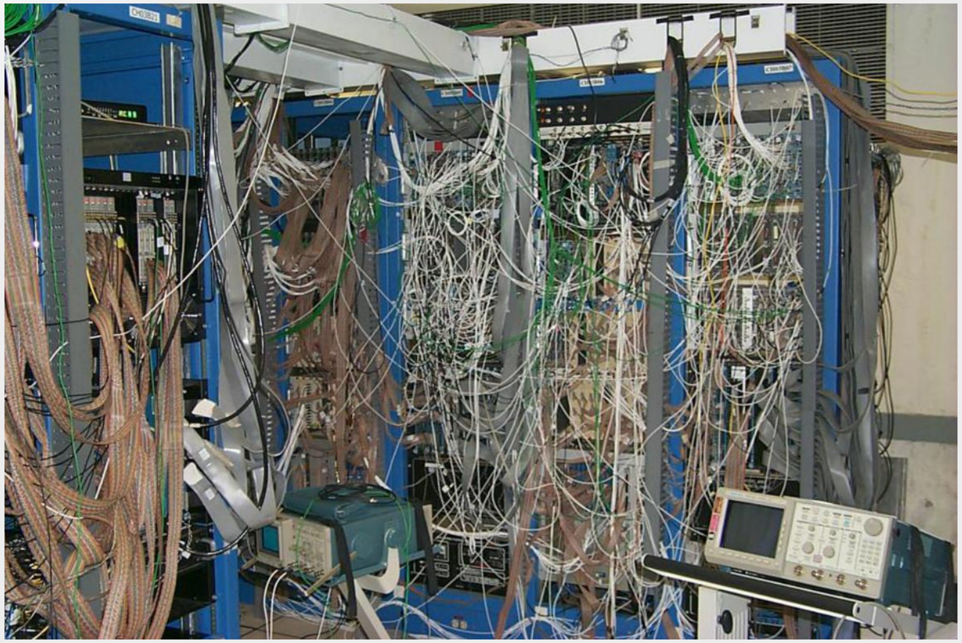
Minimal DAQ System

- The Trigger and ADC "Gate" begin the conversion and readout phase
- Fast (real-time) response to trigger is important to minimize dead-time



The Real World





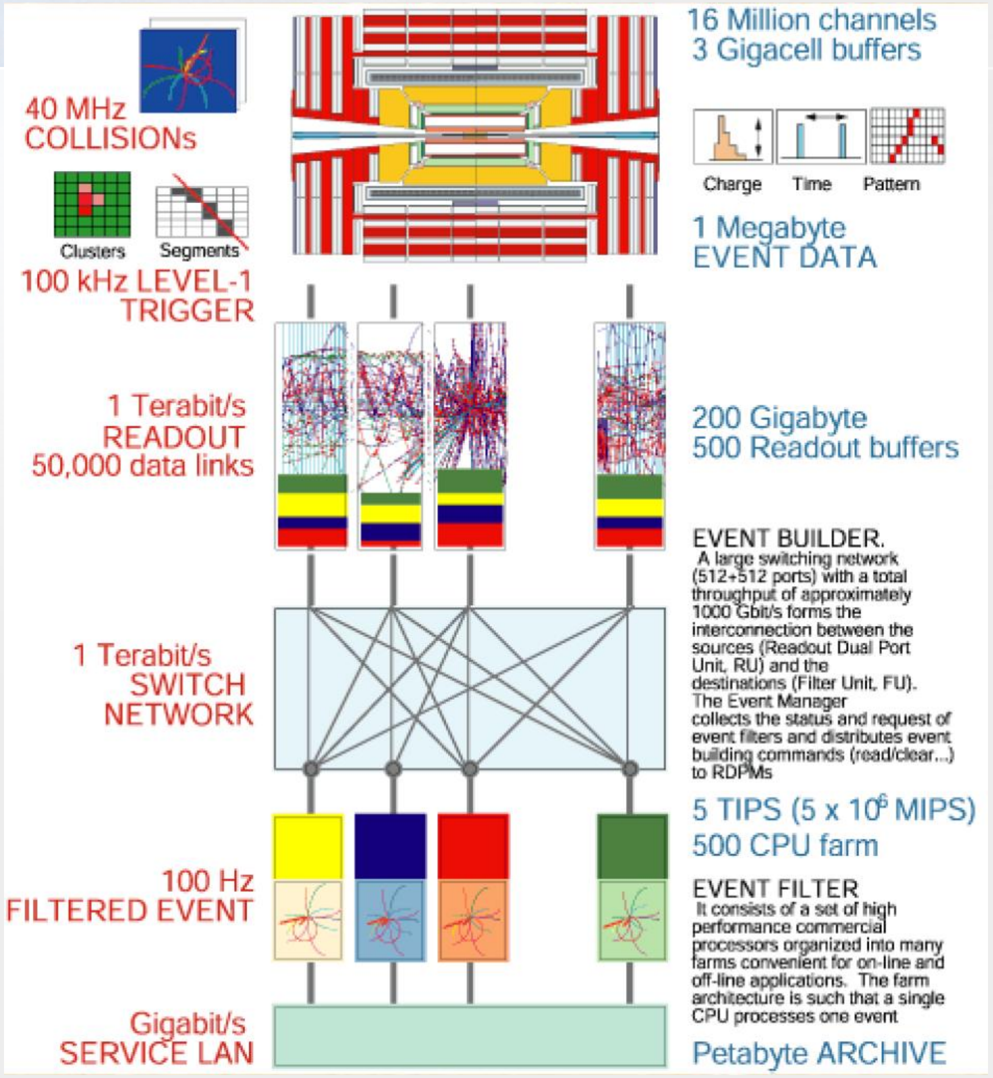
Event Formatting

- The data comes from different detectors.
  - Need to identify the detector ➔ Add channel number, module number …
  - Need to identify which event these data come from ➔ Add event identifier
  - Need to make analysis easier ➔ Sort data
- There is a lot of data
  - Format must be compact
- Analysis can take years
  - Format should be *self documenting*

Data flow

- Once all the data has been digitized them must be collected into a central place for storage
- How the data is moved from the detector readout to the storage medium depends on several factors
  - Available technology
  - Event size and trigger rate
  - Your budget!!
  - Personal taste (Experiment policy)

CMS Data flow



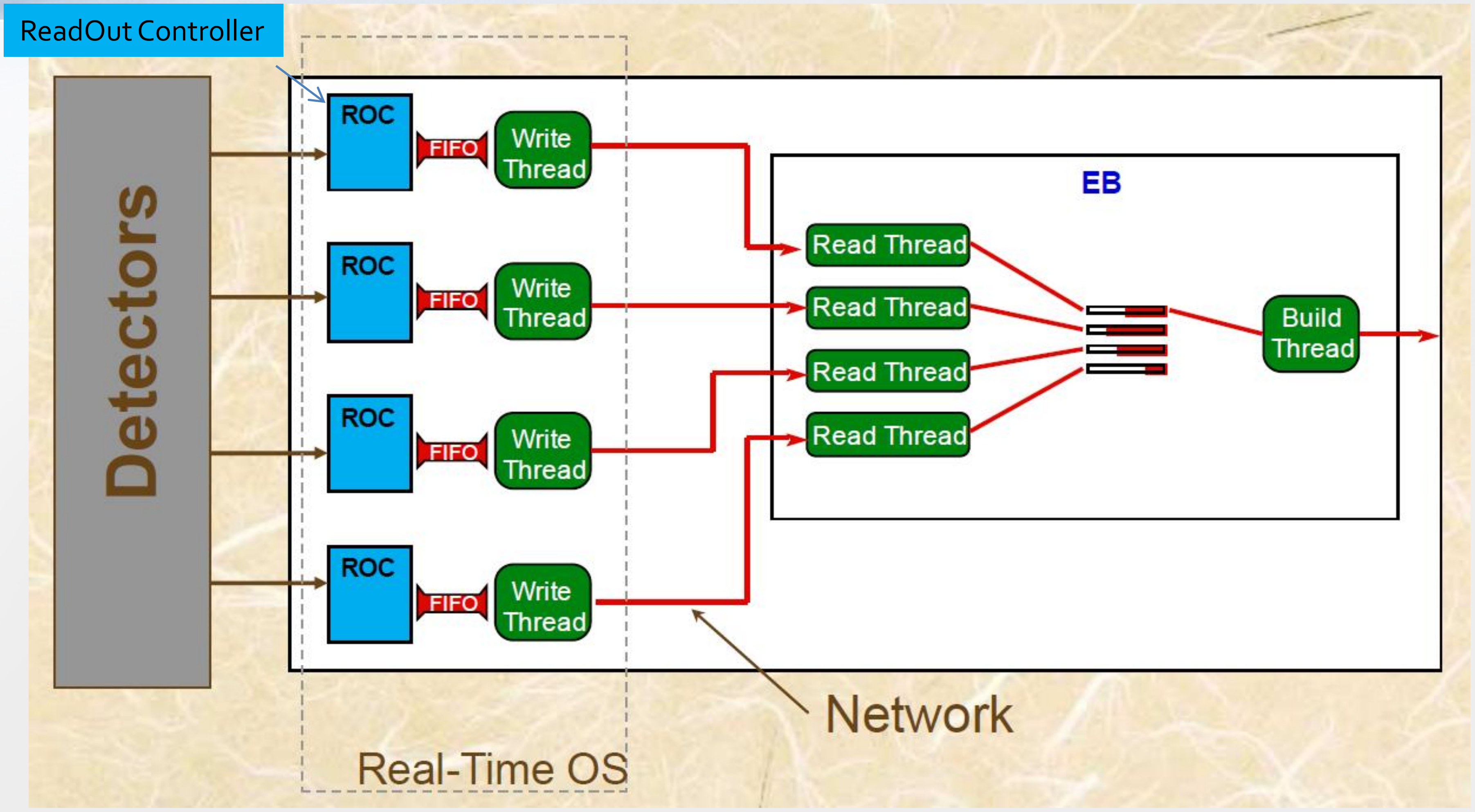
EVENT BUILDER.
A large switching network (512+512 ports) with a total throughput of approximately 1000 Gbit/s forms the interconnection between the sources (Readout Dual Port Unit, RU) and the destinations (Filter Unit, FU). The Event Manager collects the status and request of event filters and distributes event building commands (read/clear...) to RDPMs

EVENT FILTER
It consists of a set of high performance commercial processors organized into many farms convenient for on-line and off-line applications. The farm architecture is such that a single CPU processes one event

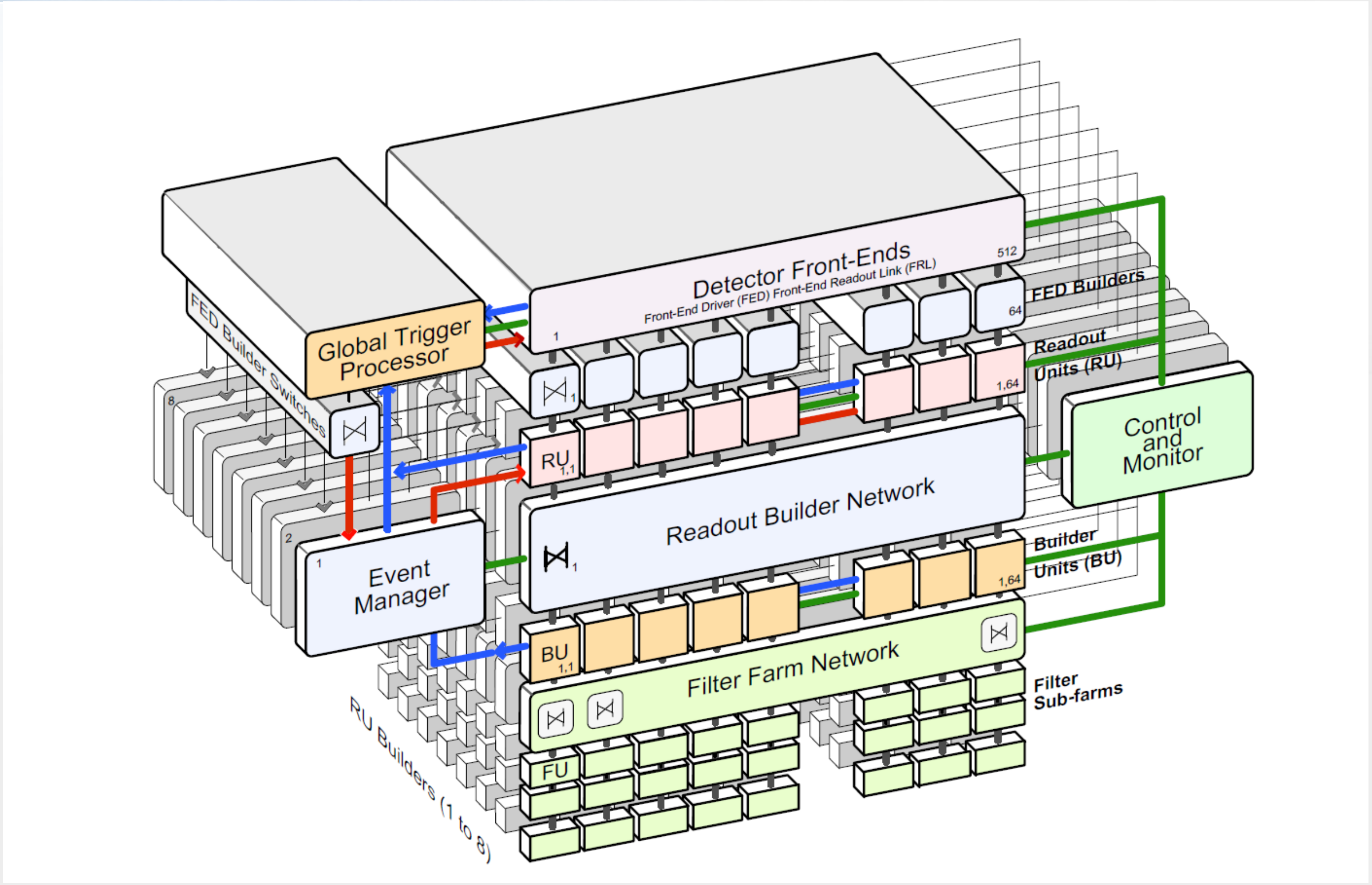
Event Building

- The detectors are spread over a physical volume.
- Bits and pieces of events arrive at different times from different places
- All the parts of the event need to be collected together and packaged with other information needed by the analysis
- The *Event builder* is a very fast collating machine

"Push" Architecture



CMS Event Builder (3D view)



Data Storage

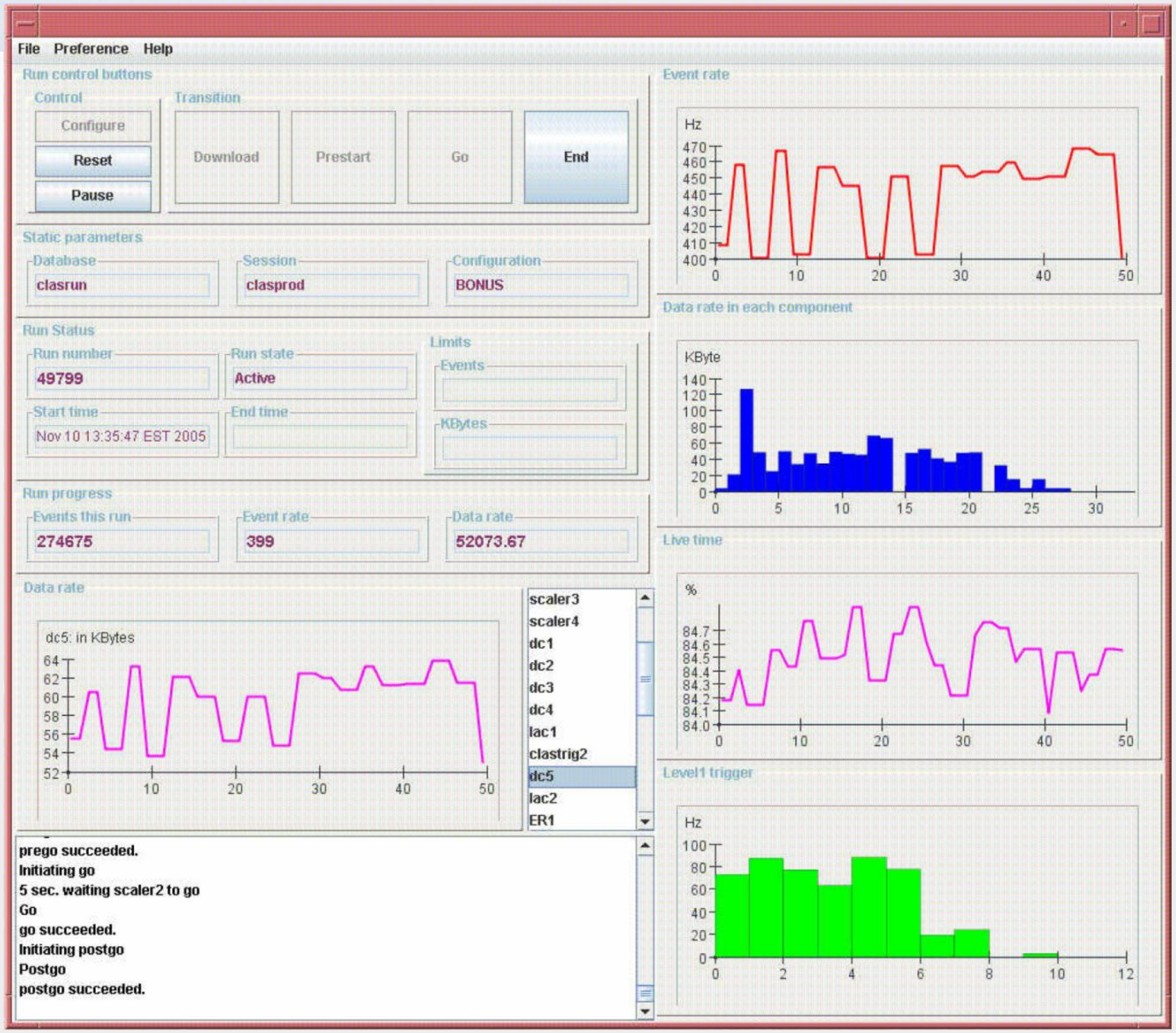
Where to store event data?

- Physics experiments generate a lot of data.
  - At CMS ~ TB/day
- The fastest method is to Disk!!
- The most cost effective method is tape

Run Control

- Need to start and stop the DAQ
- Place to input parameters which change from run to run
- Automatic monitor of the health of the DAQ system
- Something nice for the operator to look at

JAVA Run control



Database	Session	Configuration
clasrun	clasprod	BONUS

Run number	Run state
49799	Active

Start time	End time
Nov 10 13:35:47 EST 2005	

Events this run	Event rate	Data rate
274675	399	52073.67

```
prego succeeded.
Initiating go
5 sec. waiting scaler2 to go
Go
go succeeded.
Initiating postgo
Postgo
postgo succeeded.
```

Monitoring/Analysis

- Need to monitor the data quality as it is read
- Monitoring must not introduce dead-time
- Interface between code written by Physicists and code written by DAQ experts

Slow Controls

- Configuration of frontend electronics and readout modules
- All other data about the experiment which needs to be acquired
  - Power supply voltages
  - Magnetic fields
  - Beam position
  - Target position
  - Vacuum pressure
  - Gas

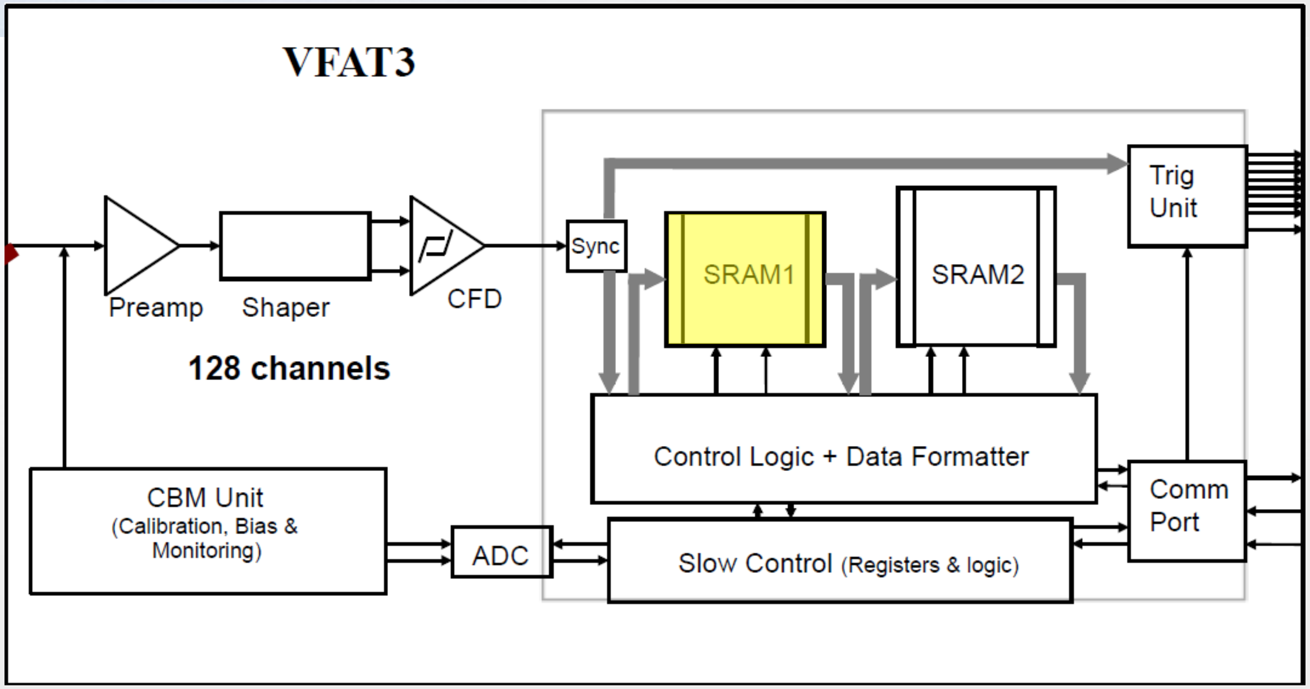
What next?

- Interesting Physics becomes more experimentally difficult (“good” events are more rare)
  - Multilevel trigger
  - Smart hardware trigger
- Technology is always changing.
  - Computer hardware becoming faster (CPU, RAM, NET)
  - More can be done in software (Real time moves to HW)
  - Direct link from readout to CPU memory
- “Customizable” hardware is becoming a viable option (FPGAs , DSPs, ASICs)

GEM detectors for CMS upgrade

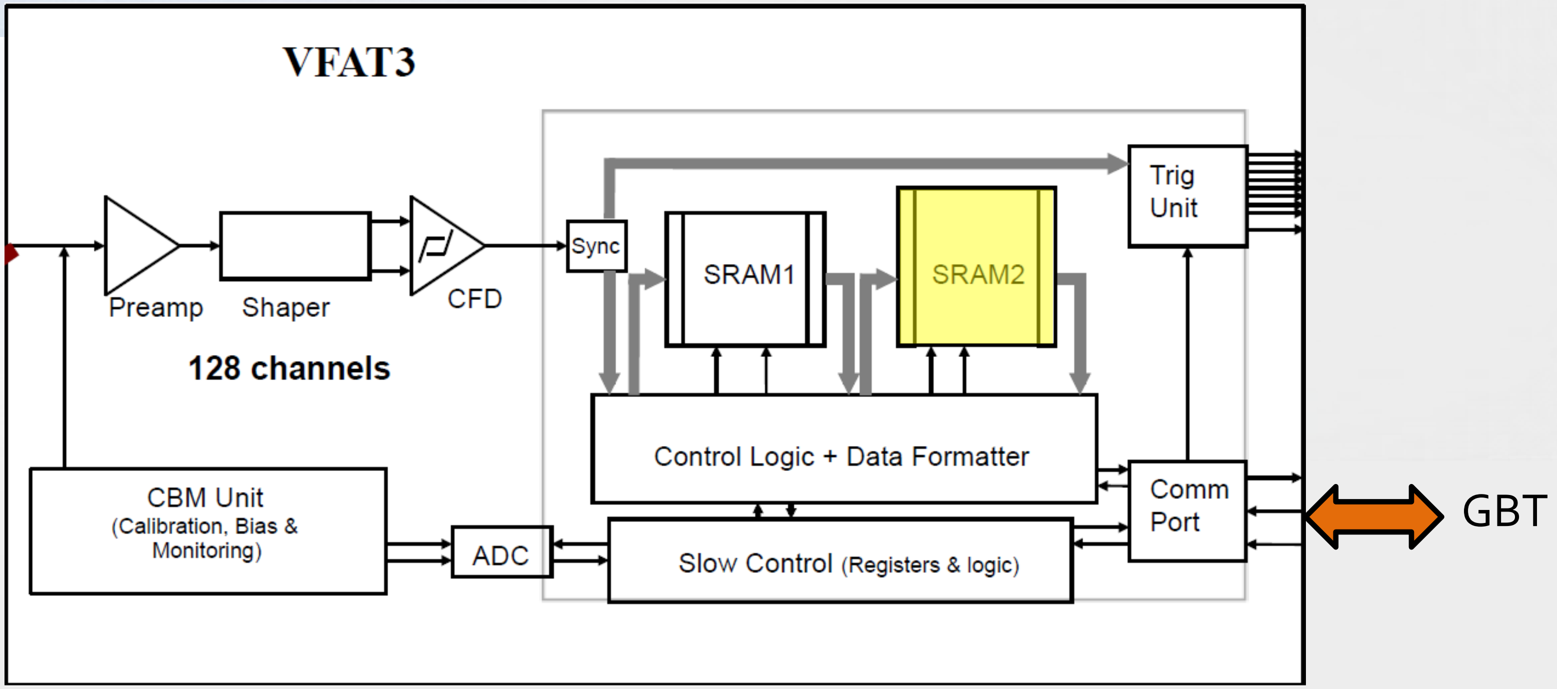
- Frontend chip (VAFT3)
- GEM Chamber electronics
- Communication channel (GBT chipset)
- Readout board (CTP7)

Multi event buffer (dead-timeless DAQ)


VFAT3
Preamp
Shaper
CFD
128 channels
Sync
SRAM1
SRAM2
Trig Unit
Control Logic + Data Formatter
Comm Port
CBM Unit (Calibration, Bias & Monitoring)
ADC
Slow Control (Registers & logic)

- SRAM1: Circular buffer continuously sampling all channels every clock cycle
- Max trigger latency = 1024 clock cycles (~25.6 us)
- At trigger data is read according to "LAT register"

Data Buffer



VFAT3
Preamp
Shaper
CFD
128 channels
Sync
SRAM1
SRAM2
Trig Unit
Control Logic + Data Formatter
Comm Port
CBM Unit (Calibration, Bias & Monitoring)
ADC
Slow Control (Registers & logic)
GBT

- SRAM2: FIFO 176x512
- At each trigger data are moved from SRAM1
- Data are read from transmit logic as soon as possible
- Many triggers can be processed on consecutive clock cycles

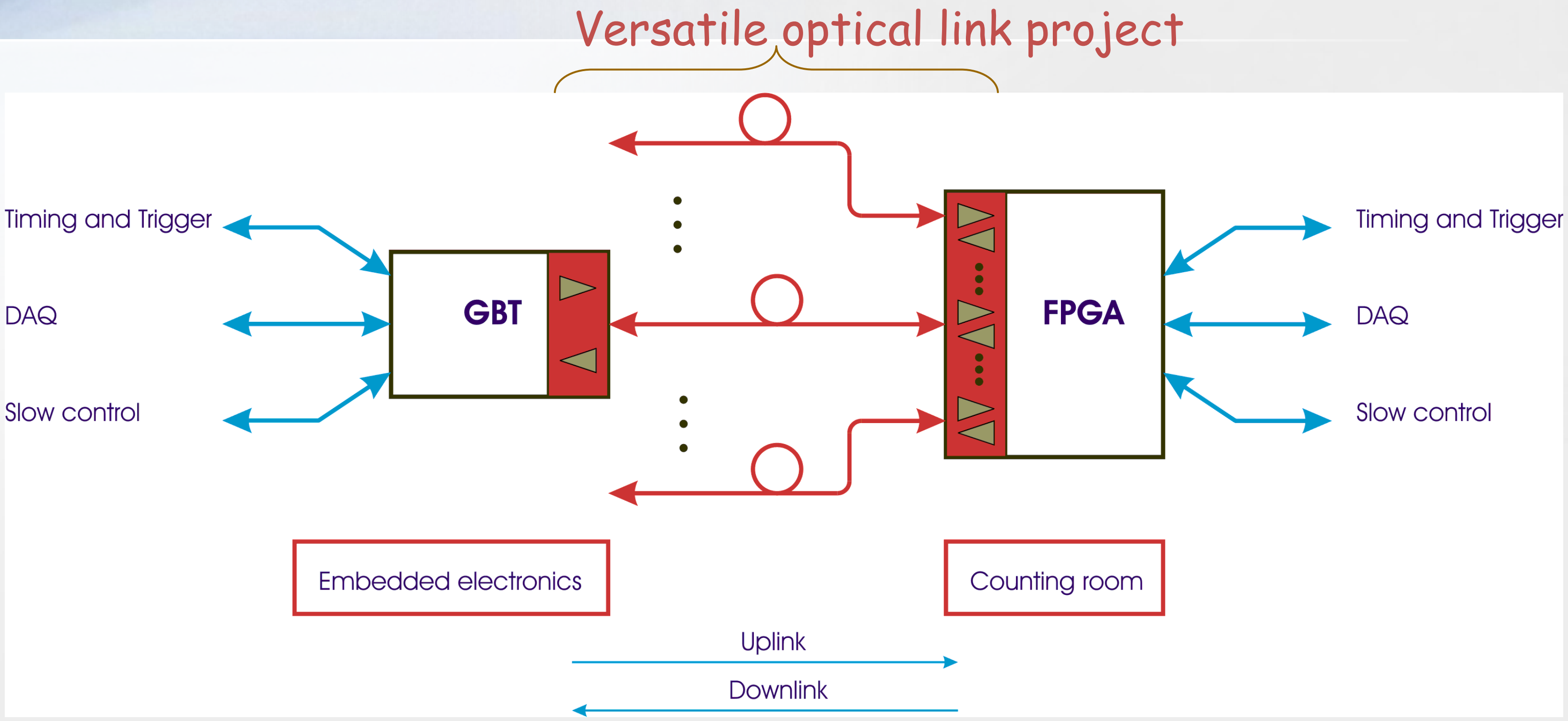
Complications

You have to store/send data with additional information to check data integrity, trigger distribution and data read correctness:

- Trigger number
  - Initialized at run start and incremented at each trigger
- Time of event
  - A counter is used to keep trace of trigger arrival time
  - Timestamp is the value of this counter store along with data
  - A global synchronization signal is needed to initialize time counter (in CMS BCN0)

Trigger number + timestamp should match for all data

GBT link architecture



- Bidirectional point-to-point optical fiber links
- The heart of the link is the GigaBit Transceiver (GBT13)
- Each link carries simultaneously: DAQ, Timing, Trigger and SC data
- Embedded transceiver: the GBT Chipset ➔ Radiation-Hard Hardware components
- Counting room transceivers: FPGA and COTS optoelectronics components ➔ Soft block

CMS GEM Electronic System

GEM fiber links consist of:

1. GBTx-based DAQ and control path provides full resolution hit information to DAQ via GEM uTCA crate

2. Trigger data is half-resolution OR of two adjacent strips.

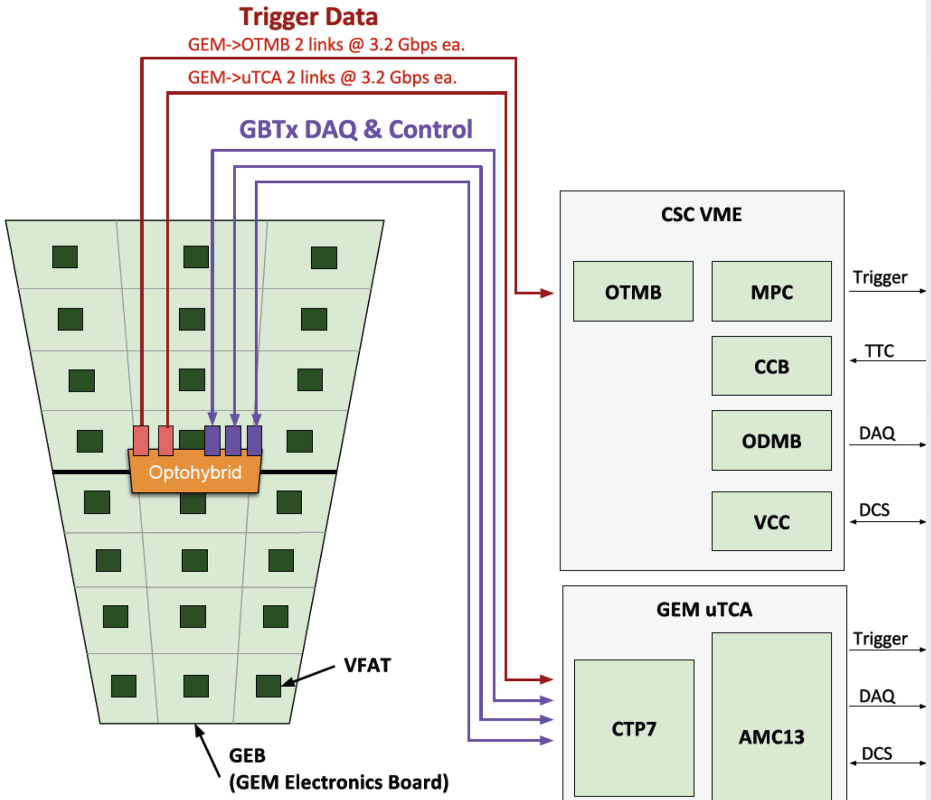
   FPGA based trigger path provides two identical copies of trigger data to:

   (1) GEM uTCA backend electronics

   - Connects GEMs to muon trackfinder

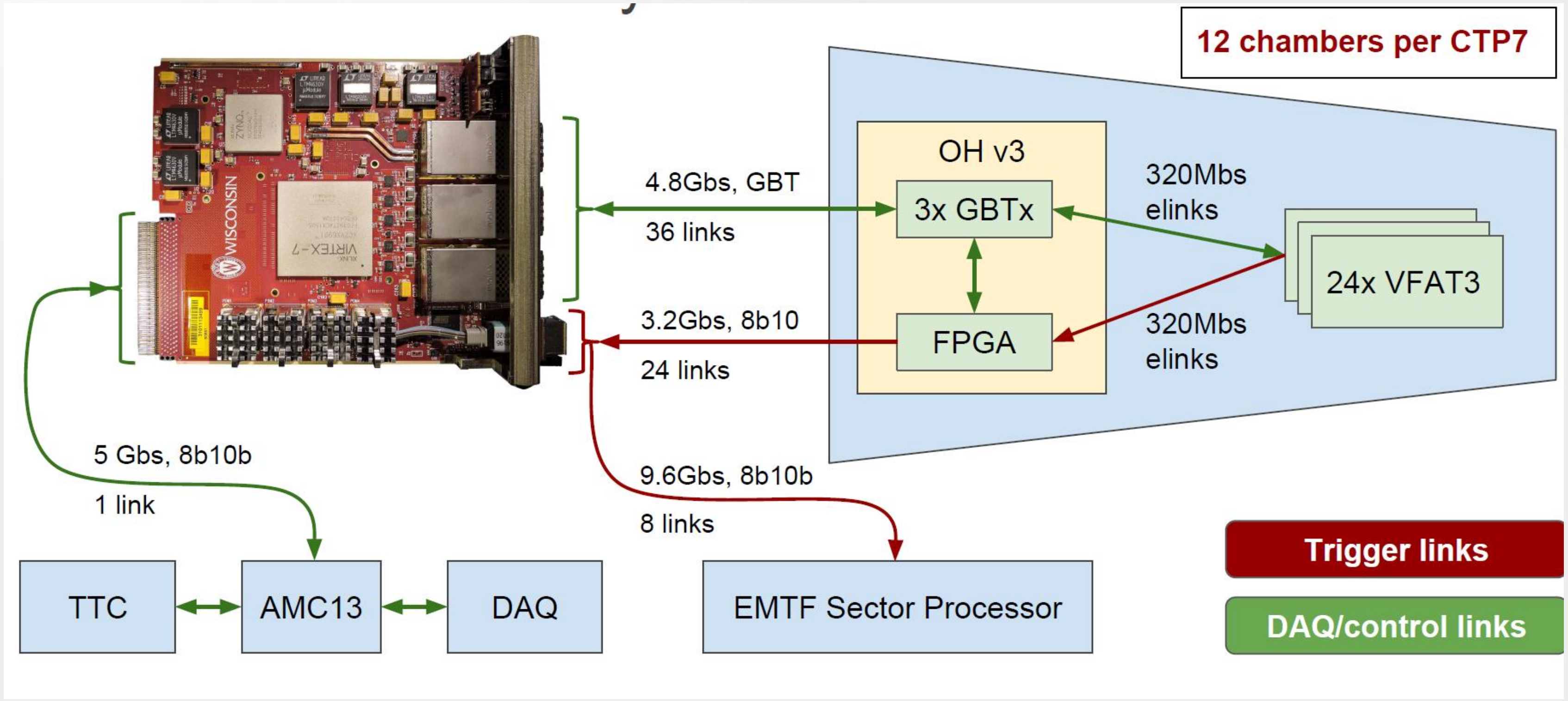
   (2) CSC trigger motherboard (OTMB)

   - Provides capabilities for combined GEM+CSC trigger primitive generation



CMS GEM Electronic System -2

- CTP7 has a large Virtex7 (690T -2) FPGA, and provides 67 optical receivers and 48 optical transmitters, all of which can run at up to 10Gb/s

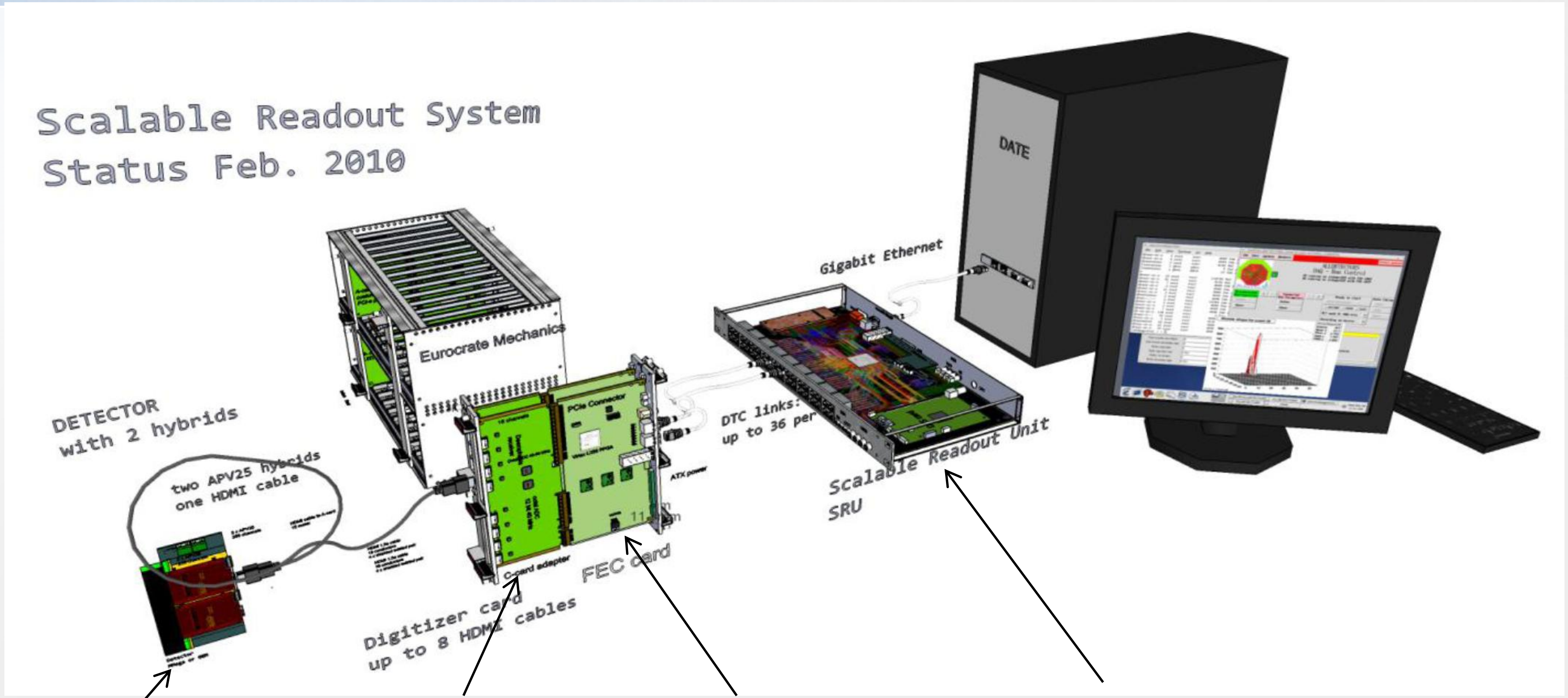


Electronics for tests

- SRS (Scalable Readout System)
  - Developed by the RD51 collaboration

- MOSAIC (MOdular System for Acquisition Interface and Control)
  - Developed in Bari for ALICE Inner Tracker System testing

Scalable Readout System (SRS)



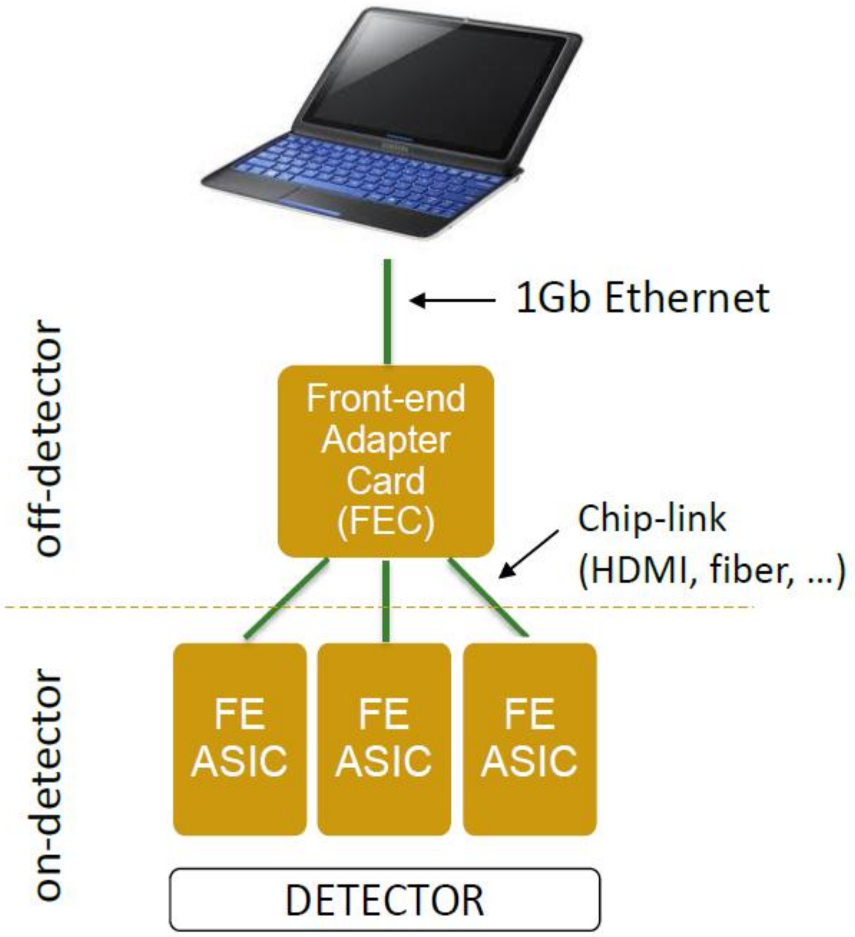
Frontend Hybrid

Digitizer Card

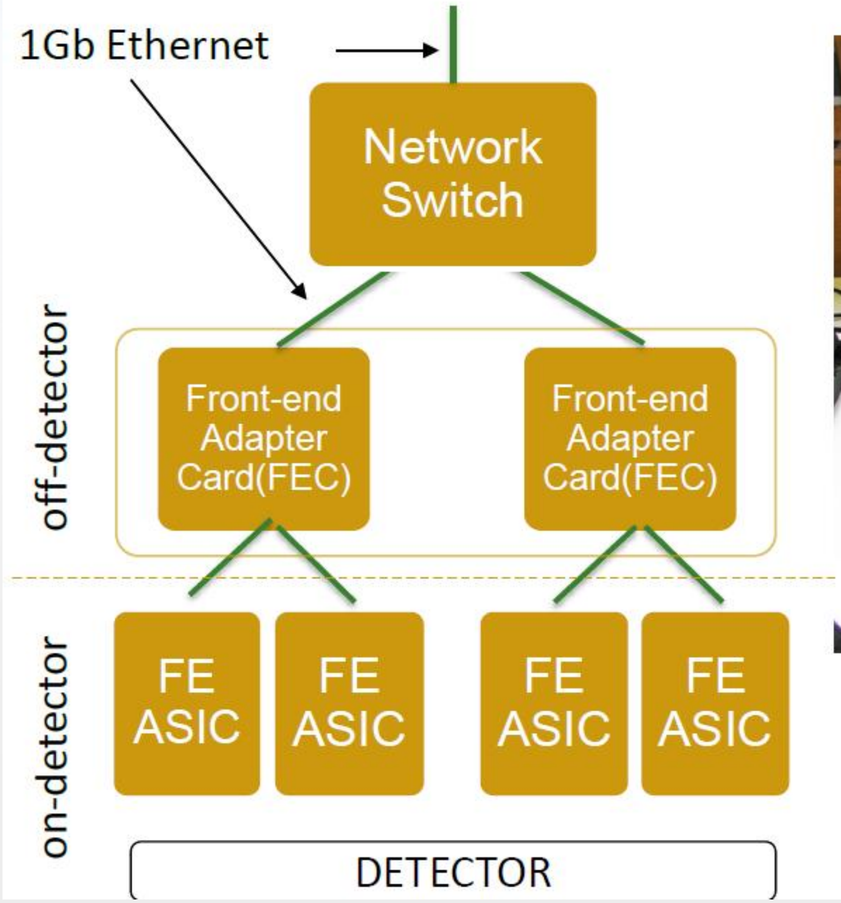
FEC Card

Scalable Readout Unit

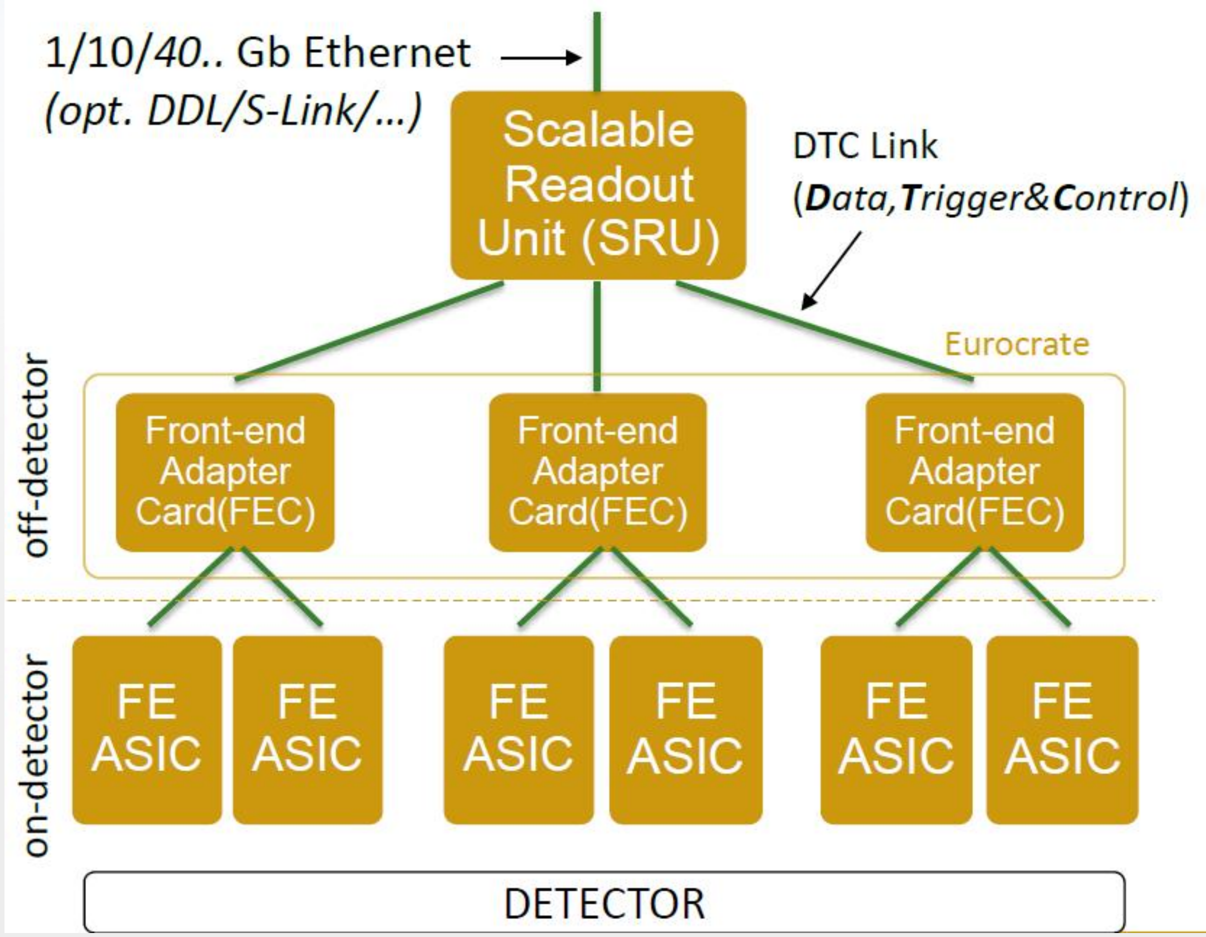
SRS Small System

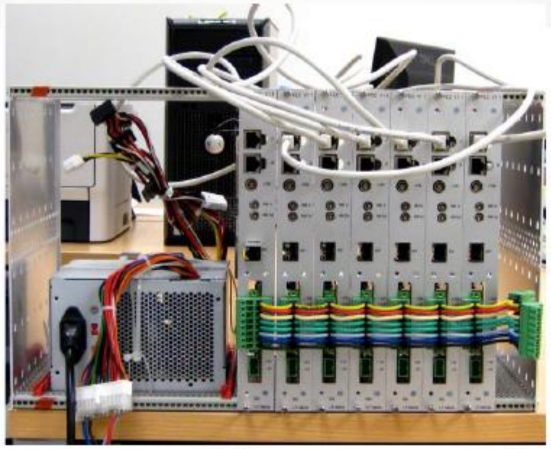


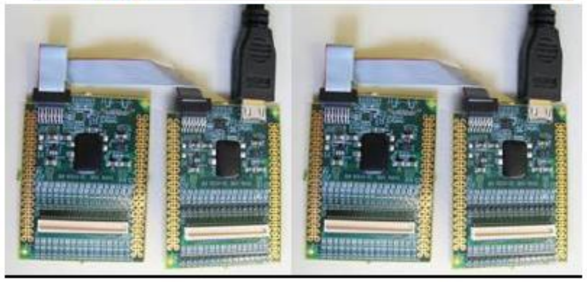
SRS Medium System



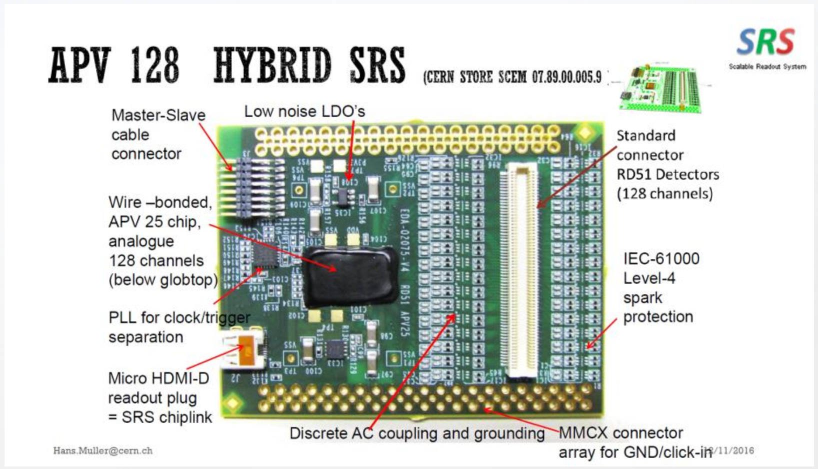
SRS Large System

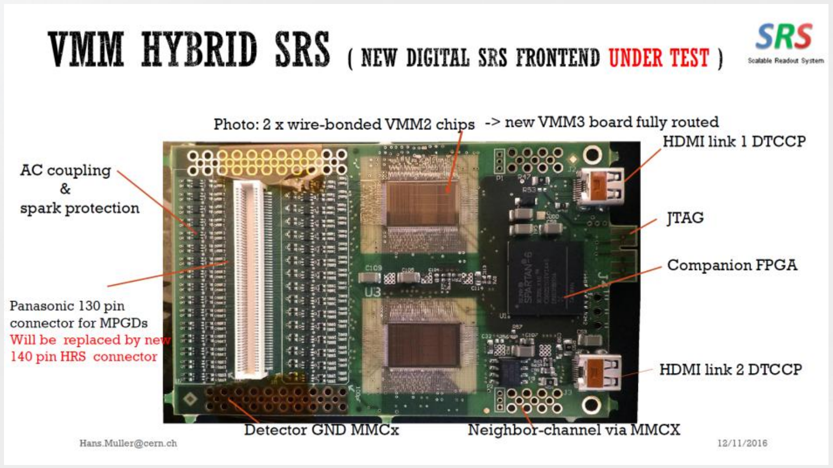






SRS - Hybrids


APV 128 HYBRID SRS (CERN STORE SCEM 07.89.00.005.9
SRS
Scalable Readout System
Master-Slave cable connector
Low noise LDO's
Standard connector RD51 Detectors (128 channels)
Wire –bonded, APV 25 chip, analogue 128 channels (below globtop)
IEC-61000 Level-4 spark protection
PLL for clock/trigger separation
Micro HDMI-D readout plug = SRS chiplink
Discrete AC coupling and grounding
MMCX connector array for GND/click-in
Hans.Muller@cern.ch
12/11/2016


VMM HYBRID SRS (NEW DIGITAL SRS FRONTEND UNDER TEST)
SRS
Scalable Readout System
Photo: 2 x wire-bonded VMM2 chips -> new VMM3 board fully routed
HDMI link 1 DTCCP
AC coupling & spark protection
JTAG
Companion FPGA
Panasonic 130 pin connector for MPGDs
Will be replaced by new 140 pin HRS connector
HDMI link 2 DTCCP
Detector GND MMCx
Neighbor-channel via MMCX
Hans.Muller@cern.ch
12/11/2016

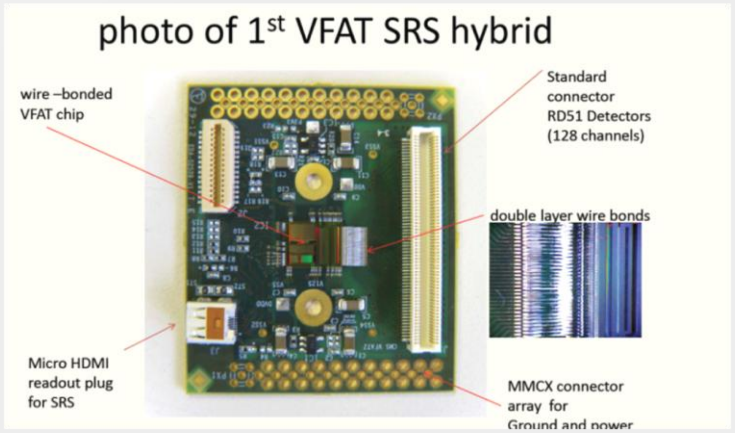
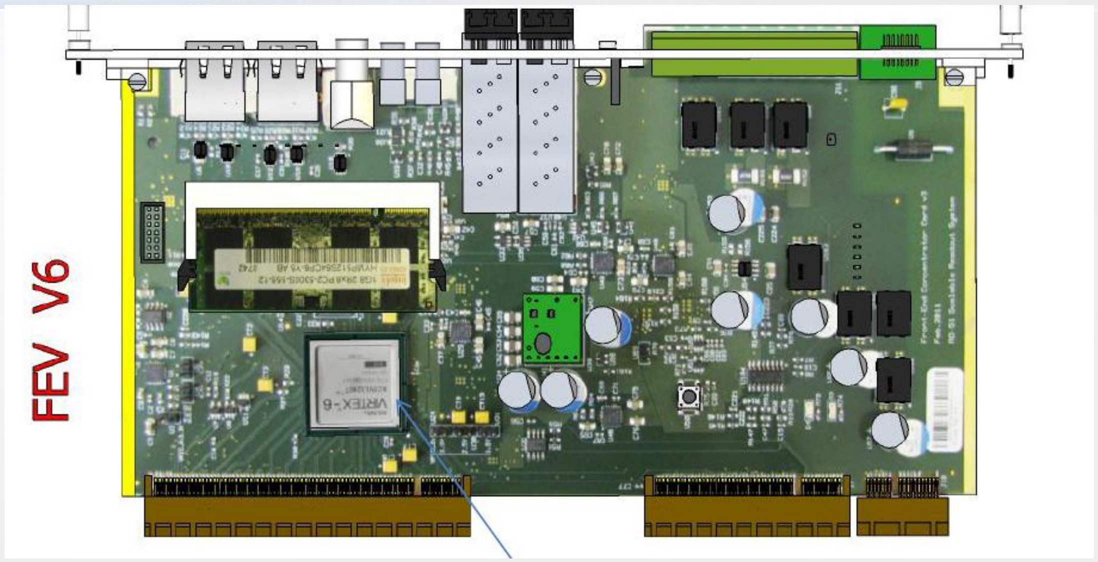


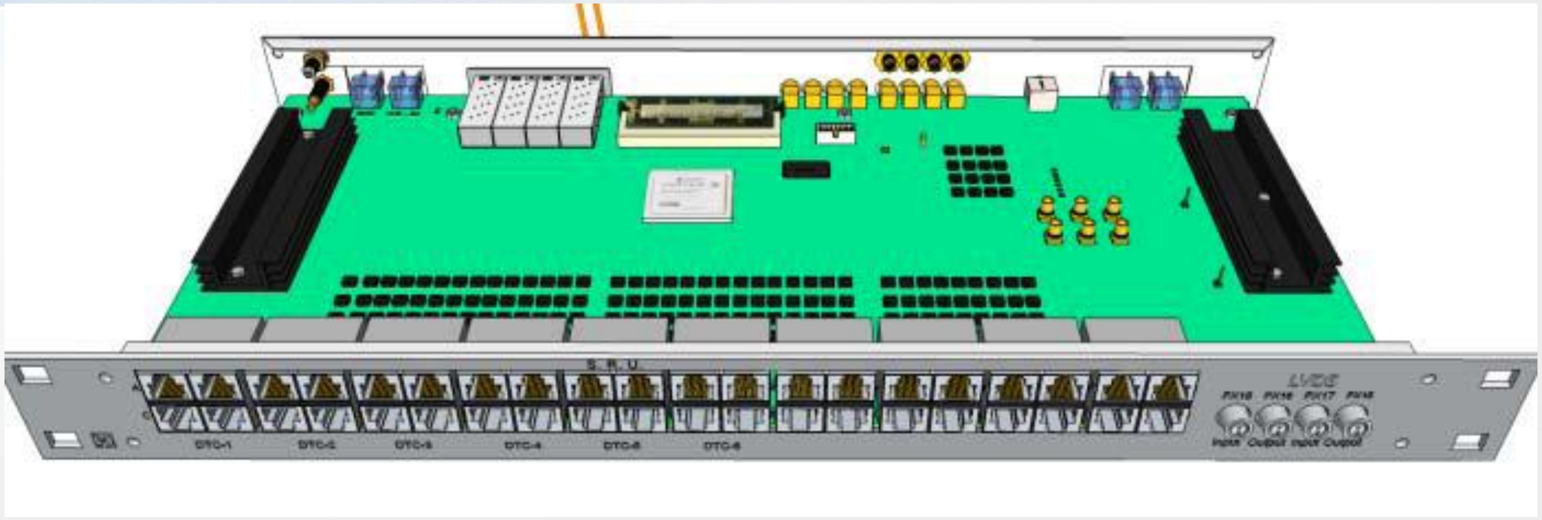
photo of 1st VFAT SRS hybrid
wire –bonded VFAT chip
Standard connector RD51 Detectors (128 channels)
double layer wire bonds
Micro HDMI readout plug for SRS
MMCX connector array for Ground and power

FEC V6 Frontend Card



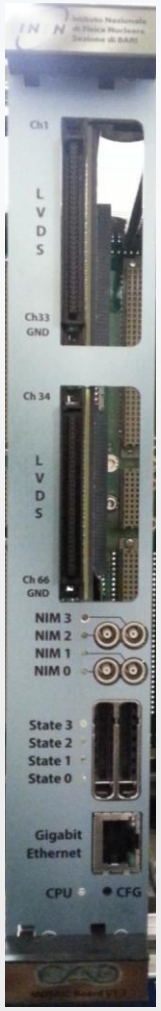
- Virtex6 FPGA
- 2 x SPF+ connectors (1 Gb Ethernet)
- 2 NIM inputs
- 1 LVDS input
- Rear connectors PCIex1, PCIex8, PCIex16

Scalable readout Unit (SRU)



- Front-panel with 40 DTC links to the FEC cards
- 4 SFP+ ports that can be used either with 10 Gigabit Ethernet copper or fibre links
- user-defined pulse interface for 100 OHM LVDS or 50 OHM NIM signals

MOSAIC Board



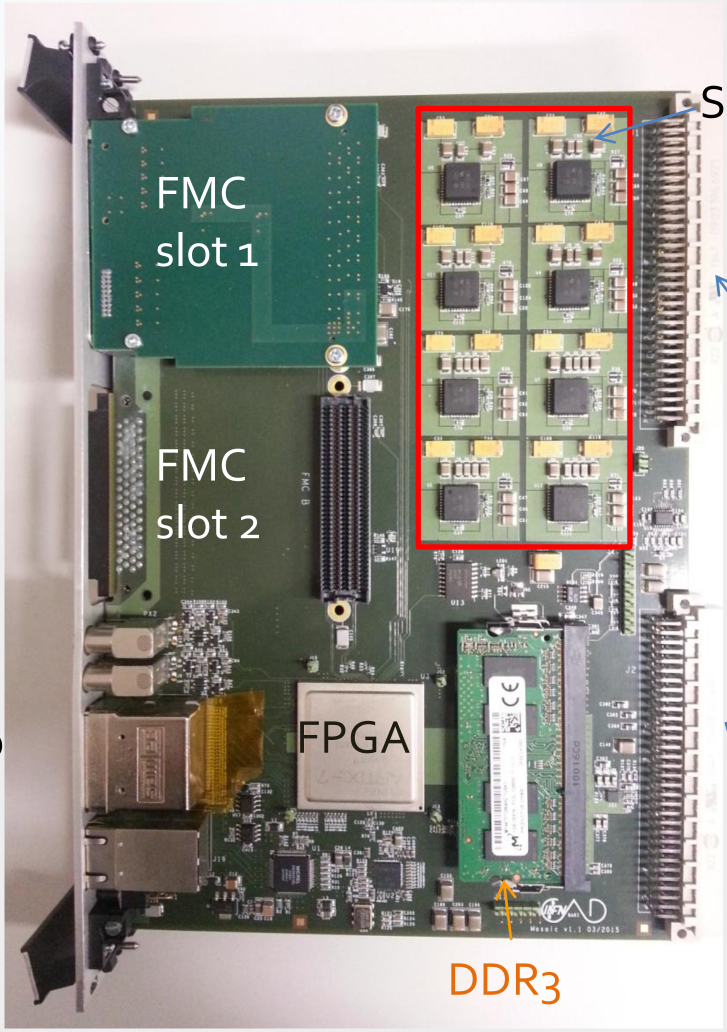
General purpose LVDS I/O

General purpose LVDS I/O

NIM I/O

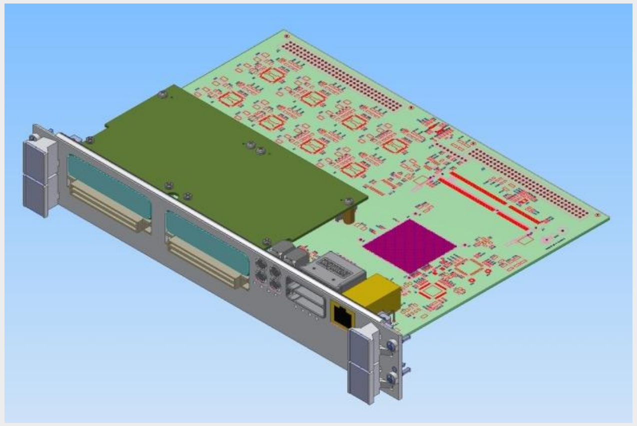
HI Speed I/O

Gigabit Ethernet



Switching voltage regulators

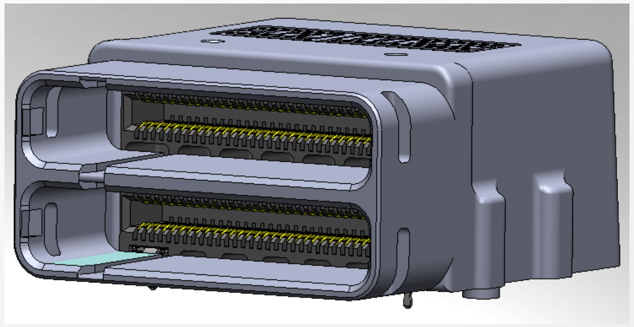
Backplane connectors

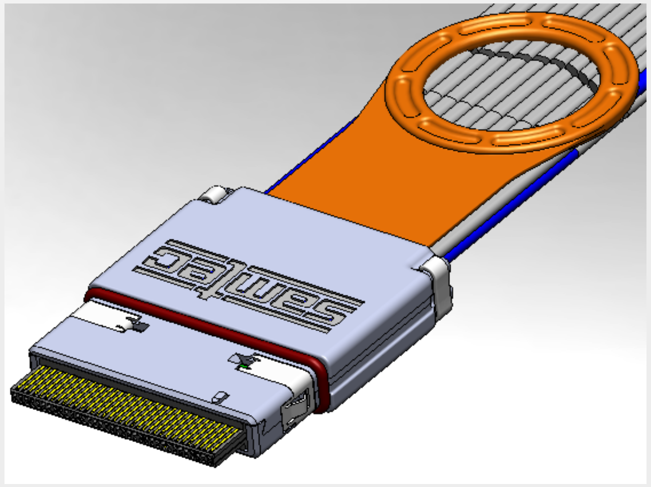


MOSAIC board overview

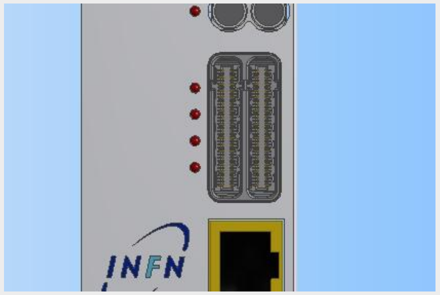
- VME board (only for power supply)
- 10 Hi speed Input + 10 Hi speed output Up to 6.6 Gb/s
- 2 x 33 general purpose LVDS I/O
- 2 expansion slots – FMC Mezzanine boards
- 4 NIM configurable I/O
- 1 DDR3 – SODIMM connector with 1GB module
- Gigabit Ethernet interface
- Xilinx FPGA: XC7A200T-2 FFG1156C

Hi speed connector & cable

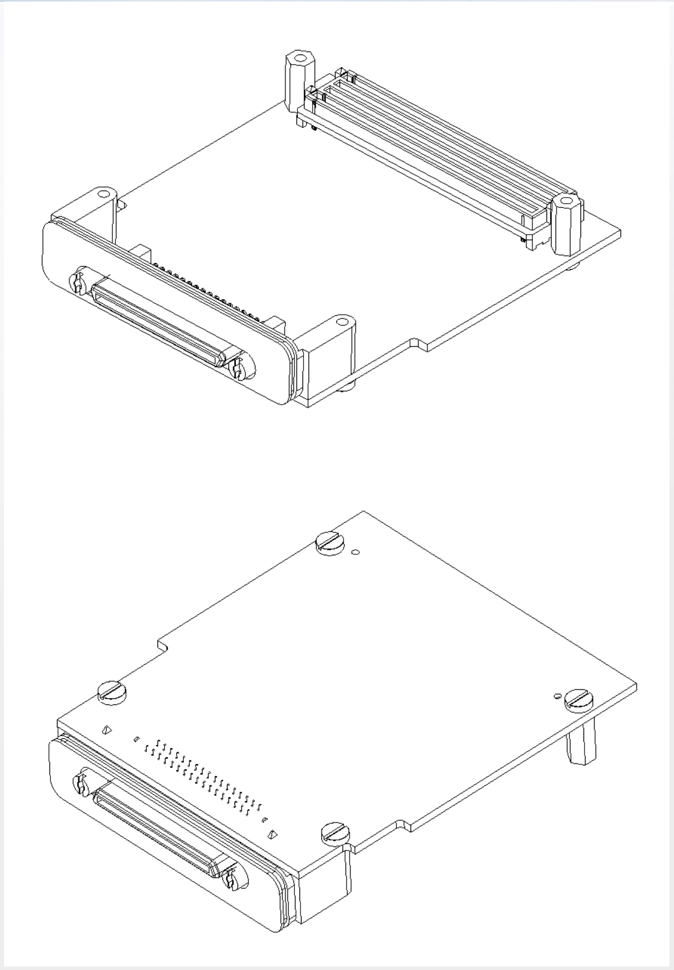




- 24 differential pairs for each cable (12 for side)
- 2 Cable for connector (1 Unused)
- Compact assembly (28 mm x 32 mm with shield)
- Standard cable length:
  - 1 m -> 6.44 GHz bandwidth
  - 2m -> 2.56 GHz bandwidth
- Signals
  - 10 Hi speed TX
  - 10 Hi speed RX
  - 2 LVDS Clock
  - 2 MLVDS Control

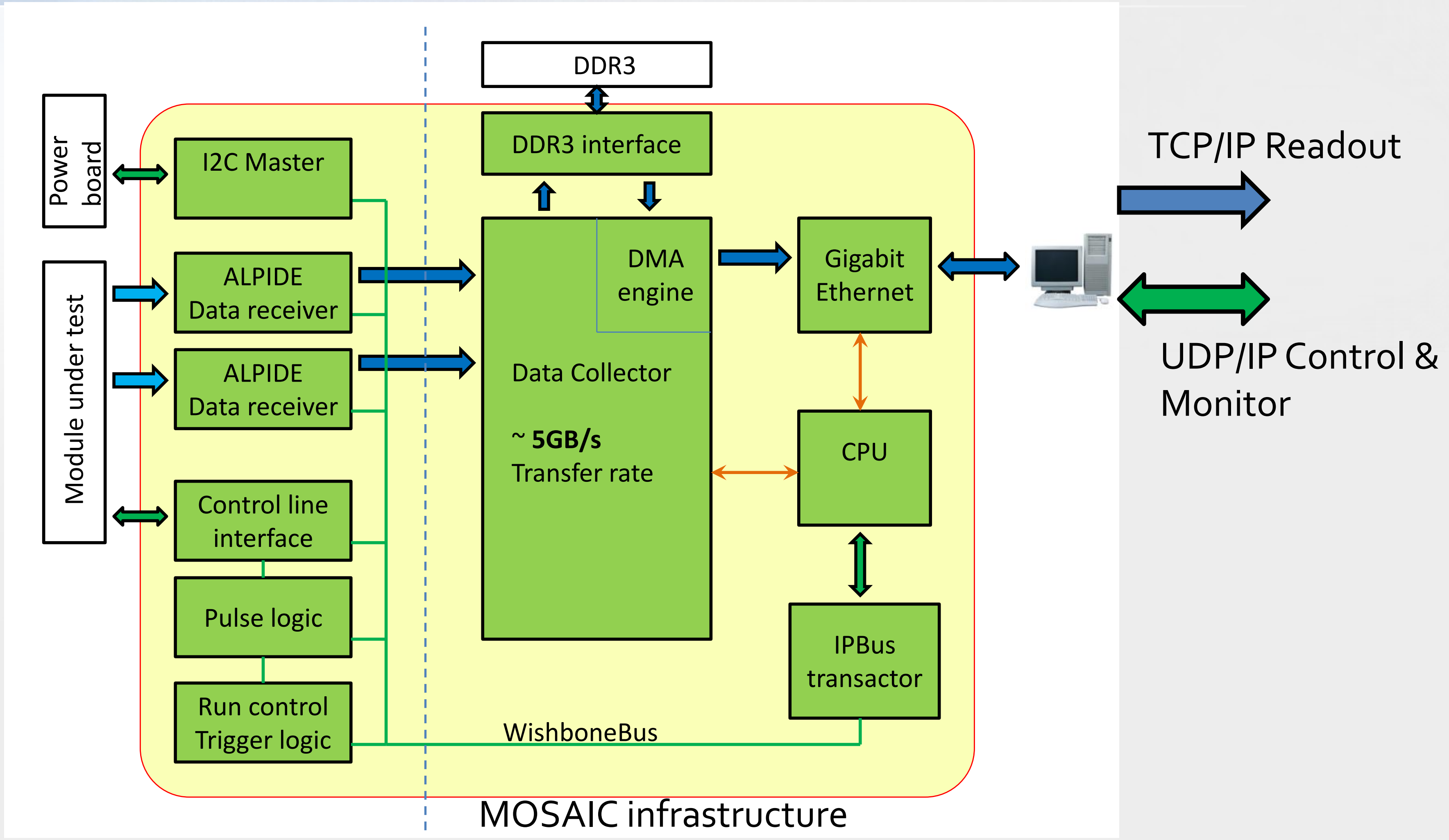


FMC Expansion slots

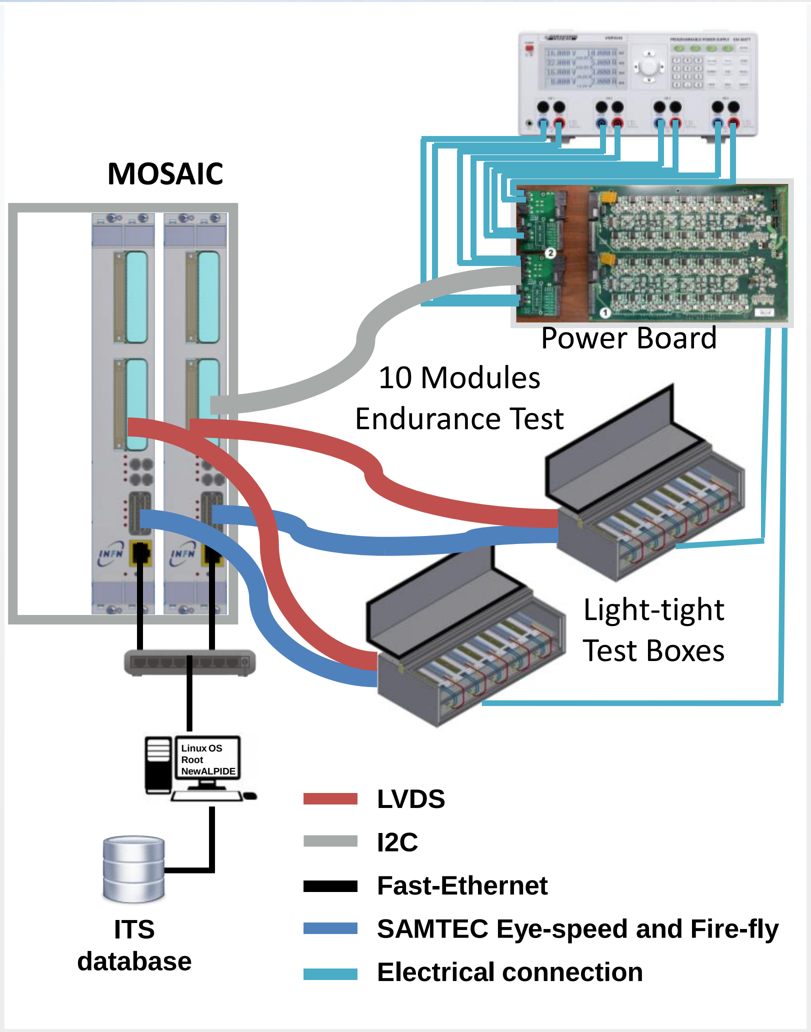


- 2 FPGA Mezzanine Card (FMC) Low Pin Count or 1 Double Width Card
  - 34 diff pairs lines or 68 single ended => User defined
  - 2 diff clock lines => Clock
  - 1 Bidirectional High speed transceiver (up to 6.6 Gbps)
- Conform to the VITA Standard ANSI/VITA 57.1

Firmware block diagram – ALPIDE TB



ALICE ITS Test Setup - Endurance



2	MOSAIC Boards
20	High Speed link
20	Control interfaces
1	I2C interface

10	Modules under test
140	ALPIDE chips
73.4	Million of channels

The End