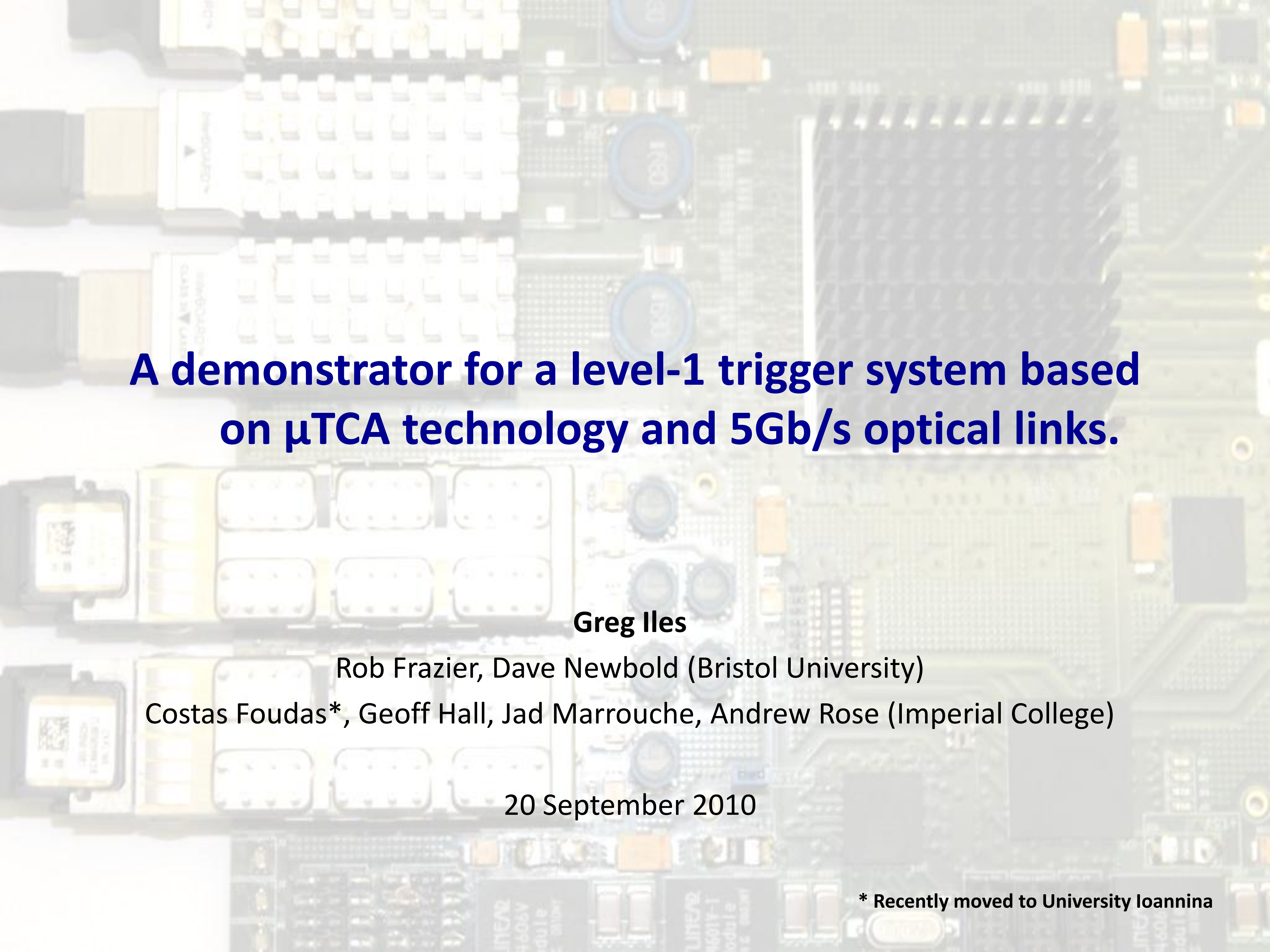


A demonstrator for a level-1 trigger system based on µTCA technology and 5Gb/s optical links.

Greg Iles

Rob Frazier, Dave Newbold (Bristol University)

Costas Foudas*, Geoff Hall, Jad Marrouche, Andrew Rose (Imperial College)

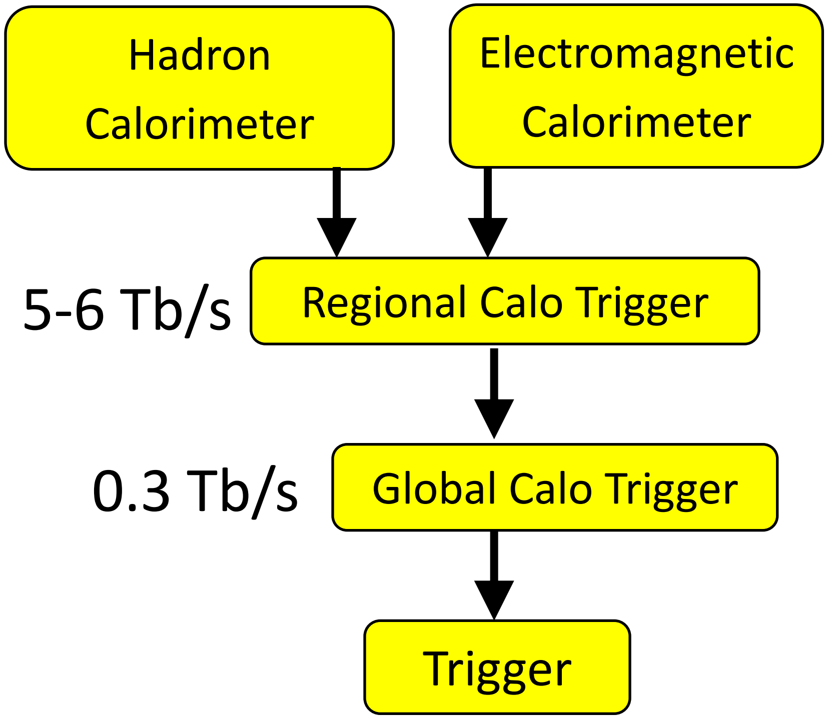
20 September 2010

*** Recently moved to University Ioannina**

CMS Calorimeter Trigger

Main task:
Find jets and electrons from energy depositions inside the calorimeters.

Sort in order of importance and send 4 most significant to global trigger.



Hadron Calorimeter

Electromagnetic Calorimeter

5-6 Tb/s

Regional Calo Trigger

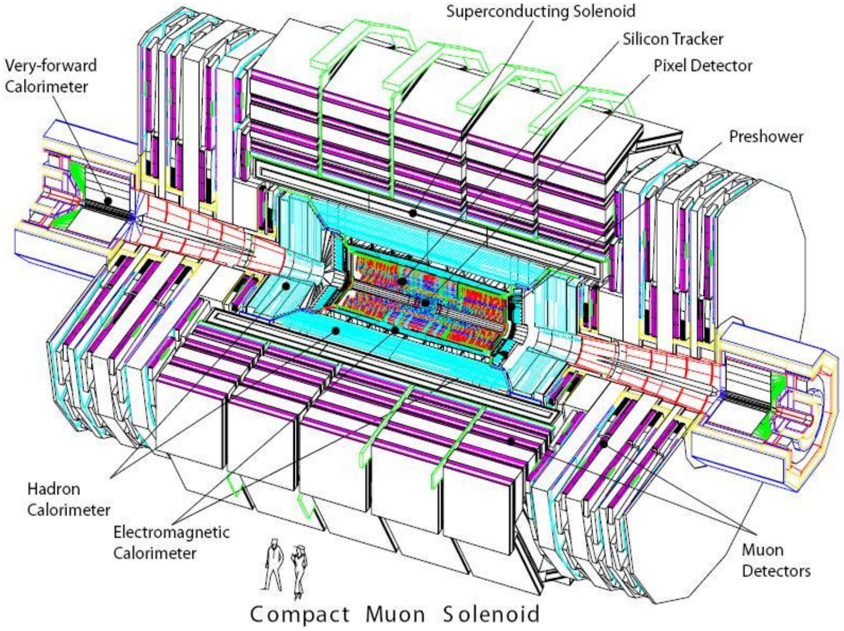
0.3 Tb/s

Global Calo Trigger

Trigger

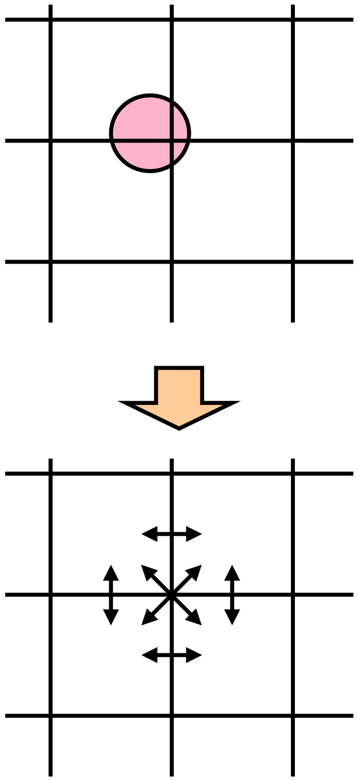
Potential benefits from upgrade:

- The trigger is effectively an image processor. Better resolution?
- Extra physics quantities?
- Better reliability?

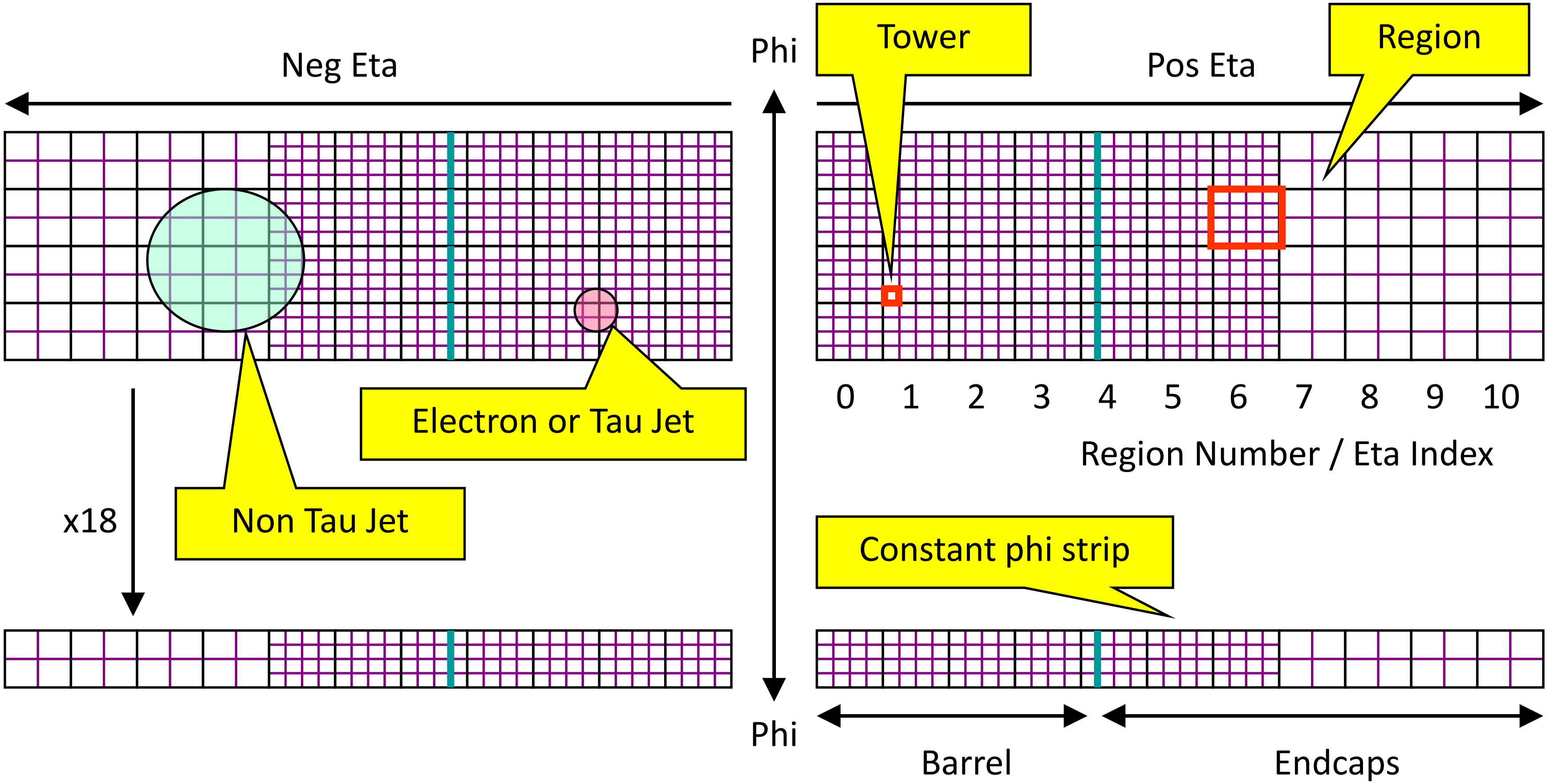


Requirements for an upgrade

- Must process 6 Tb/s
- Not a problem, just make it parallel, but....
  - Need to build physics objects, which don’t observe detector granularity!
    - Data sharing
    - Data duplication
  - Need to sort physics objects
    - Avoid multi stage sort to minimise latency
    - Restricts number of “physics builders” due to fan in constraints
  - Only have approx 1us
    - Each serialisation is 100ns - 200ns



Incoming Trigger Data

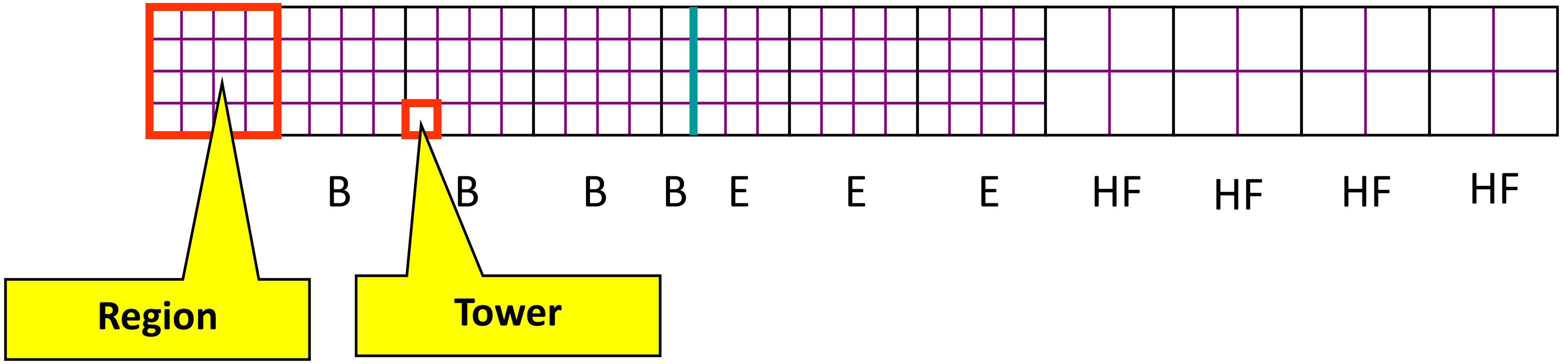


Data per tower, per bx

ECAL: 8bits energy + FGV (FineGrainVeto)

HCAL: 8bits energy + MIP (MinIonisingParticle)

Geometry & link capacity



- Number of Regions: 22 (eta) x 18 (phi)
- 1 Region = 4x4 Towers
- 1 Link at 2.4Gb/s (8B/10B)
  - Equivalent to 4 towers
  - 6 bytes per bx
  - 12bits per tower
  - Currently use 9bits per tower

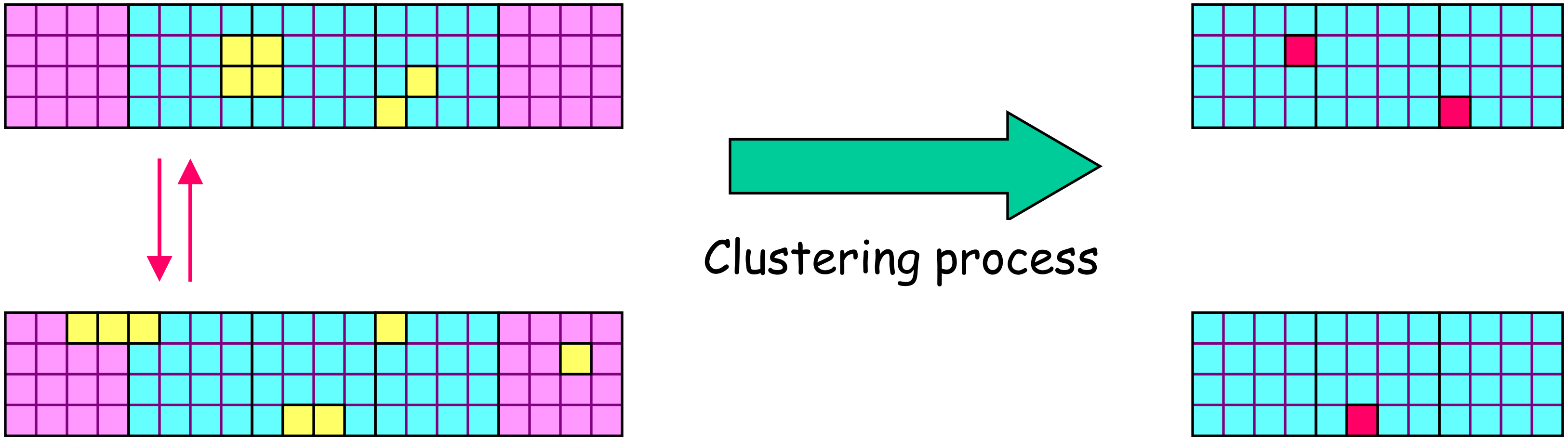
- Total number of links = 2160 links @ 2.4Gb/s
- Total BW = 5.2Tb/s

Build Level-1 Calo Trigger Demonstartor

- Why?
  - Real life always far more complicated than gant chart
    - e.g. Local clock routing from MGT to fabric in Virtex II Pro took 6 months to fully understand in last calorimter project.
  - Understand system issues
    - Some things only obvious in hindsight
  - Enable firmware and software to get underway
    - Always lags in any project...

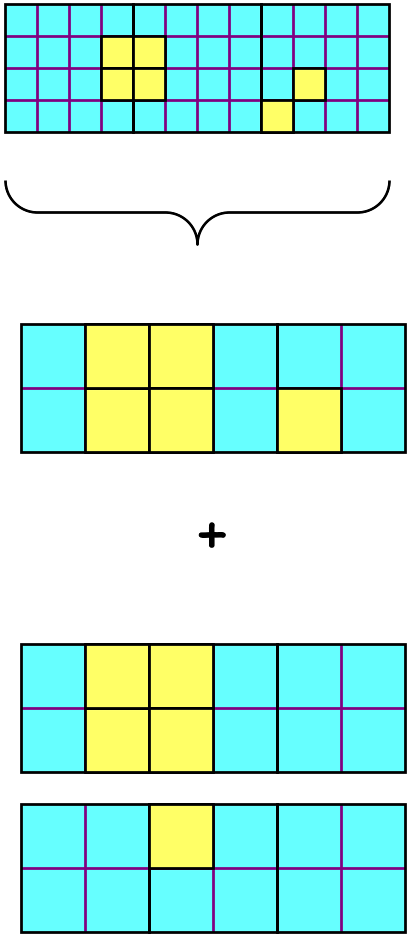
Original Plan: Fine Processing

- CMS TriggerUpgradeWorkshop: 22 July 2009 **
  - Split problem into two stages
  - Fine processing (electrons)
  - Coarse processing (jets)

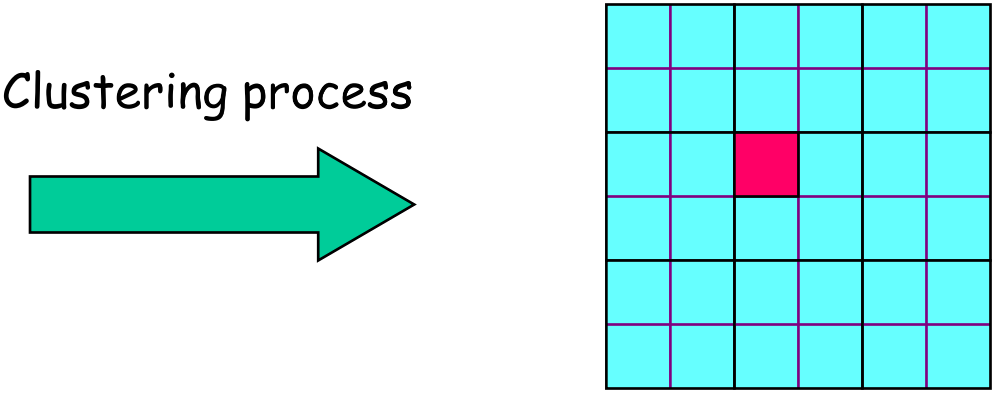


** http://indico.cern.ch/getFile.py/access?contribId=24&sessionId=2&resId=1&materialId=slides&confId=55141

Plan B: Coarse Processing



- For jets reduce resolution by factor of 2
  - Perfectly adequate
  - Nicely matches HF resolution



Conventional

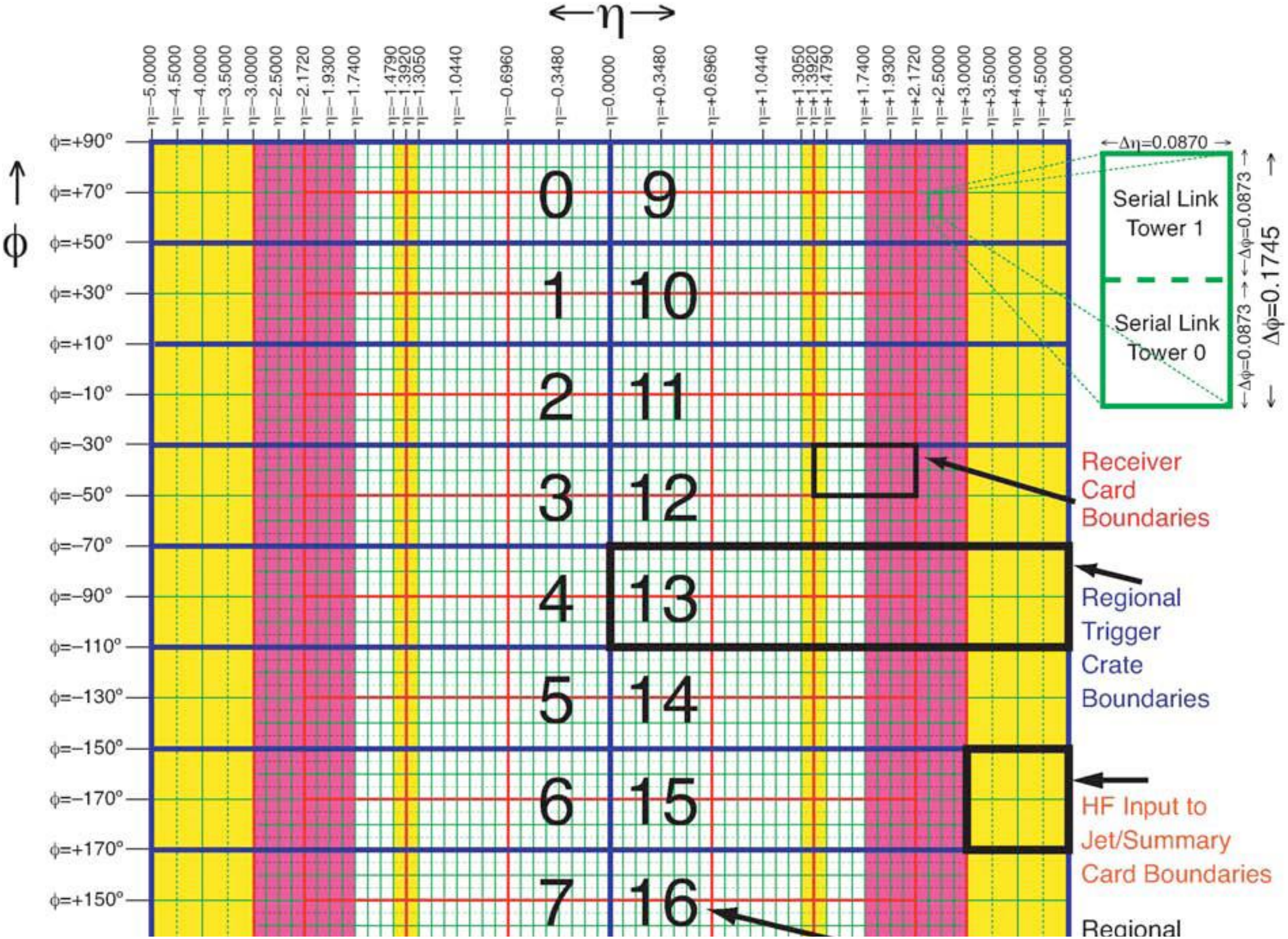
- Process entire detector on each bx
- Deal with electrons/taus first.
- Build jets from earlier energy clustering or take tower data and coarse grain to 2x2 towers.
- Pros:
  - Simple design
- Cons:
  - Less flexible, but does it matter...

Time Multiplexed *

- Akin to High Level Trigger processor farm
  - Process entire detector over ≈9bx
  - Switch between 10 systems operating in round robin method.
- Latency impact of 9bx regained by fewer serialisation steps
- Pros:
  - Jet+Elec processing in single fpga
  - Max processed/boundary area
  - Scalable: 10 identical systems
  - Redundant: Trigger keeps operating if partition lost, albit at 90%
  - Flexible: Based on single card
- Cons:
  - Never been done before

*** First proposed at TWEPP last year by John Jones**

Calorimeter Trigger Tower Mapping



Receiver Card Boundaries

Regional Trigger Crate Boundaries

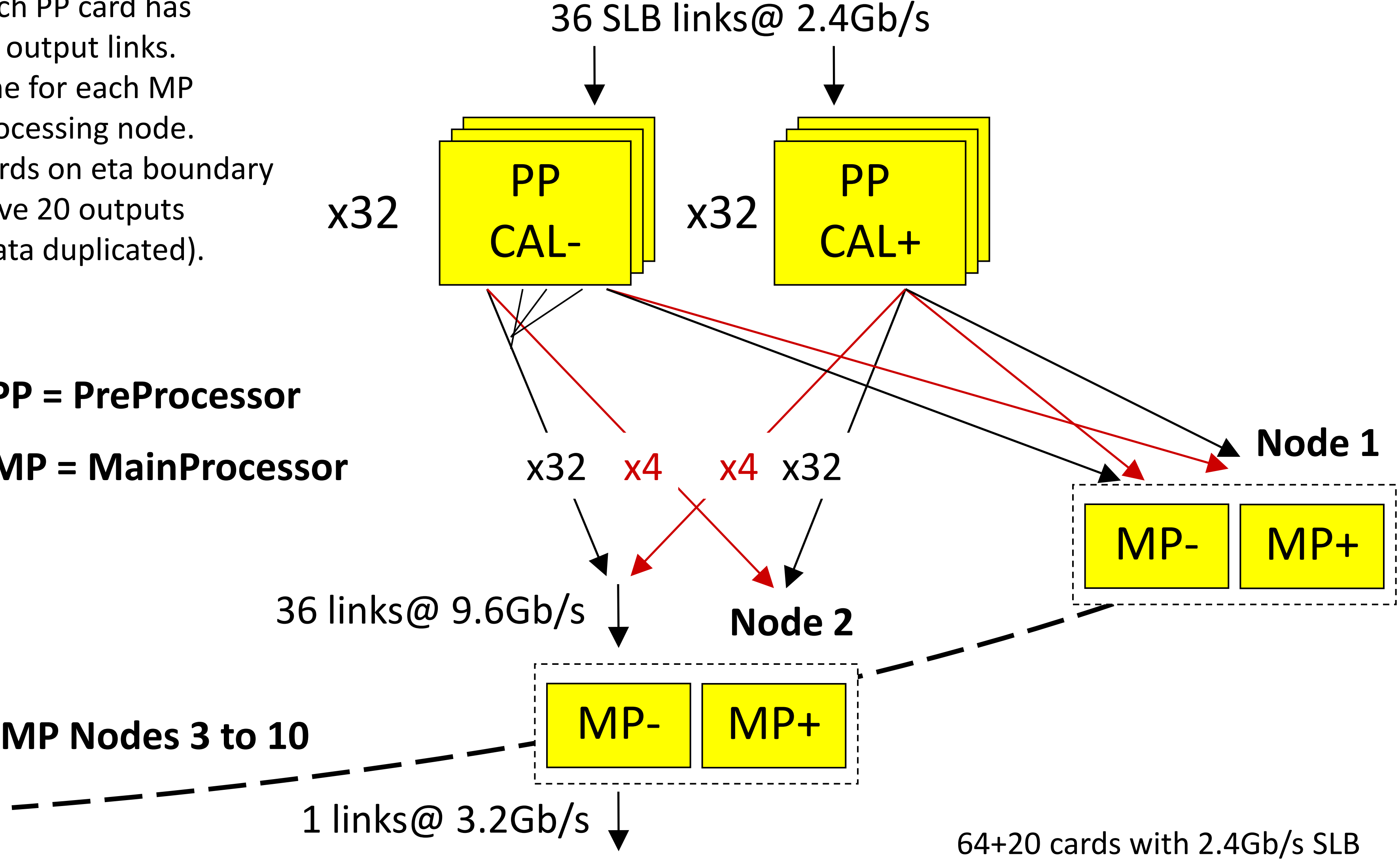
HF Input to Jet/Summary Card Boundaries

Regional

Time Multiplex example

Each PP (PreProcessor) card spans 1 tower in eta and 72 towers in phi. It receives ECAL & HCAL data.

Each PP card has 10 output links. One for each MP processing node. Cards on eta boundary have 20 outputs (data duplicated).



36 SLB links@ 2.4Gb/s

x32

PP CAL-

x32

PP CAL+

PP = PreProcessor

MP = MainProcessor

x32 x4 x4 x32

Node 1

MP- MP+

36 links@ 9.6Gb/s

Node 2

MP Nodes 3 to 10

MP- MP+

1 links@ 3.2Gb/s

64+20 cards with 2.4Gb/s SLB
34+20 cards with 4.8Gb/s SLB

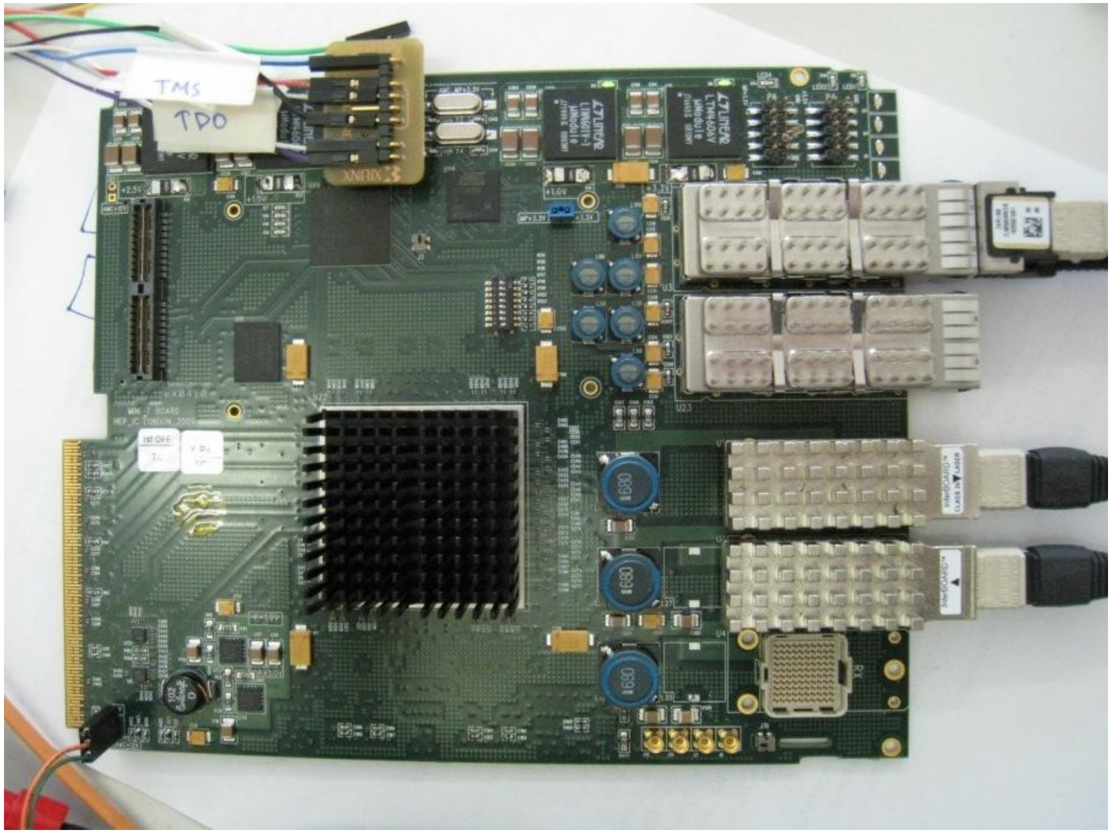
Objective of demonstartor system

- R&D project into both **Conventional** and **Time Multiplexed**
  - Develop common technology
  - 90% of hardware, firmware and software common to both

- Focussing on Time Multiplexed Trigger at present
  - Gain better understanding of different approach
  - Belief that it offers significant benefits to conventional trigger
    - Flexible: All trigger data available at tower resolution
    - Redundant: Loss of a trigger partition has limited impact on CMS
      - i.e. Trigger rate reduced by 10%. No regional area loss.
    - Scalable: Can prototype with single crate
      - i.e. Just 10% of the final hardware
      - Uses single uTCA card throughout system

Based around MINI-T5

- Double Width AMC Card
- Virtex 5 TX150T or TX240T
- Optics
  - IN = 160 Gb/s (32x 5Gb/s)
  - OUT = 100Gb/s (20x 5b/s)
- Parallel LVDS
  - BiDir 64Gb/s
  - 2x40Pairs at 800Mb/s
- AMC
  - 2x Ethernet, 1x SATA
  - 4x FatPipe, 1x Ext FatPipe



Can be used for Standard or Time Multiplex Architecture

Latency

- Modern serial links have many advantages, but latency isn't one of them

- Latency measuremnets based on Virtex 5 GTX and GTP transceivers
  - No tricks
    - e.g. No elastic buffer bypass
    - Why? Tricks sometimes backfire!
  - Measured **5.3 bx** at 5Gb/s
    - Includes sync firmware
    - Be conservative - assume **6.0 bx**

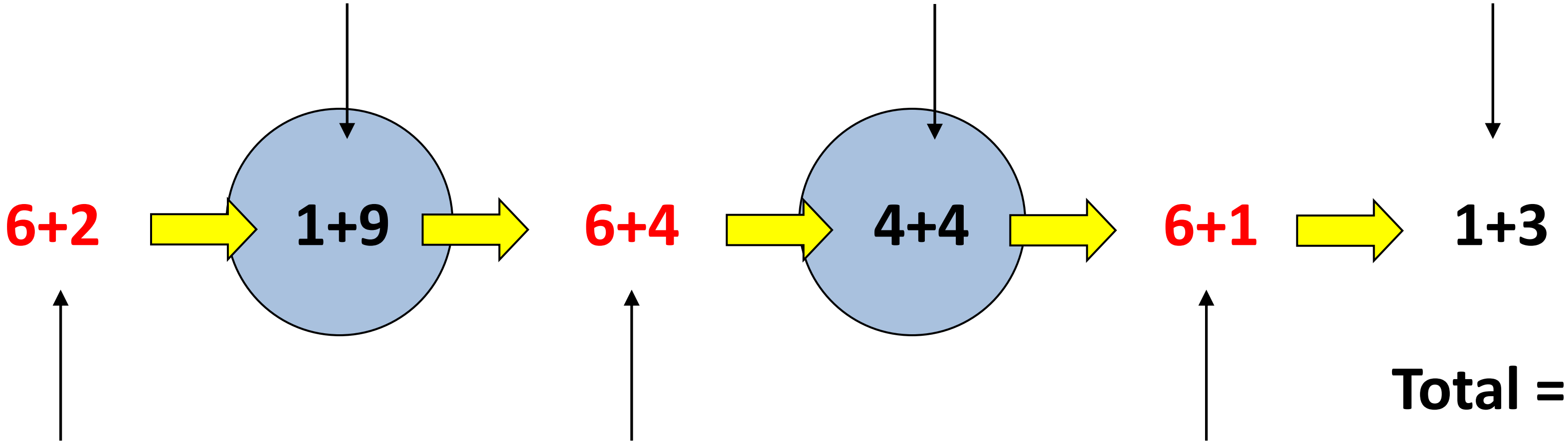
Current latency at Pt5 (bx)

+ 2	SLB Tx (estimate)
+ 2	SLB to RCT cable
+ 40	RCT+GCT
+ 3	GCT to GT

= 47	Total

Latency: Time Multiplex

All numbers in bx



PP processing plus time multiplexing

MP processing:
4bx for electrons measured.
4bx for jets & sort estimate.

GT procesing plus latency contingency

6+2 → **1+9** → **6+4** → **4+4** → **6+1** → **1+3**

Total = 47

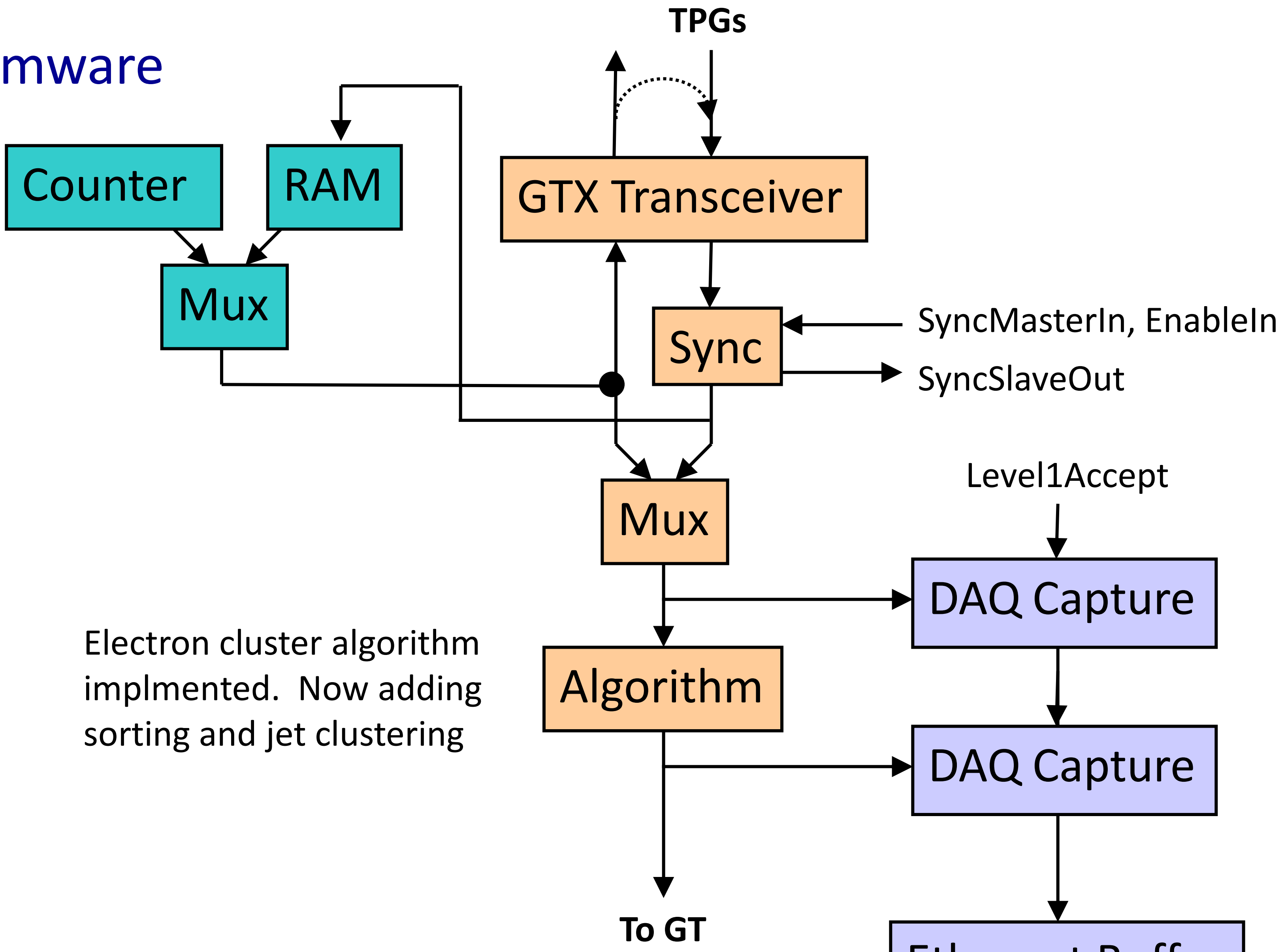
SLB to PP Serdes plus 10m cable delay

PP to MP serdes plus 20m cable delay

MP to GT serdes plus 5m cable delay

No Lateny penalty

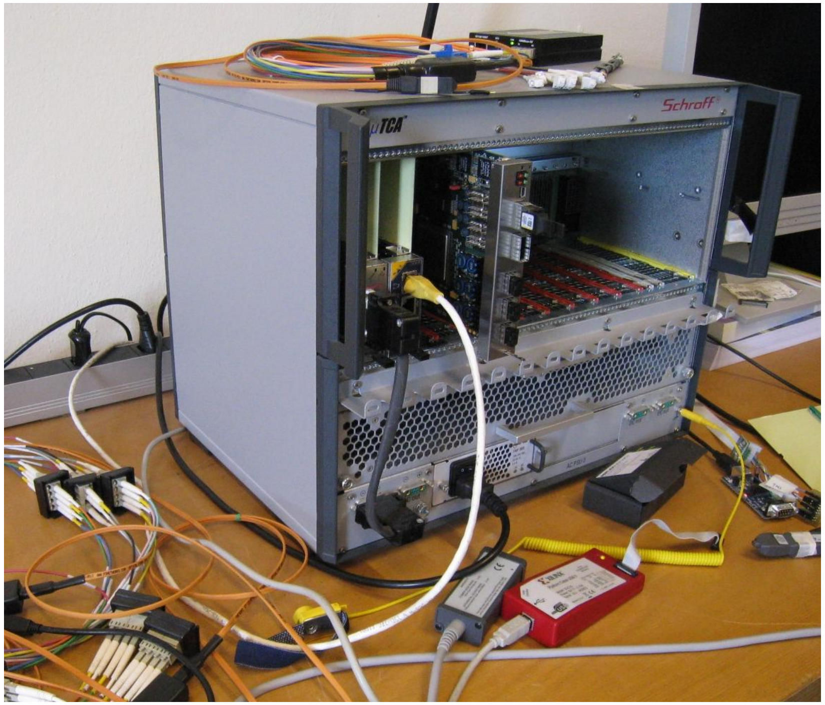
Firmware



Electron cluster algorithm implmented. Now adding sorting and jet clustering

Firmware Infrastructure - Finished

- Optical transeivers:
  - Up to 32 GTX transceivers at 5Gb/s
  - Pattern RAM injection & capture
  - CRC check
  - Low latency automatic synchronisation
    - aligns links on a user defined signal
    - e.g. beginning of a packet or BC0
- DAQ:
  - Event pipeline with automatic capture and buffering
    - i.e no more adjusting the latency!
- Ethernet:
  - UDP interface - Thanks to Jeremy Mans

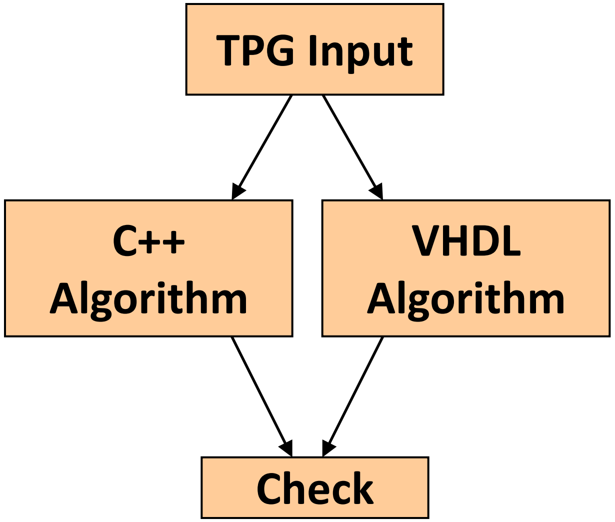


Lab development system

Firmware Algorithms – Work in progress

- Based on Wisconsin algorithms
  - Monika Grothe, Michalis Bachtis & Sridhara Dasu
  - Fully implemented & **<u>verified</u>**
    - 2×2 cluster finder
    - Electron/Photon ID module
    - Cluster overlap filtering
    - Cluster weighting
  - Work started, but postponed to allow SW development
    - Cluster isolation
    - Jet finding

- Uses Modelsim Forgein Language Interface (FLI) to verify algorithms

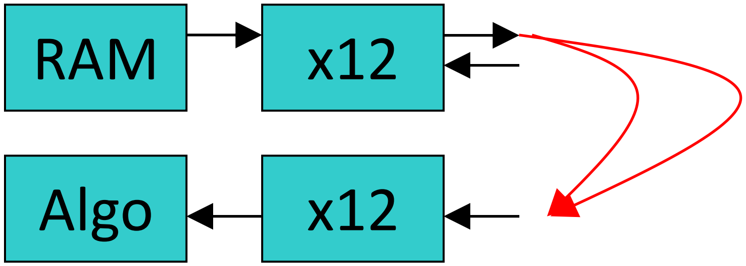


All work done by Andrew Rose

Firmware verification: Test setup

- Test setup slightly different to proposed system, but concept the same.
- Uses 8bit rather than 12bits energies (i.e. current CMS)
  - Would require 28 links at 5Gb/s to load data in 24bx.
  - Final system uses 72 links across 2 boards running at 10Gb/s to load 12bit data in 9bx.
- Different length fibre to make sure sync unit operating correctly.

```
Full electron algo in xc5vtx150t
Slice Registers 12460/92800 = 13%
Slice LUTs 17167/92800 = 18%
```



```
Partial electron algo in xc5vtx150t
Slice Logic Utilization:
  Number of Slice Registers:                  21,143 out of  92,800   22%
  Number of Slice LUTs:                       27,571 out of  92,800   29%

Specific Feature Utilization:
  Number of BlockRAM/FIFO:                       220 out of     228   96%
  Number of GTX_DUALs:                            13 out of      20   65%
```

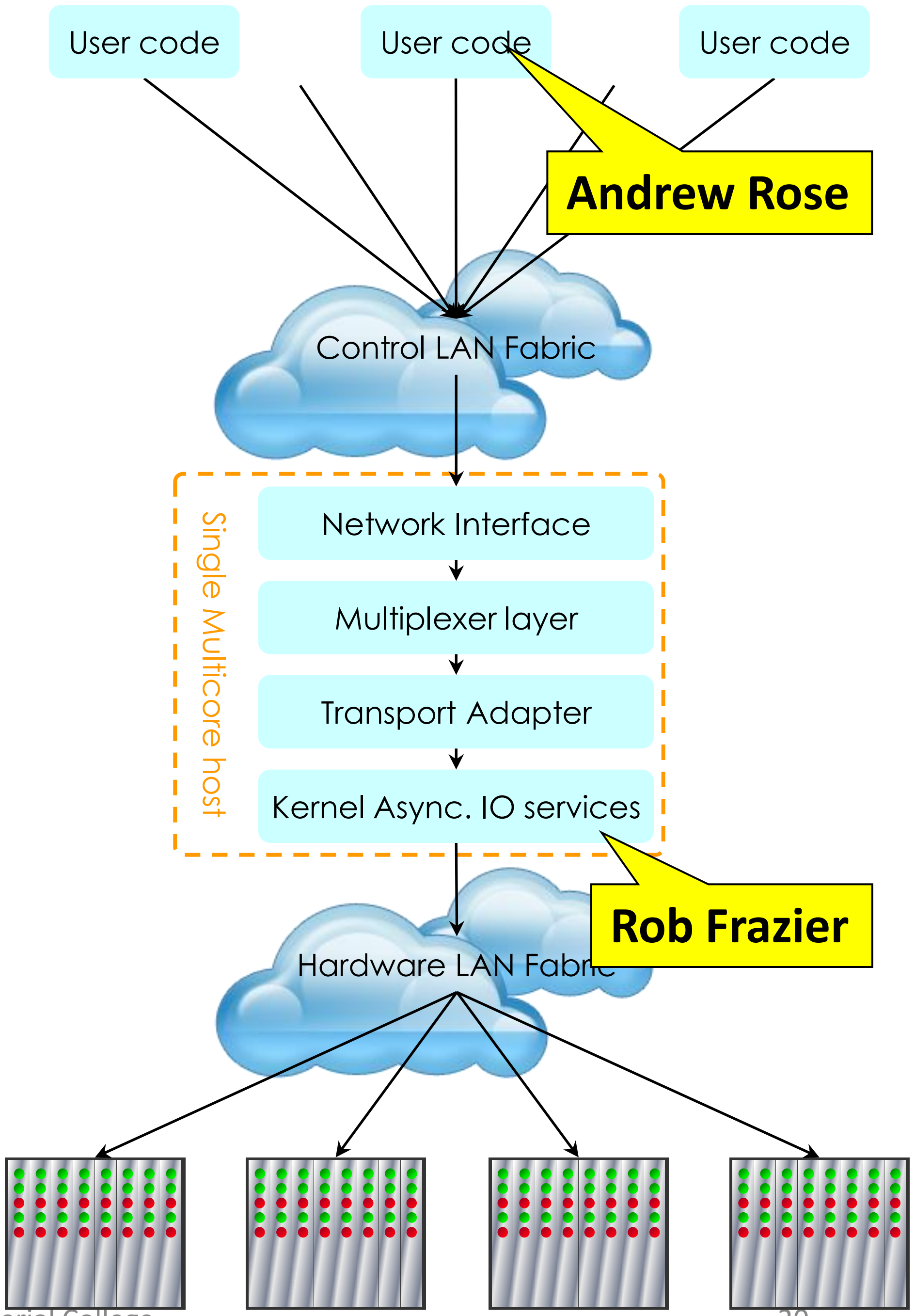
Software: Architecture

- Hardware controller PC separates the Control LAN and the User code from the Hardware LAN and the devices

- Unlike current Trigger software architecture, all network traffic hidden from end user

- Made possible by common interface layer within the firmware and mirrored within the software

- Simply python interface also available.



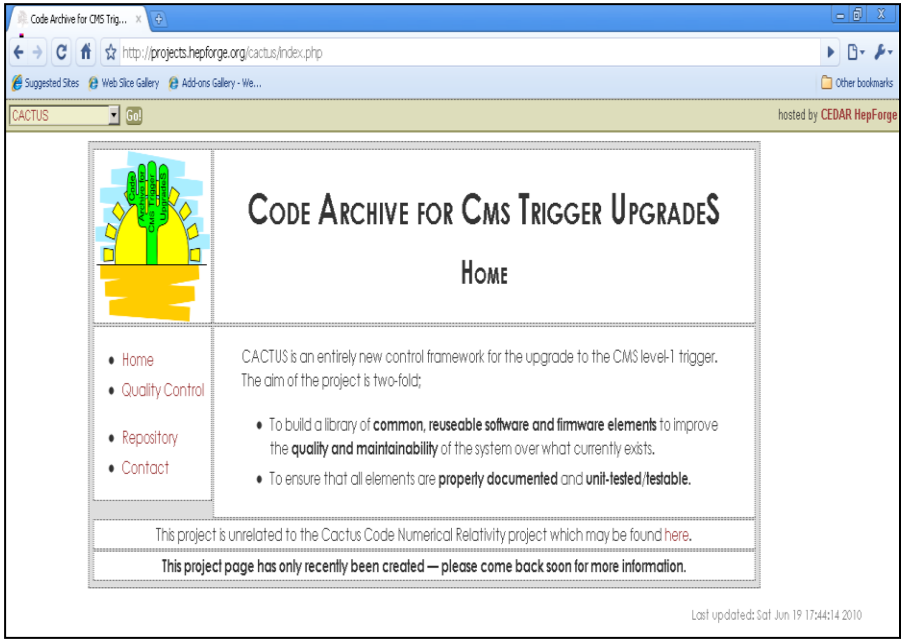
Software: Work in progress

- Rob Frazier (Bristol) writing the Controller Hub Software within a high performance web server framework

- Andy Rose (Imperial) writing the low level client side software

- Project hosted on HepForge

- Low Level Client Side software finished
  - Hierachial register map
    - Allows VHDL blocks to be reused.
    - i.e. make simple VHDL block and then instantiate multiple times.



CODE ARCHIVE FOR CMS TRIGGER UPGRADES

HOME

CACTUS is an entirely new control framework for the upgrade to the CMS level-1 trigger. The aim of the project is two-fold;

- To build a library of **common, reuseable software and firmware elements** to improve the **quality and maintainability** of the system over what currently exists.
- To ensure that all elements are **properly documented** and **unit-tested**/**testable.**

This project is unrelated to the Cactus Code Numerical Relativity project which may be found here.

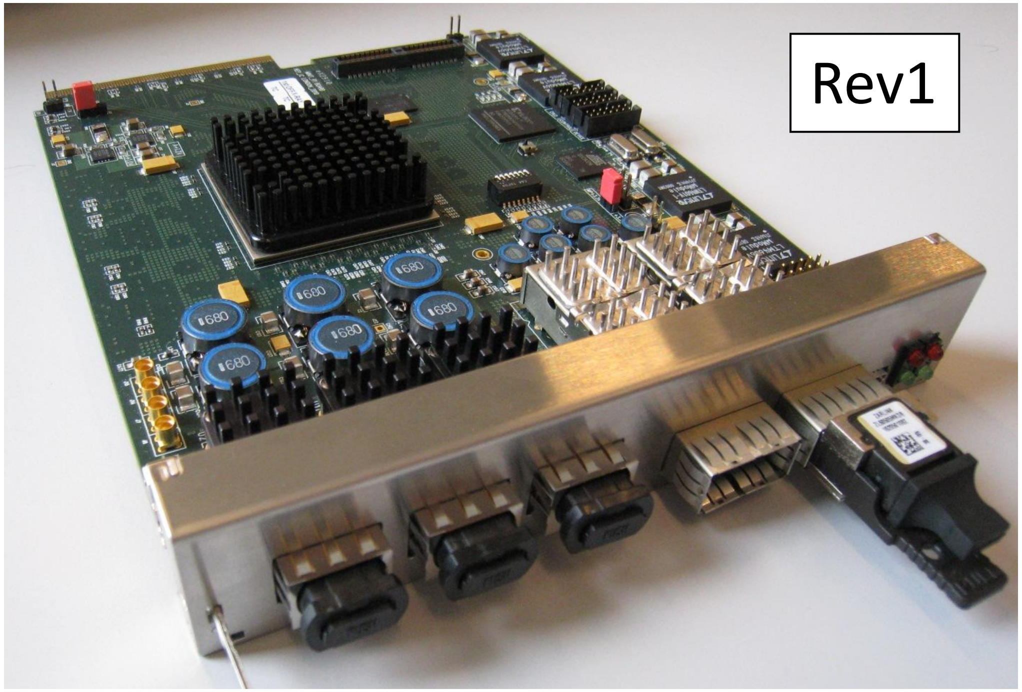
This project page has only recently been created — please come back soon for more information.

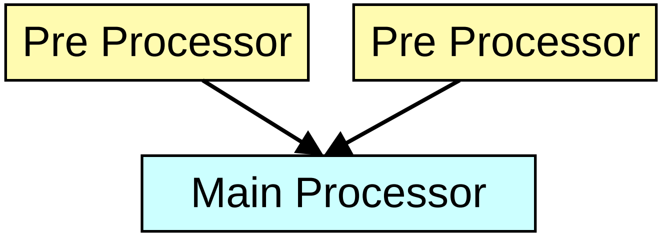
Last updated: Sat Jun 19 17:44:14 2010

Current Status

- MINI-T5 Rev1 back
  - USB2 capability
  - Avago optics
  - It even has a front panel!

- Next
  - Sort and Jet clustering
  - Build system





Current focus has been on the **MainProcessor** firmware/software.

Next focus on **PreProcessor** and buildining a system of multiple cards.

Questions ?