Design of constrained Digital Signal Processing Systems

Marco Ghibaudi

Thesis Defence, 16. October 2014

CERN

Outline

- Introduction: Constrained Digital Signal Processing
- Digital Signal Processing in HEP
  - The ATLAS experiment
  - The Central Trigger Processor
- ATLAS Upgrade
  - Motivations
  - Upgrade of the Central Trigger Processor
- The CTPCORE+
  - Design of the hardware components
  - Design of the firmware components
  - System Verification and Validation
- Conclusions

Introduction

- Information are generated mainly from processing of Digital Signals
- Data processing comes with constraints:
  - $\mathscr{C}$ - Total Cost
  - $\mathscr{Q}$ - Throughput
  - $\mathscr{T}$ - Latency

Introduction

Cost

- In case of an Digital Processing system, the Total Cost of Ownership represents the sum of the hardware, software and maintenance costs.
- In various fields, cost minimization is the primary target.
- Example: Intelligent Transport Systems (ITS) must be pervasive and appealing for government and society.
- To minimize cost: reusable low cost HW, cost efficient SW, highly adaptive/extendible systems.

Introduction

Throughput

- Throughput represents the amount of data that can be generated per unit of time.
- Traditional computing architectures have limits that are incompatible with some system expectations.
- Modern Telecommunication Systems are posing immense design challenges.
- Reconfigurable Parallel architectures, while more complex to use/design, are the solution in a large subset of high throughput systems.

Introduction

Latency

- Latency for a Digital Processing System can be seen as the time that takes to convert the inputs into the expected outputs.
- Traditional classification of hard/soft realtime applies.
- Most of the High Energy Physics experiments are for good measure tightly latency constrained.
- The full processing chain, HW and firmware/software had to be designed considered this limitation. Cost for missing computation deadlines may be extremely high.

Outline

- Introduction: Constrained Digital Signal Processing
- Digital Signal Processing in HEP
  - The ATLAS experiment
  - The Central Trigger Processor
- ATLAS Upgrade
  - Motivations
  - Upgrade of the Central Trigger Processor
- The CTPCORE+
  - Design of the hardware components
  - Design of the firmware components
  - System Verification and Validation
- Conclusions

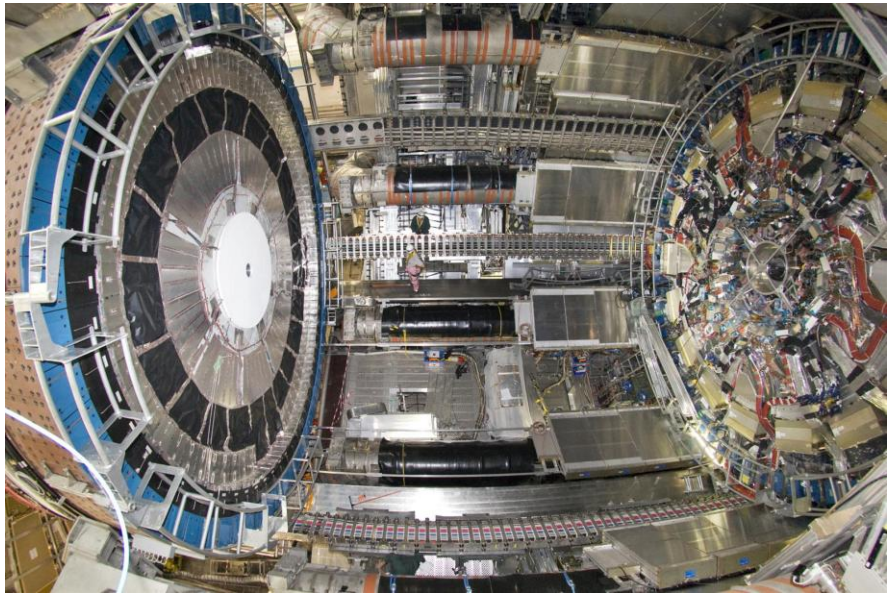
Digital Signal Processing in HEP

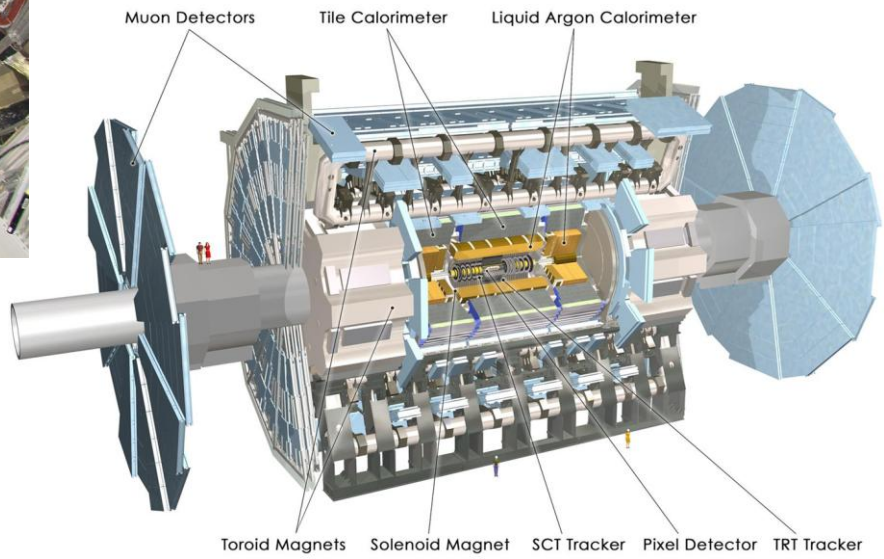
- Particle Physic Experiments can generate hundreds of Terabits of data per second (i.e. ATLAS and CMS, CERN).
- In ATLAS every 25ns (Bunch Crossing), tens of interactions occurs when protons beams collide.
- Roughly 1 collision out of 1 million generates events of interest.
- Events are rare + Level of confidence of the results must be extremely high ➔ Good data can not be lost

Digital Signal Processing in HEP

Digital Signal Processing in HEP

The ATLAS experiment

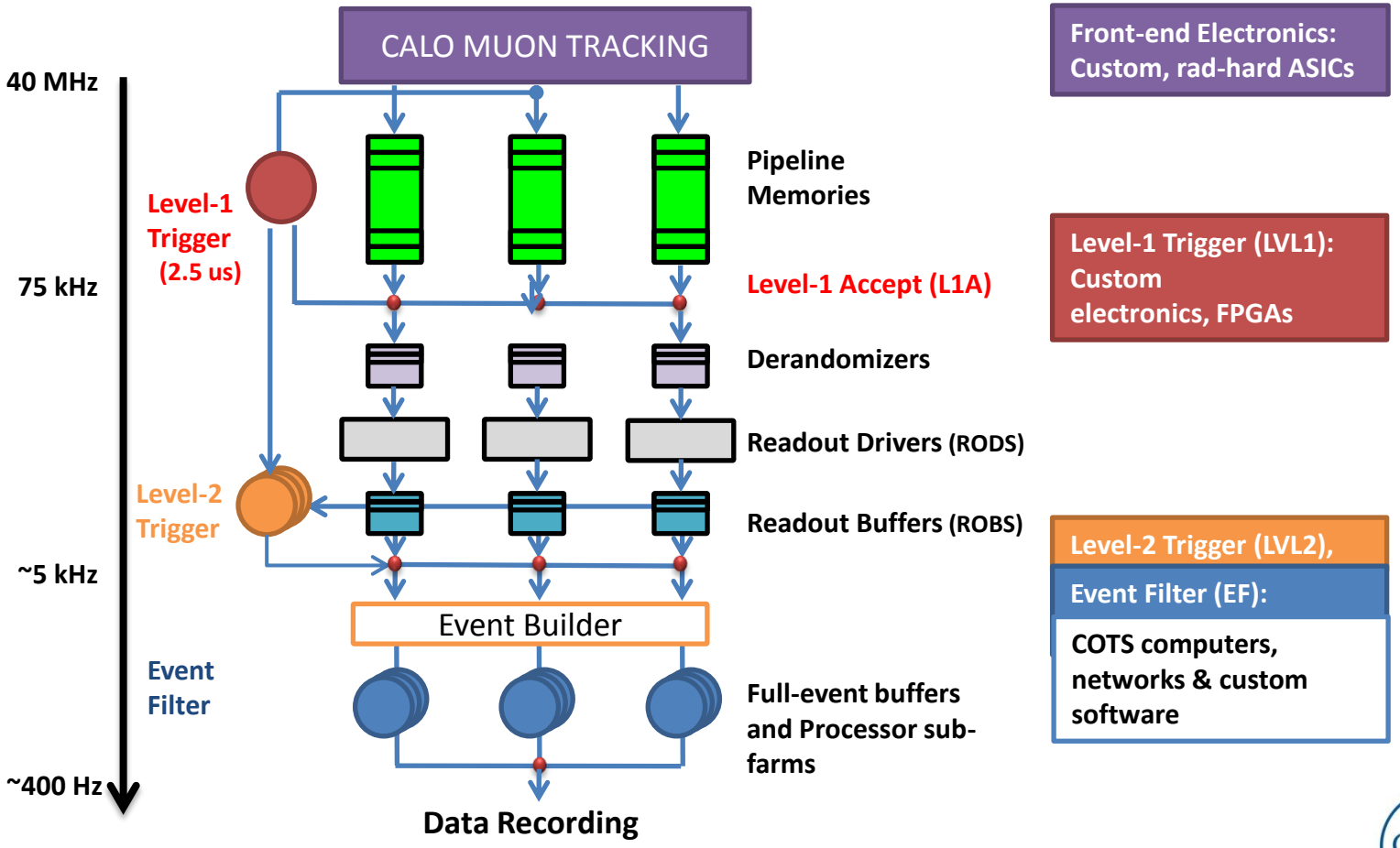




- Weight: 7000 tons
- Dimensions: diameter 25mt, length 46mt
- 3000 km of cables

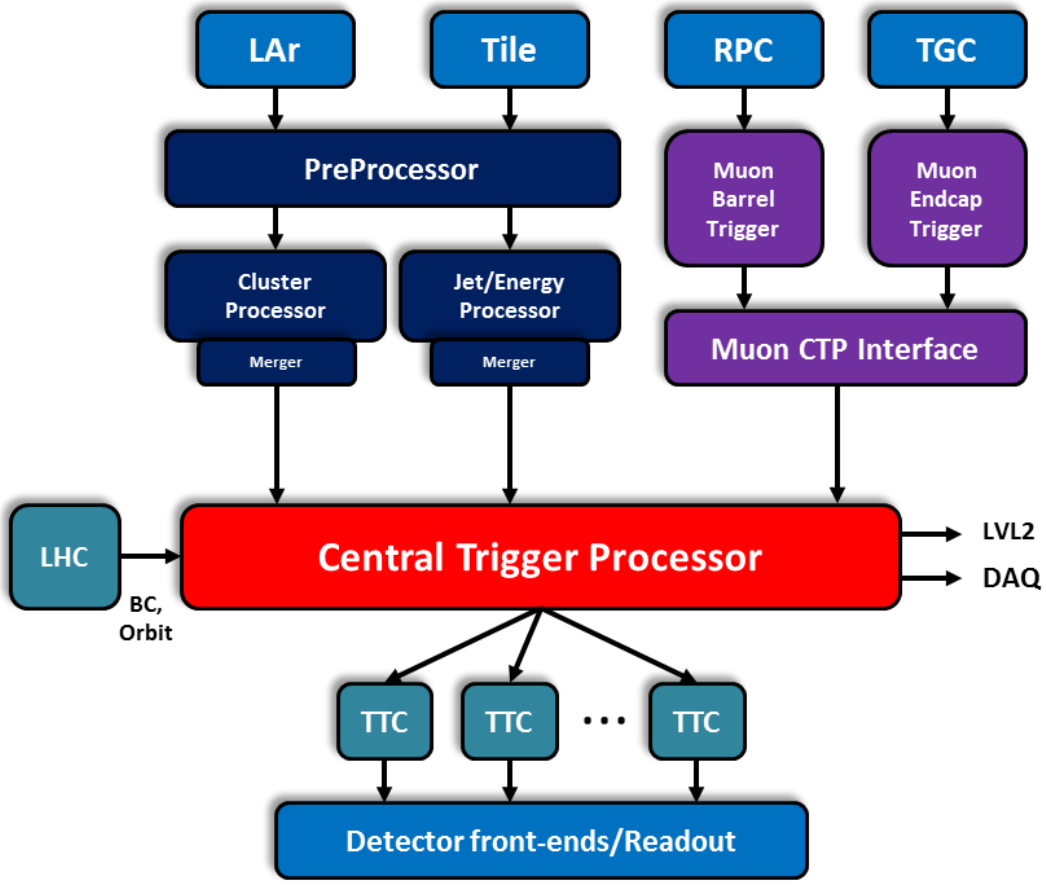
Digital Signal Processing in HEP

The ATLAS Experiment



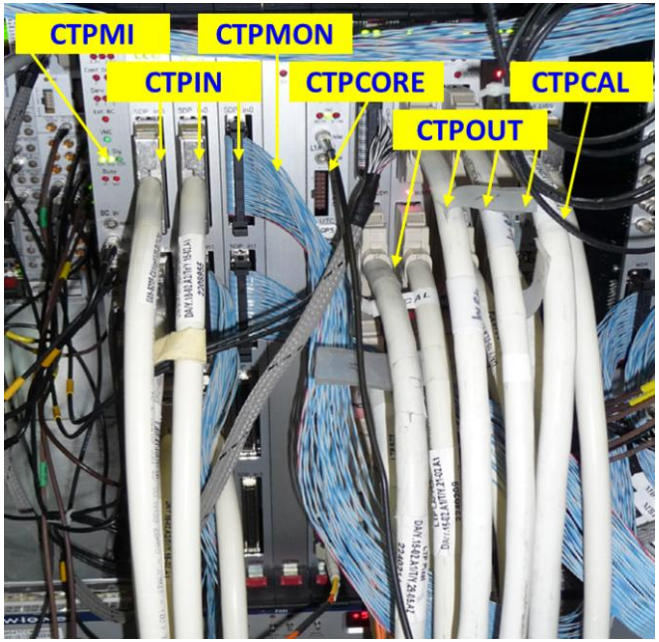
Digital Signal Processing in HEP

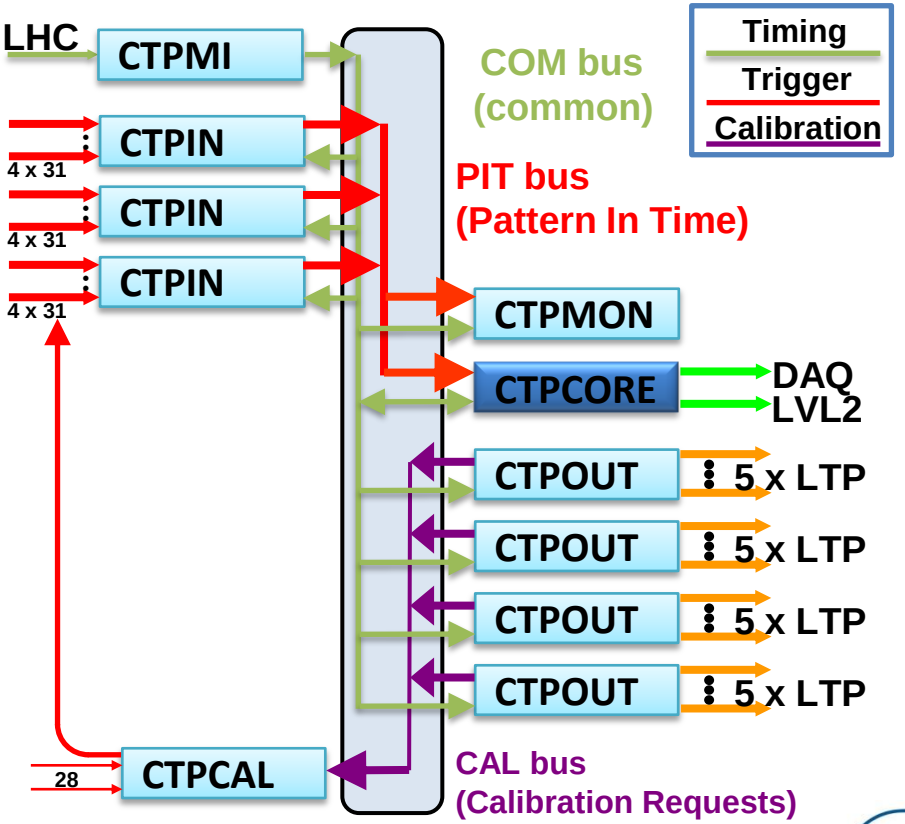
The ATLAS Level 1 Trigger



Digital Signal Processing in HEP

The ATLAS Central Trigger Processor





Outline

- Introduction: Constrained Digital Signal Processing
- Digital Signal Processing in HEP
  - The ATLAS experiment
  - The Central Trigger Processor
- ATLAS Upgrade
  - Motivations
  - Upgrade of the Central Trigger Processor
- The CTPCORE+
  - Feasibility studies
  - Design of the hardware components
  - Design of the firmware components
  - System Verification and Validation
- Conclusions

ATLAS Upgrade

Motivations

- In 2012, the LHC has been temporary shutdown for conducting maintenance and consolidation activities.
- The system is expected to operate at higher energies (higher luminosity)
- More energy implies more collisions and requires improved filtering.
- ATLAS had to be upgraded to cope with the new requirements
- Significant changes will affect the back-end electronics.

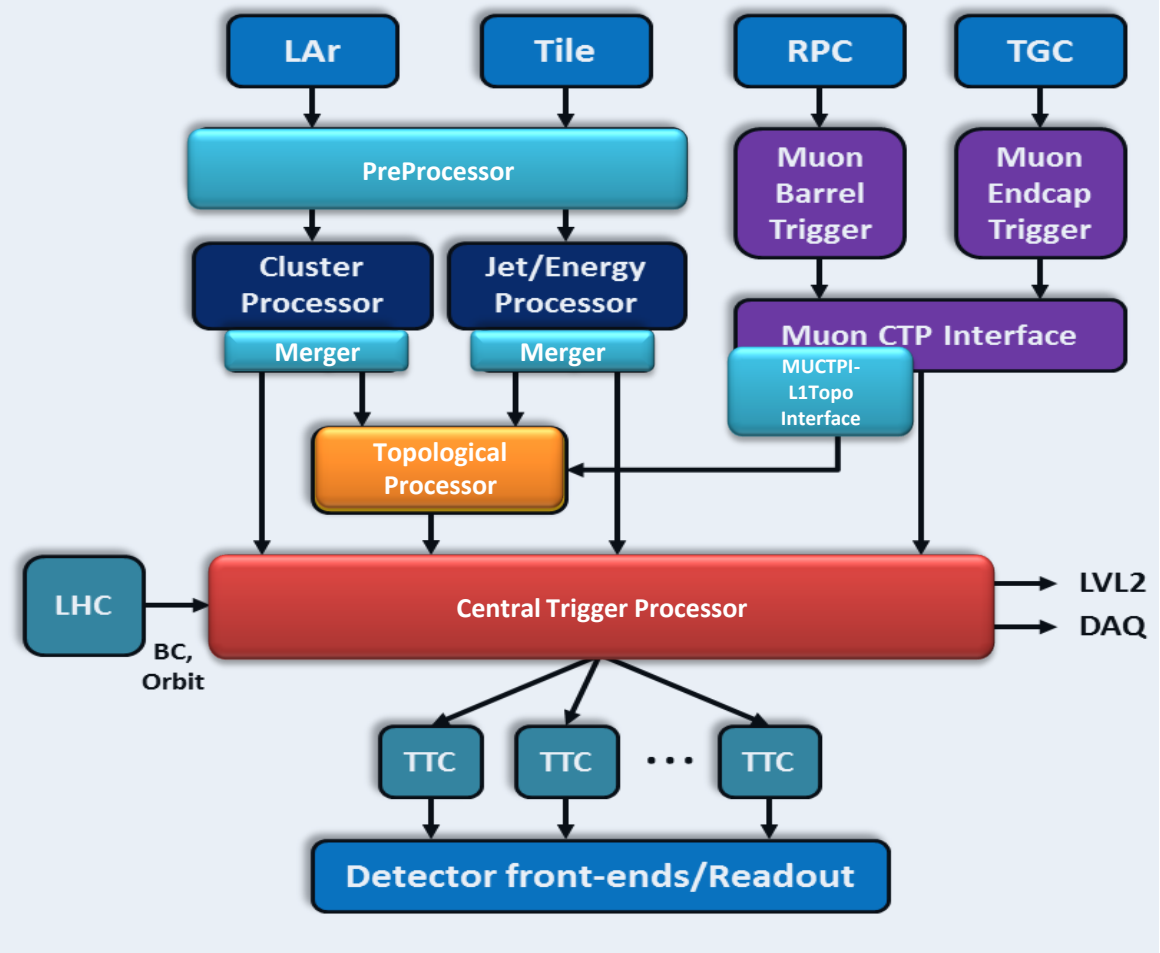
ATLAS Upgrade

Motivations

- The Level-1 Trigger must operate with the same latency constraints.
- A new components, the L1Topo will extend the Level-1 Trigger, introducing new filtering algorithms
- After the shutdown the Level-1 Trigger will be able to detect more types of events.
- This phase will also be used to investigate new technologies

ATLAS Upgrade

Upgrade of the Central Trigger Processor



ATLAS Upgrade

Upgrade of the Central Trigger Processor

What is required?	Why?	What changes?
More trigger items	Improve event selection	CTPCORE
New component (L1Topo)	Testing new algorithms	MUCTPI, CTPCORE
New optical links	Feasibility analysis	CTPCORE
Enhanced logging features	Tests and system “health monitoring”	CTPCORE

Imposes the complete redesign of the CTPCORE, CTPOUT and significant changes of the MUCTPI modules

Outline

- Introduction: Constrained Digital Signal Processing
- Digital Signal Processing in HEP
  - The ATLAS experiment
  - The Central Trigger Processor
- ATLAS Upgrade
  - Motivations
  - Upgrade of the Central Trigger Processor
- The CTPCORE+
  - Design of the hardware components
  - Design of the firmware components
  - System Verification and Validation
- Conclusions

The CTPCORE+

- The **CTPCORE** has been upgraded => **CTPCORE+**
- For handling more inputs/ implementing more sophisticated filtering - triggering operations.
- Activities:

1. Hardware redesign. New CTPCORE+ module has been produced and tested
2. Firmware has been redesigned - extended
3. Integration and verification

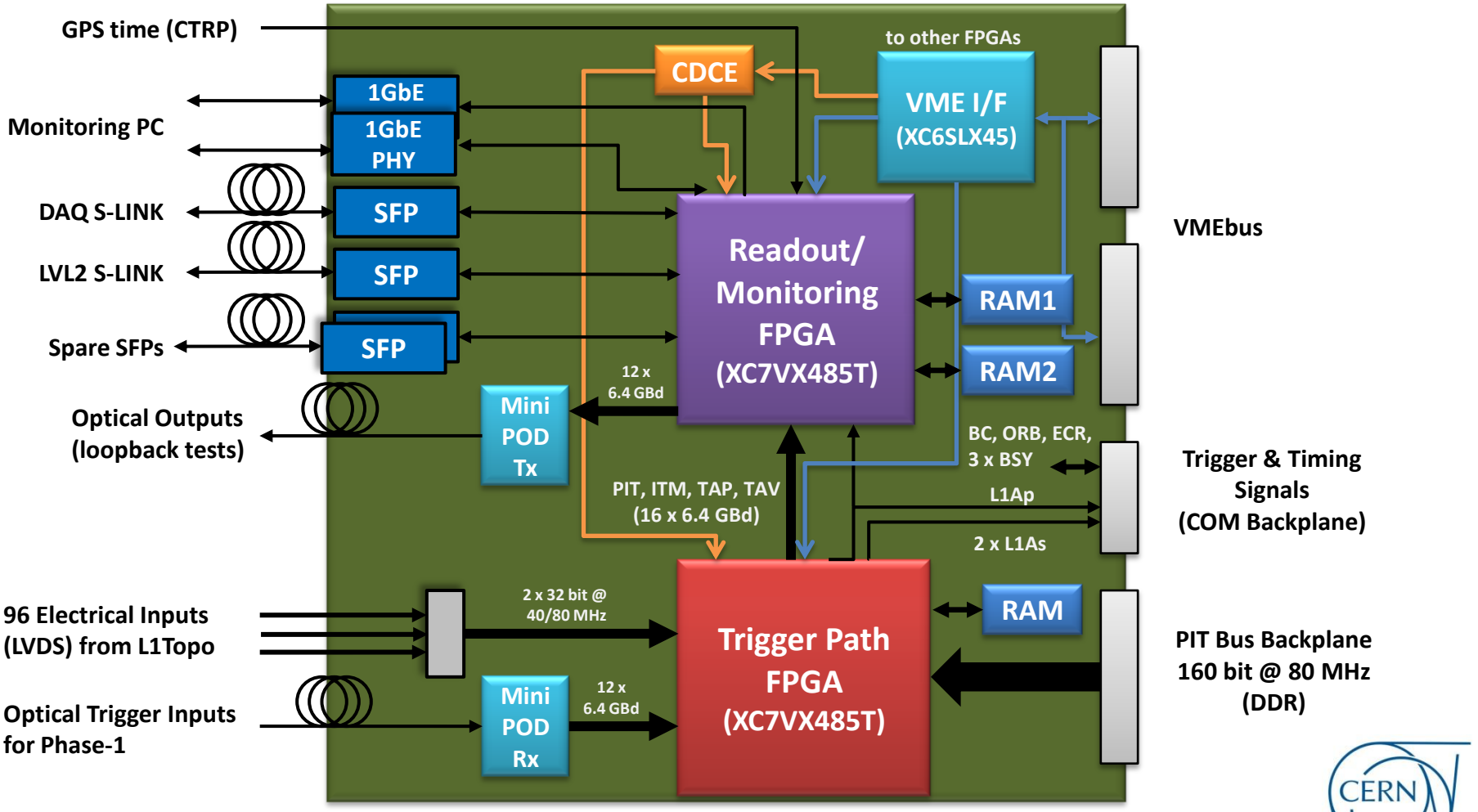
The CTPCORE+

CTPCORE+ Hardware Design

- Trigger Path FPGA, functionalities:
  - Generates L1A signal
  - Receives busy signals
  - Generates preemptive deadtime
  - Sends full information about trigger decision to readout & monitoring FPGA (>2300 bit/BC > 90 Gbit/s)
  - Playback and diagnostics memory
- Readout FPGA, functionalities:
  - Receives detailed information from trigger path FPGA via 16 serial links
  - Send trigger summary information to LVL2 and DAQ for every L1A
  - Add precise timestamp to each event (GPS reference)
  - Monitoring functionality

The CTPCORE+

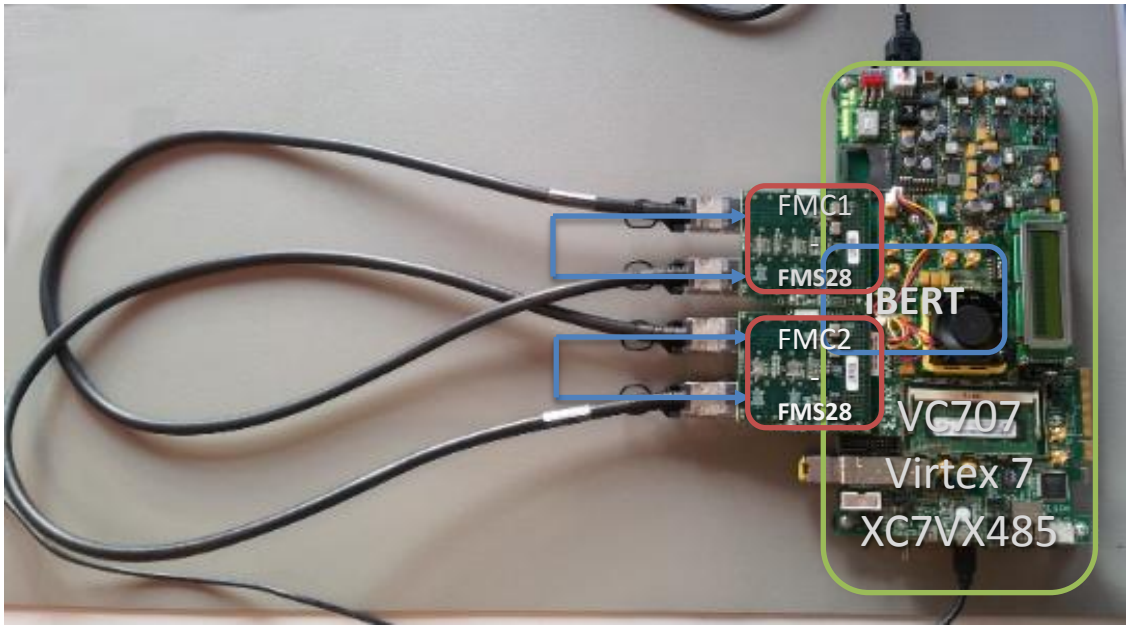
CTPCORE+ Hardware Design



The CTPCORE+

CTPCORE+ Hardware design – Validation Setup

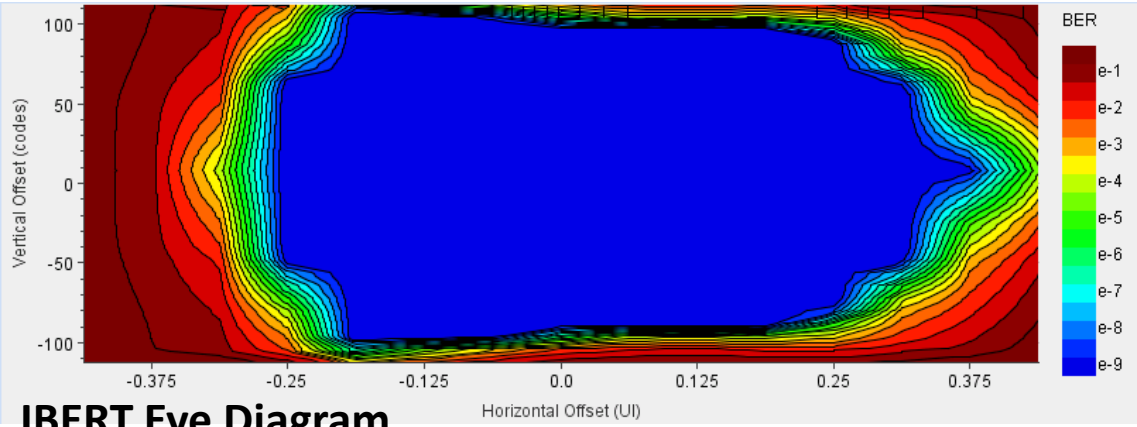
- VC707: Virtex-7, DDR3 memory, Ethernet, 16 MGTS
- Full Mezzanine Card Electrical links adapters/cables



The CTPCORE+

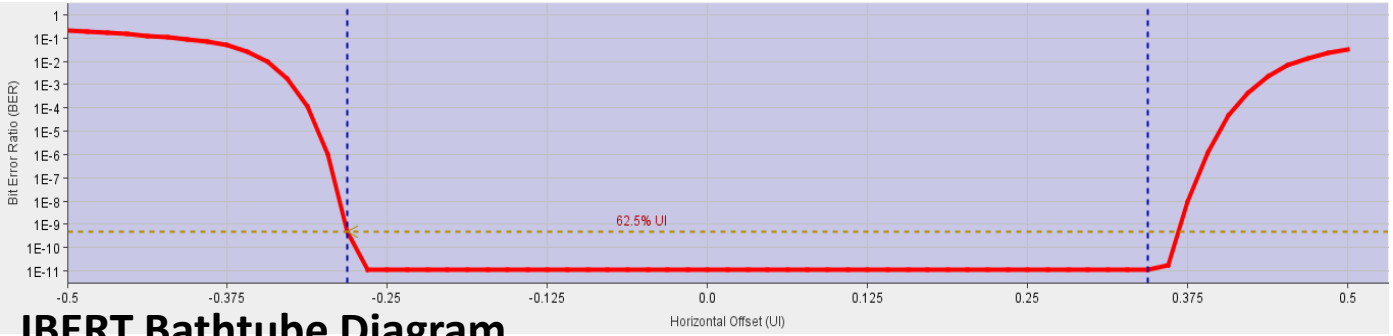
CTPCORE+ Hardware design – Feasibility Studies

- Fast link connectivity



IBERT Eye Diagram

16 Links, 6.4 Gbps
Measured BER:
< 10e-13

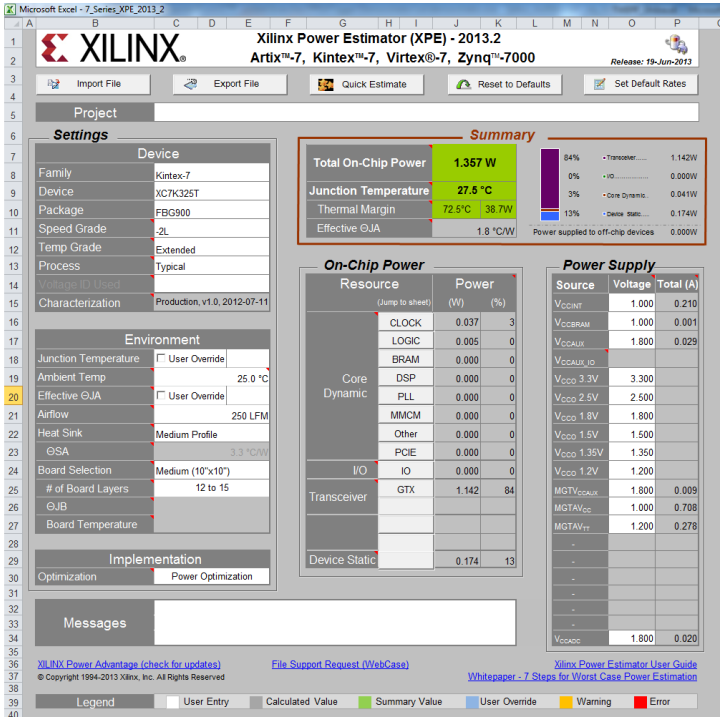


IBERT Bathtube Diagram

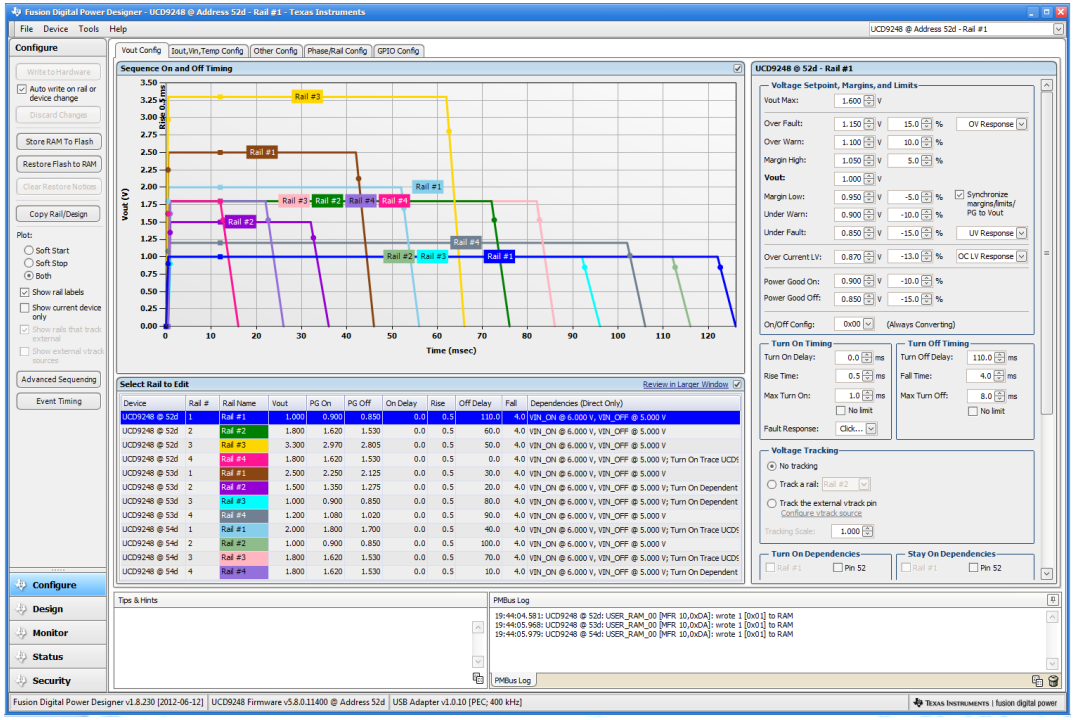
The CTPCORE+

CTPCORE+ Hardware design – Feasibility Studies

- Power consumption



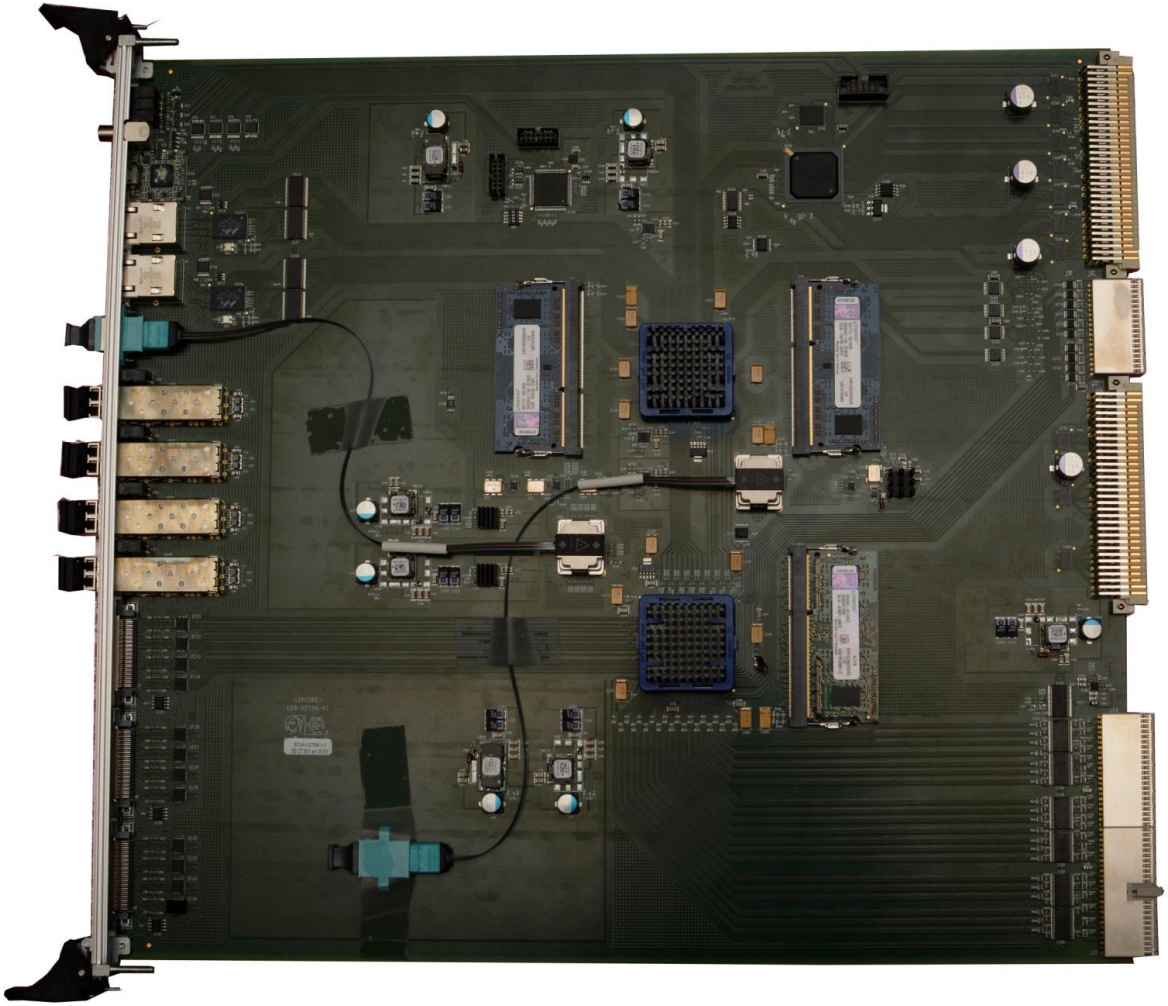
Power Estimation tools



Power Measurement tools

The CTPCORE+

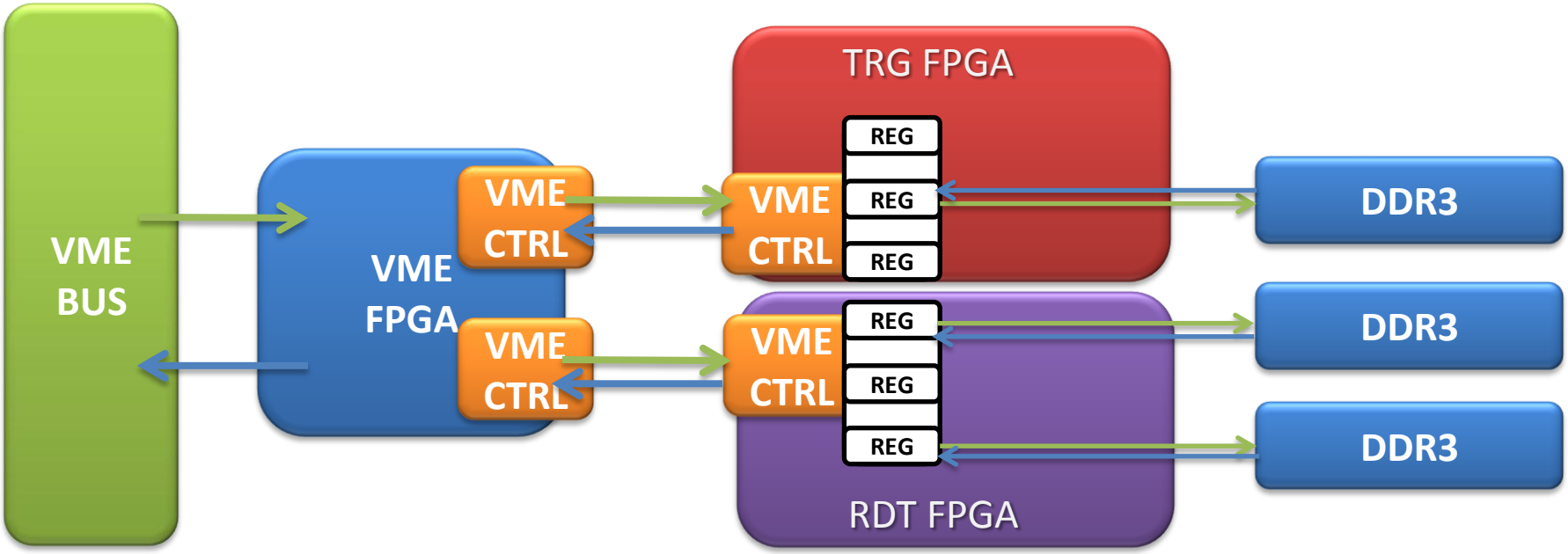
CTPCORE+ Hardware design – 1'st Prototype



The CTPCORE+

CTPCORE+ Firmware design

- VME Interface/controller



The CTPCORE+

CTPCORE+ Firmware design

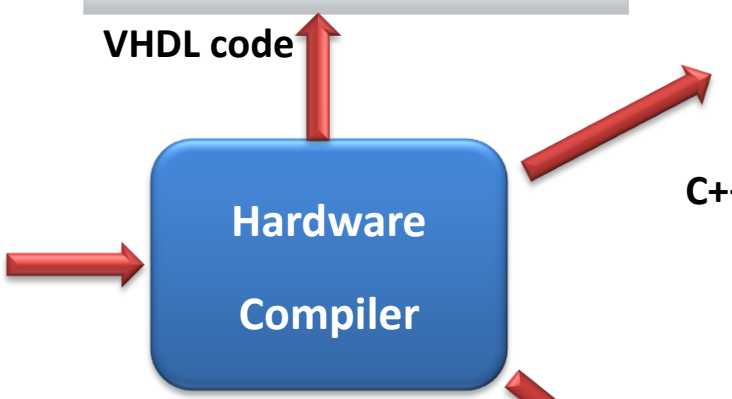
- VME Interface/controller

```
<!DOCTYPE module SYSTEM "L1CTHardwareModule.dtd">

<module name="CTPCOREPLUS" amod="A32" type="D32"
size="0x00100000" addr="0x00000000"
desc="CTPCOREPLUS register map">

  <!-- TRG-VME interface FPGA registers -->
  <block name="VME" addr="0x00008000">
    <register name="ResetControl"    addr="0x00000000">
      <field name="DDR3FullReset"    mask="0x00000001"
                          form="BOOLEAN"/>
      <field name="TrgFpgaReset"       mask="0x00000002"
                          form="BOOLEAN"/>
      <field name="RdtDllReset"        mask="0x00000004"
                          form="BOOLEAN"/>
      <field name="RdtFpgaReset"       mask="0x00000008"
                          form="BOOLEAN"/>
    </register>
  </block>

<!-- SPI interface and control for TI CDCE62005 jitter cleaner -
->
  <block name="CDC" addr="0x000c0000">
    <register name="SpiConfig"
addr="0x00000000"/>
    <register name="Status"                addr="0x00000004" >
      <field name="SpiBusy"                mask="0x00000001"
                          form="BOOLEAN"/>
      <field name="PllLocked"              mask="0x00000002"
                          form="BOOLEAN"/>
    </register>
  </block>

</module>
```

Hardware description file

```
subtype ResetControlT is std_logic_vector (31 downto 0);
constant ResetControlMask  : std_logic_vector (31 downto 0) :=
0x00000003;
```

VHDL code




```
VME_ResetControl_Write(0xf);
res = VME_DllStatus_Read(&val);
If (val== 0x3)
   return NO_ERROR

else
   return CRIT_ERROR
```

C++ library and definitions

Python wrapper

```
core.VME_ResetControl_Write(0xf);
res, val = core.VME_DllStatus_Read();
If (val == 0x3):
   return NO_ERROR
else
   return CRIT_ERROR
```

The CTPCORE+

CTPCORE+ Firmware design

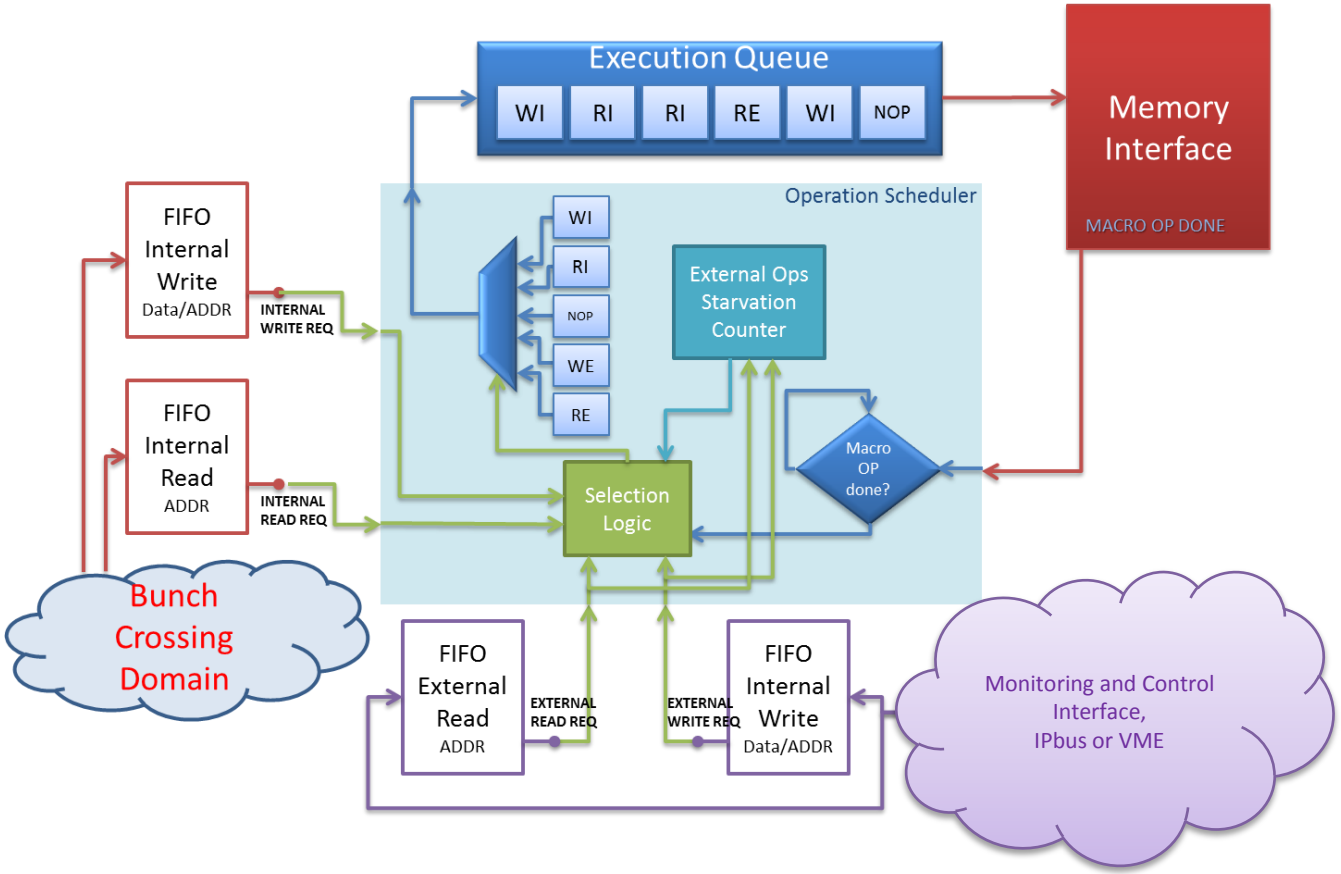
- Ethernet Interface/controller

- Control and monitoring achieved through Ipbus
- IPbus provides a mechanism to access FPGA registers through Ethernet/UDP. Used by CMS
- Supports standard operations
- Developed model to speed up simulation

The CTPCORE+

CTPCORE+ Firmware design

- Dual port DDR controller



The CTPCORE+

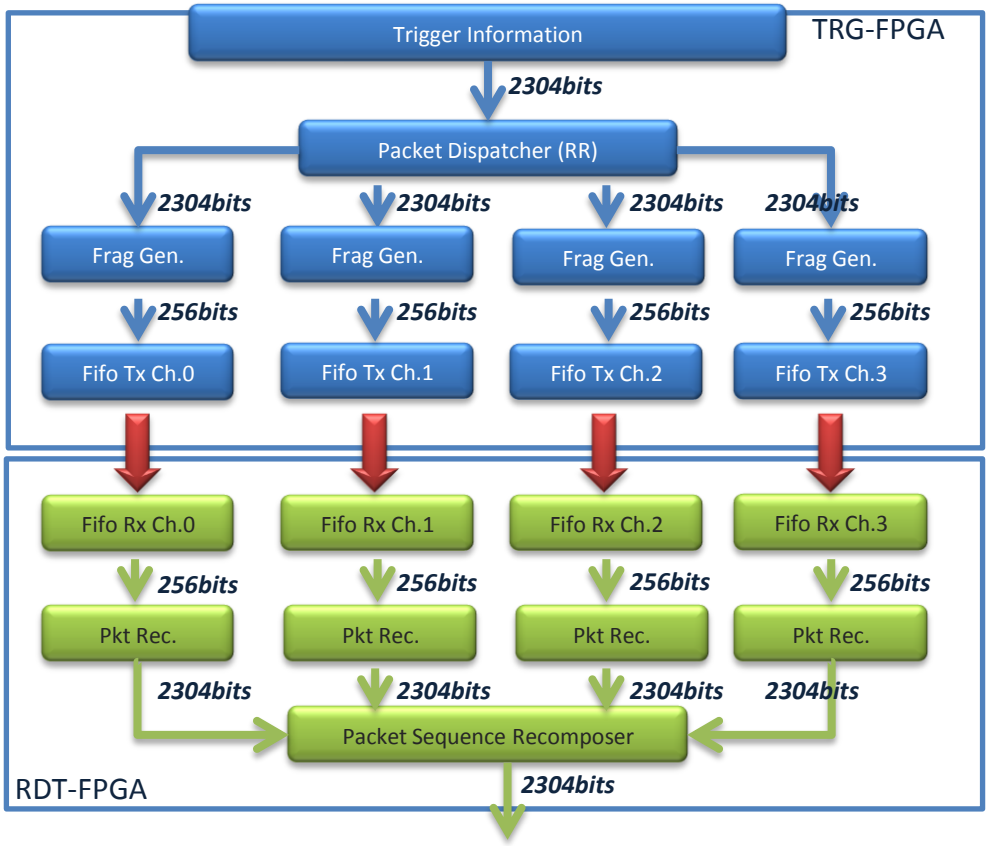
CTPCORE+ Firmware design

- High Bandwidth chip-to-chip link

16 x 6.4 Gbps

64B66B encoding

> 2200 bits @ 40MHz



The CTPCORE+

System validation and integration

- New firmware blocks have been integrated with legacy logic
- The **CTPCORE+** has been tested on an CTP replica
- Experiment commissioning is expected to be completed by 4Q-2014
- Some results may be reused in other experiment (Alice, CMS)

Outline

- Introduction: Constrained Digital Signal Processing
- Digital Signal Processing in HEP
  - The ATLAS experiment
  - The Central Trigger Processor
- ATLAS Upgrade
  - Motivations
  - Upgrade of the Central Trigger Processor
- The CTPCORE+
  - Design of the hardware components
  - Design of the firmware components
  - System Verification and Validation
- Conclusions

Conclusions

- A significant number of Digital Processing Systems have strong constraints
- Possible solutions for some specific projects have been investigated
- Cost critical systems => Use of ad-hoc low cost SW and HW. Investigation of low complexity algorithms
- Throughput critical systems => Maximization of hardware parallelism, model based design for reuse
- Latency critical systems => Specific hardware, specific software/firmware, implementation of real time systems results

Conclusions

Questions??