Upgrade of the ATLAS DAQ System

William Panduro Vazquez
(on behalf of RHUL TDAQ group)
RHUL, 14th May 2014
Outline

• DAQ Basics
• ATLAS Readout Challenge
• ATLAS DAQ System in Run 1
• RHUL and ATLAS TDAQ
• Run 2 Challenges
• New Readout System – RobinNP
  – FPGA’s and their role in DAQ
• Installation Schedule
DAQ Basics

• Simple example: temperature sensor
DAQ Basics

• Simple example: temperature sensor
DAQ Basics

• Simple example: temperature sensor

Limited by:
  Analogue-to-Digital conversion rate
  Data recording rate
DAQ Basics

• What if process is unpredictable?
  – e.g. radioactive decay

Can waste a lot of time recording nothing of interest!
DAQ Basics

• What if process is unpredictable?
  – e.g. radioactive decay
DAQ Basics

• What if process is unpredictable?
  – e.g. radioactive decay

Need to delay data to ADC to synchronise with trigger: latency
DAQ Basics

• Not whole story
  – What if new signal arrives while still processing old one?

Time when system not available to new signals: **deadtime**

Dominated by slow component
DAQ Basics

- Solution: ‘Derandomising’ buffer
  - ‘First-in First-out’ structure (FIFO)

Decouple front-end (fast) from readout (slow)
No need for more expensive and complex readout
DAQ Basics

• Collider Experiment:
  – We know beam crossing rate, can predict when data available
  – No need for initial delay, but still have deadtime from readout
DAQ Basics

• Collider Experiment:
  – No problem if trigger latency less than bunch crossing period
  – Not guaranteed with more complex triggers – add more levels
DAQ Basics

- Collider Experiment:
  - Better results if you can do more complex calculations after readout stage – add a ‘high level’ trigger
  - Readout now functions also as a buffer

With thanks to Brian Petersen (CERN) for the inspiration!
DAQ Basics

• Increased complexity when reading out many detectors
• Each component of previous diagram becomes a subsystem in itself!
• In ATLAS this is particularly challenging...
ATLAS Readout Challenge
ATLAS Readout Challenge: Tracking

- 80M channels
- 6M channels
- 350k channels
ATLAS Readout Challenge: Calorimetry

185k channels

500k channels

185k channels

500k channels
ATLAS Readout Challenge: Muons

440k channels → Thin-gap chambers (TGC) → 350k channels

Cathode strip chambers (CSC) → 70k channels

Monitored drift tubes (MDT) → End-cap toroid → 380k channels

Barrel toroid → Resistive-plate chambers (RPC) → 350k channels
## ATLAS Readout Challenge: Rates

<table>
<thead>
<tr>
<th>Property</th>
<th>Design Expectation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total channels to read out</td>
<td>~100M</td>
</tr>
<tr>
<td>p-p collision rate</td>
<td>~$10^9$ per second</td>
</tr>
<tr>
<td>Incoming data volume (front end)</td>
<td>~1 PB/s (!)</td>
</tr>
<tr>
<td>Data written to disc</td>
<td>~1 GB/s (~3.2 PB per year)</td>
</tr>
<tr>
<td>Average event size</td>
<td>1.6 MB</td>
</tr>
</tbody>
</table>
## ATLAS Readout Challenge: Rates

<table>
<thead>
<tr>
<th>Property</th>
<th>Design Expectation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total channels to read out</td>
<td>~100M</td>
</tr>
<tr>
<td>p-p collision rate</td>
<td>~$10^9$ per second</td>
</tr>
<tr>
<td>Incoming data volume (front end)</td>
<td>~1 PB/s (!)</td>
</tr>
<tr>
<td>Data written to disc</td>
<td>~1 GB/s (~3.2 PB per year)</td>
</tr>
<tr>
<td>Average event size</td>
<td>1.6 MB</td>
</tr>
</tbody>
</table>

High performance DAQ crucial!
ATLAS TDAQ System (Run 1)

20 MHz (*)

75 kHz

4-600 Hz

(*) due to 50ns bunch crossing rate instead of 25ns lower rate but more interactions per crossing (pileup)

~1 PB/s

120 GB/s

1 GB/s
ATLAS TDAQ System (Run 1)

- 20 MHz (*) due to 50ns bunch crossing rate instead of 25ns lower rate but more interactions per crossing (pileup)
- 75 kHz
- 4-600 Hz
- ~1 PB/s
- 120 GB/s
- 1 GB/s

(*) due to 50ns bunch crossing rate instead of 25ns lower rate but more interactions per crossing (pileup)
Front-end Readout

- Channels from front end aggregated and analysed through custom electronics
  - Perform pulse shaping, ADC etc.
Front-end Readout

- Channels from front end aggregated and analysed through custom electronics
  - Perform pulse shaping, ADC etc.
  - Example: LAr calorimeter front-end board (128 channels)
Front-end Readout

- Channels from front end aggregated and analysed through custom electronics
  - Perform pulse shaping, ADC etc.
  - Example: LAr calorimeter front-end board (128 channels)
- In calorimeter and muon system, data then passed to **Level 1 trigger**
Front-end Readout

- Channels from front end aggregated and analysed through custom electronics
  - Perform pulse shaping, ADC etc.
  - Example: LAr calorimeter front-end board (128 channels)
- In calorimeter and muon system, data then passed to Level 1 trigger
- Based on L1 decision data then aggregated further within readout drivers (RODs)
  - 2 x FEB per ROD = 256 channels
  - ~715 RODs for whole LAr calo
- RODs also perform more advanced signal processing as needed, assemble data into ‘fragments’
Front-end Readout

- Channels from front end aggregated and analysed through custom electronics
  - Perform pulse shaping, ADC etc.
  - Example: LAr calorimeter front-end board (128 channels)
- In calorimeter and muon system, data then passed to Level 1 trigger
- Based on L1 decision data then aggregated further within readout drivers (RODs)
  - 2 x FEB per ROD = 256 channels
  - ~715 RODs for whole LAr calo
- RODs also perform more advanced signal processing as needed, assemble data into ‘fragments’
- Finally, data passed to readout system (ROS) via 160 MB/s optical readout link (ROL)
  - ~1600 ROLs in run 1

Where we come in!
Readout Drivers

- RODs are detector specific
- All pass data at varying rates to ROS

Lab test of muon ROD crate with readout links
ATLAS TDAQ System (Run 1)

20 MHz (*)

75 kHz

4-600 Hz

1 PB/s

120 GB/s

1 GB/s

(*) due to 50ns bunch crossing rate instead of 25ns lower rate but more interactions per crossing (pileup)
ATLAS High Level Trigger

• Entirely software-based system
• Based in large server farm on surface above ATLAS
• Approaching 20k cores

• Split in run 1 into 2 conceptual blocks
  – Level 2 (manages ‘region of interest’ event building decision)
  – Event Filter (more complex analysis of built events before writing to disc)
Region of Interest Triggering

- ATLAS High-Level Trigger based on ‘region of interest’ ROI triggers
  - L1 trigger passes information on segments of the detector containing events for further study to ‘ROI builder’
  - Jets, EM, muon candidates etc.
- HLT requests data for interesting regions from readout system
  - Fragments passing selection read out for event building (EB)
- Incoming rate (L1) ~75 kHz while full EB rate ~3-6 kHz but similar overall bandwidth thanks to ROI building
  - Similar technology can process both phases, keeping down costs
ATLAS TDAQ System (Run 1)

20 MHz (*)

75 kHz

4-600 Hz

1 PB/s

120 GB/s

1 GB/s

(*) due to 50ns bunch crossing rate instead of 25ns lower rate but more interactions per crossing (pileup)

Today's focus
Readout System

• Function:
  – Receive and buffer event data after L1 accept
  – Serve data to high level trigger (HLT) and hold buffered data for entire HLT latency period
  – Read out desired data to event building system on HLT accept, delete rejected events

• Based on off-the-shelf PCs with custom hardware for the buffers
• ~150 PCs in run 1
• Each PC can service up to 12 readout links
Readout System: ROBIN Card

• Buffering and management of event data implemented in ‘ROB-IN’ card – designed largely at RHUL
• Up to 4 cards per ROS PC
• Services 3 readout links
  – Max 480 MB/s input rate
  – 64 MB buffer per link
• As ATLAS does not need 100% of data the ROBIN allows high link density to be achieved cost-effectively
  – PC doesn’t need to be extremely powerful (expensive) to handle multiple cards
RHUL and ATLAS TDAQ

• RHUL have been involved in the readout system of what became ATLAS since early 90’s (J. Strong, G. Boorman, B. Green)
  – Involved throughout its evolution from VME (or earlier) based platform to current PCI cards
  – Selection available to view at the front!
  – Responsible for ‘winning’ design which evolved into current ROBIN
  – Leading the design of RobinNP (B. Green, W. Panduro Vazquez)
• Simon George is current ATLAS Trigger group lead
  – Working on trigger since 1995 on wide range of activities from early signature performance studies (contributing heavily to final TDR) moving on to core software (online and offline) and finally management
• Francesca Pastore (see recent seminar!) heavily involved with future tracking trigger upgrade as well as overhaul of high level trigger control framework (steering) for 2015
• Pedro Teixeira-Dias and Veronique Boisvert contributing and ROS and L1Track projects respectively as well as joint effort on SW validation and coordination
RHUL and ATLAS TDAQ
ATLAS TDAQ System (Run 1)

20 MHz (*)

75 kHz

4-600 Hz

(*) due to 50ns bunch crossing rate instead of 25ns lower rate but more interactions per crossing (pileup)
ROS Performance in Run 1

1 word = 4 bytes

10% - 15% readout capability
Changes for Run 2

• New detector components
  – Insertable B layer ‘IBL’
    • 4th layer added to pixel detector, more channels to read out
  – Fast tracker ‘FTK’ to be integrated by 2016
    • Uses fast pattern recognition to provide more complete track information to HLT
    • Precursor to potential full tracking trigger at L1 in next upgrade (2018)

• Most subsystems require more readout capacity and or increased rate from particular channels
  – Number of readout links increasing from ~1600 to over 2000

• L1 trigger components upgraded plus addition of ‘topological’ trigger to cope with higher rates – overall rate raised to 100 kHz
  – Can use event topology to form triggers based on angular separation, invariant mass, interaction hardness

• Overall requirement for larger data throughput
• High level trigger modified to merge L2 and EF functionality for more efficient use of resources
  – More nodes added taking total up to 30k
ATLAS TDAQ System (Run 2)

40 MHz

100 kHz (*)

1 kHz

1 PB/s

200 GB/s

1-2 GB/s

(*) to be demonstrated if all detectors can achieve this
Readout System Upgrade

• Existing ROS only just capable of satisfying required rates for run 2 with no contingency (at least 400 extra links)
• Rack space becoming a problem with current link density

<table>
<thead>
<tr>
<th>Readout Scenario</th>
<th>Pixels: per ROB (kHz)</th>
<th>Pixels: per ROS PC (kHz)</th>
<th>HEC: per ROB (kHz)</th>
<th>HEC: per ROS PC (kHz)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2012, 75 kHz L1A rate</td>
<td>12</td>
<td>30</td>
<td>21</td>
<td>33</td>
</tr>
<tr>
<td>Scenario 1</td>
<td>18</td>
<td>48</td>
<td>30</td>
<td>53</td>
</tr>
<tr>
<td>Scenario 2</td>
<td>21</td>
<td>51</td>
<td>20</td>
<td>43</td>
</tr>
<tr>
<td>Scenario 3</td>
<td>23</td>
<td>53</td>
<td>25</td>
<td>48</td>
</tr>
<tr>
<td>Gen II ROS PC, 10 GBE</td>
<td>22</td>
<td>50</td>
<td>24</td>
<td>45</td>
</tr>
</tbody>
</table>

• Run 1 suggests requirements will evolve beyond expectation
• Beginning to run into hardware obsolescence and end-of-life problems
• Large number of existing spare ROBINs would be needed to satisfy increased link requirements
• Requirement: manage 100 kHz input rate from L1 while reading out 50% of this data (~50 kHz aggregate output rate)
ROBIN obsolescence

• Increasingly difficult to find support for 64-bit PCI in new server motherboards
  – Harder to upgrade ROS PCs to improve performance
  – RobinExpress (PCIe version) built as a solution but not used

• On-board PPC cannot be upgraded

• Increasing number of HLT nodes place greater demands on memory capacity

• Boards will reach 10 years of age before next upgrade opportunity
  – Risk of end-of-life problems with minimal spares
RobinNP

• Hardware based on board developed in parallel by ALICE (Common Readout Receiver Card or ‘C-RORC’)
  – Custom ATLAS firmware to implement enhanced ‘Robin’ functionality
• 12 ROLs running at up to 200 MB/s each
• PCI express
  – Standard with long term support
  – Capable of meeting highest predicted rates
• 8 GB on board RAM
• On-board PPC no longer present, manage its tasks on host PC
  – Easier to maintain, can upgrade CPU for performance boost
FPGA’s and their role in DAQ

• ‘Field Programmable Gate Array’
  – Advanced programmable logic chip
  – Evolved from earlier, more basic programmable logic devices (PLDs)
• Faster than a regular microprocessor
  – Contains fixed logic for faster routing
• More flexible than an integrated circuit
  – Internal logic can be changed retaining majority of speed
FPGA’s and their role in DAQ

- Earliest PLDs were just large arrays of AND and OR gates with ‘flip-flops’ to store state
- User could connect gates up in different ways to achieve different logic
FPGA’s and their role in DAQ

• As things evolved ‘complex’ PLDs (CPLDs) were created
  – Essentially an array of PLDs with more complex interconnection structure
• The ultimate evolution of this was the FPGA
  – An array of small logic units with many different available functions
  – Extremely complex interconnection structure

• In modern FPGA’s each logic block is made up of lookup tables for a pre defined set of logical operations
• Multiple flip-flops per block to store more complex state
• Potentially millions of logic blocks per chip
FPGA’s and their role in DAQ

• Newest FPGAs also contain more advanced structures
  – Clocks, memory, microprocessor emulations...

• Extremely popular in R&D both in academia and in industry
  – Combination of flexibility and speed make them idea for prototyping
  – Used extensively throughout ATLAS hardware
    • RobinNP, RODs, L1 trigger etc...
FPGA Example: RobinNP Structure

Complex logic all built from the basic logic blocks seen earlier
Old ROS vs New ROS

<table>
<thead>
<tr>
<th>Card</th>
<th>ROBIN</th>
<th>RobinNP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Links</td>
<td>3</td>
<td>12</td>
</tr>
<tr>
<td>Memory per link (MB)</td>
<td>64</td>
<td>682</td>
</tr>
<tr>
<td>Output Bandwidth (MB/s)</td>
<td>266</td>
<td>1600 (3200 possible)</td>
</tr>
<tr>
<td>Max Input Bandwidth (MB/s)</td>
<td>480</td>
<td>2400</td>
</tr>
</tbody>
</table>

- New ROS PCs will be half the size of the old
- 2 RobinNP’s per ROS PC
- Link density 4x greater
  - Significant space, power and cooling saving
  - Much greater scope for later expansion
- 40 GbE data connections in each PC
  - No network bottleneck at ROS stage
Achieving RobinNP Performance

• As the processing is now done in the host PC the on-board control data are now across a PCIe bus.
  – Reading very slow
  – Implemented novel protocol to transfer data only using writes ‘fifo duplicator’ (see extra slides)
    • Host memory ring buffer mirroring hardware fifo with joint HW and SW control
  – Much faster performance

• Use modern interrupt technology to signal new data/control information to software

• Modular firmware allowing two parallel DMA transactions for main dataflow
  – Makes more optimal use of the bus
Achieving RobinNP Performance

• Refactor of software to merge functionality and avoid needless inter-process communication

• Improved multithreading
  – Previous readout software inefficient, many CPU cycles wasted
  – Poll and sleep model with many dual purpose threads
  – Overhauled threading model to eliminate waste using more modern signalling/semaphores/interrupts
  – Threads have single well constrained function and dormant until needed, rather than for arbitrary time period
**ROS Performance (Run 2 Predicted)**

- **X** marked line: 2 RobinNPs, 100% Readout
- **△** marked line: 2 RobinNPs, 50% Readout

ATLAS Requirement @ 50%
New ROS status and performance

• All main dataflow features implemented
• Test results very encouraging
• Currently working on final debugging and inclusion of maintenance features
• Production order for 200 C-RORC cards released
  – Hardware construction & testing ongoing at manufacturer site
• Selection of suitable host PC to be completed this week
Installation Schedule

- ROS must be ready for data on 1\textsuperscript{st} Feb 2015

<table>
<thead>
<tr>
<th>Step</th>
<th>Completion date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ordering C-RORCs, memory and QFSPs</td>
<td>1 May</td>
</tr>
<tr>
<td>Ordering PCs</td>
<td>15 May</td>
</tr>
<tr>
<td>Delivery of C-RORCs, memory, QFSPs and PCs</td>
<td>15 August</td>
</tr>
<tr>
<td>Upgrade of the first ROS rack (rack yet to be nominated)</td>
<td>15 September (with some contingency)</td>
</tr>
<tr>
<td>Stability test with this rack</td>
<td>1 October</td>
</tr>
<tr>
<td>Upgrade of the remaining racks</td>
<td>15 November</td>
</tr>
<tr>
<td>Decision: Keep GEN-III or revert to GEN-II</td>
<td>1 December</td>
</tr>
</tbody>
</table>

On track
LHC Long Shutdown 1 ‘Dashboard’

Physics by end of March 2015
Thanks for listening!

Any questions?
Extra Slides
FIFO Duplicator Schematic

RobinNP Firmware

- Combined Used Page FIFO
  - FIFO Fill

- FIFO Duplication Control
  - Write Index
  - Read Index Copy
  - Read Index

DMA Engine

- DMA Descriptor
  - DMA Write

PCIe Bus

- Write Index Copy
  - PIO Write

Ring Buffer

- Host system Memory

Duplicator Handling Software
FIFO Duplicator Protocol

• FIFO Duplicator Control (Firmware)
  – Compares write index and read index copy to establish if space available in ring buffer
  – Compare available space with FIFO fill
    • Transfer smaller of the two values with write index as starting point
    • Stop at space before read index copy
  – Update write index and write index copy (using DMA)
  – Issue interrupt to wake software thread

• Ring Buffer Control (Software)
  – Dormant until interrupt received
  – Compares write index copy and read index to establish fill of ring buffer
  – Read data from ring buffer starting at read index (but stopping in space before write index copy)
  – Update read index and read index copy (using PCIe write)
  – Reset interrupter and await new signal
Gen III ROS SW Architecture

Descriptor Flow
Page Flow
DMA & DMA Control
HLT Traffic and Data Collection

Request Processor Thread (1 per ROBgroup)
Request Queue (Semaphore)
Receive DMA (FIFO + Interrupts)

DMA Monitoring Thread (1 per ROBgroup)

Data Collector Thread
Free Descriptor Queue (Semaphore)

Async IO Threads

HLT Traffic

RobinNP Hardware

Queue DMA

RobinNP Readout Module

Memory Pages (FIFO + Interrupts)

Processed Descriptor Queue (Semaphore)