21st IEEE Real Time Conference

iFDAQ for the COMPASS experiment

Bodlák Martin, Frolov Vladimir, Huber Stefan, Jarý Vladimír, Konorov Igor, Květoň Antonín, Nový Josef, Steffen Dominik, Šubrt Ondřej, Tomsa Jan, Virius Miroslav

Sponsored by:
1. Motivation and Concept of Hardware Event Building

2. Design of the intelligent FPGA-based DAQ
   - System
   - Hardware
   - (Firmware)
   - (Software)

3. iFDAQ – integration and performance in COMPASS
   - Setup of COMPASS spectrometer
   - iFDAQ – Performance and Status in 2017/2018

4. Switching Network Topology
   - Required Hardware Developments
   - Software Developments

5. Outlook and Conclusion
1. **Motivation and Concept of Hardware Event Building**

2. **Design of the intelligent FPGA-based DAQ**
   - System
   - Hardware
   - (Firmware)
   - (Software)

3. **iFDAQ – integration and performance in COMPASS**
   - Setup of COMPASS spectrometer
   - iFDAQ – Performance and Status in 2017

4. **Switching Network Topology**
   - Required Hardware Developments
   - Software Developments

5. **Outlook and Conclusion**
Different DAQ Systems

**Motivation/Concept**

- Design of the iFDAQ
- Performance
- Switching Network Topology
- Conclusion

Concept of iFDAQ:
Minimize amount of real-time processes and push Event Building into FPGA-world

Feasibility due to recent FPGA developments:
- Increased I/O bandwidth
- Support for high-performance SDRAMs

Expected Advantages:
- Increased Compactness
- Increased Reliability
- Higher Scalability
- Reduced Costs
Event Builder – CPU vs. FPGA

**Event Building:** combination of logically connected, but physically split data fragments

**Buffer PCs:**
- Buffers data until receiving control msg to forward to specific builder unit
- Replicated over number of computers to fit performance needs and application scenario

**Event Builder PCs:**
- Collects event fragments and combines them into complete event
- Replicated over number of computers to fit throughput requirements

**Motivation/Concept**
- Design of the iFDAQ
- Performance
- Switching Network Topology

**Usage of FPGAs and exploiting its properties:**
- Parallel processing
- Pipeline architectures

**Optional multiplexing stages to reduce number of incoming links**

**Distribution of data flow to different computer nodes**
Event Builder – CPU vs. FPGA

**CPU (sequential)**
- Buffer PCs
- Ethernet Network Switch
- Event Builder PCs
- Data Storage Network
- CDR

**Advantages**
- Easy integration of redundancy elements
- Uses mass-produced components
- Knowledge and templates available

**Disadvantages**
- Throughput limited by EB-network
- Performance of sequential execution strongly depends on algorithm complexity
- Recovery of hanging processes takes significant time

**FPGA (parallel)**
- MUX stages
- MUX

**Advantages**
- Continuation of the pipeline architecture in FEE
- Only FPGA allows to build real real-time systems
- High scalability
- High reliability
- Reduced costs

**Disadvantages**
- Long development
  But: Progress in higher-level tools (System Verilog, OSVVM)

⇒ **Motivation for**
- Minimizing real-time processes
- Development of highly automatized and reliable DAQ

---

Dominik Steffen | Real Time – Williamsburg | 14/06/2018
Scaling Possibility of Hardware EB

Motivation/Concept

Design of the iFDAQ

Performance

Switching Network Topology

Conclusion

Scenario for:

- Xilinx 7-series FPGA
- SLINK interfaces replaced by Aurora

Graph:

- costs [k€] vs. data throughput [GB/s]
- 1.2 M€
- 80 k€
- 4 k€
1. Motivation and Concept of Hardware Event Building

2. Design of the intelligent FPGA-based DAQ
   • System
   • Hardware
   • (Firmware)
   • (Software)

3. iFDAQ – integration and performance in COMPASS
   • Setup of COMPASS spectrometer
   • iFDAQ – Performance and Status in 2017

4. Switching Network Topology
   • Required Hardware Developments
   • Software Developments

5. Outlook and Conclusion
FPGAs take full responsibility for reliable data transmission from FEEs to PC.

Data buffering:
- 4GB RAM on each module => possibility to average data rate over spill cycle
- 1 GB/s sustained rate

Independent interfaces:
- synchronization → TCS (Trigger Control System)
- data flow (event building) → SLINK
- configuration and data flow control → IPbus
iFDAQ – System Design

FPGAs take full responsibility for reliable data transmission from FEEs to PC

Intelligence Elements in Hardware:
- Self-synchronized event building data flow (back pressure and throttling)
- FEE error diagnostic and handling to prevent DAQ crash
  \[\Rightarrow\] monitoring and localization of failing FEEs
- Automatic resynchronization of FEEs
  \[\Rightarrow\] FEEs can be attached/detached at any time

Independent interfaces:
- synchronization \(\rightarrow\) TCS (Trigger Control System)
- data flow (event building) \(\rightarrow\) SLINK
- configuration and data flow control \(\rightarrow\) IPbus
# Data Handling Card (DHC)

## AMC module
- **form factor:** AMC standard
- **FPGA:** Virtex6 XC6VLX130T
- **memory:** 4 GB DDR3 SDRAM
- **firmware:**
  - [DHCmx](#) 12:1 multiplexer
  - [DHCsw](#) 8x8-switch
- **data rate:**

![Graph showing data rate vs. event size]

## VME carrier card
- **form factor:** 6 U VME
- **interfaces:**
  - TCS (Trigger Control System) receiver
  - 1 Gb Ethernet for control network (IPbus)
  - 16 serial data links (SLINK)
  - JTAG for backup programming of FLASH

---

[DHCmx](#): 12:1 multiplexer

[DHCsw](#): 8x8-switch
1. Motivation and Concept of Hardware Event Building

2. Design of the intelligent FPGA-based DAQ
   • System
   • Hardware
   • (Firmware)
   • (Software)

3. iFDAQ – integration and performance in COMPASS
   • Setup of COMPASS spectrometer
   • iFDAQ – Performance and Status in 2017 and 2018

4. Switching Network Topology
   • Required Hardware Developments
   • Software Developments

5. Outlook and Conclusion
COMPASS – Overview

- Fixed target experiment at SPS accelerator at CERN (M2 beamline)
- High intensity beams: $4 \cdot 10^7 \frac{\mu}{s}; 2 \cdot 10^7 \frac{\text{hadrons}}{s}$
- Multi-purpose experiment
- Start of data-taking: 2001
- Since 2014: New DAQ with hardware event builder (iFDAQ)
COMPASS – Spectrometer Setup

spectrometer facts
- Length: 60m
- Amount of channels: 300,000
- Trigger rate: 30 kHz
- On-spill data rate: 1.5 GB/s
- Event size: 20-50 kB

- Fixed target experiment at SPS accelerator at CERN (M2 beamline)
- High intensity beams: $4 \cdot 10^7 \mu \text{s}^{-1} ; 2 \cdot 10^7 \text{hadrons s}^{-1}$
- Multi-purpose experiment
- Start of data-taking: 2001
- Since 2014: New DAQ with hardware event builder (iFDAQ)

Source [2]
iFDAQ in COMPASS – Hardware Parts

- **Very compact:** 30 online PC in **former** DAQ
  **Now:** 1 VME crate (6-U) + one rack (8 computers)

- **Highly flexible:** Easily adaptable to different spectrometer setups (e.g. DVCS 2017 vs. Drell Yan 2018)
Performance – Accumulated Data 2017

Collected Data

Total data size: 1389 TB
Mean data rate: 414 GB/h
Performance – Uptime in 2017

Effects:
1. Detectors more stable during run than during commissioning phase
2. Upgrades of RCCAR software

Highly reliable
1. Motivation and Concept of Hardware Event Building

2. Design of the intelligent FPGA-based DAQ
   - System
   - Hardware
   - (Firmware)
   - (Software)

3. iFDAQ – integration and performance in COMPASS
   - Setup of COMPASS spectrometer
   - iFDAQ – Performance and Status in 2017

4. Switching Network Topology
   - Required Hardware Developments
   - Software Developments

5. Outlook and Conclusion
Disadvantages:
- Throughput limited by EB-network switch
- Inefficient usage of max. bandwidth due to:
  - Improper comm. pattern (N senders -> 1 receiver) => network congestion
  - Data overhead due to addressing etc.

Advantages:
- Independence of network switch
- Efficient usage of link bandwidth (no addressing etc.)
- High reliability

Disadvantages:
- Strong dependence on reliability of network nodes
  (no rerouting possibility in case of hardware failure)
- No possibility for dynamic network optimization (e.g. load balancing)
Crosspoint Switch - Integration

Crosspoint Switch

- connects:
  - FE electronics
  - DHCmx modules
  - DHCsw module
  - Spillbuffers

- purpose:
  - Ease of load balancing
  - System redundancy to compensate hardware failures

⇒ provides fully customizable network topology
Crosspoint Switch – Hardware Design

Crosspoint Switch Components

- **Interfaces:**
  - 12 x 12 channel CXP transceiver (MPO fiber connectors)
  - Ethernet for IPbus
  - JTAG
  - TCS (Trigger Control System) receiver

- **Switching and Control:**
  - **Vitesse VSC3144-02** – fully configurable 144x144, asynchronous, 6.5 Gbps crosspoint switch
  - **Xilinx Artix-7 FPGA** for switch control and monitoring

- **Interface FPGA – Crosswitch:**
  - 90 MHz, 11-bit parallel data bus
  - Multiple program assignments can be queued and issued simultaneously ⇒ fast programming (≪ 1us)
Crosspoint Switch – Hardware Test

<table>
<thead>
<tr>
<th>Line rate</th>
<th>Signal BEFORE switch</th>
<th>Signal AFTER switch</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.13</td>
<td>![Before 3.13]</td>
<td>![After 3.13]</td>
</tr>
<tr>
<td>5.00</td>
<td>![Before 5.00]</td>
<td>![After 5.00]</td>
</tr>
<tr>
<td>6.25</td>
<td>![Before 6.25]</td>
<td>![After 6.25]</td>
</tr>
</tbody>
</table>

Input Signal Equalization on Switch

No ISE

ISE
1. Motivation and Concept of Hardware Event Building

2. Design of the intelligent FPGA-based DAQ
   • System
   • Hardware
   • (Firmware)
   • (Software)

3. iFDAQ – integration and performance in COMPASS
   • Setup of COMPASS spectrometer
   • iFDAQ – Performance and Status in 2017

4. Switching Network Topology
   • Required Hardware Developments
   • Software Developments

5. Outlook and Conclusion
Ongoing development:

- Integration of crosspoint switch
  - test of crosspoint switch successful and promising
  - integration of the hardware for the crosspoint switch (results to be expected soon)
  - upgrade of Software for automatic identification of malfunctioning hardware parts

Ideas for the Future:

- upgrade of TCS to bidirectional PON (passive optical network) with use of Universal Communication Framework (UCF) developed at TUM for on-the-fly reconfiguration of interconnections
- Minimizing of real-time processes
  - direct writing of data onto SSD
Improvements since commissioning of iFDAQ in 2014:

- Increased reliability (Uptime around 99%)
- Extended intelligence elements in software:
  - Automatic safe stop of the run for self-recovery
  - Continuously running
- No event size limit due to upgrades in firmwares

Performance in 2017:

- Data rate: 91.7 MB/s (average)
  - 250 MB/s (in stable beam conditions)
  - 380 MB/s (peak sustained rate)
- On-spill data rate: 1.5 GB/s

iFDAQ transferred to other HEP experiment (NA64)
Possible DAQ Systems

- Motivation/Concept
- Design of the iFDAQ
- Performance
- Switching Network Topology
- Conclusion

Diagram:
- Trigger Logic -> Front-End
- Front-End -> MUX
- MUX -> Event Builder
- Event Builder -> HLT
- HLT -> Storage

- iFDAQ
- HLT (beyond 2020)
- FPGA WORLD
- PC WORLD
THANK YOU for your Attention
References

[1] Y. Bai et al., *New data acquisition system for the COMPASS experiment*, in proceedings of *Topical Workshop on Electronics for Particle Physics (TWEPP)*, Lisbon Portugal September 2015

Backup slides
DHC – Data error handling in firmware

Data Consistency Check:
- Transmission errors detected by S-Link
- Truncation, i.e. mismatch between real and declared data block size
- Inconsistency of event label
- Missing data -> timeout

Error handling:
- Discarding/throttling of wrong data
- Adding of specific header for empty frame
- Setting error flag in local register (diagnostics of FEE errors)
Firmware – DHCmx

Data Consistency Check:
- Transmission errors detected by S-Link
- Truncation, i.e. mismatch between real and declared data block size
- Inconsistency of event label
- Missing data -> timeout

Error detected
- Discarding/throttling of wrong data
- Adding of specific header
- Setting error flag in local register
Firmware – DHCsw

Preallocated memory banks

Motivation
Design of the iFDAQ
iFDAQ in COMPASS
Redundancy Logic
Conclusion

Event size [B]

Absolute limit

Data rate [MB/s]

0 500 1000 1500 2000 2500 3000 3500

10 100 1000 10000
Run Control, Configuration, And Readout Software (RCCARS):

- **Multilayer system** around master process
  - **master:** Middleman between GUI, DB, all other processes
  - **GUIs** display overall status of iFDAQ, control behavior of iFDAQ (when in Control mode)
  - **readout slaves (only real-time processes)** readout of data, analysis of events, error check, transformation into DATE-format, distribution to monitoring tools, writing to HDD
  - **control slaves** monitoring and control of hardware nodes
  - **message logger** collection of msgs from processes, storing msgs in DB
  - **message browser** display of msgs and support for advanced filtering
  - **Central Data Recording (CDR)** transfer of raw data to CASTOR, disk cleaning for new data

- Inter-process communication via **DIALOG library** (Custom-developed server/client communication)

**ICHEP 2017 Paris**
Run Control, Configuration, And Readout Software (RCCARS):

- **Multilayer system** around master process
  - **master**: Middleman between GUI, DB, all other processes
  - **GUIs**: display overall status of iFDAQ, control behavior of iFDAQ (when in Control mode)
  - **readout slaves (only real-time processes)**: readout of data, analysis of events, error check, transformation into DATE-format, distribution to monitoring tools, writing to HDD
  - **control slaves**: monitoring and control of hardware nodes
  - **message logger**: collection of msgs from processes, storing msgs in DB
  - **message browser**: display of msgs and support for advanced filtering
  - **Central Data Recording (CDR)**: transfer of raw data to CASTOR, disk cleaning for new data

- Inter-process communication via **DIALOG library** (Custom-developed server/client communication)

**Intelligence Elements in Software:**
- Data format consistency check for whole stream
- Reduced data stream to monitoring tools (higher level check)
- Automatic stop of run and notification of shift crew on critical issues
- Continuous data processing through EB and to monitoring tools (Start of run just enables data writing to disk)
Run Control & Node Status GUI

Motivation

Design of the iFDAQ

iFDAQ in COMPASS

Redundancy Logic

Conclusion

Link status

<table>
<thead>
<tr>
<th>Node Status GUI:</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>System overview:</strong> shows status of all nodes of hardware event builder</td>
</tr>
<tr>
<td><strong>Control of front-end modules:</strong> allows easy online including, excluding of front-end modules into DAQ and reloading of front-end modules</td>
</tr>
</tbody>
</table>
Hardware: Vitesse VSC3144

- 144 x 144 strictly non-blocking cross-point switch
- Up to 6.5 Gbps bandwidth per port
- No registers used in data path i.e. asynchronous data path => no restrictions on the phase, frequency, or signal pattern of any input (protocol independent)
- 45mm x 45mm 1072-pin BGA package
- Core programming on port-by-port basis OR simultaneous issuing of multiple queued assignments