DCS Detector Control System

- overview
- hardware
- firmware
- software
- status
- future
- people
DCS Overview

DCS-board
- single-board computer based on an Excalibur EPXA1
- ARM922T hardwired processor
- running Linux stored in a flash memory
- integrated PLD for custom logic with direct connection to the ARM

Tasks:
- configuration of the RCU-FPGA
- configuration of the FEE/ALTROS
- monitoring/controlling of the FEE
- configuration/readout of the TTCrx
- distributing the L1trigger
- distributing the L2trigger and the eventheader
DCS Hardware

**EPXA1**
- ARM922T processor
- APEX 20K100E embedded PLD

**Flash**
- 4 Mbyte

**SD-Ram**
- 32 Mbyte

**TTCrx chip + OptoLink**
- receives global detector clock
- transfers L1 trigger
- transfers L2a/r message

**Ethernet chip**
- physical layer between logic and medium

**JTAG**
- cross-configuration of DCSboards

**ADC**
- voltage monitoring on DCSboard

**DIMM144**
- connection to RCU
**Modules**

**Ethernet**
- provides the access to the ETH

**TTCrx**
- interface to the TTCrx chip
- write/read register over I2C
- maps all the TTCrx registers in a registerfile in the PLD
- decodes the L2a/r message. In case of a L2a message the trigger information is directly transferred over the RCU bus to the Data- assembler module

**FPGA-Config**
- configures the RCU-FPGA.
  worst case time: ca. 800ms
  (488722 bytes á 1.6µs)

**ADC-Readout**
- monitors the voltages on the DCS

**JTAG-Master**
- configures the chained DCS boards

---

**Diagram Description**

- **ARM922T**
- **PLD**
- **ETHERNET**
- **RCU-Access**
- **FPGA-Config**
- **TTCrx**
- **JTAG-Master**
- **ADC-Readout**
- **DIMM**
- **Error**
- **Warning**
- **RCUbus**
- **L2a**
Features
- using „message“ system. The complete datapath is transparent. Each transaction is acknowledged by the target.
- complete configuration data can be stored in the SD-Ram
- in case of a change of the configuration data only the necessary parts must be changed
- system is technology-independent, modular and independent from the RCU clock
- pipelined: next message is prepared while waiting for the target acknowledge
- system is fully integrated and in use
DCS Module RCU-ACCESS

Message-System for Configuration and Monitoring

- 6 different kind of messages:
  - Single Read/Write
  - Multiple Read/Write
  - Random Read/Write

- Payload consists of Address/Data

- Each Message is acknowledge and contains an Status report

- Following features are prepared and will be implemented soon:
  - Safety Message: This message is executed immediately and overwrites „protected mode“
  - CRC check
  - Pedestal compressing.
  - „protected mode“

Diagram:
- Header
- Payload
- Marker
- Message-Buffer
- Endmarker

31 1 2 3 0

FF
DCS Module RCU-ACCESS

RCU-bus
- bidirectional
- 16bit address space
- 32bit data width
- asynchronous protocol
- common strobe to validate addr/data
- acknowledge to indicate valid transfer

Transfer rate
- without synchronizer:
  32bit x 40MHz/4clkcycle = 40MByte/s
- with synchronizer: +4clkycle/trx
  32bit x 40MHz/8clkyclce = 20MByte/s

Reliability
- bus has been tested in a 12h stress test
  with approx. 200 billion transactions
  without any error

Synchronizer
- since the DCS board and the RCU are running on different clock domains,
  synchronizer has been implemented to guarantee the data integrity
DCS Module Trigger

Trigger-Modul
- reading out the trigger chip
- keeps all the info's in a registerfile
- different modes of readout
  - continually
  - intervall
  - complete/partial
- Error checking/correction optional (CRC/Huffman)

L2-Transfer
- In case of L2 accept, the trigger is getting the bus immediately to transfer the eventheader to the DA
- 2 L2-transfer-modes:
  - normal: 56 clkcycles
  - boosted: ca. 10 clkcycles
Status

- using OpenSource BusyBox Linux
- drivers for the different modules are written, tested and already in use
- DIM-Server and DIM-Client are implemented and tested but have to be modified due to the format of the configuration data format.
- Subscriber mechanism. The client subscribes to the server the desired services.
- 2 software routines to configure the RCU modules and the RCU-FPGA directly from the bash. Supports different dataformats and is able to execute scripts.
### DCS Status

**DONE!**
- Ethernet is implemented and has been fully tested
- Linux is up and running
- Communication to the TTCrx chip established
- Configuration and monitoring of the FEE/ALTROS is possible including a feedback from the modules if the data was accepted or not.
- The monitoring does not interfere with the datapath of the readout FEE-ALTRO-DA-SIU-DDL and can be used while the data-aquisition is in progress.
- Soft- and firmware to configure the RCU-FPGA have been developed and is currently under test.
- DIM-Server/Client architecture has been developed and tested and is currently been modified to match the format of Roland Bramms configuration data
- System has been used to debug and configure the full datapath within the RCU/DCS

**TO DO!**
- Test of the RCU-FPGA configuration controller
- Upgrade to the next version of the RCU-access module
- Integration of Roland Bramm’s configuration data into DIM
- Import of the TTCrx module into the CERN DCSboard
<table>
<thead>
<tr>
<th>Location</th>
<th>Names</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bergen</td>
<td>D. Röhrich, K. Ullaland, B. Pommersche, J. Lien, A. Martinsen, T. Alt, M. Richter, J. Alme</td>
</tr>
<tr>
<td>Köln</td>
<td>T. Krawutschke</td>
</tr>
<tr>
<td>Oslo</td>
<td>Sten Solli</td>
</tr>
<tr>
<td>Worms</td>
<td>S. Bablock</td>
</tr>
<tr>
<td>Christian Kofler</td>
<td></td>
</tr>
<tr>
<td>CERN</td>
<td>L. Musa, R. Campagniolo, C. Gonzales, B. Leto Mota</td>
</tr>
<tr>
<td>Heidelberg</td>
<td>V. Lindenstruth, D. Gottschalk, M. Stockmaier, V. Kiworra, T. Alt</td>
</tr>
</tbody>
</table>
DCS Firmware

ARM922T

Com/Stat

Mem-CTRL

M1

RCU-master

RCU-Config CTRL

M2

Mem-CTRL

FPGA-Config CTRL

Sequenzr

RCU board