Upgrade of the Data Read-Out and Control Systems of the HADES Detector

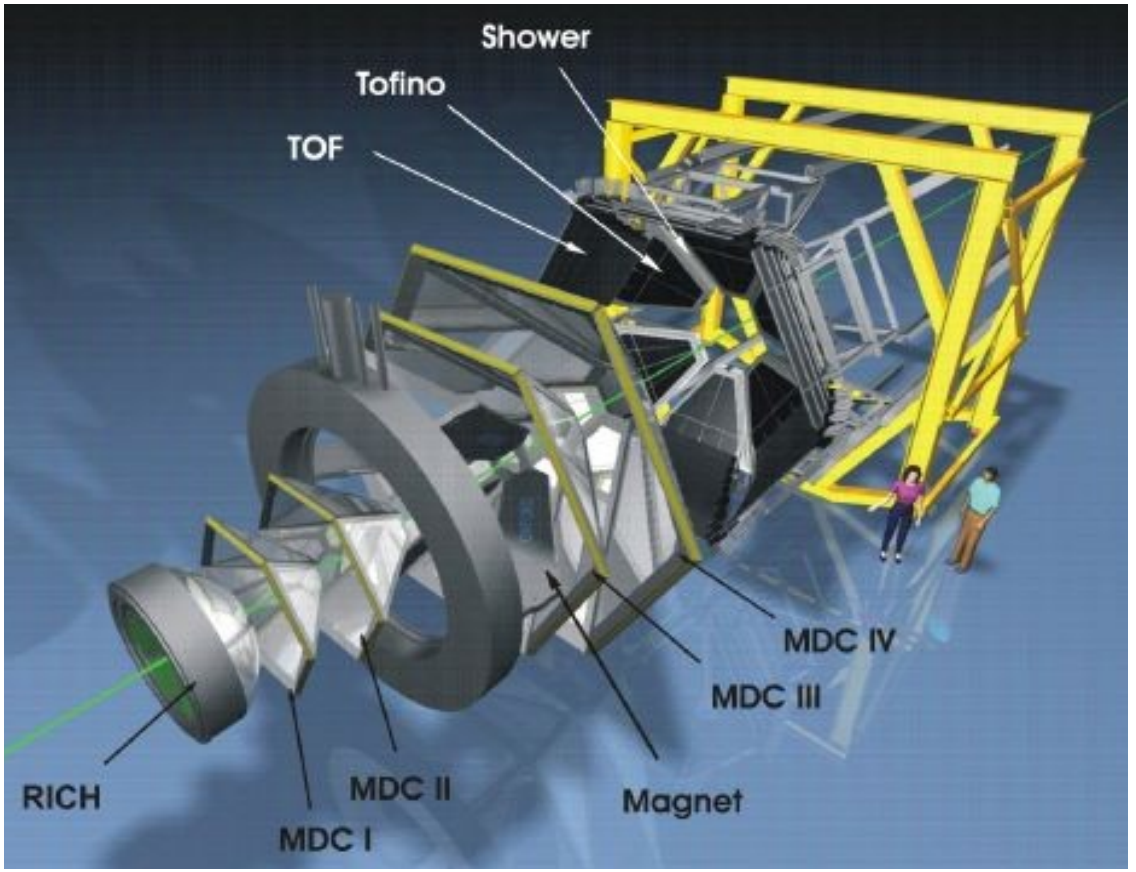
Jan Michel

HADES Collab. Meeting, Sesimbra 2009

HADES – High Acceptance DiElectron Spectrometer

- 8 distinct detector systems
- 80.000 data channels
- Passed many succesful beamtimes in the last years

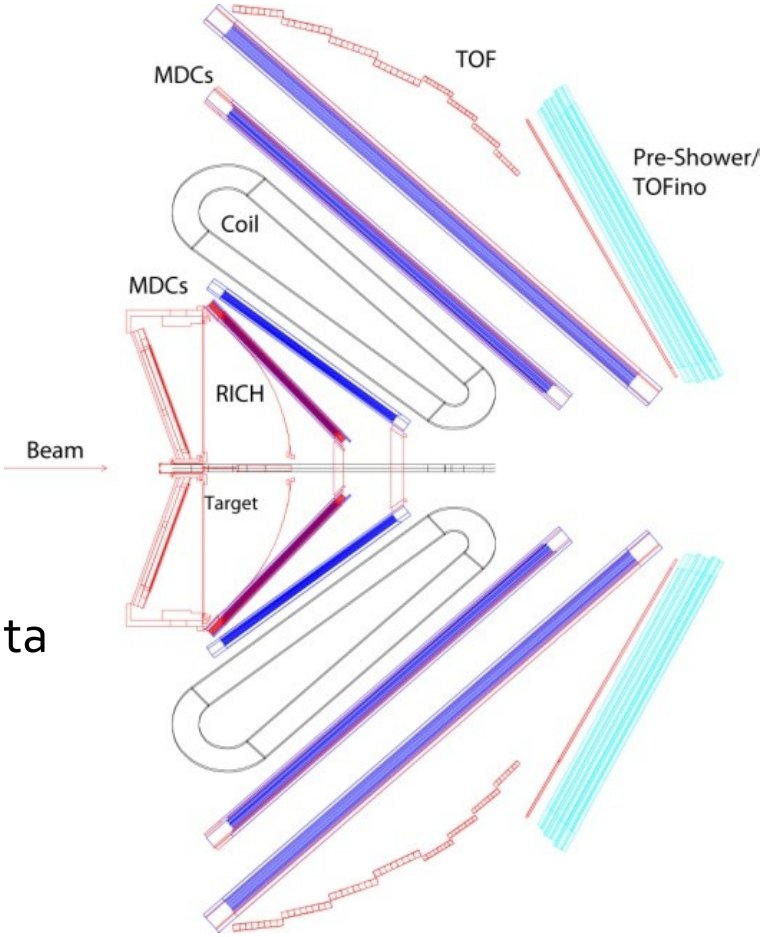
2002	**C** + **C** 2 AGeV
2004	**C** + **C** 1 AGeV
2004	**p** + **p** 2.2 GeV
2005	**Ar** + **KCl** 1.75 AGeV
2006	**p** + **p** 1.25 GeV
2007	**p** + **p** 3.5 GeV **d** + **p** 1.25 AGeV
2008	**p** + **Nb** 3.5 GeV



- Medium sized systems @ 1-3 AGeV
  - Events written to storage: 10k/s
  - Size per event: 1-2 kbyte
  - Data : up to 15 MByte/s

The DAQ System

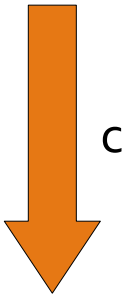
- LVL 1 trigger
  - Hit in Start but no hit in VETO detector
  - Analog electronics count number of charged particles in TOF
- If positive, Central trigger system distributes information to all detectors
- Detector data is digitized / parts of it are read-out
- LVL2 trigger
  - Based on TOF, RICH and Pre-Shower detector data
- If positive, all data is read-out and sent to processing nodes (VME grates)
- Data is sent via Ethernet to PC (EventBuilder)



DAQ Data Rates

current DAQ system

	Particles / Event	LVL1 Trigger Rate	Data Rate
p + p	5	10 – 20 kHz	10 MByte/s
Ca + Ca	40	3 – 4 kHz	10 MByte/s
Au + Au *	200	0.7 kHz	10 MByte/s

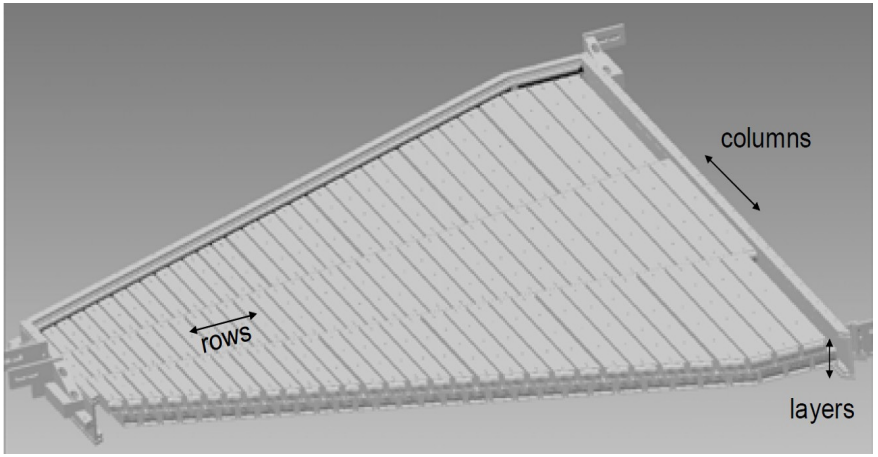
Improve data acquisition  capabilities by a factor of 20

upgraded DAQ system (design performance)

	Particles / Event	LVL1 Trigger Rate	Data Rate
p+p	5	100 kHz	100 MByte/s
Au+Au	200	20 kHz	200 MByte/s

Upgrade Program

- Replace TOFino with Resistive Plate Chambers (RPC)
  - Higher granularity
  - Better time resolution
- Rebuild one layer of drift chambers (MDC 1)
  - Higher efficiency than old chambers



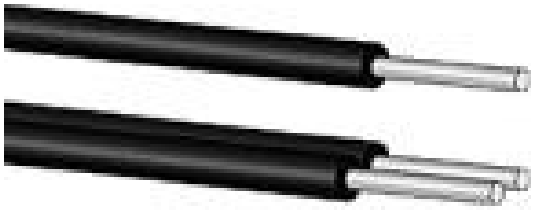
- New readout electronics for all detectors
- Integration of better control- and configuration possibilities (SlowControl)
- Development of a combined network transporting triggers, data and monitoring

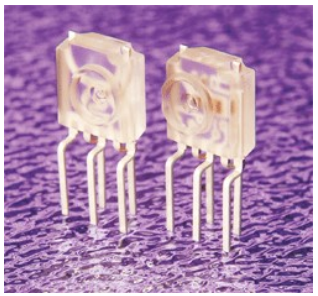
Data Transport Medium

- Fast switching signals cause noise in analog electronics
- Electrical connection causes ground loops
- Limited distances
- ~ 0.5 Gbit/s under good conditions
- Comparably high weight of cables
- Cable failures hard to detect

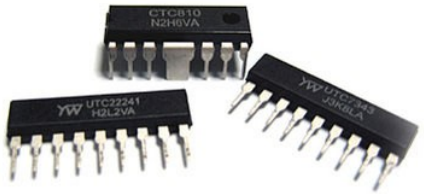
- Optical signals
- Sender and receiver electrically decoupled
- > 100 m length
- Up to 4 Gbit/s per fibre (SFP)
- Light-weight
- Small footprint on PCB

- Now available: Even smaller & cheaper transceivers and cables based on plastic fibres
  - Will be used for readout of MDC
  - Even cheaper than regular copper wires!





FPGA – Field Programmable Gate Array



x 1.000.000

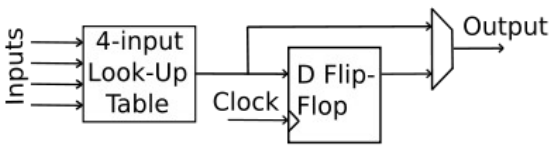
+

+

+ fixed special blocks like memory, multipliers...

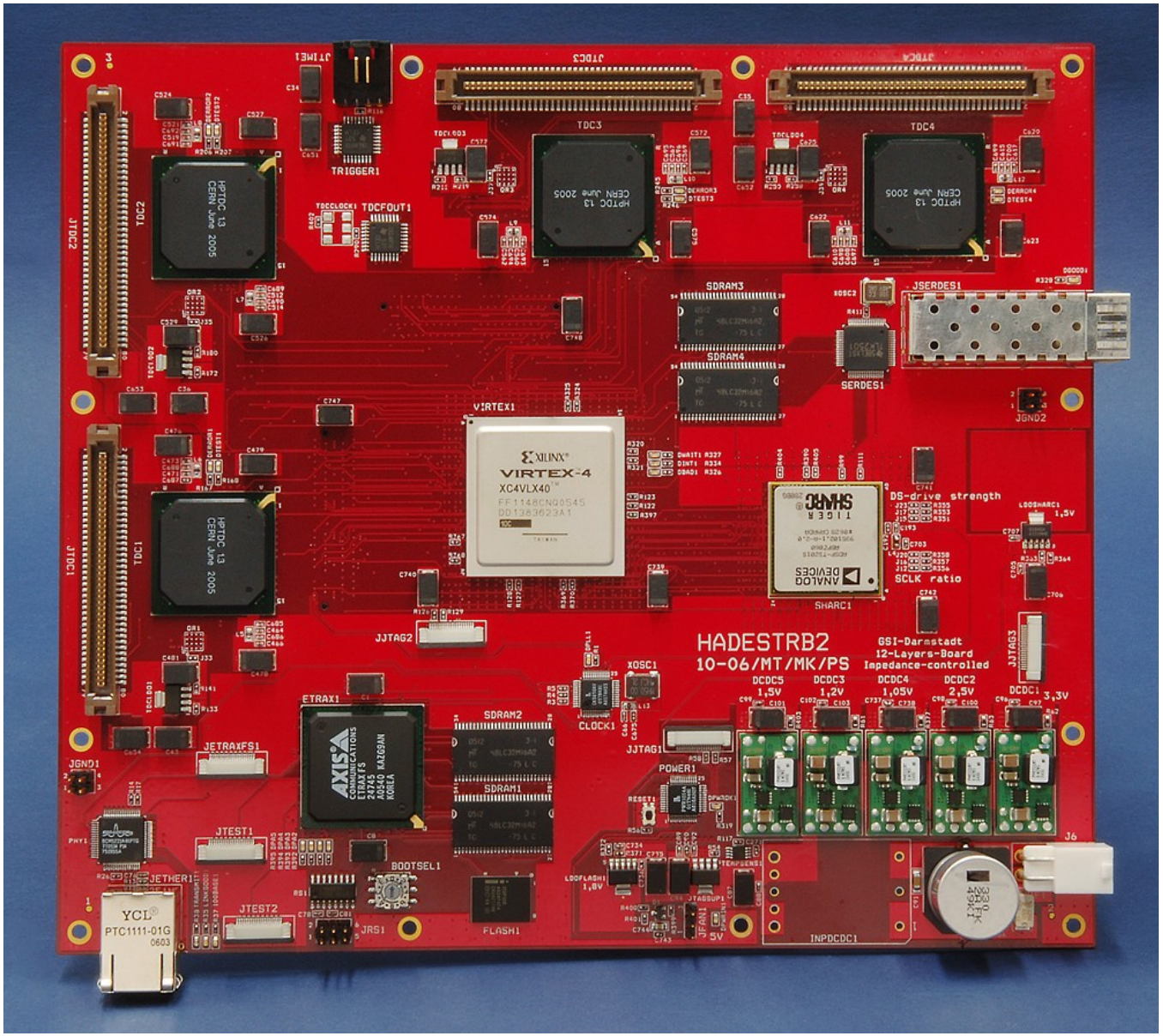
=

- Programmable in VHDL / Verilog
- Typical numbers: 100 – 200 MHz, 500 I/O pins, equiv. 6 million logic gates
- Like in normal electronics, all the logic runs in parallel, not serialized as in a CPU
  - A big number of small tasks can be done at the same time



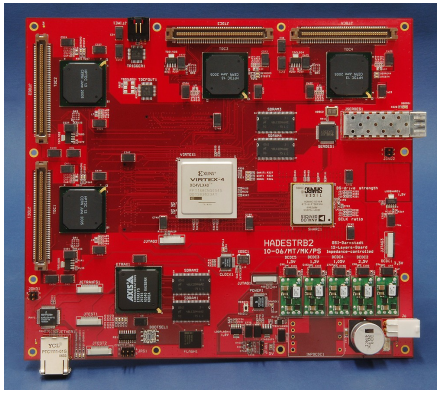
The TRB – A common platform

- Virtex 4 FPGA
- TigerSharc DSP
- Etrax CPU (Linux & Ethernet)
- 2 GBit/s optical link
- 128 TDC channels (time resolution: 40 ps)
- On the back: Connector for AddOn-Boards
- Already used in several beamtimes
- … and by other experiments, e.g. for detector tests

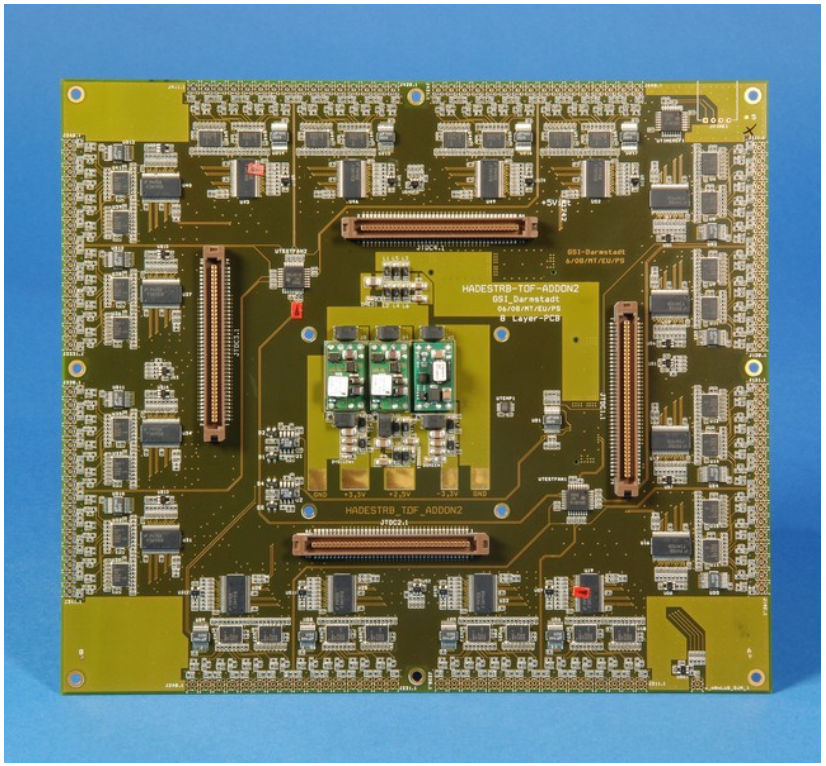


Readout Electronics – Part 1

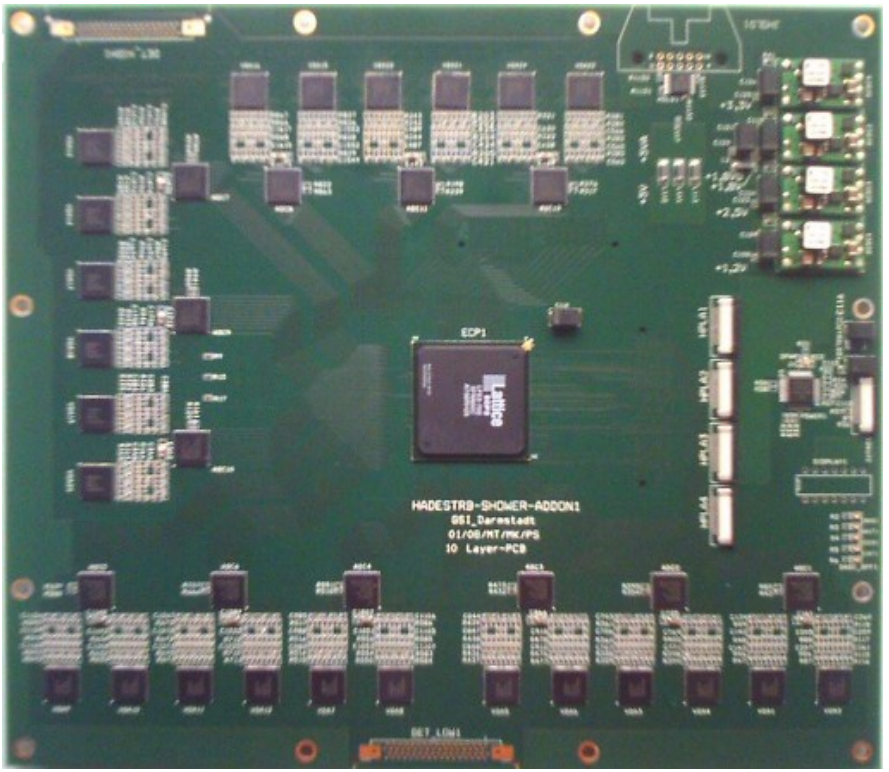
- RPC, Forward Wall, Start & Veto detectors: directly read out by TRB



TOF: TRB with AddOn converting charge in detector to a pulse width

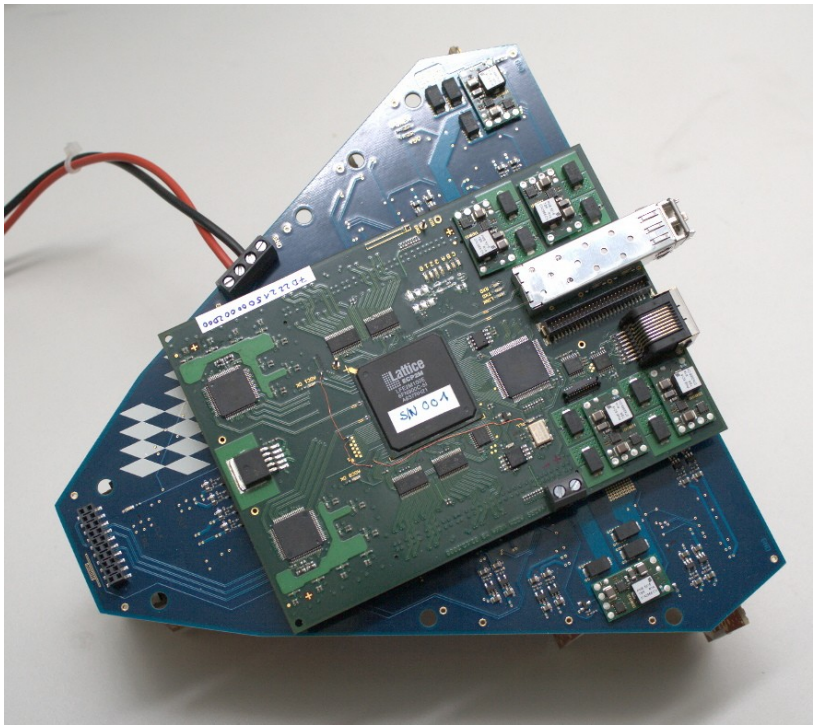


Shower: AddOn with 96 ADC channels

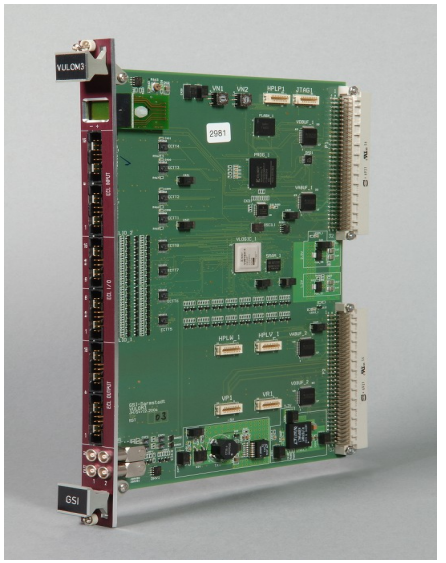


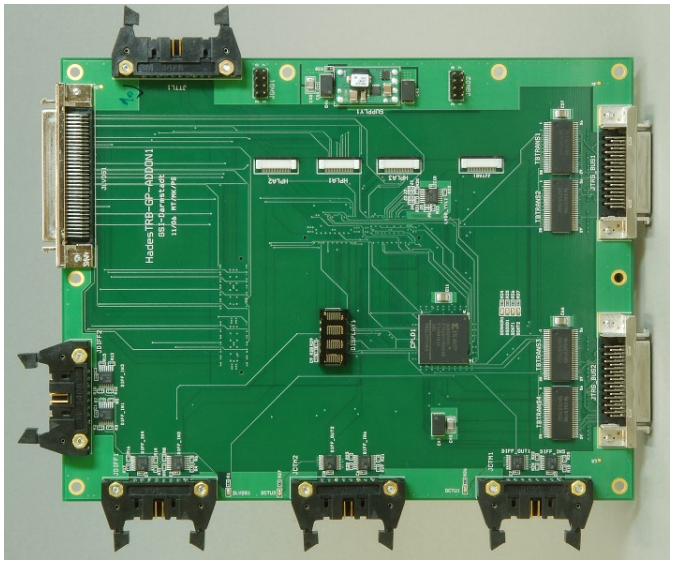
Readout Electronics – RICH & CTS

- RICH ADCM
  - Stand-alone board
  - FPGA, Optical link
  - 2x 8 channel, 12 Bit, 40 MSPS ADC



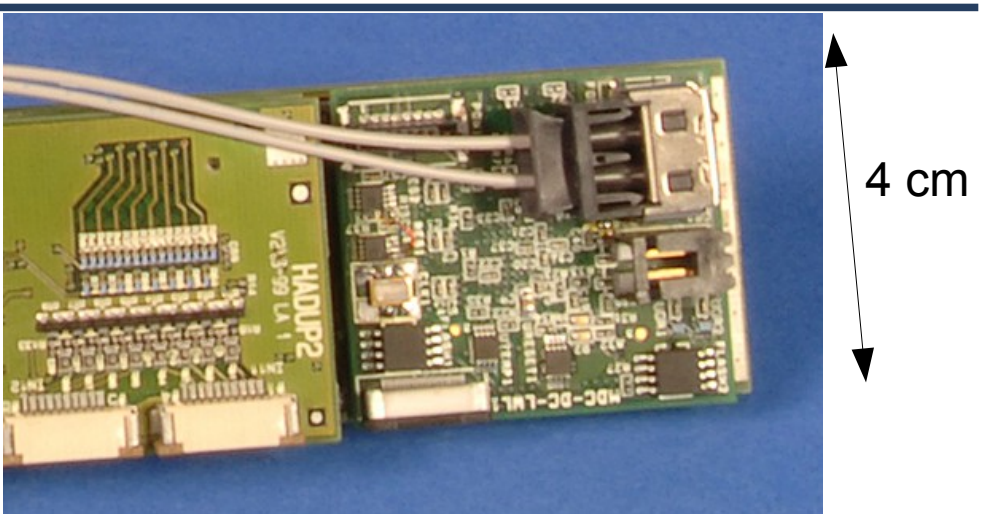
- General-Purpose AddOn
  - Connection to VULOM – the central "brain" of the trigger system



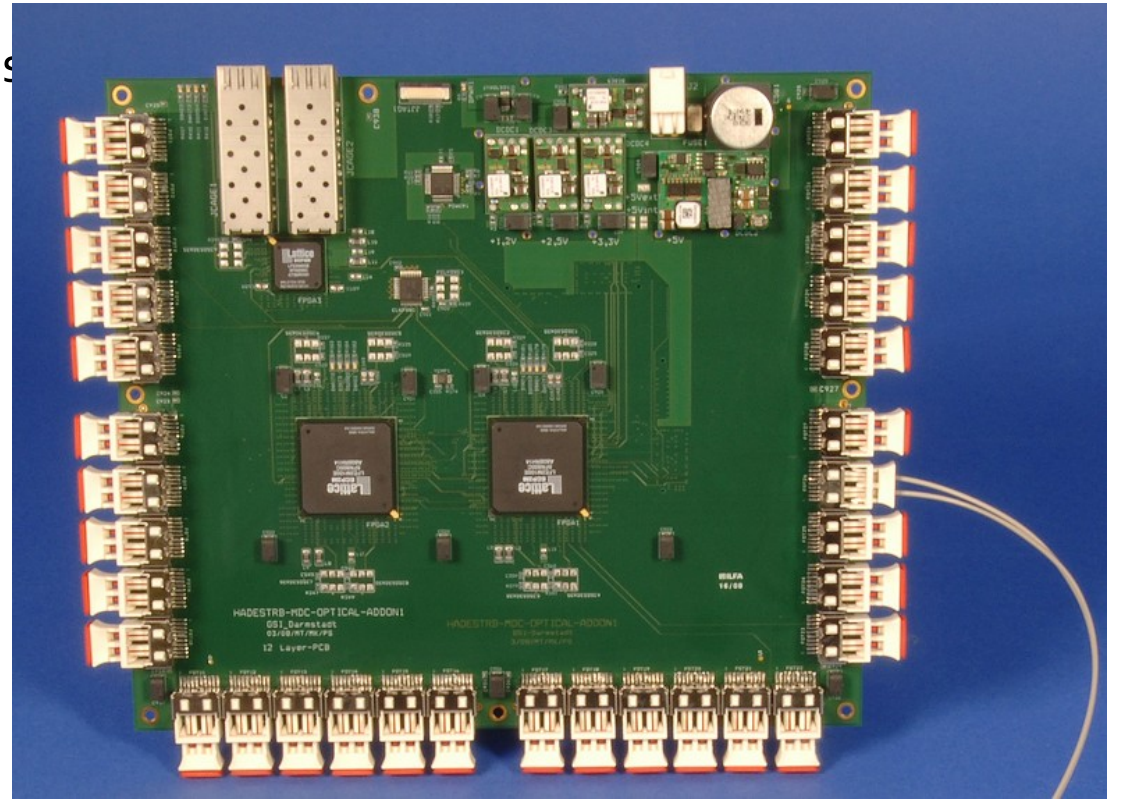


Readout Electronics - MDC

- Driver Card (OEPB)
  - Only 4 x 5 cm size (constraint by given detector setup)
  - FPGA + 250 MBit/s optical link
  - Bootloader feature
  - Reads out front-end digitizer chips
  - 380 boards for MDC system

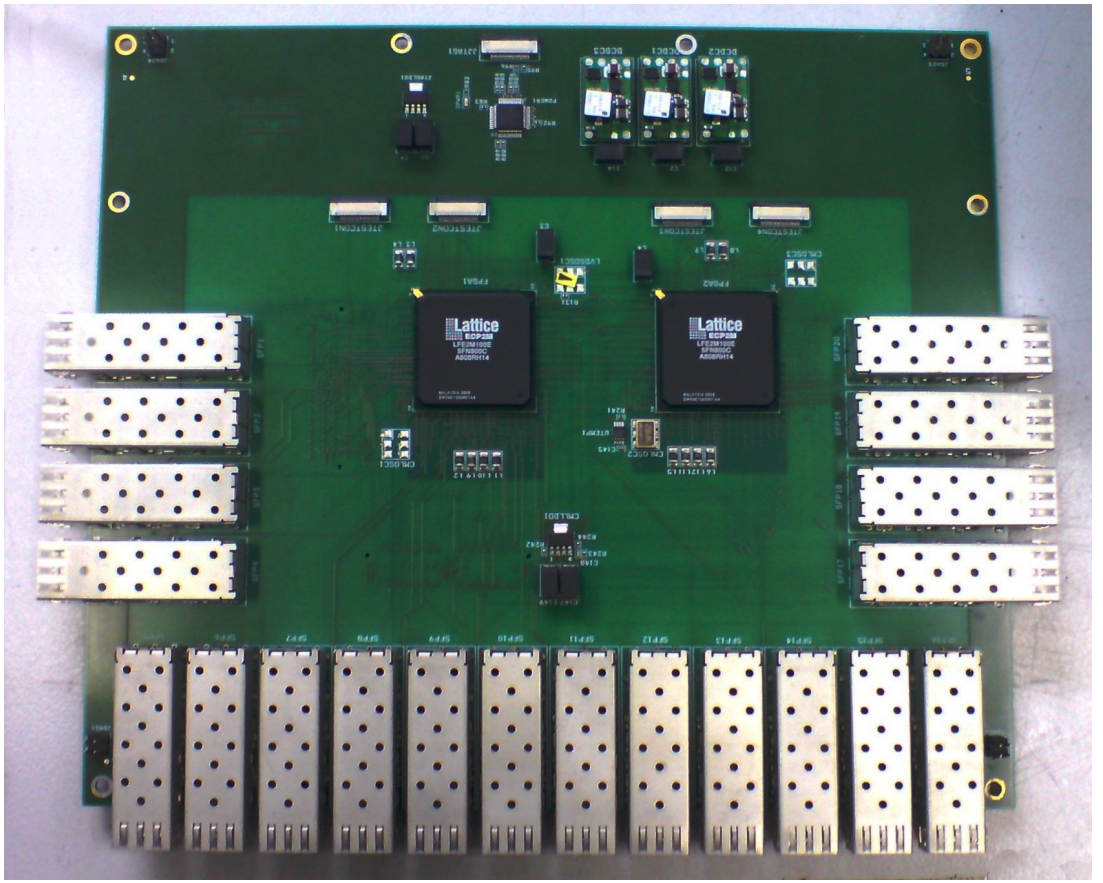


- AddOn
  - 32x 250 MBit/s + 2x 3.125 GBit/s optical links
  - Controlled by 3 FPGAs

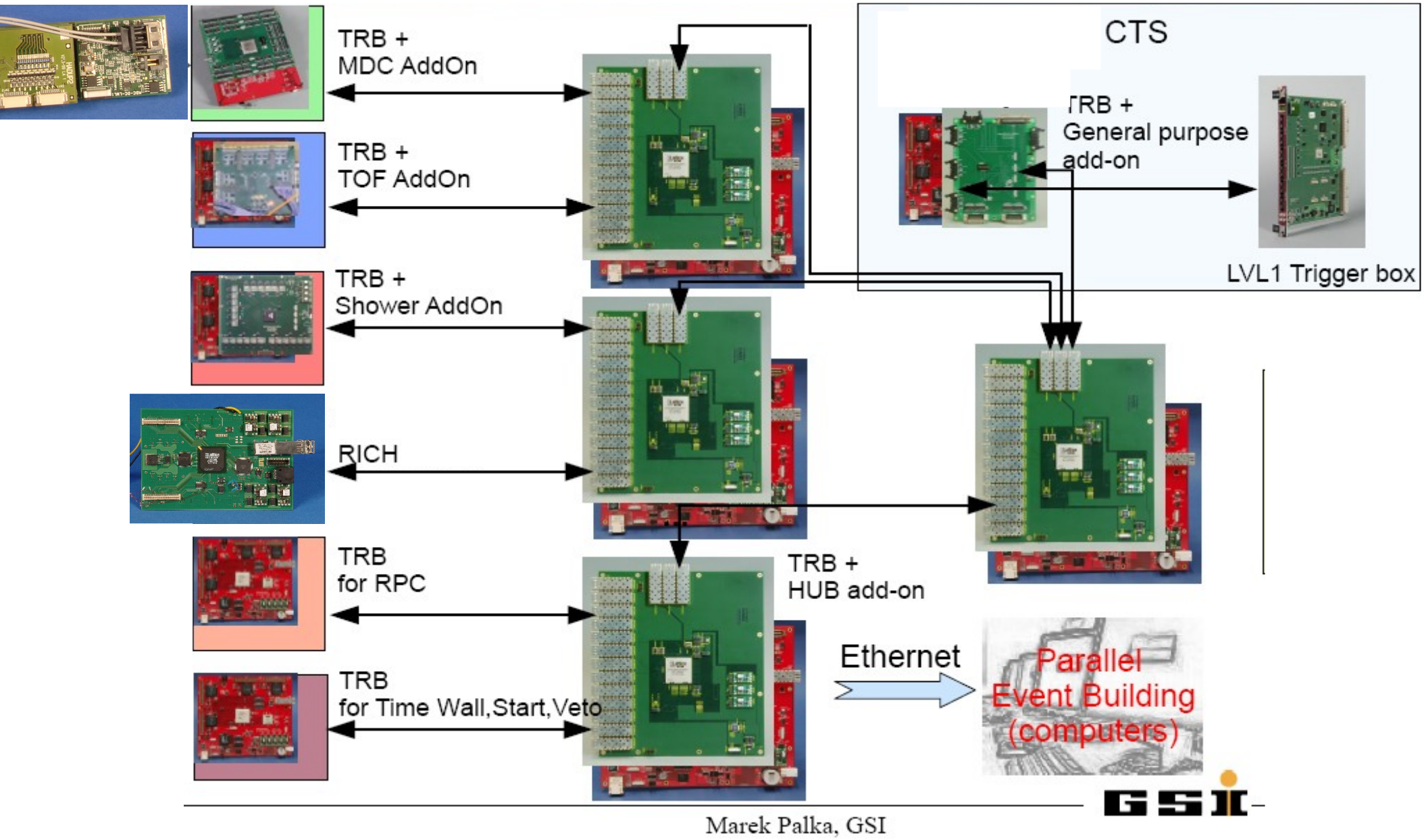


Readout Electronics – Network Hub

- The main "working horse" of the network
- 20x up to 3.125 GBit/s
- Capable of Gigabit-Ethernet to send data to standard PC
- Implements basic data processing features

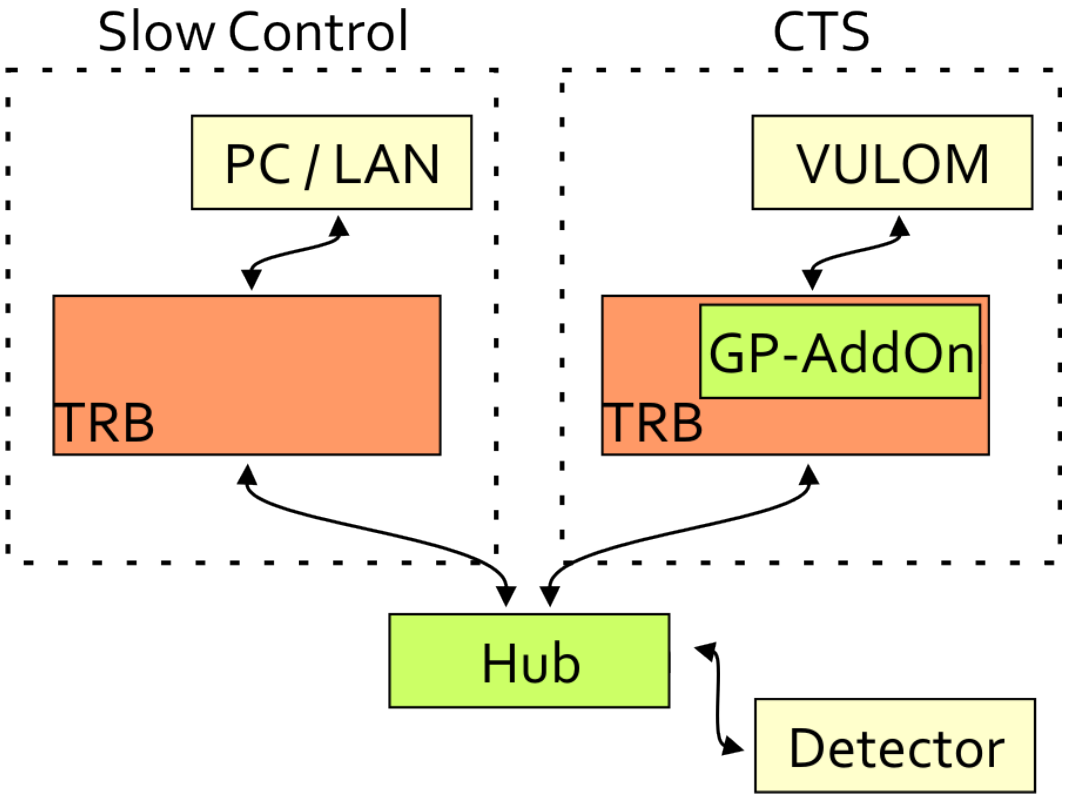


The Complete Network

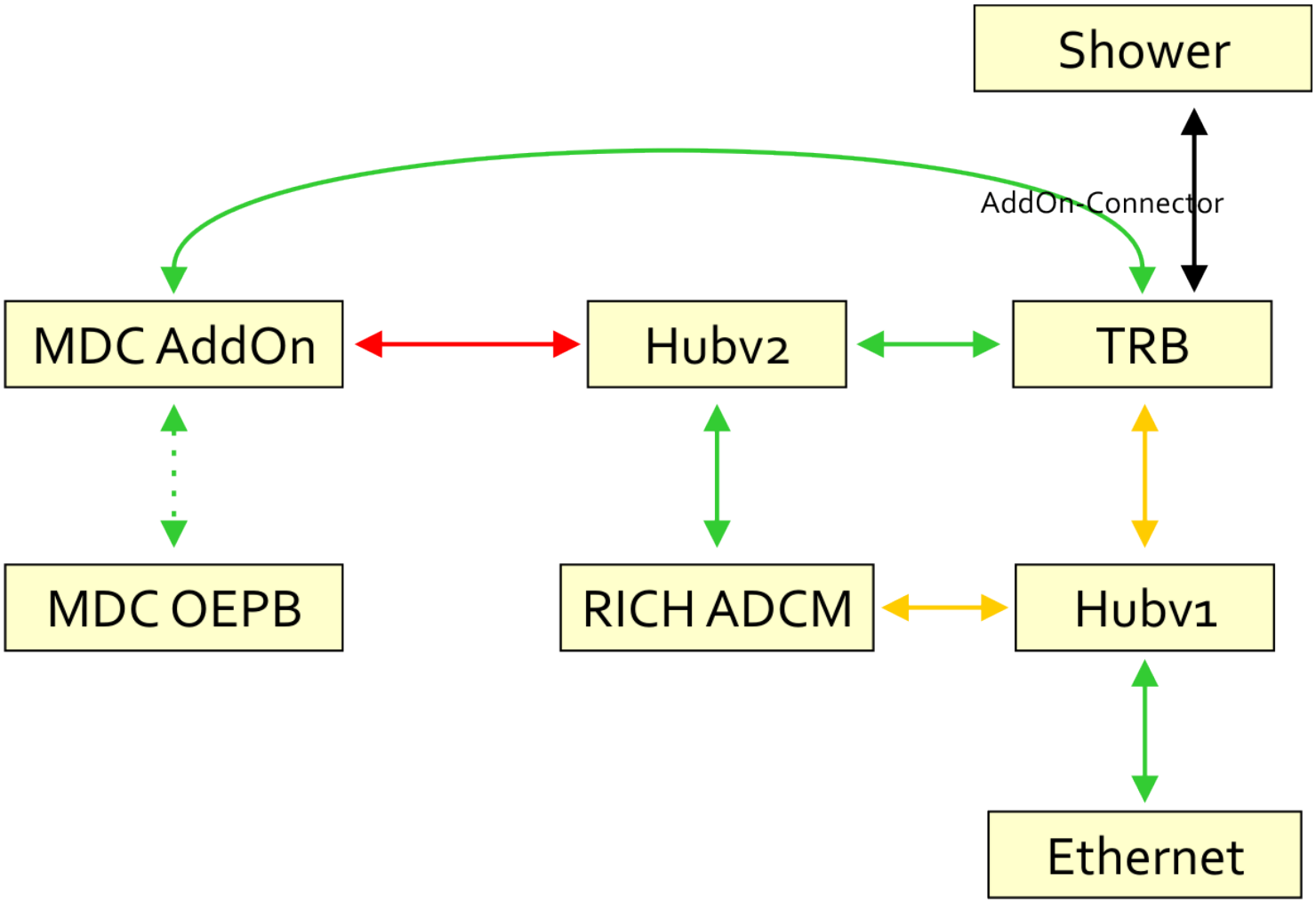


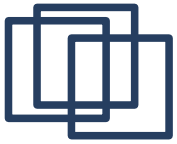
Central Network Parts

- CTS-TRB
  - Connects to Vulom
  - Controls trigger and data channels
- SlowControl-TRB
- Cross connection between both boards



Status of (TrbNet)-Links between Boards





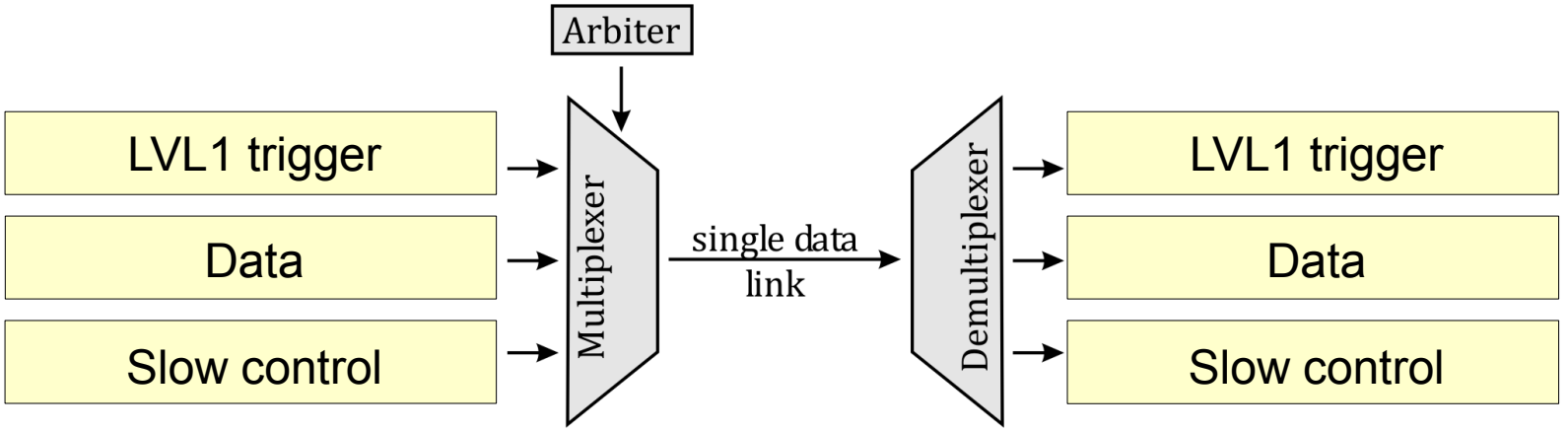
Requirements on Network & -Protocol

- Integration of trigger distribution, data read-out and slow control into one system
- Short latencies (3 µs between two endpoints)
  - Latency of trigger transport directly affects detector deadtime
- Minimized electromagnetic influence on analog electronics
  - Usage of optical fibers instead of copper wires
- Controlled transfers
  - no arbitrary data loss or corruption
- Individually accessible boards with feedback path for integrated monitoring and controlling
- Flexible to adapt to different FPGAs and media types
  - Modular design to adapt to different media (POF, glass fiber, LVDS) and different hardware

Network: Requirements & Solutions Part 1

? Triggers, Data & Slow Control on the same link ?

- Network is divided into several logical channels
  - Internal handling of each channel is completly decoupled
  - Different kinds of transfer are able to run in parallel



Network: Requirements & Solutions Part 2

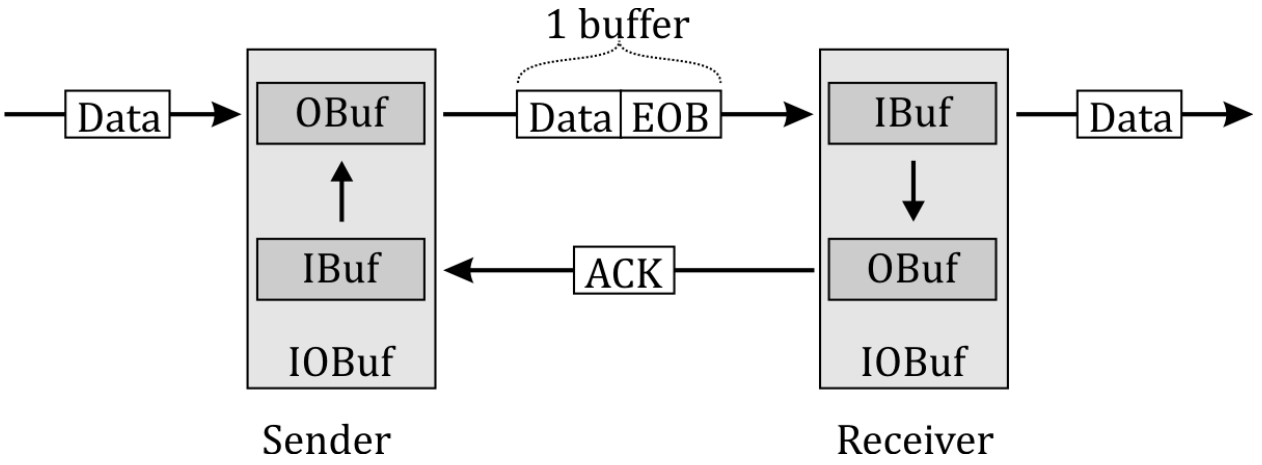
? Low Latency transfer during transport of big data blocks ?

- All transfers are divided into small packets (80 bit)
  - Channel can be switched after each packet
  - Transfers can easily be interrupted for high priority signals
  - Different packet types e.g. for data, flow control, error detection
  - 64 bit payload per packet (4x 16 bit words)

Name	Type	Description	H0	F0	F1	F2	F3
DAT	0x0	Normal data word		Data	Data	Data	Data
HDR	0x1	Transfer start / source changed		Source adress	Target adress		SEQNR / DTYPE
EOB	0x2	End of Buffer		Checksum		Data count	Buffer number
TRM	0x3	Transfer Terminated		Checksum	Error pattern	Error pattern	SEQNR / DTYPE
ACK	0x5	Buffer acknowledge			Length of buffer		Buffer number

? Controlled transfers – no dataloss ?

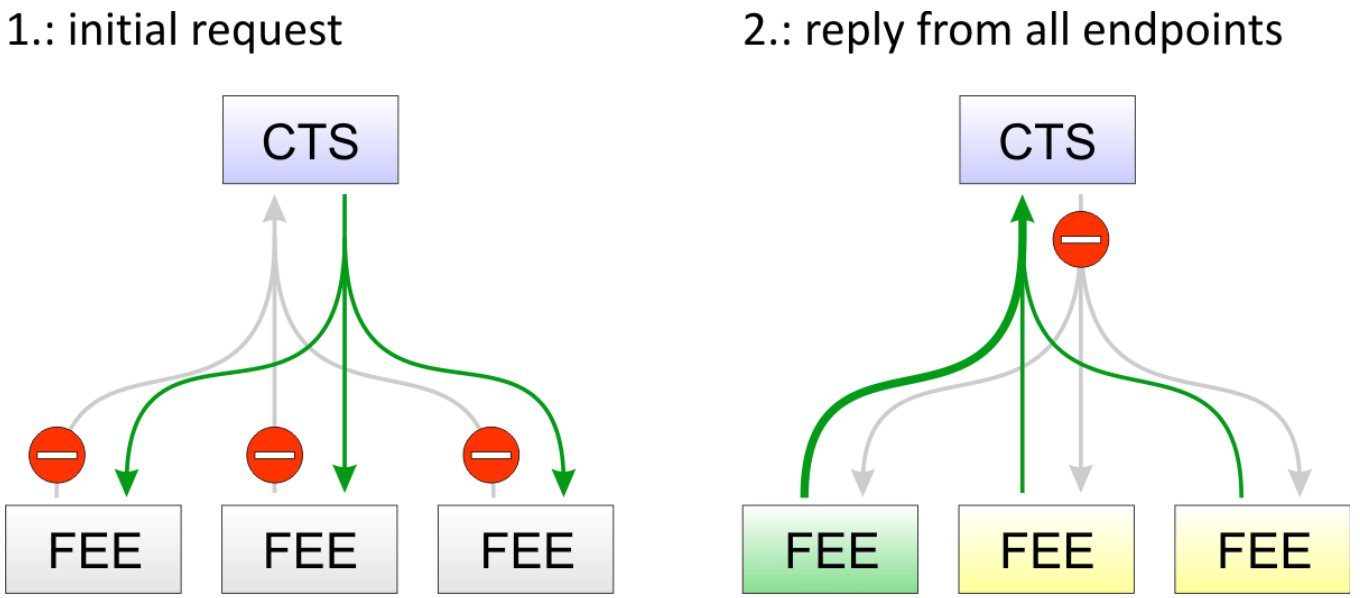
- Data Blocks with defined size
  - Receiver is able to store this in its buffers
  - Sender waits until receipt is acknowledged
  - Block size is negotiated between both sides



Network: Requirements & Solutions Part 5

Check for all nodes being active

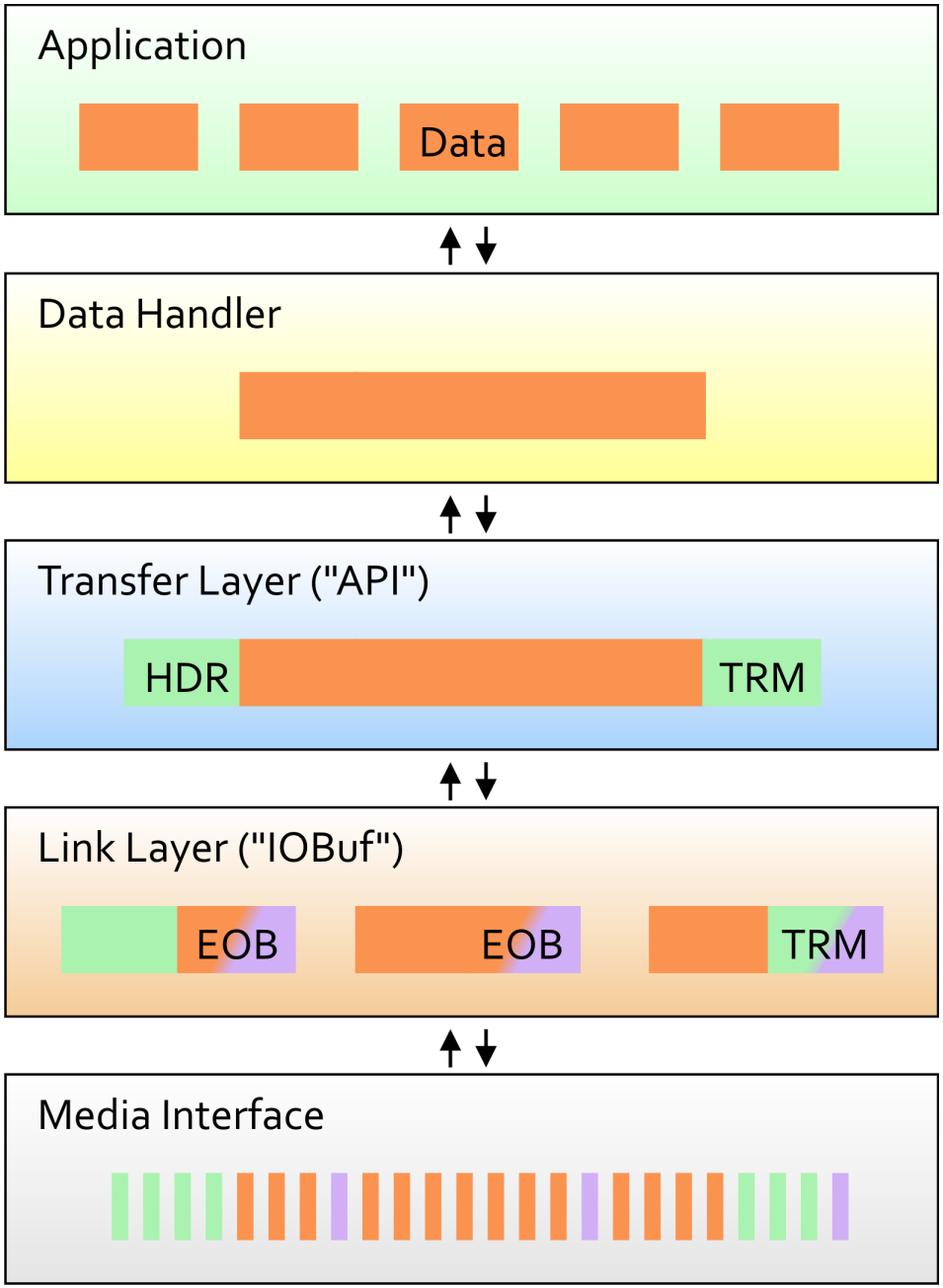
- Each session (one data transfer) consists of two parts:
  - The CTS sends a data transfer to the network (distributed to all endpoints)
  - All endpoints answer the transfer to acknowledge receipt
  - Only addressed endpoints send a longer answer (if necessary)
  - Before all endpoints have answered, no other transfer on this channel is allowed
    - One blocked channel doesn't influence other running transfers



Network Control Layers

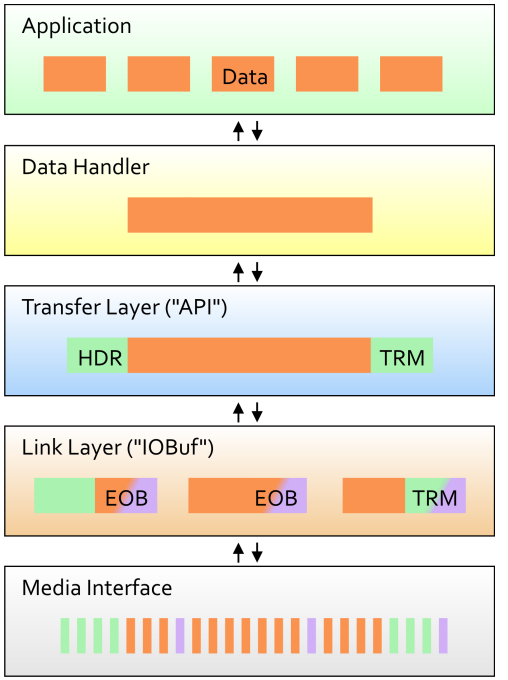
Flexibility – Modular Design

- Data Handler prepares provided data
  - Adds information needed to process data
- API adds header and termination to data
  - HDR: target address & type of data
  - TRM: end of each transfer & gives basic status information
- Link layer divides data into blocks
  - Receiver buffers must be able to store a whole block
  - Sender waits for acknowledge (buffer cleared) before
- Blocks contain CRC error checking
- Media Interfaces control the physical connection between FPGAs



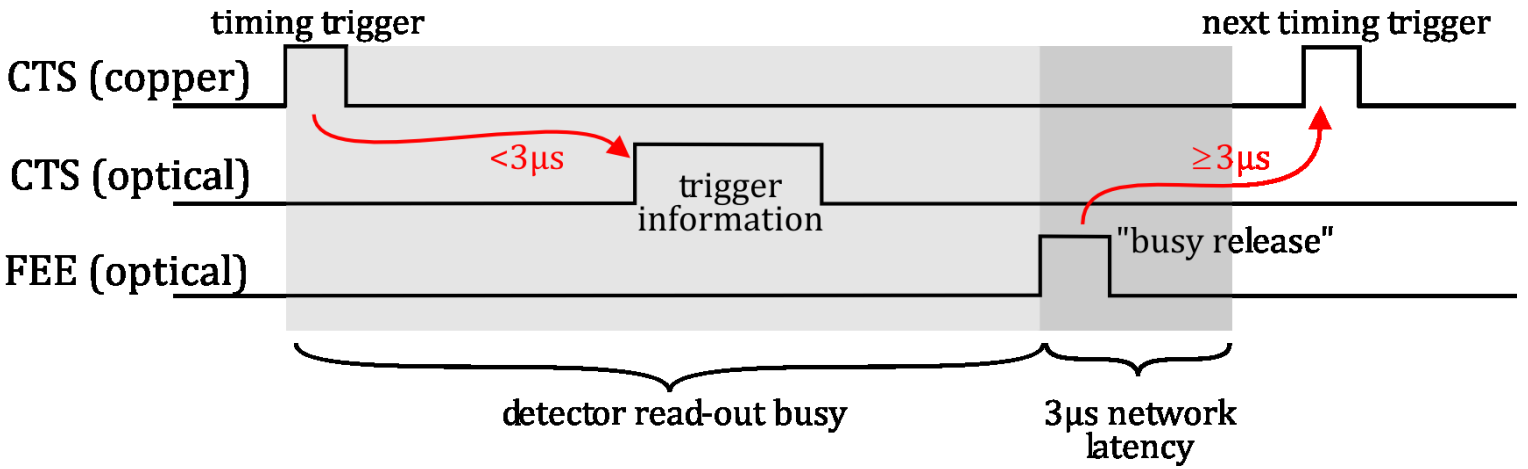
Access to the Network: Data Handlers

- One data handler for each channel
  - LVL1 trigger (“trigger”)
  - Data (“IPU-data”)
  - Slow Control (“RegIO”)
- The whole network endpoint is encapsulated in one entity
  - trb_net16_endpoint_hades_full.vhd
  - Only the media interface and own logic has to be connected



Trigger LVL1

- Each board gets two trigger signals
  - Timing trigger signal using a dedicated copper wire
  - Trigger information packet via the main network:
    - Trigger number, Trigger Type
- Network protocol includes "busy-logic":
  - No more triggers may be sent before all FEE have acknowledged to be ready to take data again
- Data is stored in memory & marked with trigger information

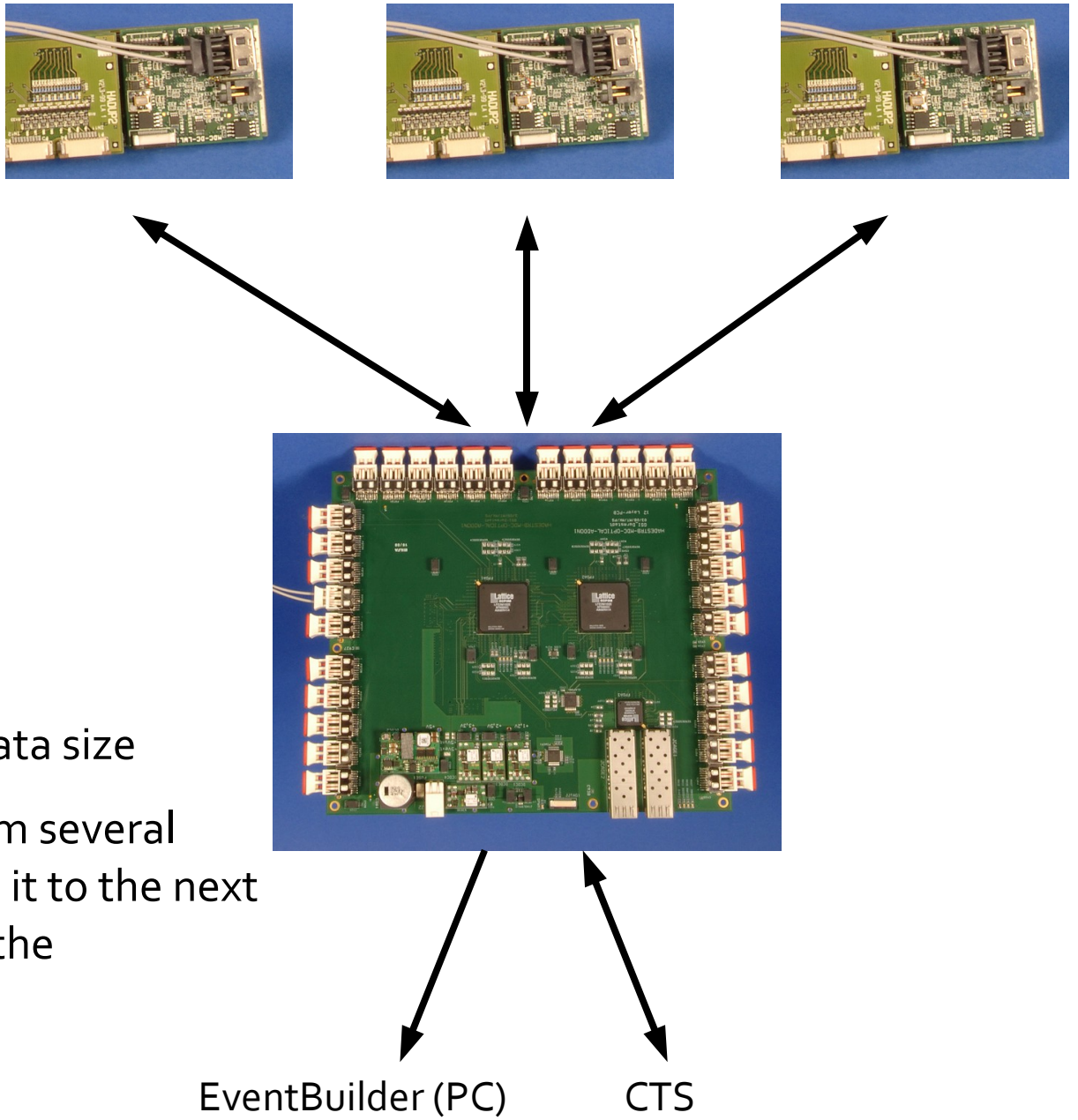


Trigger LVL1 interface

- **LVL1_TRG_RECEIVED_OUT**
  - Goes high when all information is valid and goes low after busy release was set
- **LVL1_TRG_TYPE_OUT(3..0)**
- **LVL1_TRG_NUMBER_OUT(15..0)**
- **LVL1_TRG_CODE_OUT(7..0)**
- **LVL1_TRG_INFORMATION_OUT(7..0)**
  - A random code to help preventing event mixing
  - Additional information byte for future extensions

- **LVL1_ERROR_PATTERN_IN(31..0)**
  - Basic status information
- **LVL1_TRG_RELEASE_IN**
  - "busy release" - strobe, 1 clock cycle

Data Readout



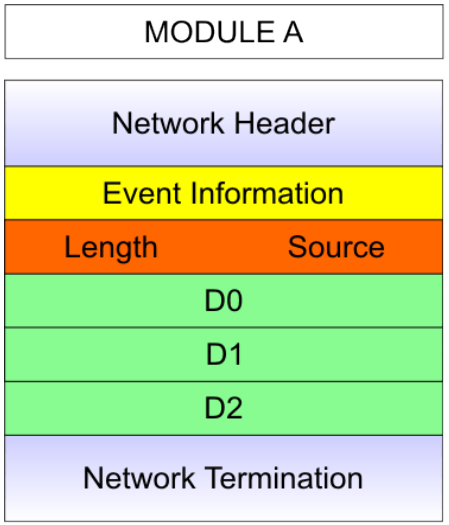
- Upon request the FEE sends event data to the next data concentrator (Hub)
- Data is preceded by an event information header
  - Contains event number & data size
- A concentrator merges data from several FEE to one stream and forwards it to the next concentrator or via Ethernet to the Eventbuilder

Data Readout

- Upon request the FEE sends event data to the next data concentrator (Hub)
- Data is preceded by an event information header
  - Contains event number & data size
- A concentrator merges data from several FEE to one stream and forwards it to the next concentrator or via Ethernet to the Eventbuilder

Data Interface - Ports

- Data Handler provides event number & additional information
- Application provides DHDR1 (Event Information)
- Handler generates DHDR2
- Application sends event data
- Finally application set finished signal

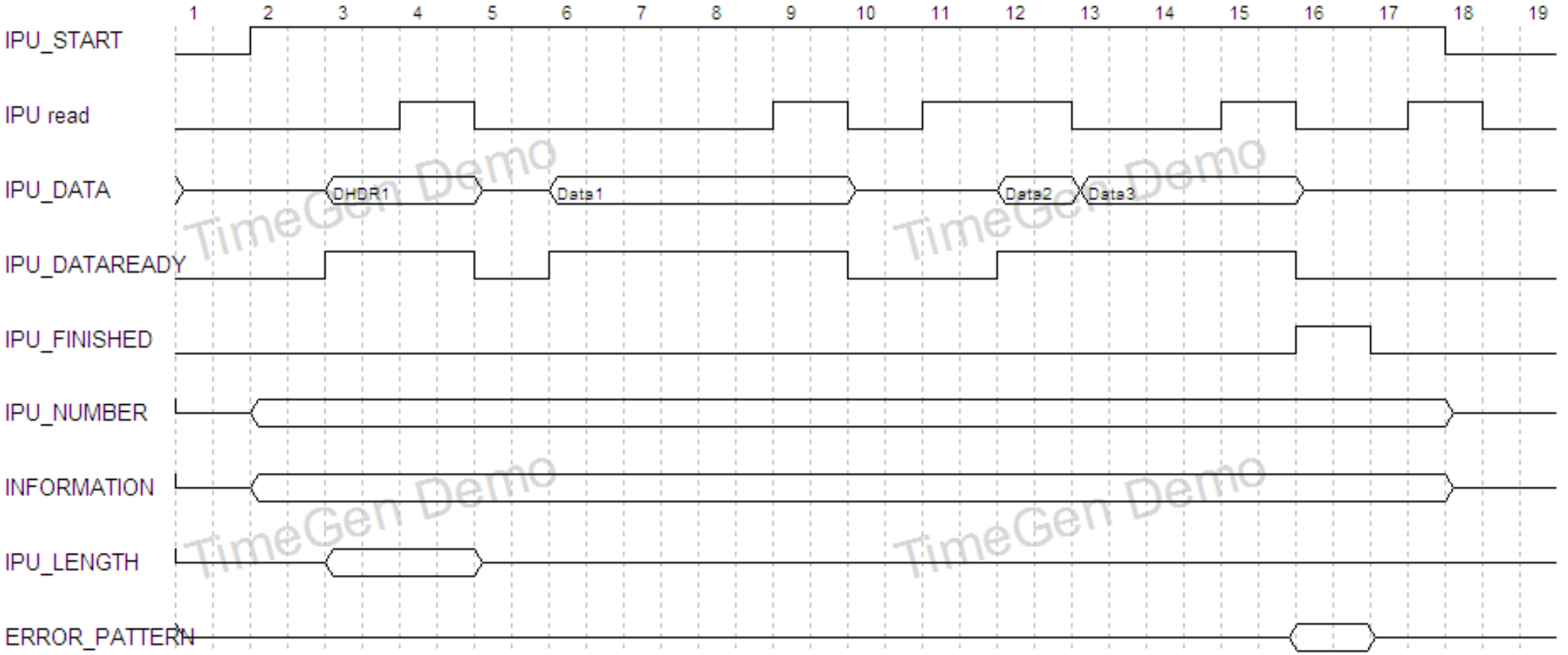



```
IPU_NUMBER_OUT(15...0)
IPU_INFORMATION_OUT(7..0)
IPU_START_READOUT_OUT

IPU_DATA_IN(31..0)
IPU_DATAREADY_IN
IPU_READOUT_FINISHED_IN
IPU_READ_OUT

IPU_LENGTH_IN(15..0)
IPU_ERROR_PATTERN_IN(31..0)
```

Data Interface



- IPU_START goes high after request is received and goes low some clock cycles after IPU_FINISHED was high
- IPU_READ goes high only after IPU_DATAVALID is high

Slow Control – Main Features

- Aim: Provide an standardized interface for monitoring software
  - Easy-to-run during beamtime shifts

- Standardized registers give fast overview of current status of the whole detector system
- Board information memories give information about firmware version etc.

Address Range	Description
0000 – 001F	common status registers
0020 – 003F	common control registers
0040 – 0048	board information ROM
0050 – 005F	board information RAM
0080 – 00FF	user defined registers
0100 – FFFF	internal data port

Slow Control Address Map

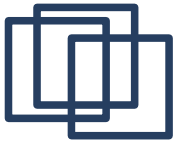
- PC software periodically polls certain registers to monitor the systems status
- Monitoring & Controlling algorithms are directly built into the FPGA design
- Additional features:
  - Advanced monitoring features
  - Methods to automatically monitor the change of a given value over time

Deadtimes, data amounts of each FEE …

Board temperatures & over-heating prevention

Blocking data from erroneous boards …

Possible monitoring & controlling features



RegIO User Interface I

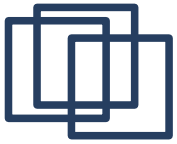
- Accesses to the low addresses (00 – FF) are independently handled by RegIO
  - RegIO provides control registers, User provides status registers
  - Width is configurable via generics

```
REGIO_COMMON_STAT_REG_IN          0x00
REGIO_COMMON_CTRL_REG_OUT         0x20
REGIO_REGISTERS_IN                0x80
REGIO_REGISTERS_OUT               0xA0
```

- Standard data/address port for r/w operations (addresses 0100 - FFFF)
- 5 control signals to identify different kinds of reactions from the user logic

```
REGIO_ADDR_OUT(15..0)
REGIO_READ_ENABLE_OUT
REGIO_WRITE_ENABLE_OUT
REGIO_DATA_OUT(31..0)
REGIO_DATA_IN(31..0)

REGIO_DATAREADY_IN
REGIO_NO_MORE_DATA_IN
REGIO_WRITE_ACK_IN
REGIO_UNKNOWN_ADDR_IN
REGIO_TIMEOUT_OUT
```



RegIO User Interface II

- DAT_DATAREADY_IN
  - DATA_OUT is valid after read
  - High until READ_OUT is high too
- DAT_NO_MORE_DATA_IN
  - Read: no more data is available from this address (FIFO)
  - Write: endpoint is not able to process more data now
- DAT_WRITE_ACK_IN
  - Write access was successful
- DAT_UNKNOWN_ADDR_IN
  - The given address does not exist
- DAT_TIMEOUT_OUT
  - Transfer terminated, since user did not respond in time
- Signals are strobes, one clock cycle long

Slow Control – Main Features 2

? All boards have to be individually addressable ?

- DHCP-like assignment of addresses based on ID-Chips on each board

1. Ids of all boards are read out
2. Central address database is consulted
3. List of ids and assigned addresses is sent
4. Receipt of address is acknowledged

Address assignment procedure

? Some boards are not accessible anymore after mounting ?

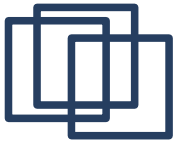
- On-board flash memories are programmable via TrbNet
- One flash contains a fixed boot-loader giving basic network functions ("Micro Kernel")
- Second flash contains updatable firmware

Software: trbcmd

- Shell-based software running on Etrax
- Connecting to special trbnet endpoint on TRB
- Allows for all request types on TrbNet
- Split into TrbNet-library, FPGA-connection library & high-level software
  - Easy to implement in own code

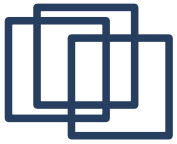
```
Commands:
   r <trbaddress> <register>                  -> read register
   w <trbaddress> <register> <data>           -> write register
   rm <trbaddress> <register> <size> <mode>   -> read register-memory
   wm <trbaddress> <register> <mode> <file>   -> write to register-memory from ASCII-file
   i <trbaddress>                             -> read unique ID
   s <uid> <endpoint> <trbaddress>            -> set trb-address
   T <input> <type> <random> <info> <number> -> trigger by slowcontrol
   I <type> <random> <info> <number>          -> read IPU data slowcontrol
   f <channel>                                -> flush FIFO of channel
   R <register>                               -> read register of the FPGA
   W <register> <value>                       -> write to register of the FPGA
```

```
> trbcmd i ffff
0xee000001e43c17c1  0x01
0x8e000001fc533228  0x01
```



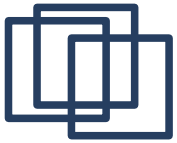
Resource Consumption

- Full Endpoint needs 2000 slices
  - 20% of available resources in smallest FPGA (OEPB)
  - 5% on biggest FPGA (RICH ADCM)
- Hub needs much more resources
  - Hubv1 can handle ~ 6 links with its 12k slices
  - Hubv2 is occupied by 50% when handling 17 links (16x optical + up-link)
  - Scales almost linearly with number of links (~ N * 1500 + 1000 slices)
- Big FPGAs are much more complicated to handle than small ones
  - 20% occupancy in ECP2M20 : No problem at all (up to 150 MHz)
  - 20% of ECP2M100: Software shows very different results (70 – 115 MHz) with identical input VHDL code
  - (Xilinx Virtex4 shows similar maximum frequencies for both)



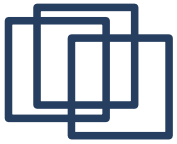
TrbNet: Nomenclature

- **Packet**: 80 Bit of data, smallest amount of data on any connection, organized in 5 16bit **Words**. The first word is **Packet Start (H0)** containing packet type and channel id. The next 4 words are the **Payload (F0 – F3)**, the „useful" data transported. Internally the **Packet Number** (H0, F0-F3) is mandatory to be transported along with each word.
- The **Medium** is divided into three c**hannels** with different **priorities**. The **Media Interfaces** provide access to the physical cable. Data then goes through the **Multiplexer** to the **IOBuf** that provides the security and divides data streams into **Blocks**. The **API** provides the interface to the network. The user logic connects to data handlers (**RegIO** on the Slow Control channel, **Trigger** on the LVL1 channel, **Ipudata** on the data channel) whoch provide additional features.
- The **packet types** the user sees are **HDR, DAT** and **TRM.** The IOBuf additionally uses **EOB (End of Block)** and **ACK**.
- Trigger Data contains the **trigger type, trigger number, trigger information** and a **random trigger code**.
- The 16bit **addresses** of each board are assigned based on the **unique ids** of its temperature sensors.
- A network **session** consists of two an **init transfer** and a **reply transfer**.



Error Detection

- Board is not reacting
  - Automatically detected, since all boards have to answer each transfer
  - Port / Endpoint can be switched off via slow control
- Bit errors in data words
  - I/O-Buffers have additional error detection:
  - 16bit CRC are calculated & checked for each buffer
  - Number of data words is checked
  - Apparently corrupted data is marked by setting a bit in the status pattern
- Word loss due to bit error in 8b/10b encoded stream
  - Damaged packets are detected and deleted
  - Automatic realignment to packet boundary
  - Optional: Error correction based on redundant encoding

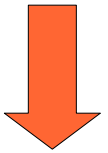


Conclusion

- Current Status of Development
  - Online monitoring features both in hard- and software are being developed
  - Tests with productive usage of all channels are ongoing
  - Adaption to different hardware requirements is in progress
  - Installation of new detector and electronics will be finished by the end of this year
- TrbNet is a versatile network protocol
  - Optimized for HADES, but the modular design makes adaption to different requirements easy
- The HADES DAQ upgrade
  - Trigger- and data-rates achieved with the new network will greatly improve the performance of the HADES detector
  - Slow Control provides all tools for efficient monitoring during experimental runs

Data Transport - Summary

- 500 boards
- 10 board types
- Very different sized FPGAs from two vendors
- Unlike media types
  - glass fibres, plastic optical fibres, AddOn-Connector, on-board communication
- Diverse requirements for transport for triggers, data, monitoring information



Urgent need for one common, flexible, adaptable network protocol:

Trigger- and Readout Board Network - TrbNet