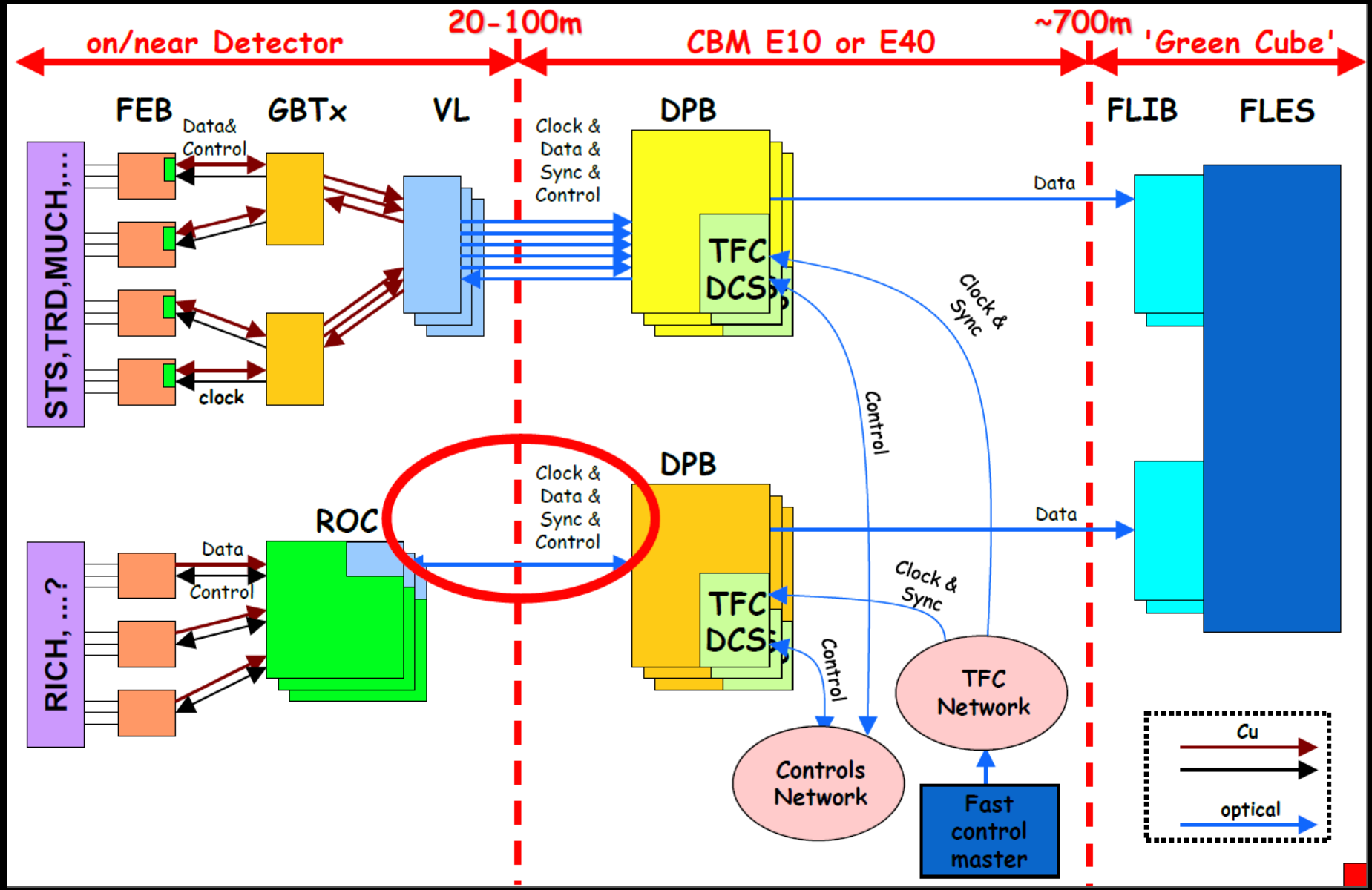
Jan Michel - Goethe Universität Frankfurt

A FPGA network for Panda & CBM

Overview

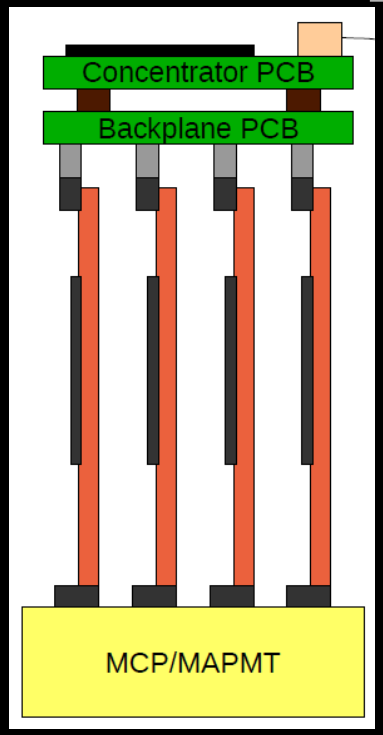
- DAQ Network Overview
  - Needed Features
  - Available Framework
- Roadmap of additional features

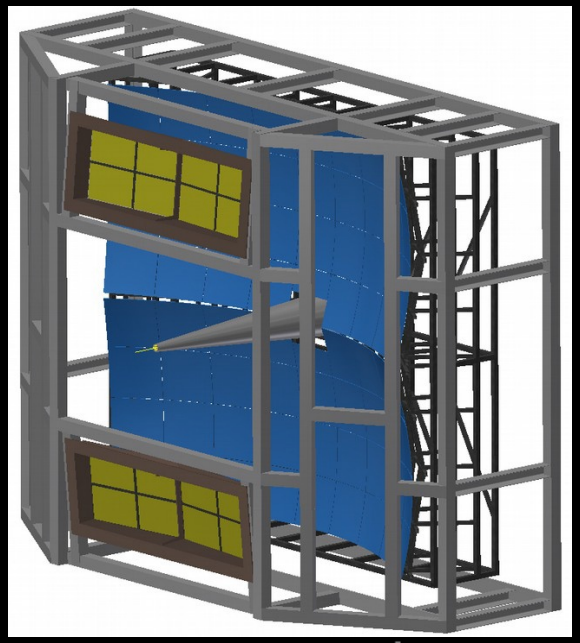
CBM DAQ Network



Walter Müller

Example: CBM RICH





800x 1600x 100x

100x
2.4 GBit/s

Control
Monitoring

20x
10 GBit/s

MC-PMT FPGA Back plane FPGA FPGA Server

64 channels

FPGA as precise TDC (20ps)

FPGA as data pre-processor

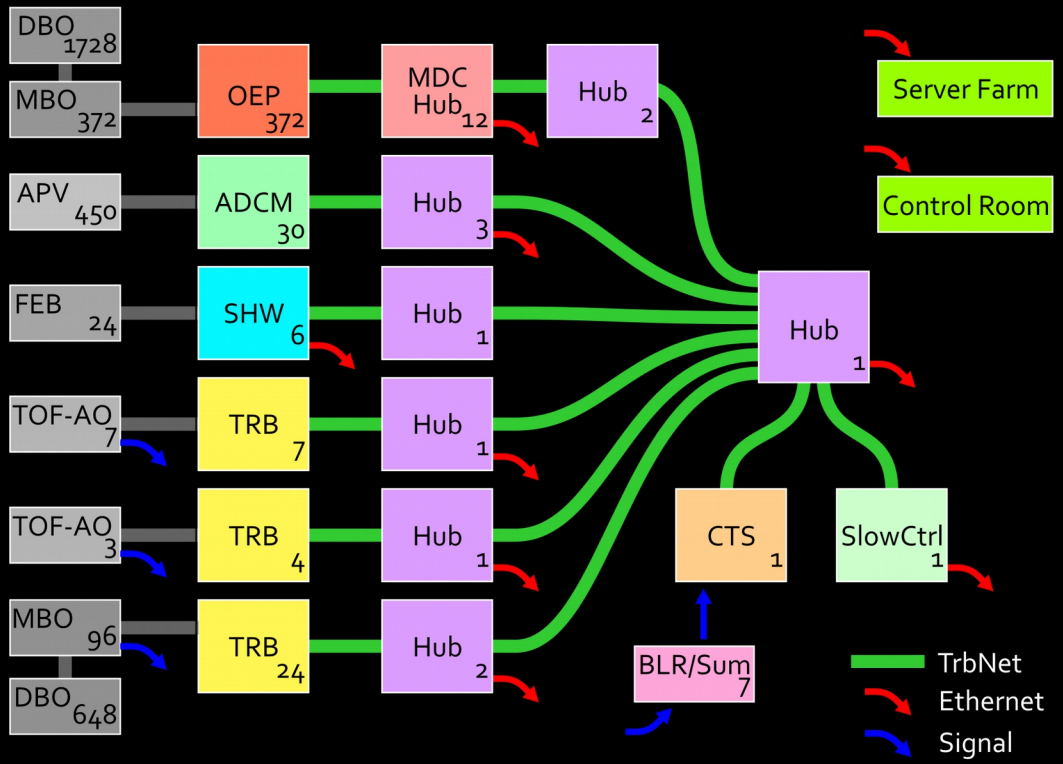
FPGA as data concentrator

Synergies: PANDA, CBM and HADES

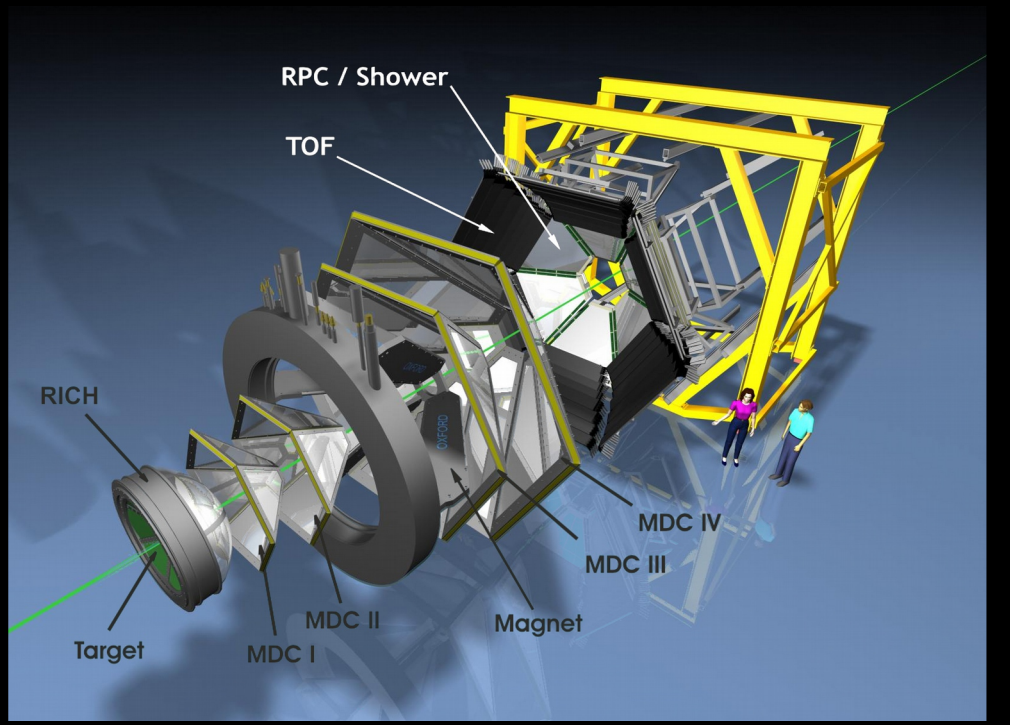
- All experiments use
  - the same FPGA-based TDCs
  - a common FEE for DIRC and RICH detectors
  - a common hardware platform
  - a common inter-FPGA network
- Detector Tests last year used the same hardware for PMT read-out

Picture: Jochen Schwiening

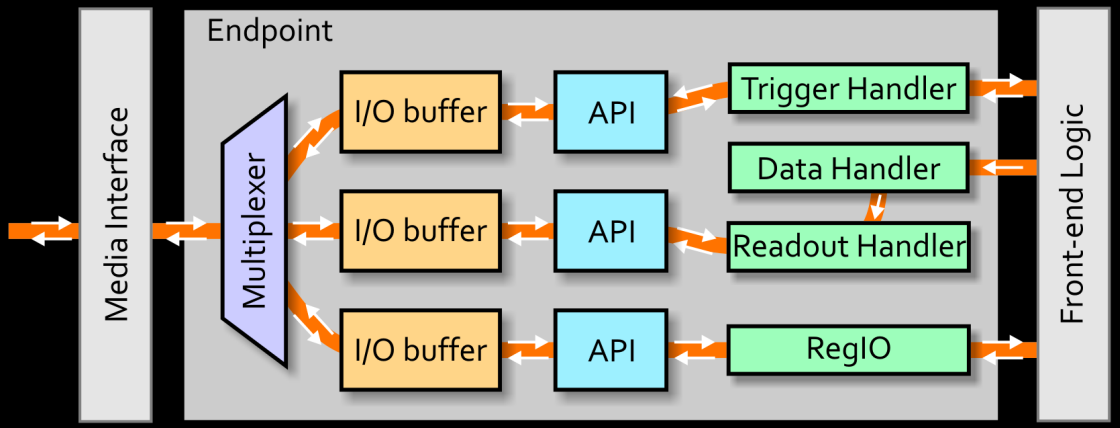
Protocol: TrbNet (by HADES)



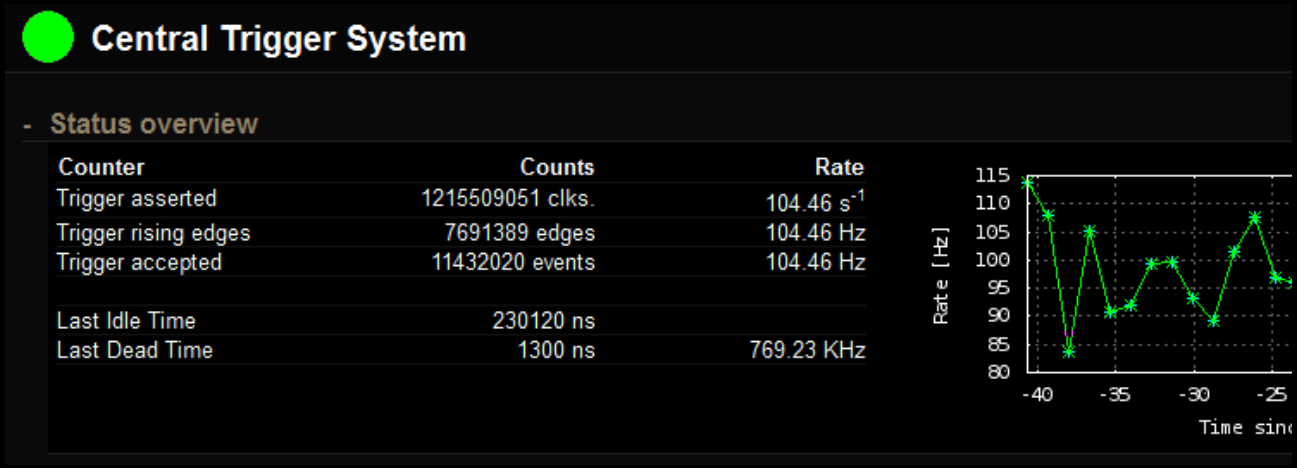
- Spectrometer
  - 7 detector systems
  - 50k ADC channels
  - 30k TDC channels
- DAQ System
  - 50 kHz trigger rate
  - 500 MByte/s
  - 500 Front-ends



A glance at TrbNet features

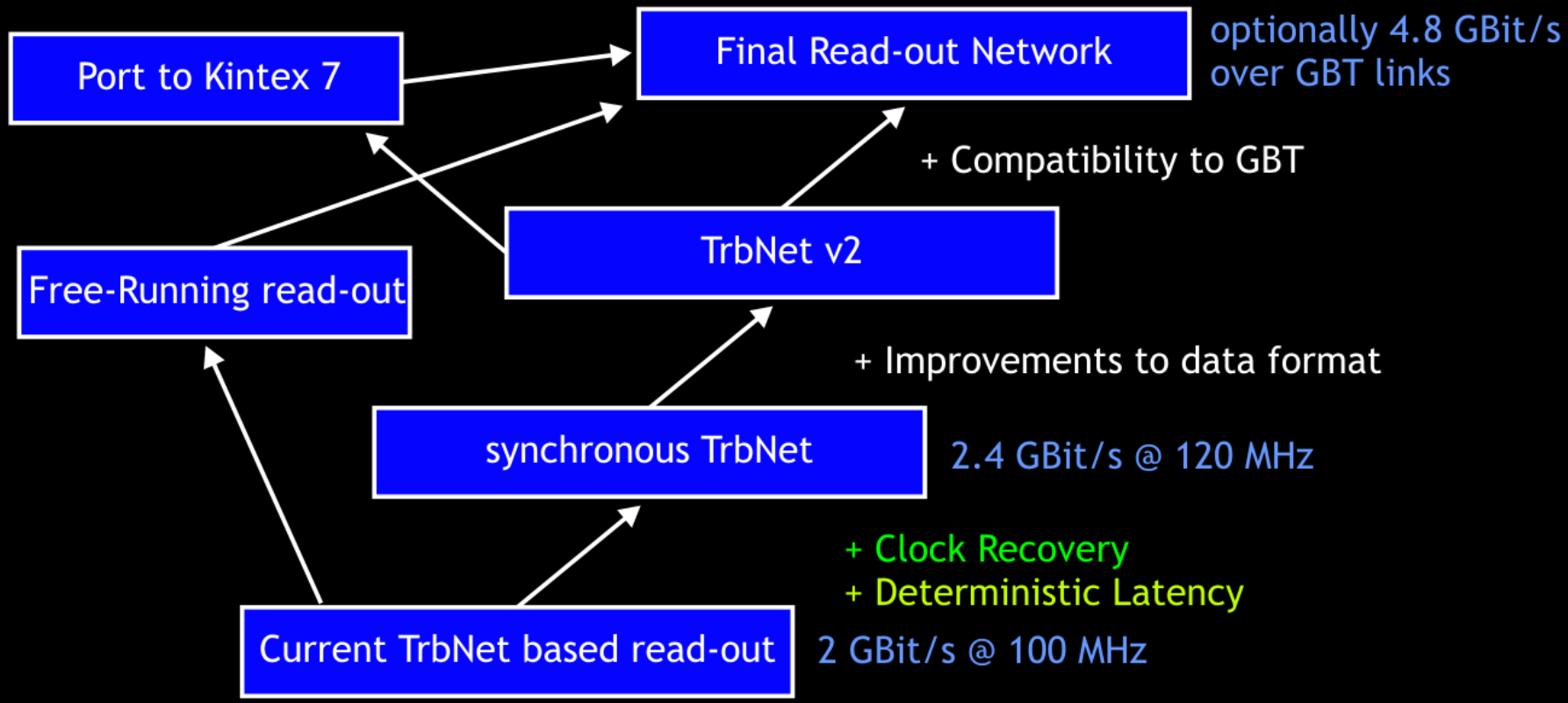


- Developed since 7 years
- Used in various setups outside of HADES
  - detector development for FAIR
- Huge software framework & hardware modules available



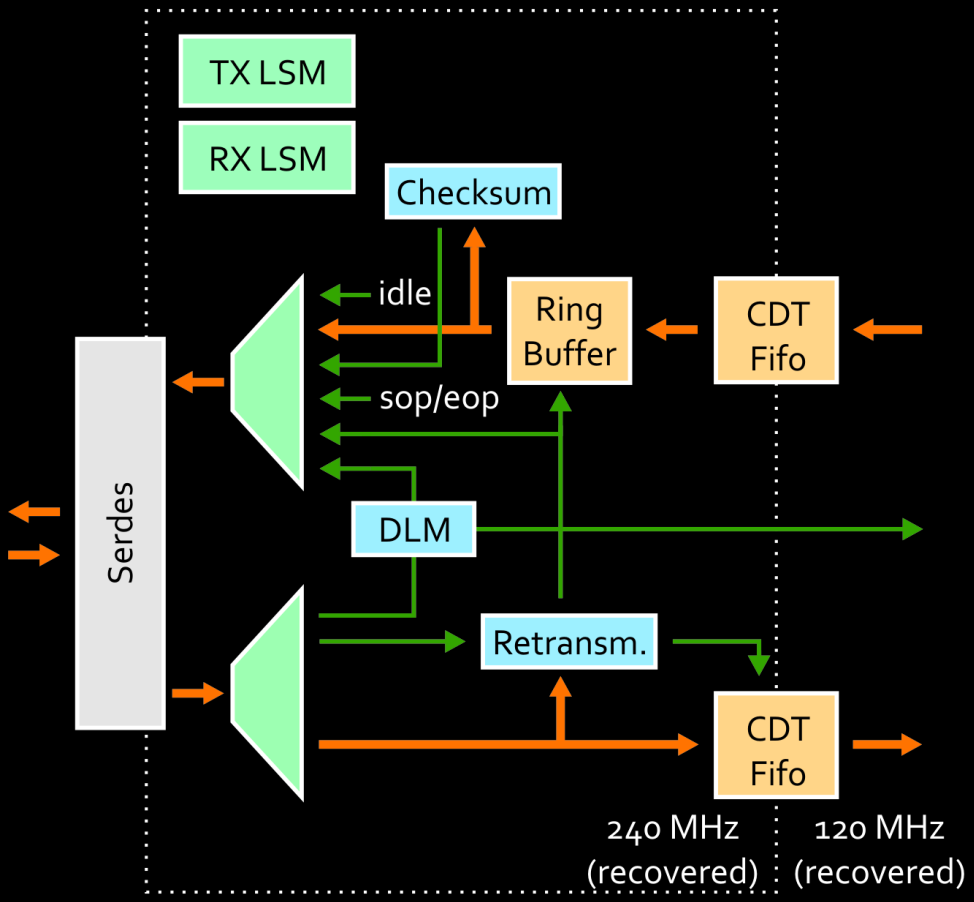
0240	0241	0242	0243	0250	0251	0252	0253
40.4	45.3	41.4	44.6	37.4	36.0	35.1	35.4
42.2	45.1	43.6	45.3	40.9	39.6	36.5	37.5
42.5	42.7	41.9	---	42.9	41.7	37.9	39.9
43.6	43.8	43.8	---	43.6	42.6	39.6	40.6

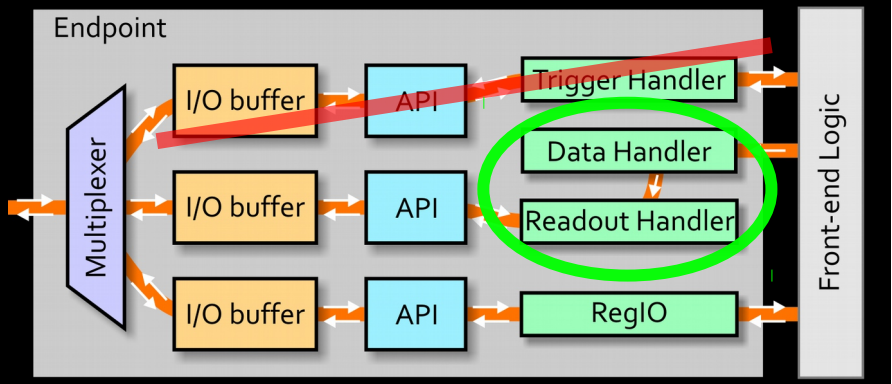
Initial Road Map



Central question: What is the best way to reach final goals?

Endpoint Architecture Architecture





+ "Synchronization Manager"

Deterministic Messages in Panda

- Define two types of messages, each with 32 Bit payload
  - "Start of burst" to mark the beginning of each super-burst (16 bursts)
    - sent in fixed intervals
  - "Control" to select different operation modes, trigger calibration…
    - checksum to prove correctness of packet
    - can be sent at any time

"Start of burst": 31 Bit super-burst number

"Control": 24 Bit for control tasks & checksum

- Selection of format as simple starting solution, needs refinement
  - can we come up with a common message container?
- Link length measurement by returning messages
  - resolution: 4 ns (byte clock) is trivial, 400 ps (bit clock) could be possible

Free-Streaming Read-out

- All front-ends send data on their own
  - no central controller
- Different schemes are possible:
  - Version 1: Continuous data stream from FEE, concentrator merges by time on a best-effort basis
  - Version 2: FEE buffers data for N byte or M µs, then sends a packet
  - Version 2a: Concentrator just forwards packets without touching them
  - Version 2b: Packets are sent synchronously by all FEE, concentrator merges all packets from same time range

- Before implementation can start, discussion about favored mode is needed

TrbNet v2

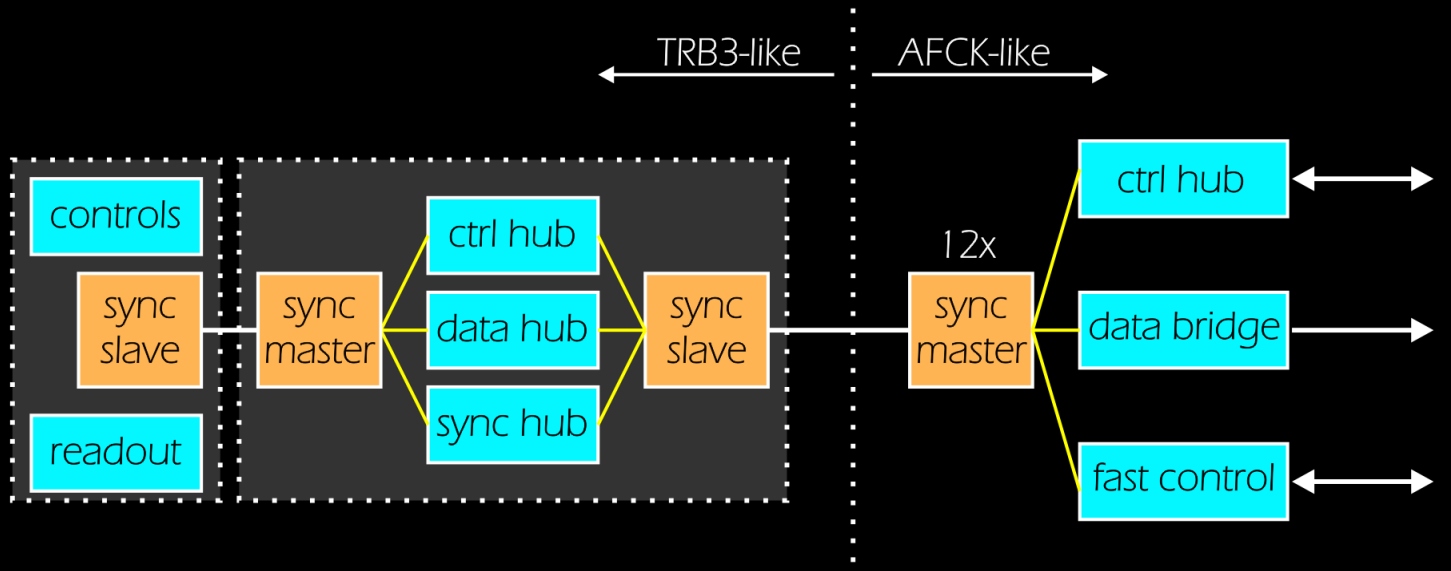
- Original version:
  - small packets to transport low latency trigger messages
  - few comma characters because of early hardware limitations

Name	ID	F0	F1	F2	F3
DAT	0x0	64 Bit data payload			
HDR	0x1	source	target	length	seq.no. & type
EOB	0x2	CRC[1]	reserved	word count	buffer number
TRM (request)	0x3	CRC[1]	payload		seq.no. & type
TRM (reply)	0x3	CRC[1]	error & status information		seq.no. & type
ACK	0x5	resv.	buffer size	reserved	buffer number

- New Version
  - variable packet size (?)
  - better inclusion of "packet start" comma and checksums
  - increase bandwidth for user data
    - 112 Bit word size instead of 80 Bit
  - optimizations in VHDL code
    - run on 32 Bit data path? different handshaking?

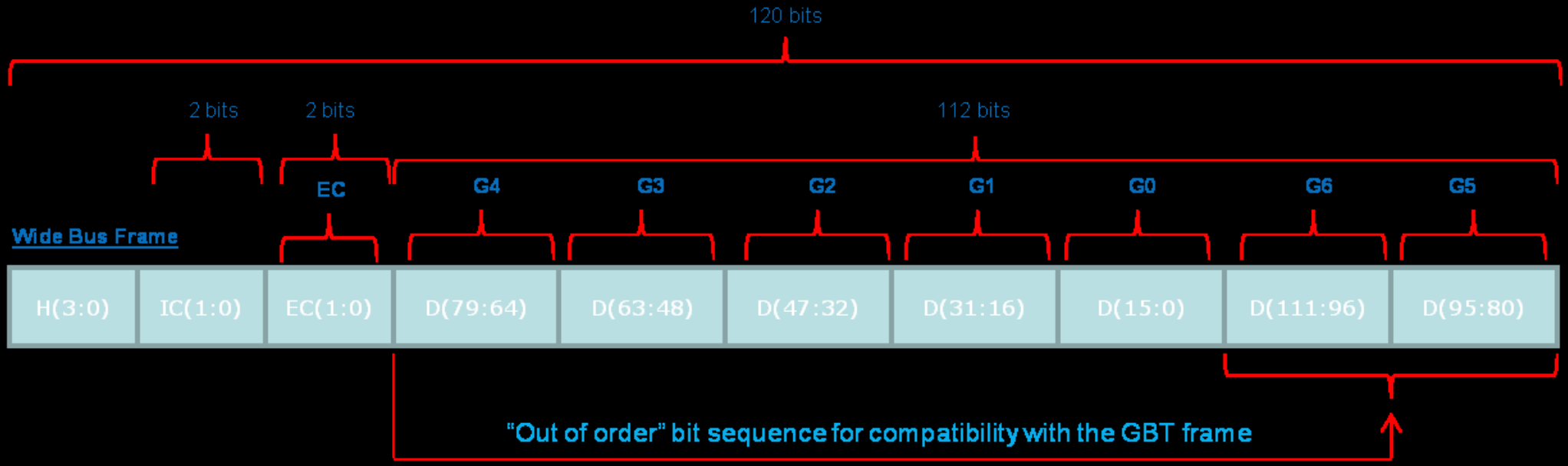
Port to Kintex

- Porting the endpoint is mostly trivial – Virtex 4 was already used
- Media Interface needs some experienced developers
- Adaption of control interfaces and data bridges
  - GbE as slow-control link proved very useful
  - data bridge: common system solution



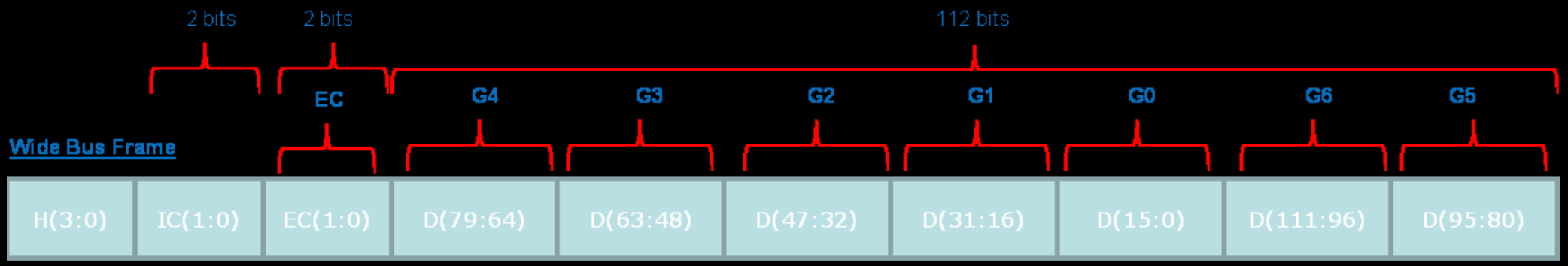
GBT compatibility

- Transceivers are the most complicated part in a sync. network
  - we need a synchronous GBTx interface either way!
  - why not combine the high-level TrbNet protocol with a low-level GBTx interface?
    - fixed latency guaranteed by Cern, freedom for any high-level protocol
    - disadvantage: not compatible to current TDC-FPGAs



TrbNet over GBTx

- 112 (116?) bits free to be used
  - 6 Bit packet type / channel information
  - 10 (14) Bit checksum
  - 3 x 32 Bit payload
- Available bandwidth for data: 80%
- Packet size perfectly fitting to plans for TrbNet v2



Work packages and status

- Data transport from 1600 FPGA ✔
  - up to 200 kHz per channel
- Slow-control for FEE (thresholds, pedestals...) ✔
- online monitoring (PMT signal quality, noise levels) ✔
- time synchronization ✘
  - better than 100ps
  - no additional cable besides optical fiber
- free-running, untriggered read-out ✘
- Port to Kintex 7 ✘
- Fast 10G links ✘
- Tunneling over GBT links ✘

Infrastructure available for all upcoming test beams

Additional features missing

And... what's its name?

Fiber-optical Advanced Inter-fpga Readout NETwork

And... what's its name?

Fiber-optical Advanced Inter-fpga Readout NETwork

FairNet

Conclusion

- Extensions to TrbNet can can provide all features needed for successful data taking in Panda & CBM
- Quite some work
  - discussion needed
  - not a one-man-show – man-power needed
- Using GBT as low-level transport layer gives a convenient, homogenous setup