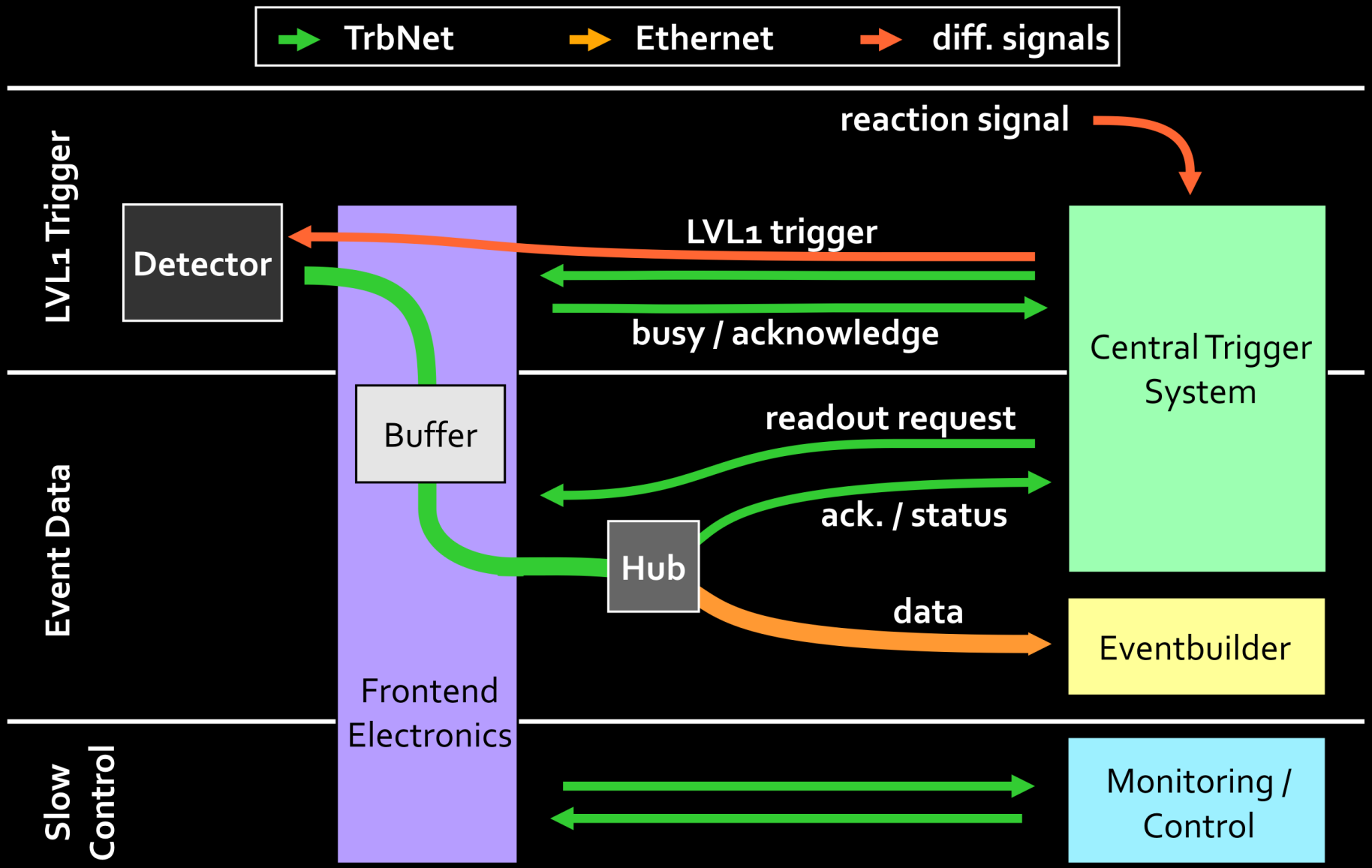
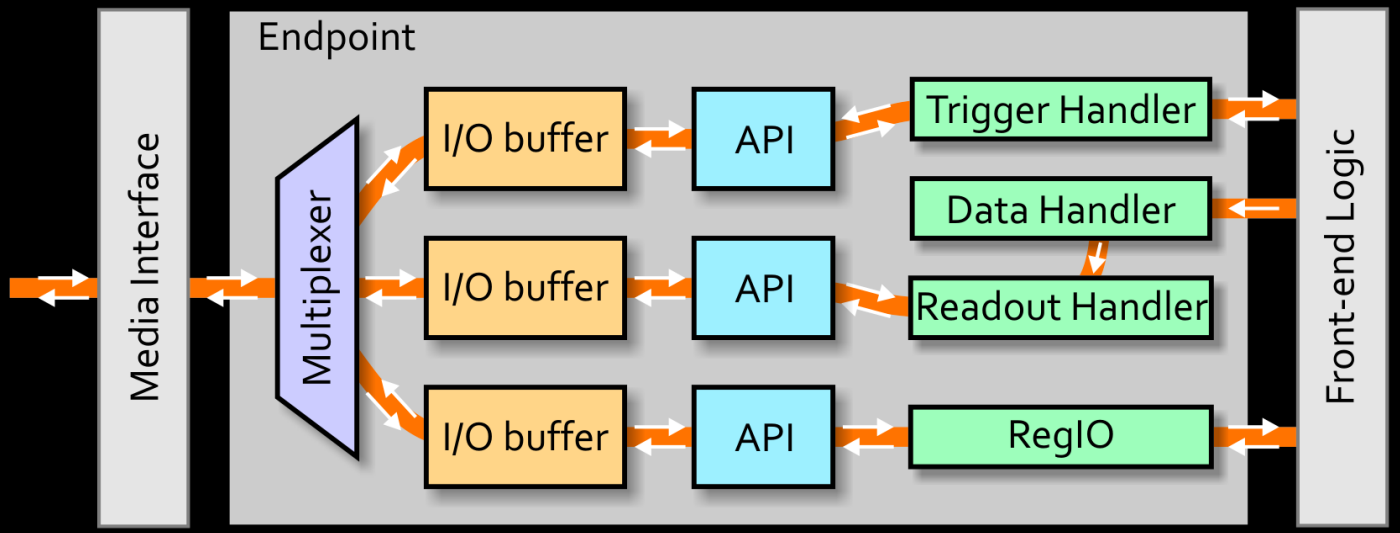
Jan Michel - Goethe Universität Frankfurt

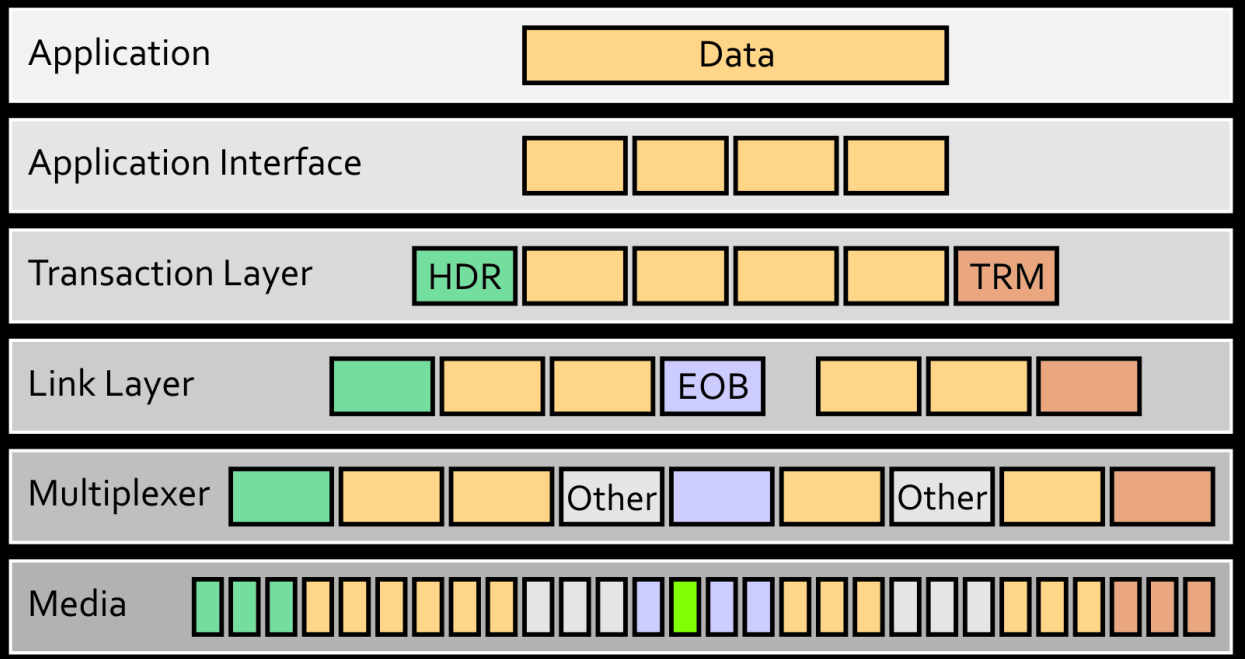
A FPGA network for Panda & CBM

TrbNet – Features (classical)

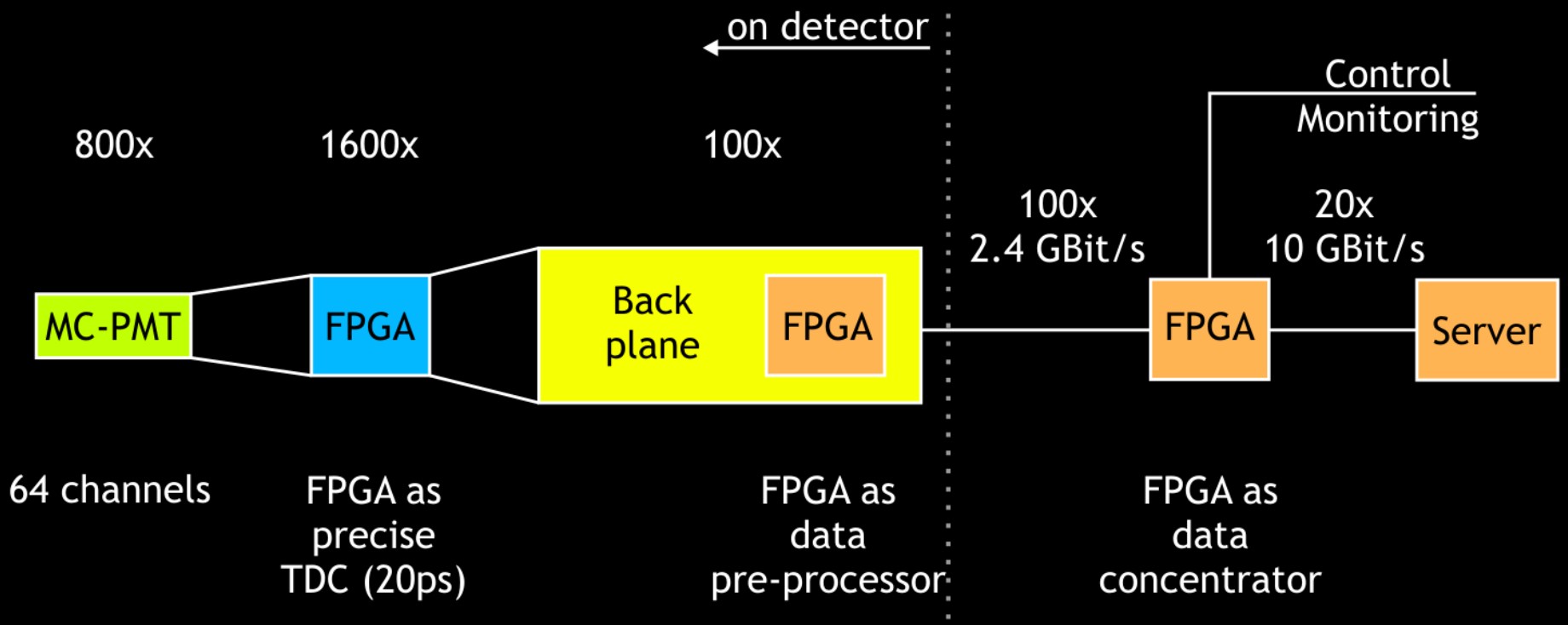


TrbNet (Summary)

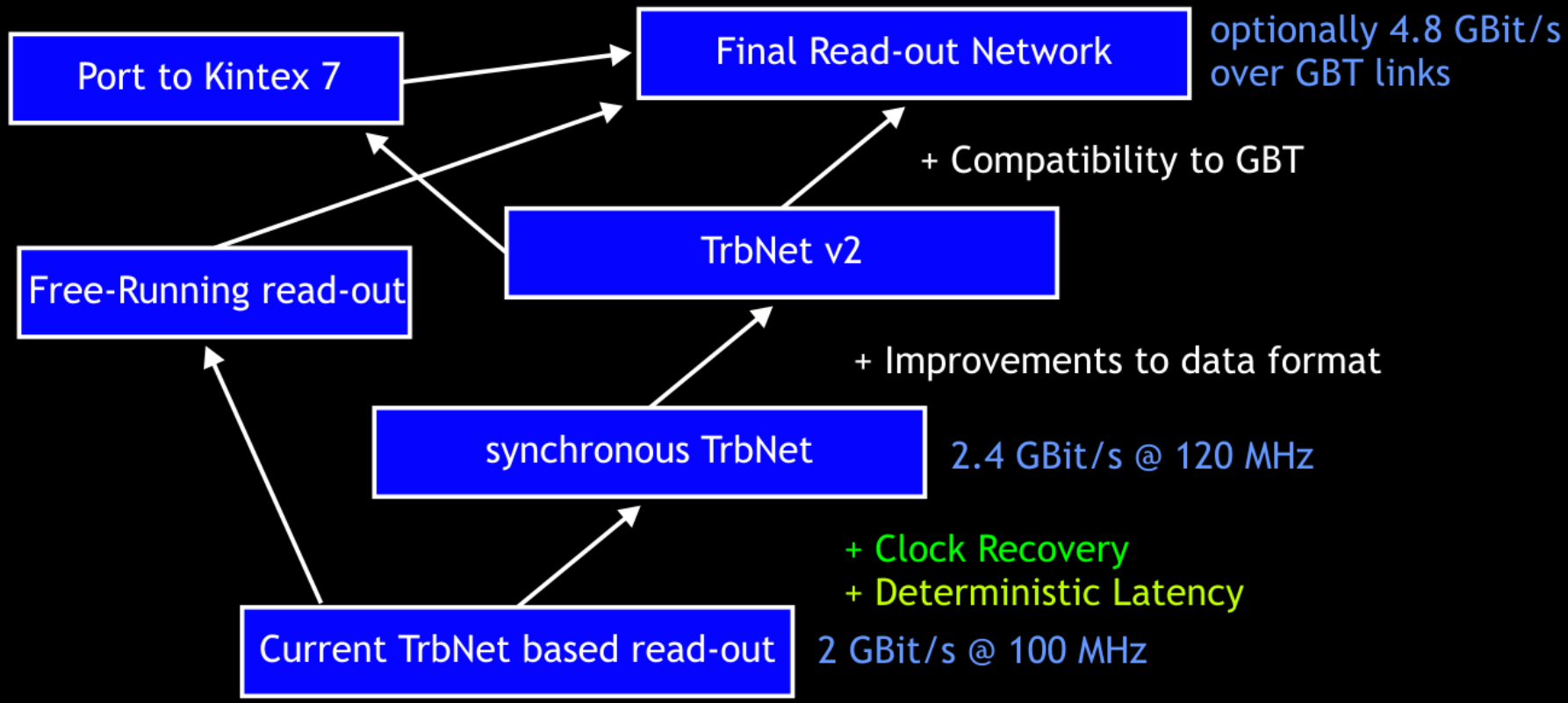




Example: CBM RICH



Initial Road Map



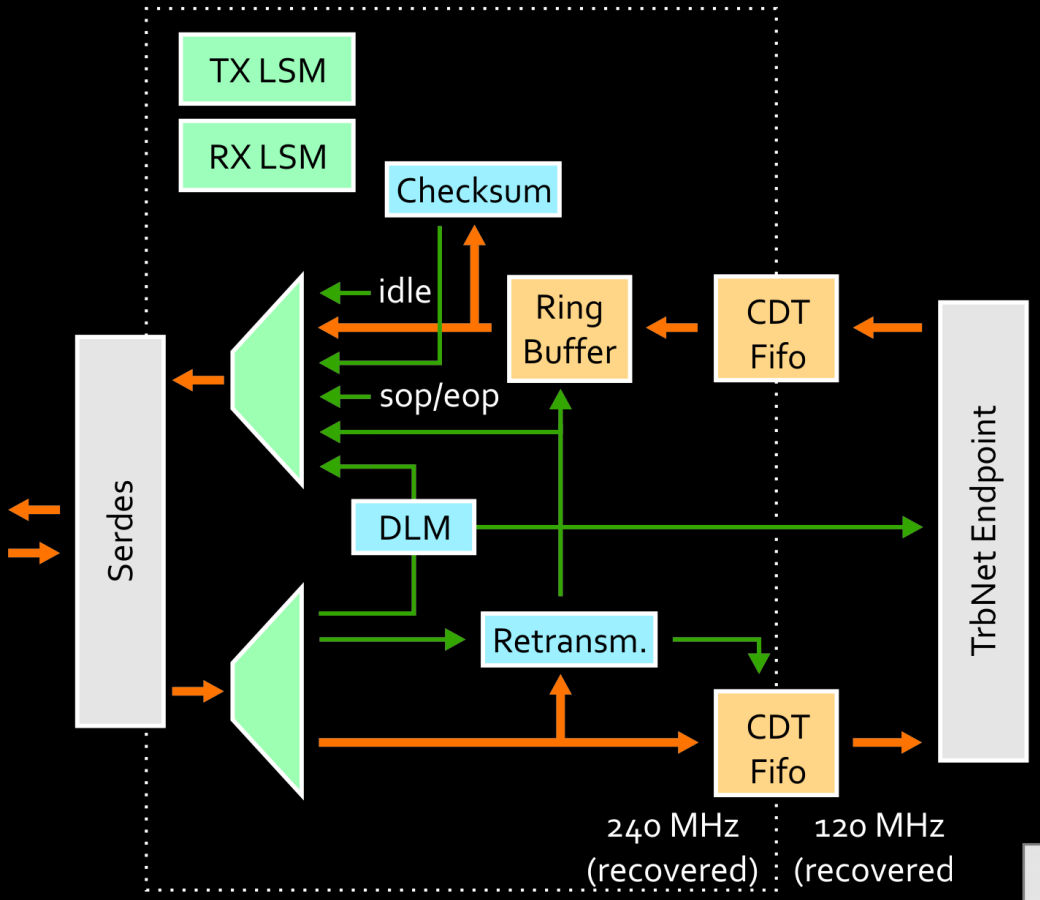
Central question: What is the best way to reach final goals?

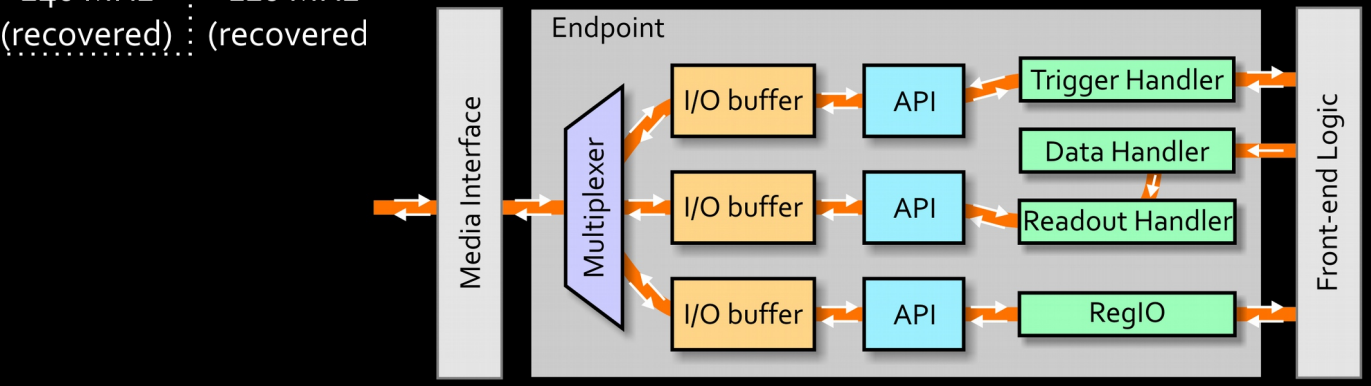
Clock Frequencies

- ECP Fpgas limited to 2.5 GBit/s in synchronous mode
  - higher speed requires RX fifo and 16Bit data paths
- Main other component: GBT
  - running 4.8 GBit/s at 40 MHz base frequency

- Integer divider between all frequencies required for deterministic operation
- use common clock source of 40 MHz
- run FPGA at 120 MHz and 2.4 GBit/s (8b/10b encoding) or 4.8 GBit/s
  - 16bit @ 120 MHz or 40bit @ 120 MHz

Media Interface Architecture





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 Modes

- Version 1: Continuous data stream from FEE, concentrator merges by time on a best-effort basis
  - needs some processing power in concentrator
  - generates overhead due to larger channel addresses
- Version 2: FEE buffers data for N byte or M µs, then sends a packet
  - reduced overhead
- Version 2a: Concentrator just forwards packets without touching them
  - simplest mode
- Version 2b: Packets are sent synchronously by all FEE, concentrator merges all packets from same time range
  - smallest overhead

- Several other schemes possible – what do we want?

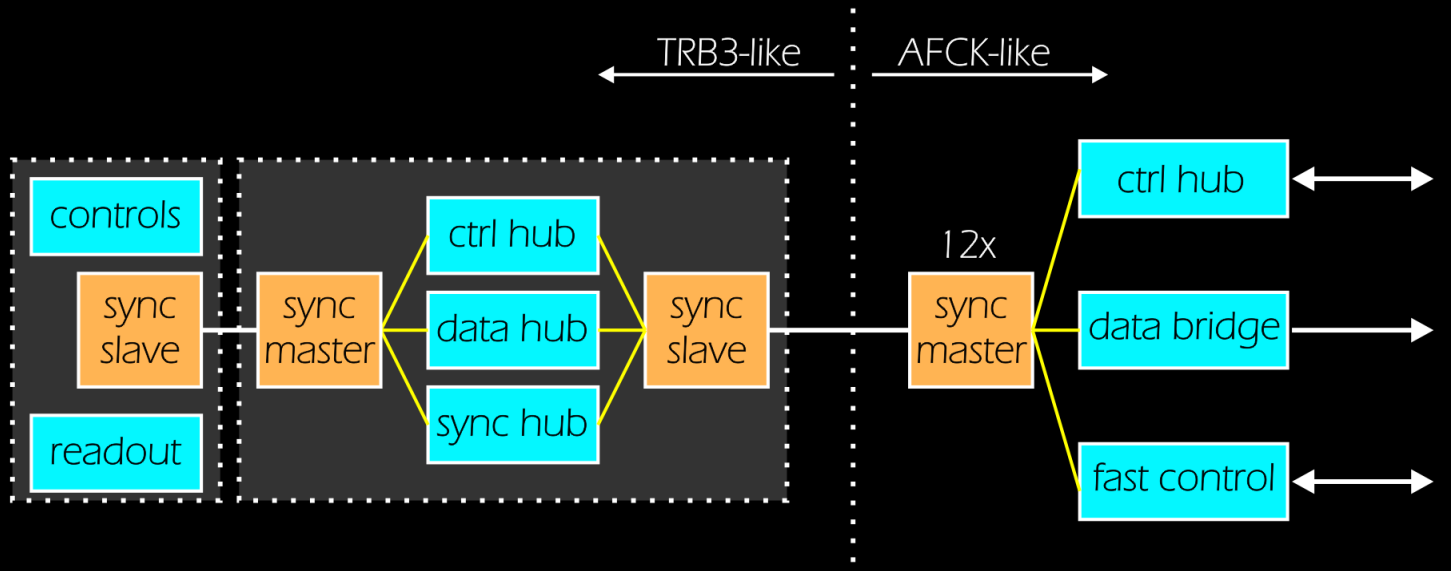
TrbNet v2

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

- version 2
  - variable packet size (?)
  - better inclusion of "packet start" comma and checksums
    - better robustness against data errors (even though seldomly experienced)
  - 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?

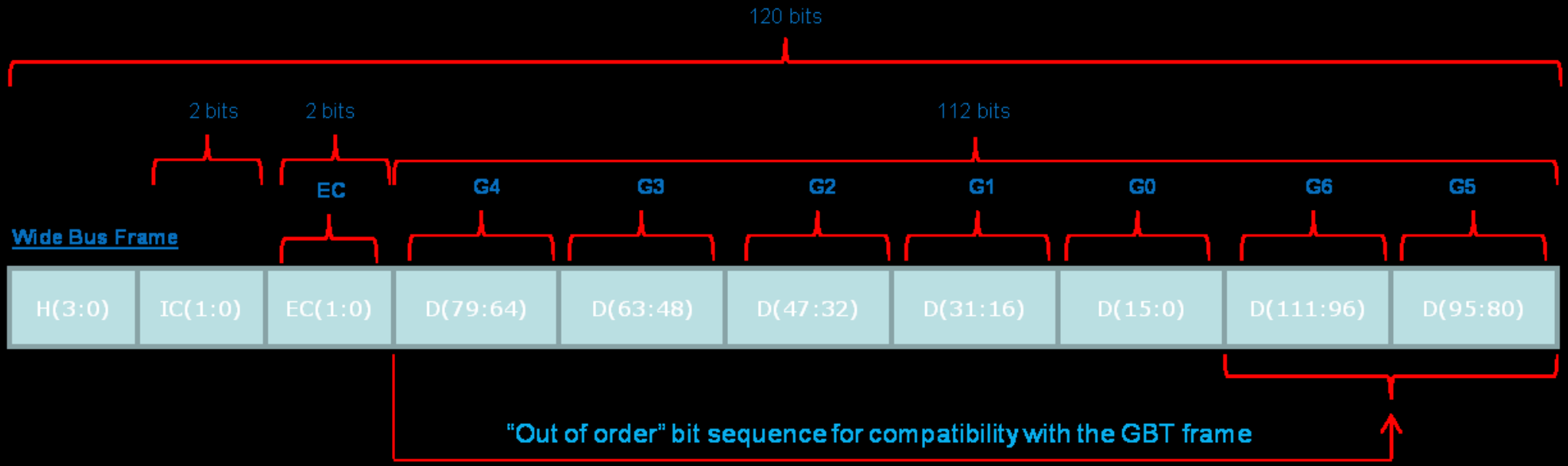
Port to Kintex

- 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



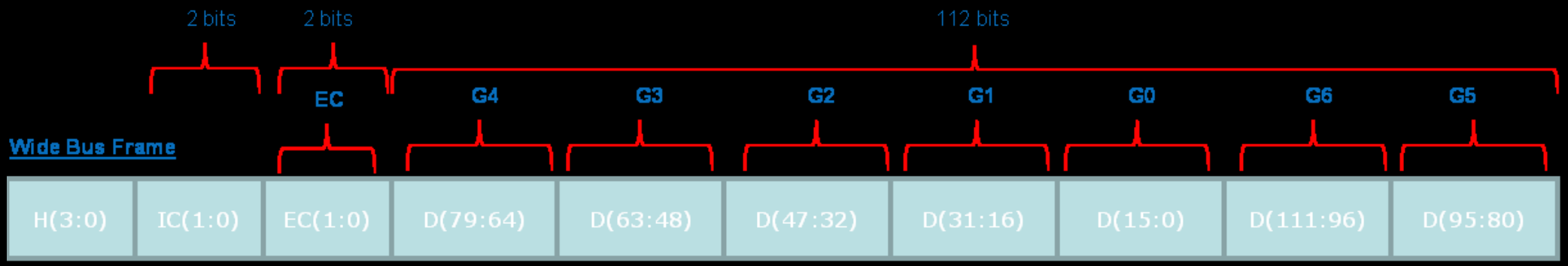
GBT compatibility

- Receivers 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



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



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
- Using GBT as low-level transport layer gives a convenient, homogenous setup