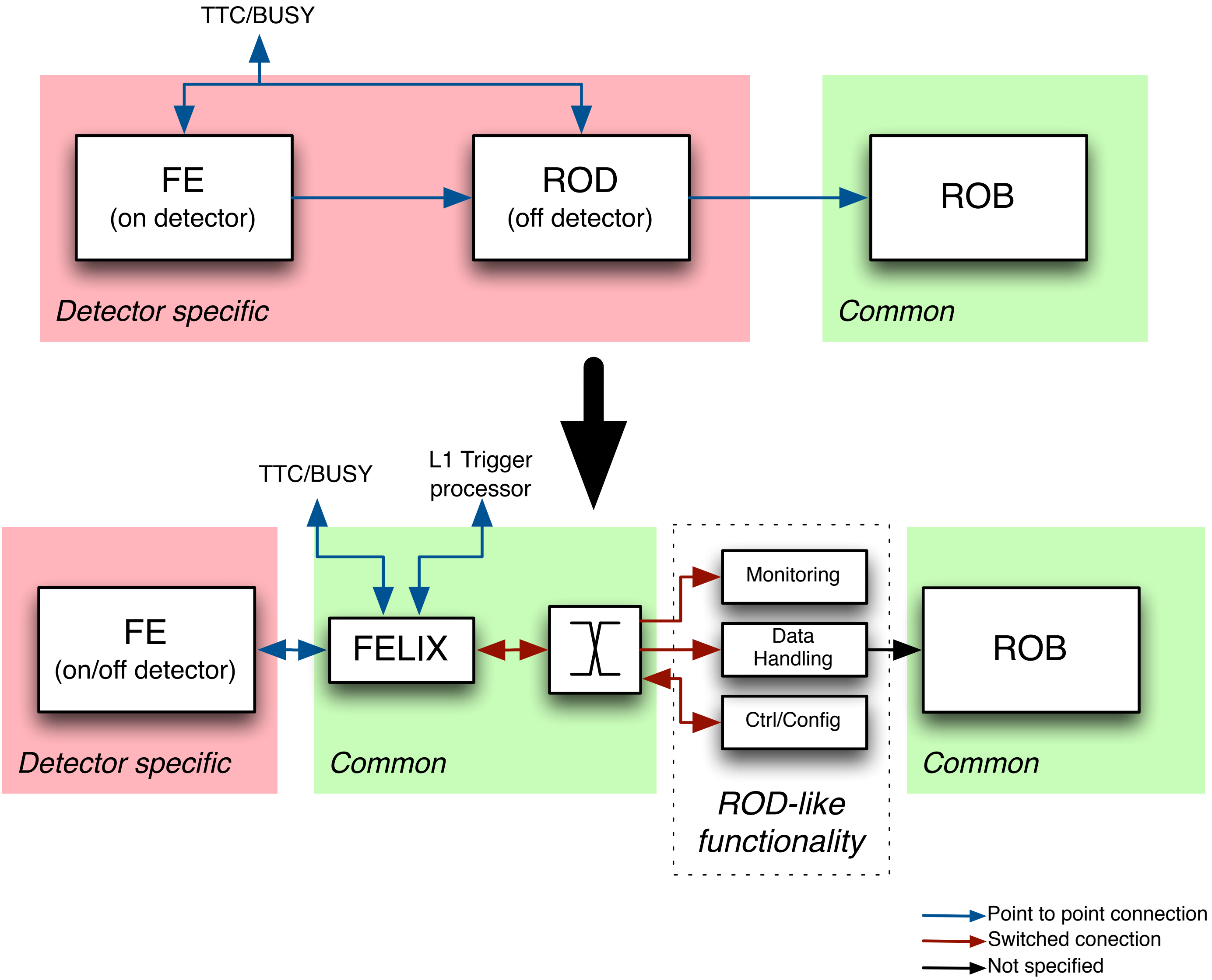
ATLAS and the FELIX

Major change in approach



New common component: FELIX

- Detector-independent
  - No knowledge of detector-specific data formats required
- Quasi-stateless
  - Configurable on a run-by-run basis
- As much as possible based on COTS components
- Supporting multiple input link implementations
  - Hot pluggable transceivers (QSFP-like)
  - Only requiring a minimal GBT protocol implementation for uniformity

- Two target H/W deployments
  (same functionality, different connectivity):
  - Data transfer
    - Not overcommitted: about 40 GBT vs 4x40 Gb/s or 2x100 Gb/s network connections per unit
  - TTC forwarding
    - As many GBT connections per unit as possible (negligible data throughput)

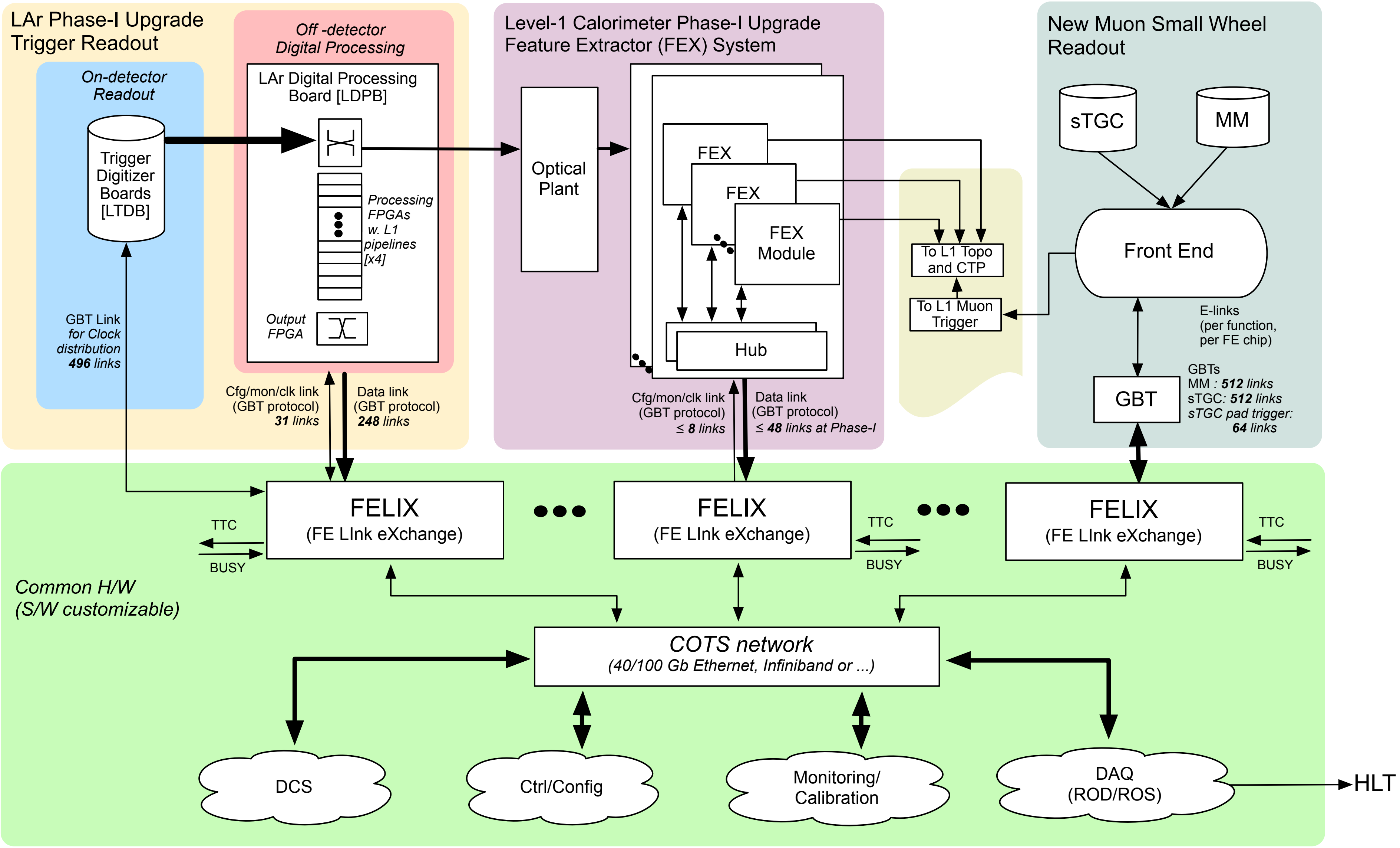
FELIX functionalities (1)

- □ Handling of high level switched protocol
  - ■ Infiniband/Ethernet
  - ■ QoS to guarantee minimal bandwidth to different traffic types
- □ Configurable data routing (flexible FE to Readout Buffer correspondence)
  - ■ Distinguishing data streams either by E-link (no constraints on FE) or by implementing a packet-oriented protocol (still detector independent)
  - ■ Advanced routing based on L1 ID or Trigger type available to detectors implementing packet-oriented protocol
  - ■ Data duplication and sampling for monitoring

FELIX functionalities (2)

- TTC handling
  - Busy handling (no back pressure on GBT)
  - Isolates FE electronics from TTC evolution
  - May provide single L1 and ECR counting with unique implementation (counters values sent upstream or downstream whether needed)
- Dedicated low latency path for critical data
  - E.g. for Level 1 trigger input data for future L0/L1 trigger architecture
- Command synchronisation for calibration
  - Programmable sequences
  - Synchronisation trough TTC

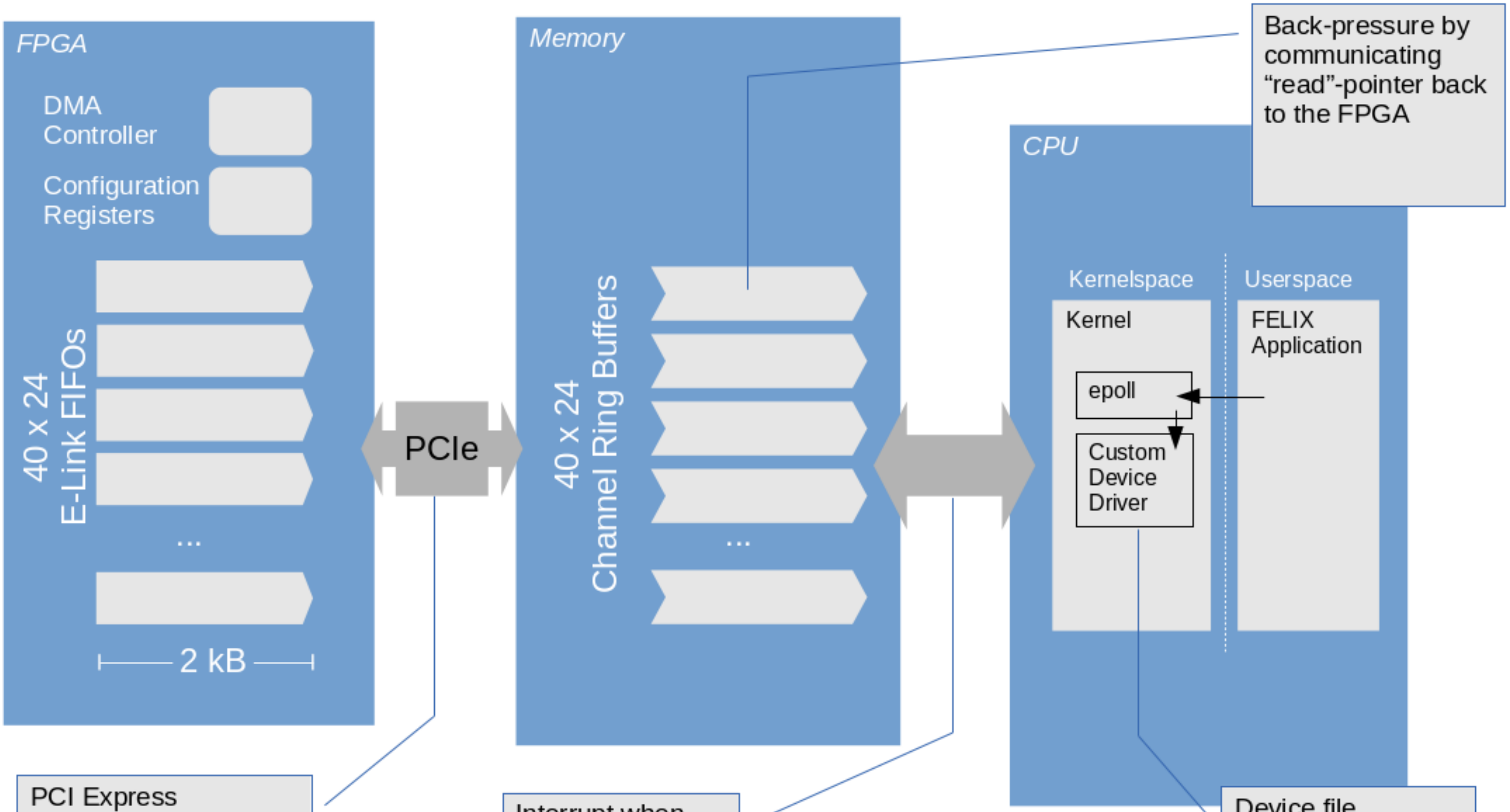
Deployment in Phase I



Status of FELIX project

- H/W implementation for demonstrator chosen:
  - Completely based on COTS components
    - Rack mountable PC + PCIe IO cards for GBT/TTC connections
- BEWARE: H/W choice not binding for final FELIX
  - We are ready to consider drastically different options
  - But must be compatible with design philosophy of using as much COTS components as possible and restricting development to F/W+S/W
- R&D work for demonstrator has started
  - F/W
    - Initial block scheme for FPGA being prepared
    - On paper evaluation of Virtex 7 Block RAM resources indicates that we should be able to implement at least 20 GBT inputs per FPGA
  - S/W
    - Interface for data transfer from/to FPGA being discussed
    - Driver architecture being discussed:
      - Performance tests of epoll mechanism to notify userspace of new data have given results exceeding requirements

FELIX demonstrator internals



Back-pressure by communicating "read"-pointer back to the FPGA

PCI Express Scatter-Gather operations constantly initiated by the DMA controller on the FPGA

Interrupt when new data arrived

OR

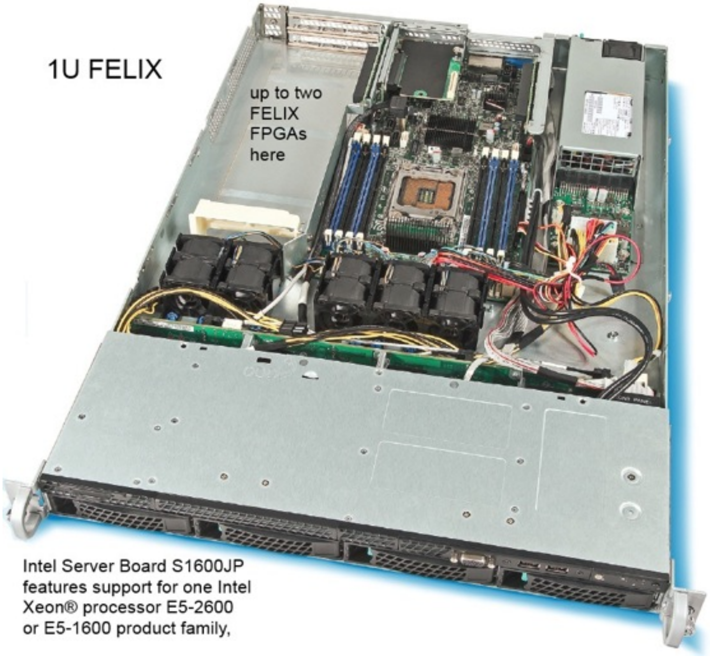
Polling

Device file interface

H/W candidates for demonstrator

Possible clock recovery issue for GBT implementation being investigated

- □ Hitech Global HTG-710
  - ■ 2 CXP cages: can connect 24 bi-directional links
  - ■ With Virtex-7 X690T FPGA (for PCIe Gen3 8 lanes)



1U FELIX

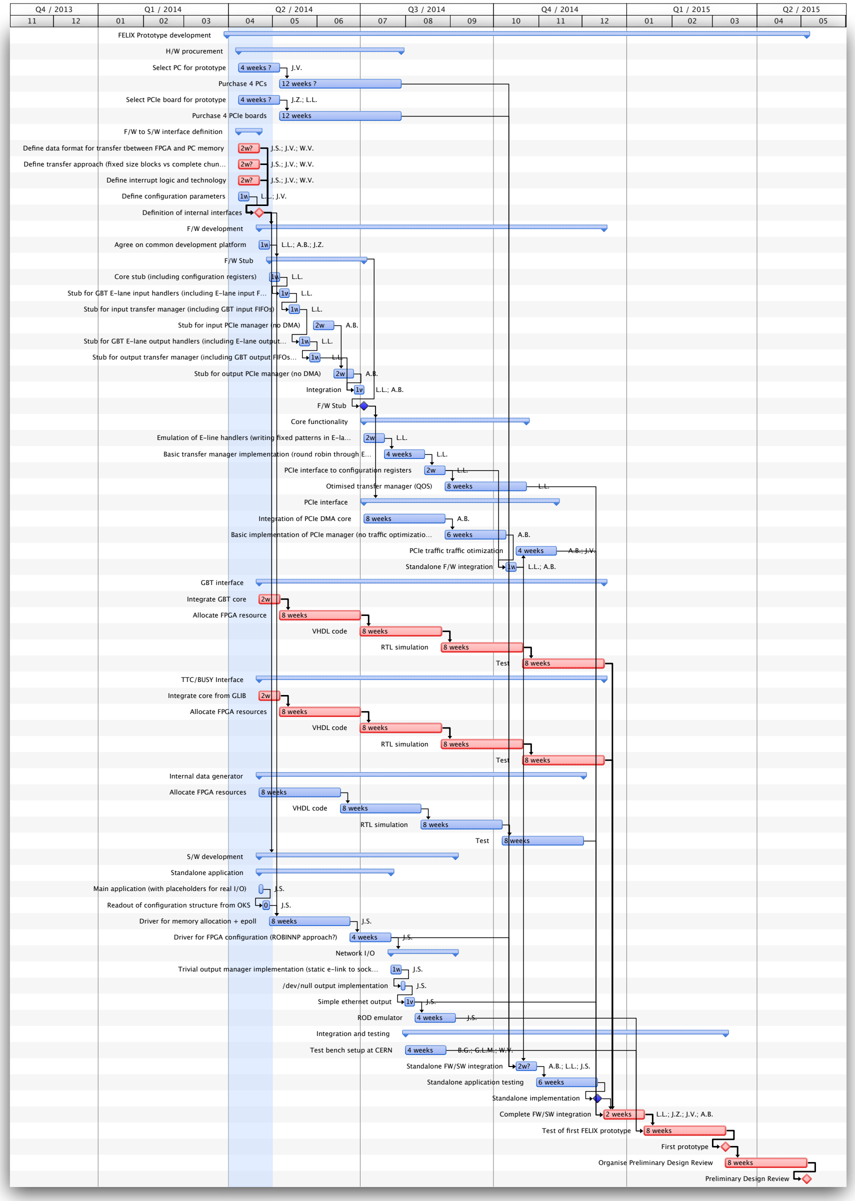
up to two FELIX FPGAs here

Intel Server Board S1600JP features support for one Intel Xeon® processor E5-2600 or E5-1600 product family,

- □ Intel server board S1600JP
  - ■ 3 PCIe Gen3 slots

FELIX demonstrator project plan

- □ Next milestones
  - ■ Formal definition of internal interfaces
    - □ **~1 week**
  - ■ F/W stub
    - □ **Jul 2014**
  - ■ Standalone prototype (only internal I/O)
    - □ **Nov 2014**
  - ■ Prototype with external I/O (non optimised)
    - □ **Q1 2015**



Areas for common discussions

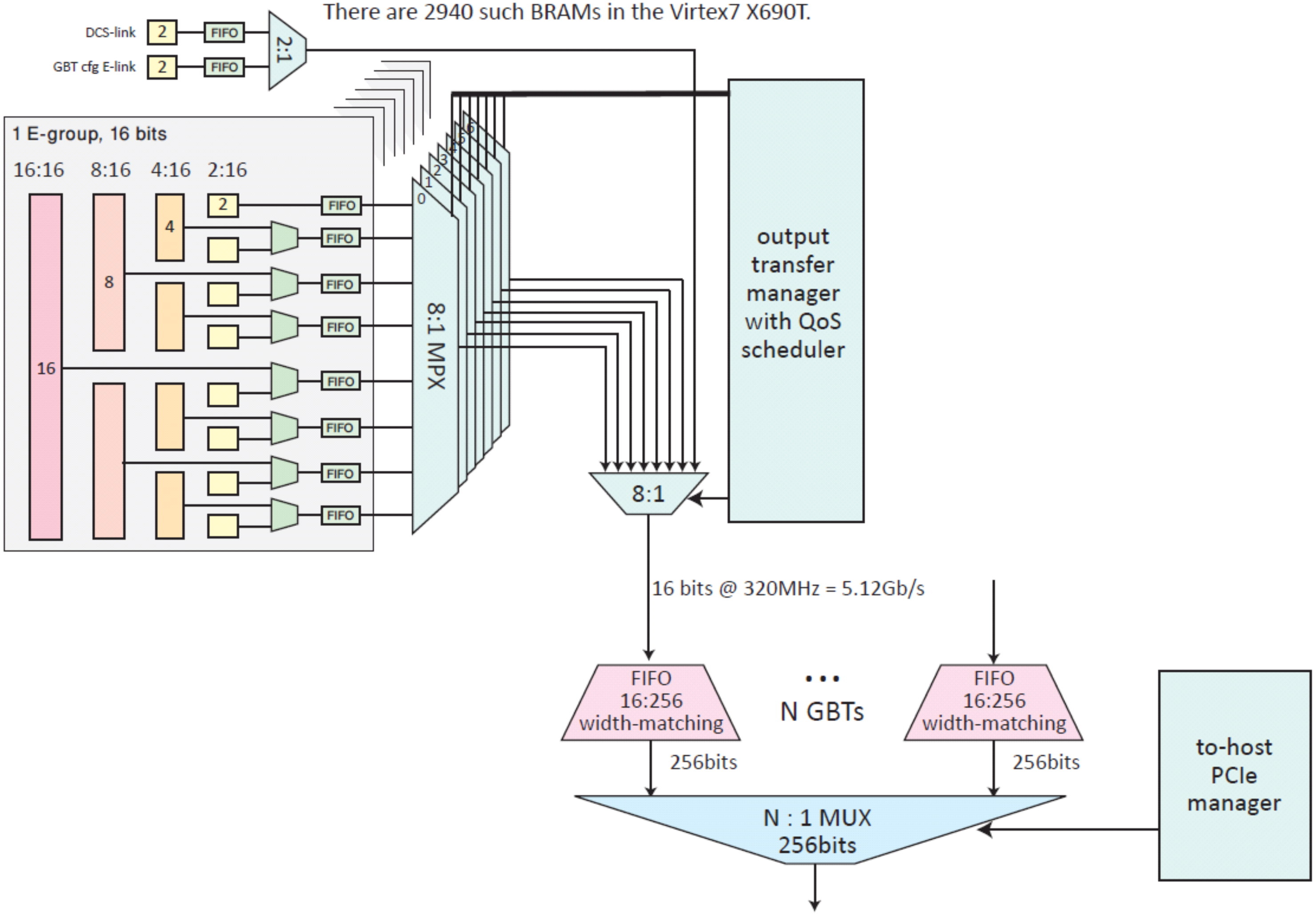
- Sharing of requirements and specifications
- Sharing VHDL in order to understand details, e.g. GBT error handling
- GBT-FPGA issues (to be resolved with CERN GBT group)
- Intel processor optimisations, e.g. memory cache options, PCIe sleep timeout
- Network interface and Linux network optimisation
    - FELIX: TCPIP; PCie40, Tell40: ?
- Requirements for TTC for Phase 2; VHDL for TTC

- *Altera-Xilinx divide precludes some, but not all, sharing*

Backup for discussion

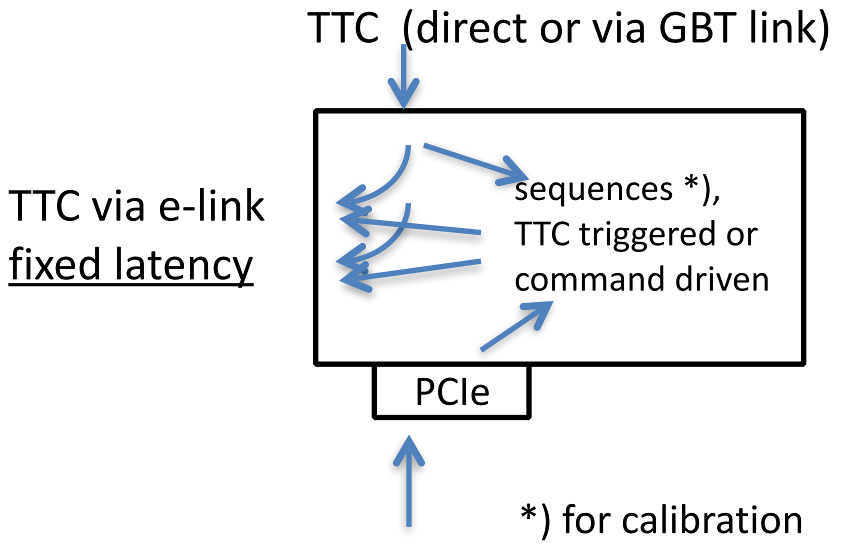
Internal F/W dataflow architecture

Each FIFO is 1K x 16-bits and uses one BRAM.
There are 2940 such BRAMs in the Virtex7 X690T.



to corresponding stream buffers in
PC host memory via PCIe gen 3 x8

TTC related



TTC (direct or via GBT link)

TTC via e-link

fixed latency

sequences *),

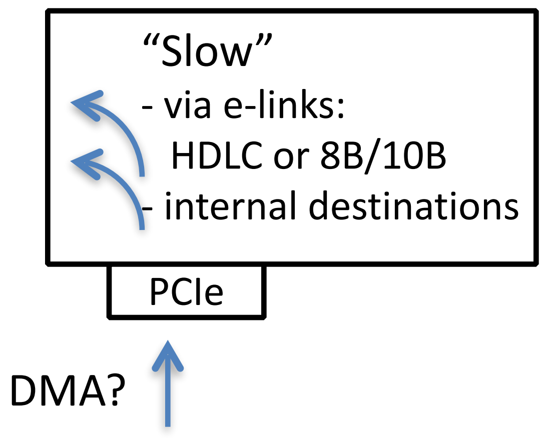
TTC triggered or

command driven

PCIe

*) for calibration

Control / Configuration commands



“Slow”

- via e-links:

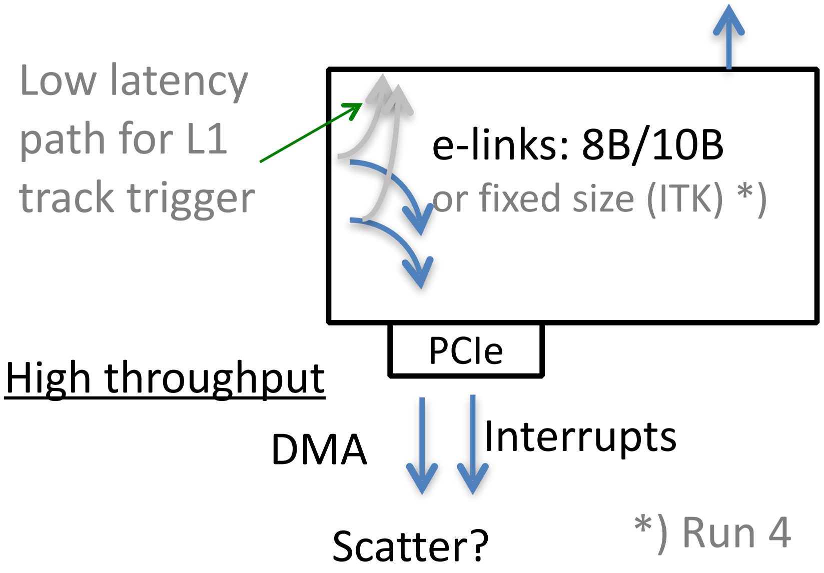
HDLC or 8B/10B

- internal destinations

PCIe

DMA?

Event data readout



BUSY

Low latency

path for L1

track trigger

e-links: 8B/10B

or fixed size (ITK) *)

PCIe

High throughput

DMA

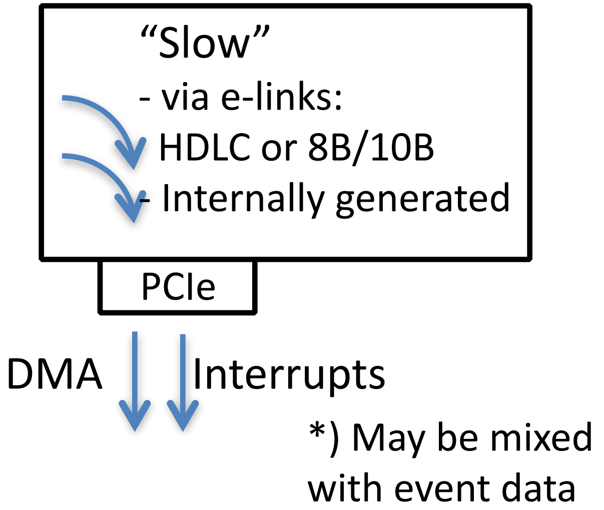
Interrupts

Scatter?

*) Run 4

Control / Configuration responses*)

Error messages



“Slow”

- via e-links:

HDLC or 8B/10B

- Internally generated

PCIe

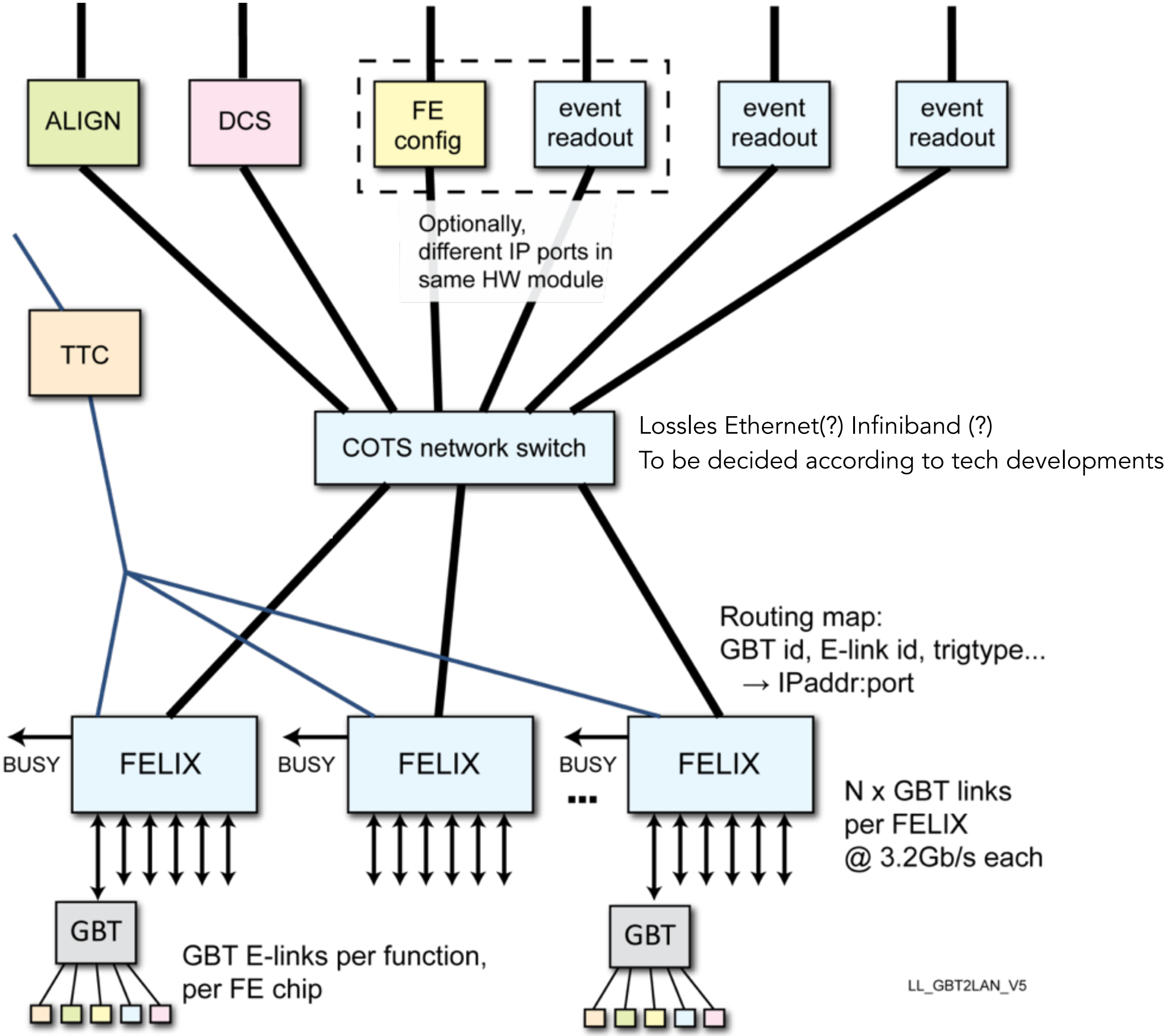
DMA

Interrupts

*) May be mixed

with event data

Upgraded Readout Architecture



Optionally, different IP ports in same HW module

Lossles Ethernet(?) Infiniband (?)
To be decided according to tech developments

Routing map:
GBT id, E-link id, trigtype...
→ IPaddr:port

N x GBT links
per FELIX
@ 3.2Gb/s each

GBT E-links per function,
per FE chip