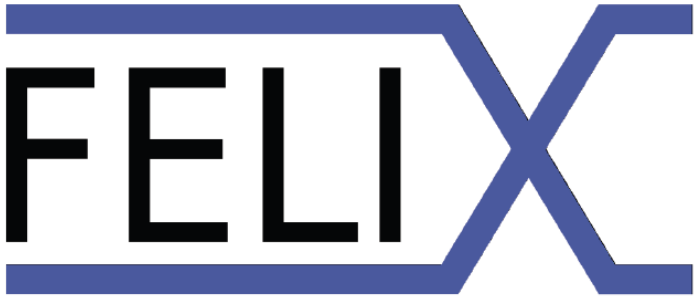


Front**E**nd **LI**nk e**X**change

Detailed Description

Andrea Borga
Nikhef
on behalf of the FELIX group

ATLAS EXPERIMENT

FELIX development team

Argonne NATIONAL LABORATORY
- John Anderson
- Soo Ryu
- Jinlong Zhang

BROOKHAVEN NATIONAL LABORATORY
- Hucheng Chen
- Kai Chen
- Weihao Wu
- Francesco Lanni

CERN
- Markus Joos
- Giovanna Lehmann
- Wainer Vandelli
- Benedetto Gorini

Nikhef
- Frans Schreuder
- Andrea Borga
- Henk Boterenbrood
- Jos Vermeulen

UNIVERSITÄT PADERBORN
Die Universität der Informationsgesellschaft
- Joern Schumacher

Radboud University
- Mark Donszelmann

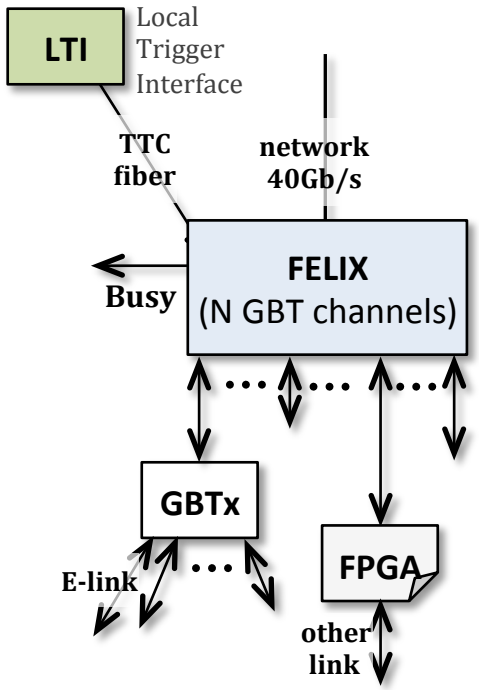
UCI University of California, Irvine
- Daniel Guest
- Daniel Whiteson

WEIZMANN INSTITUTE OF SCIENCE
- Julia Narevicius
- Alex Roich
- Lorne Levinson

ROS/ROD effort on software framework:
- Gordon Crone (UCL)
- Will Panduro (RHUL)
- Joern Schumacher (CERN / Paderborn)
- Wainer Vandelli (CERN)

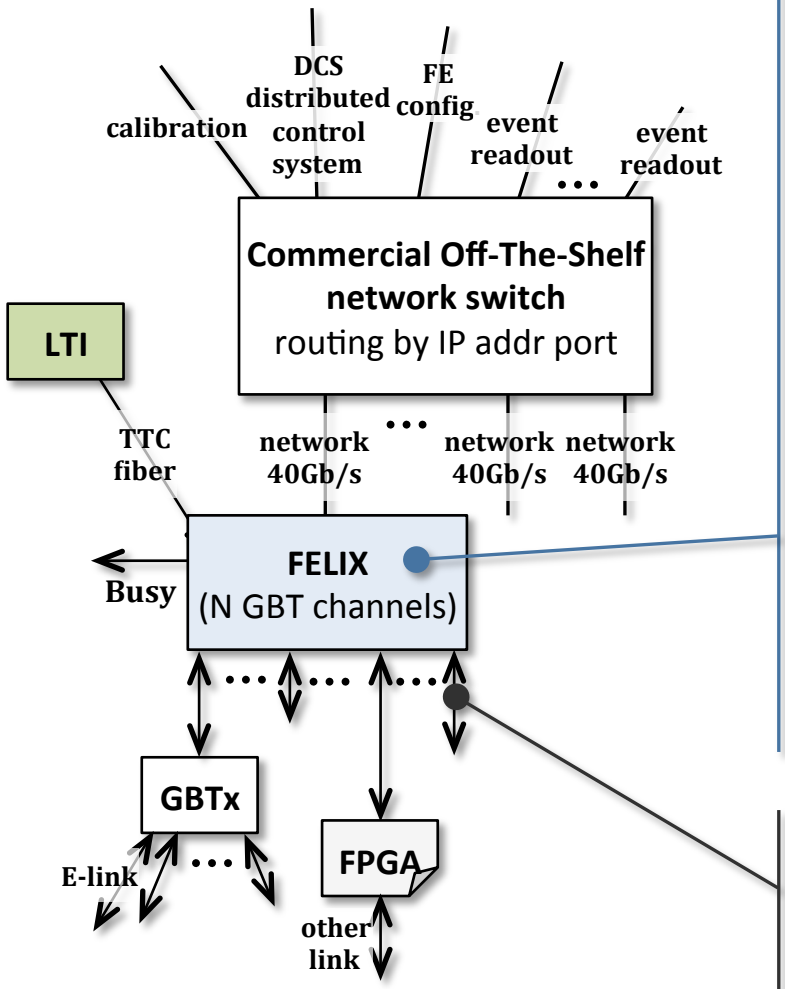
FELIX at a glance

Interfaces multiple GBT (GigaBit Transceiver) serial links to a high bandwidth standard network (40 Gbps Ethernet or 56 Gbps InfiniBand) then routes logical data flows to/from different off-detector endpoints.



- Separates GBT technology into a standard, fixed, but configurable, building block for several detectors.
- FELIX does not need to know the running state of the experiment or detailed data formats – high-reliability: always running.
- Enables different logical data flows to be handled by different off-detector end points.
  - In the past done by using separate physical interconnects.
- Reduces the amount of custom HW in ATLAS in favour of COTS HW and SW.

FELIX: Front-End LInk eXchange



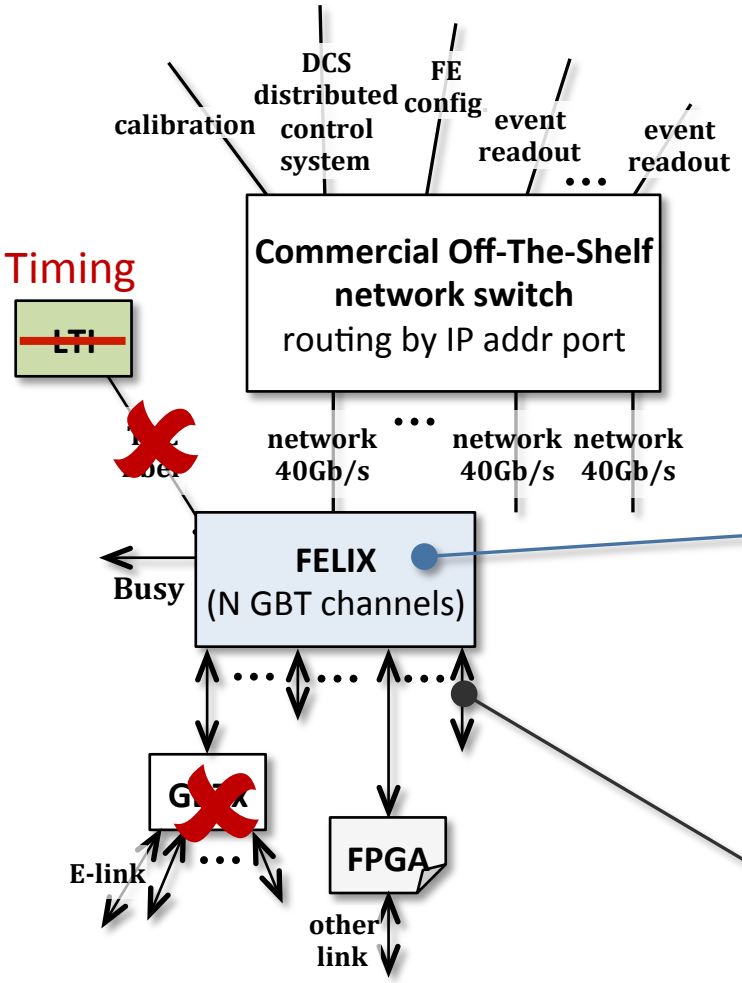
FELIX features:

- → Routing of event data, detector control, configuration, calibration and monitor
- → Connects ATLAS detector frontends to the ATLAS DAQ system, both up-links and down-links
- → SW based data processors and handlers
- → All FELIX components are independently upgradable (PCs, FPGAs, NICs)
- → Integrated Timing Trigger and Control (TTC) and LHC clock distribution
- → Scalable architecture
- → Flexible mapping of FE GBTs to data handlers
- → Detector independent

FELIX Operation Modes:

- → GBT Normal mode (with FEC, 3.2 Gb/s)
- → GBT wide mode (no error correction, 4.48 Gb/s)
- → "Full mode" (~10 Gb/s, protocol to be defined)
- → TTC distribution

FELIX: Front-End LInk eXchange



FELIX features:

- → **Routing of event data, detector control, configuration, calibration and monitor**
- → Connects ~~ATLAS~~ ANY detector frontends to the ~~ATLAS~~ ANY DAQ system, both up-links and down-links
- → **SW based** data processors and handlers
- → **All FELIX components are independently upgradable (PCs, FPGAs, NICs)**
- → ~~Integrated Timing Trigger and Control (TTC) and LHC clock distribution~~ Potential to integrate other timing systems
- → **Scalable architecture**
- → ~~Flexible mapping of FE GBTs to data handlers~~
- → **Detector independent**

FELIX Operation Modes:

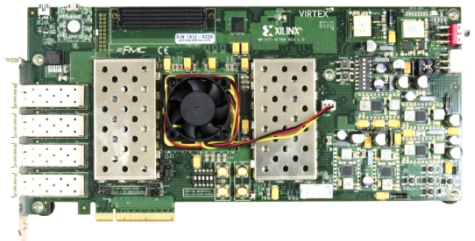
- → ~~GBT Normal mode (with FEC, 3.2 Gb/s)~~
- → ~~GBT wide mode (no error correction, 4.48 Gb/s)~~
- → "Full mode" (~10 Gb/s, protocol to be defined)
- → ~~TTC distribution~~ Timing distribution

FELIX server PC components

TTCfx

→ Custom FMC with TTC input
→ ver1: ADN2814 + CDCE62005
→ ver2: ADN2814 + Si5338

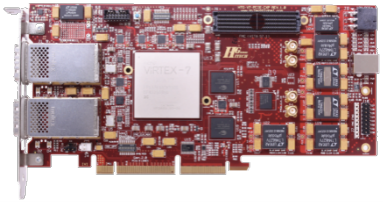
FLX-709: Xilinx VC-709



→ Subset of full FELIX, intended for FE development support
→ Virtex-7 X690T
→ 4 SPF+ connectors
→ PCIe Gen3 x8

or

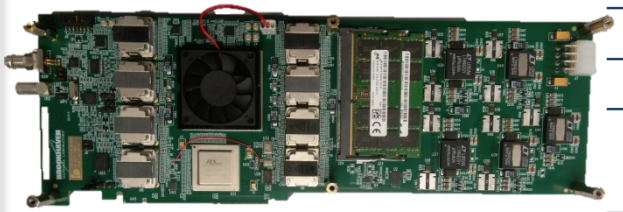
FLX-710: HiTech Global HTG-710



→ FELIX development
→ Virtex-7 X690T
→ 2x12 bidirectional CXP connectors
→ PCIe Gen3 x8

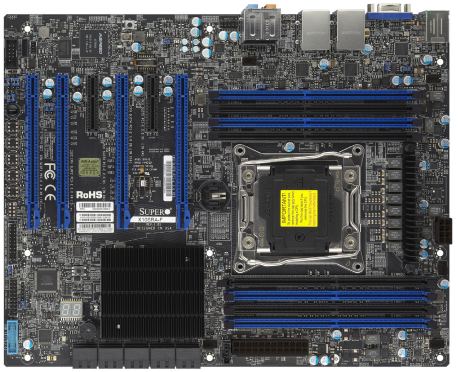
or

FLX-711 from BNL



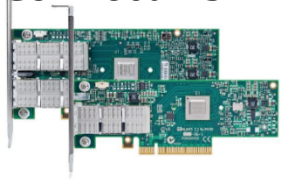
→ FELIX phase 1 prototype
→ TTC input ADN2814 + Si5345
→ Xilinx Kintex **Ultrascale** XCKU115
→ 48 duplex optical links (based on MiniPODs)
→ PCIe Gen3 x16

SuperMicro X10SRA-F



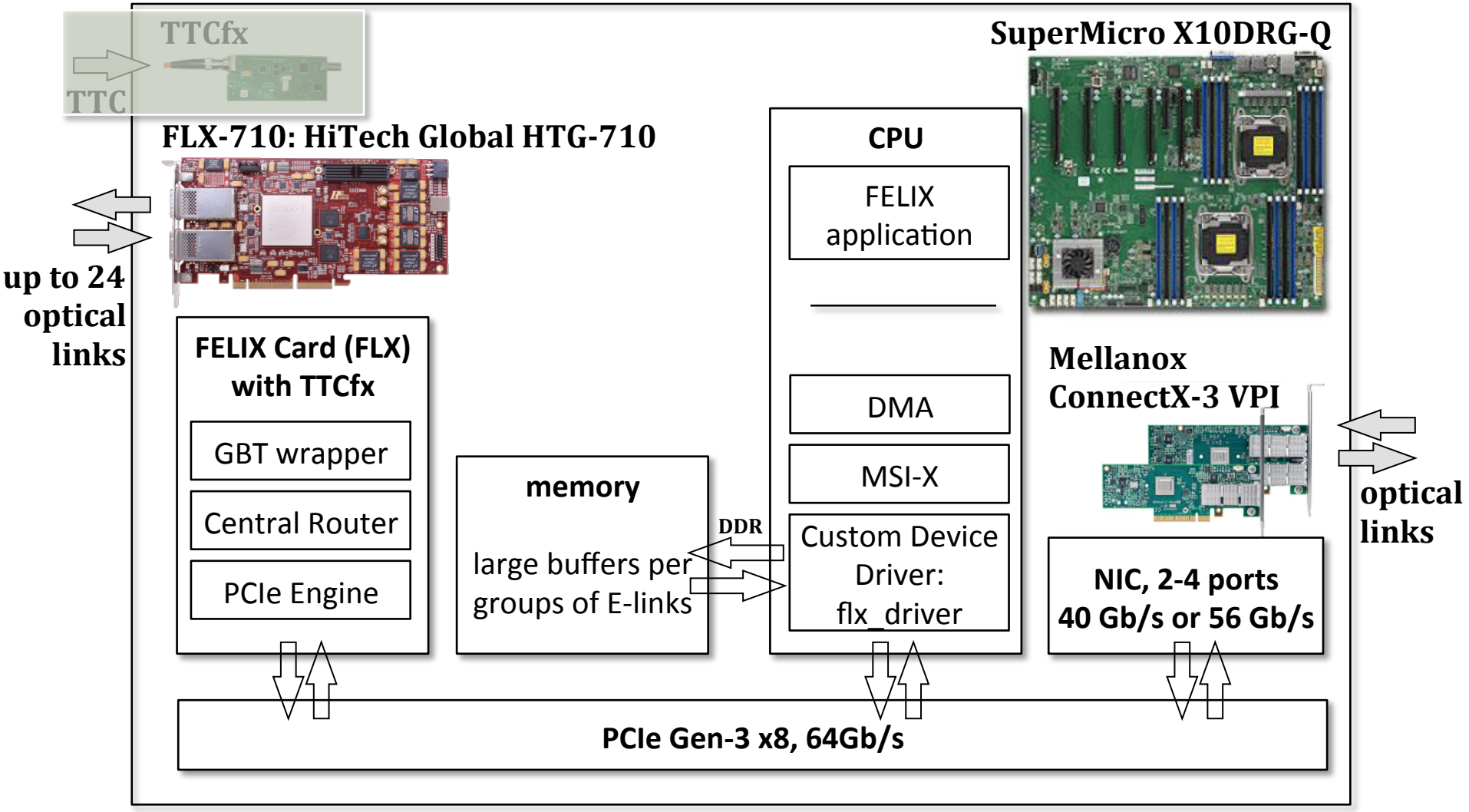
→ Haswell CPU, e.g. E5-1650V3, 6 cores, 3.5 GHz
→ PCIe Gen3 slots

Mellanox ConnectX-3 VPI

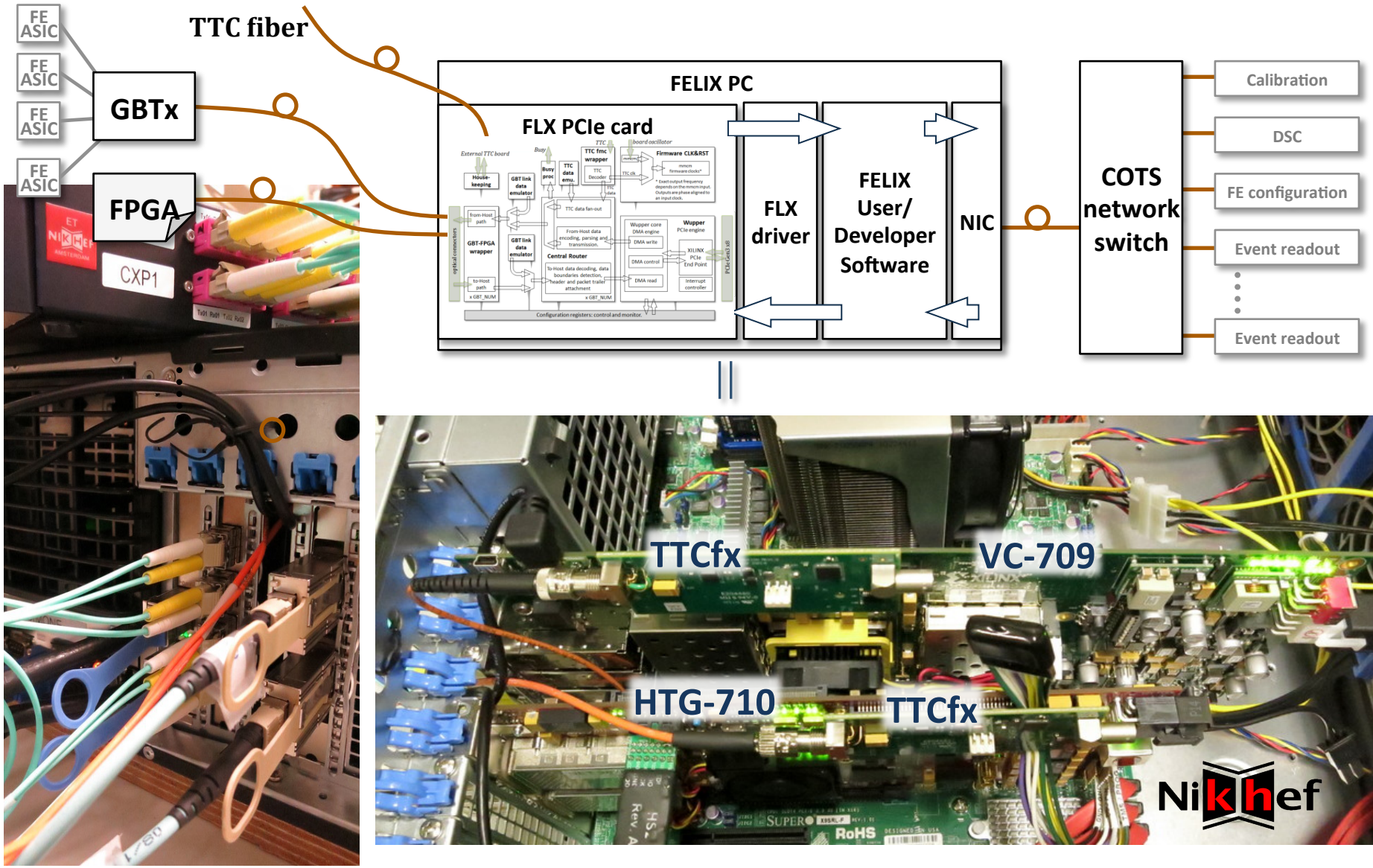


→ 2x FDR/QDR Infiniband
→ 2x 10/40 GbE

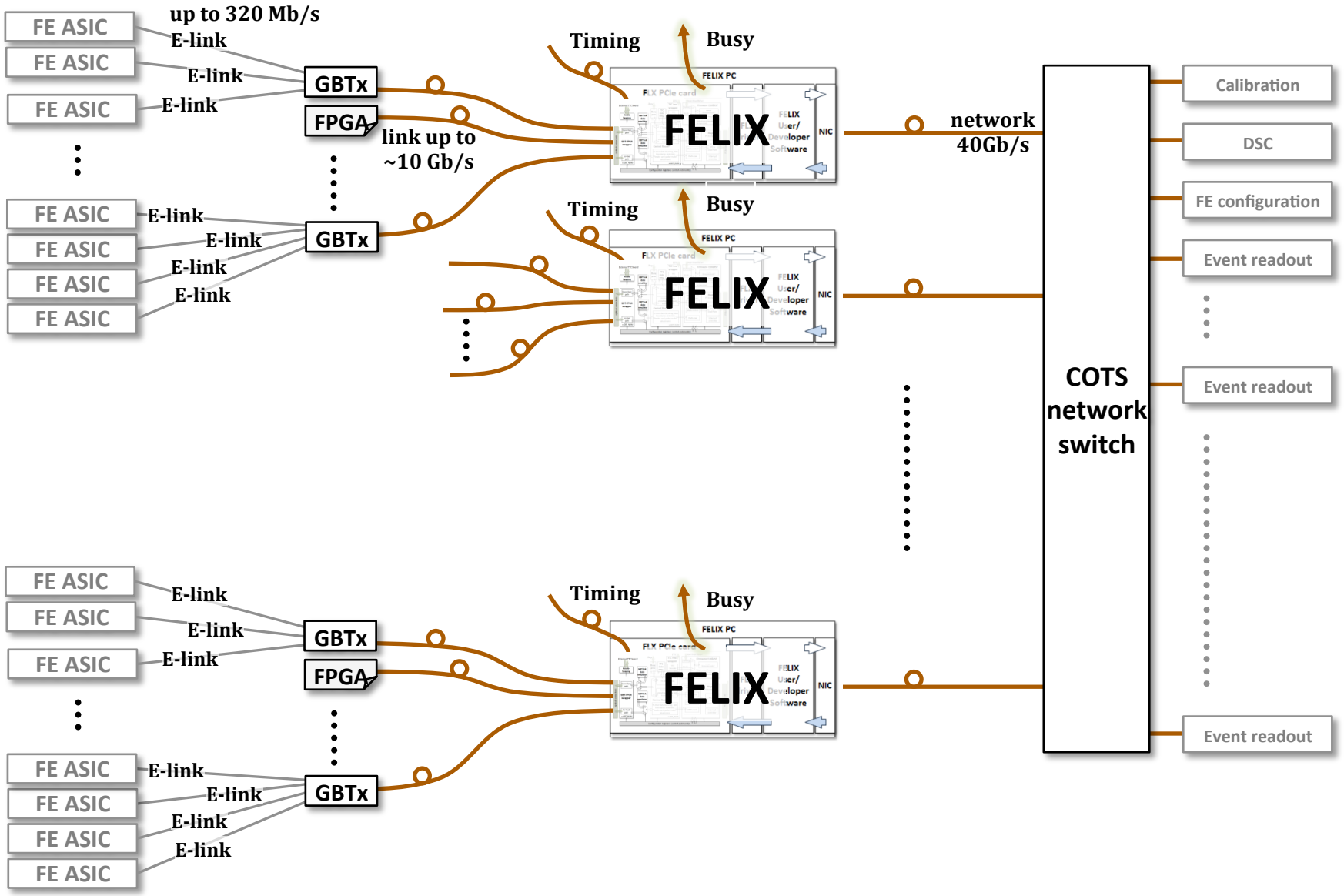
FELIX demo and development platform



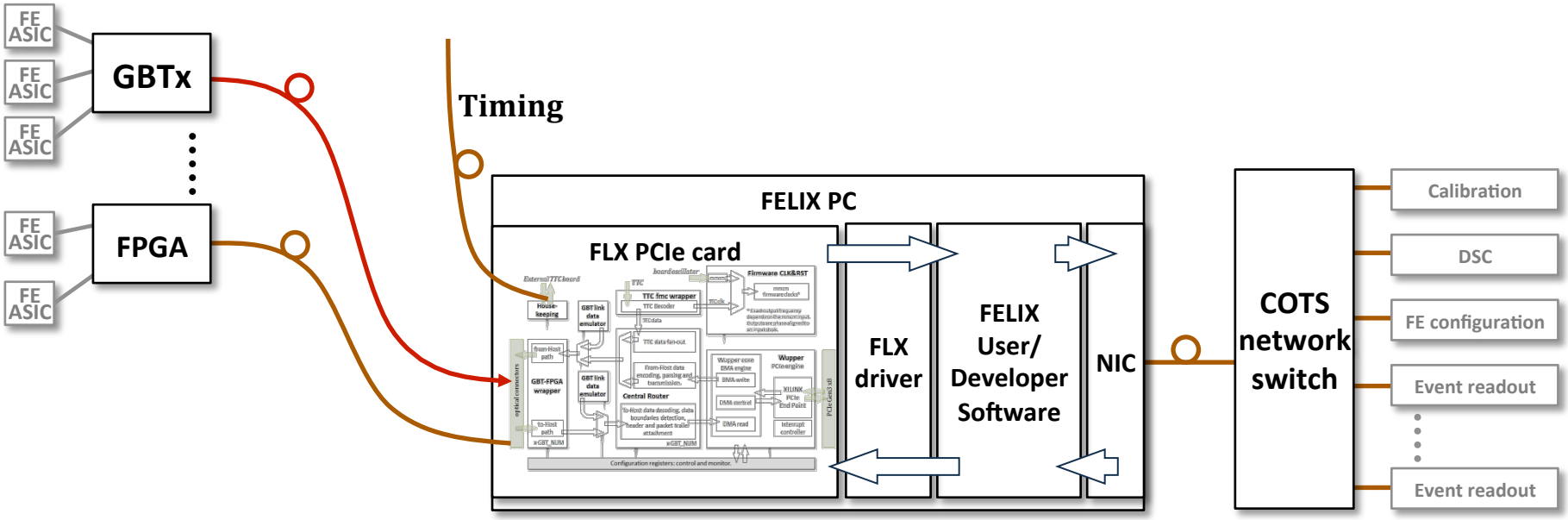
FELIX today


FE ASIC
FE ASIC
FE ASIC
FE ASIC
GBTx
FPGA
TTC fiber
CXP1
FELIX PC
FLX PCIe card
FLX driver
FELIX User/ Developer Software
NIC
COTS network switch
Calibration
DSC
FE configuration
Event readout
Event readout
TTCfx
VC-709
HTG-710
TTCfx
Nikhef

FELIX Flow overview



FELIX Flow: From-FrontEnd



The GBT protocol (simplified)

- The *GBT protocol* is a data serializier (with line encoding)
- The *GBT frame* is 120 bits @ 40 MHz
  - 4 bits are reserved for a header (data / no data)
  - 4 bits are always available for the To FrontEnd direction
- *GBT wide mode*, 4.48 Gb/s (112 bit @ 40MHz) payload
- 8B/10B symbols (<u>not</u> for line encoding) used to delimit stream boundaries, Start Of Frame (SOF), End Of Frame (EOF), comma

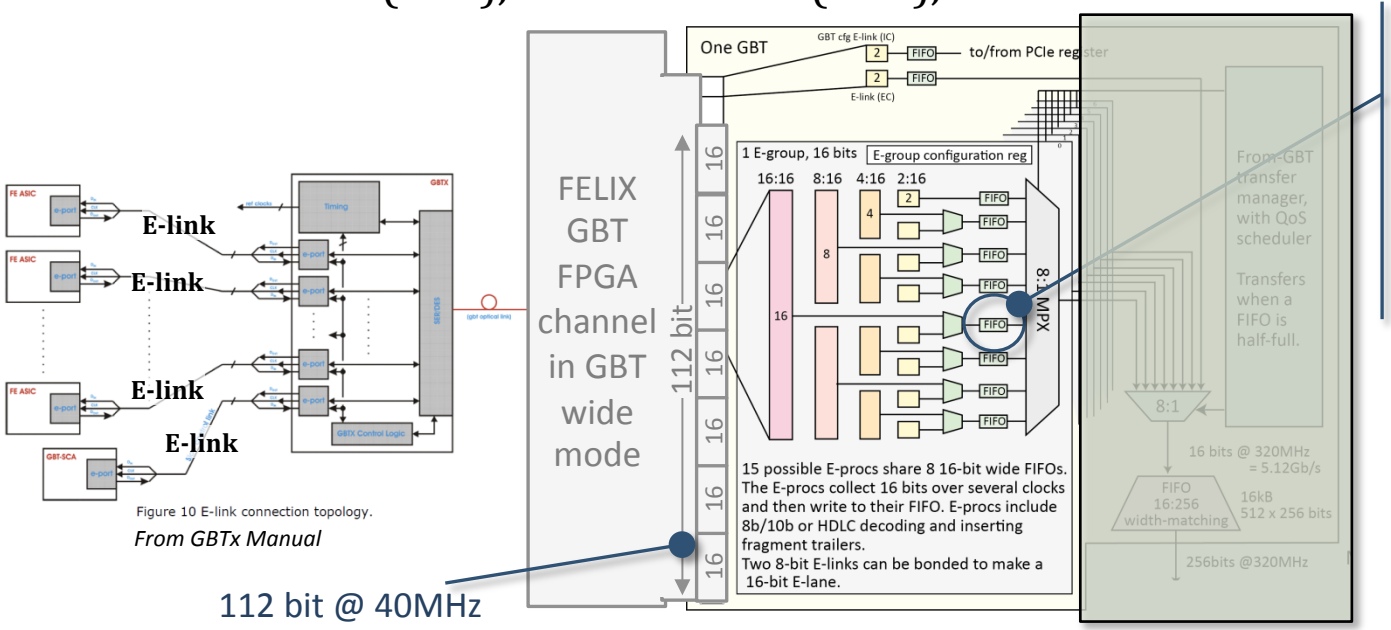
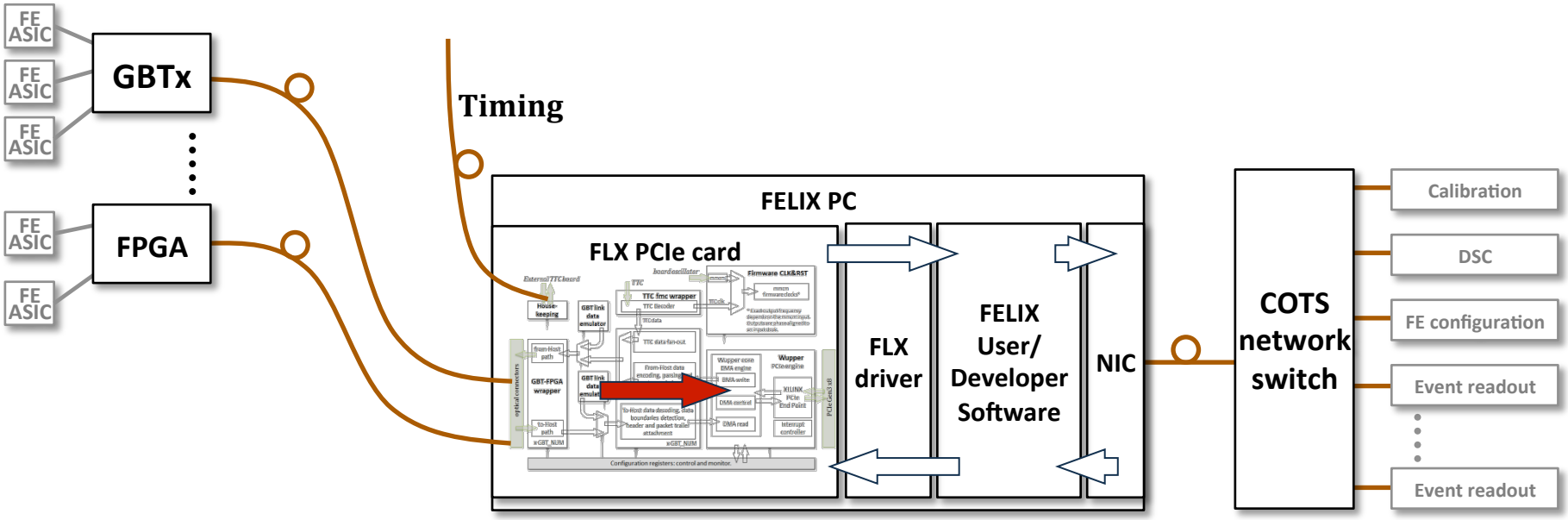


Figure 10 E-link connection topology.
From GBTx Manual

2KByte FIFO.
→ In the To-Host direction accumulates 1KByte blocks of corresponding E-link data. Transfers complete 1KByte blocks.

E-link: variable-width logical link (80, 160 or 320 Mb/s) on top of the GBT protocol. Can be used to logically separate different streams on a single physical link.
E-group: a group of e-links (any combination of 16 bits)
E-proc: in a e-group handles every e-link toward its FIFO buffer

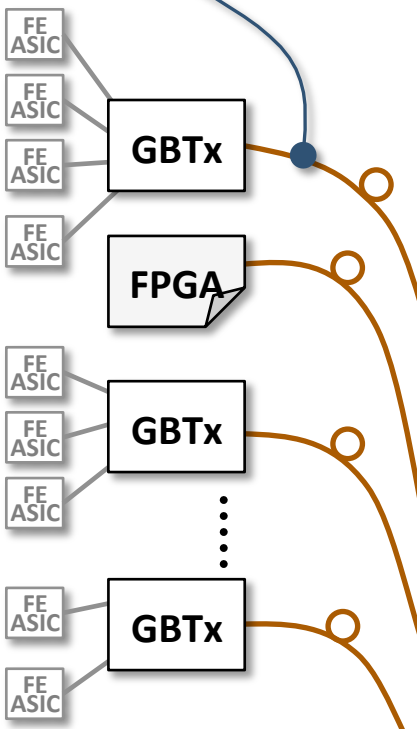
FELIX Flow: To-Host

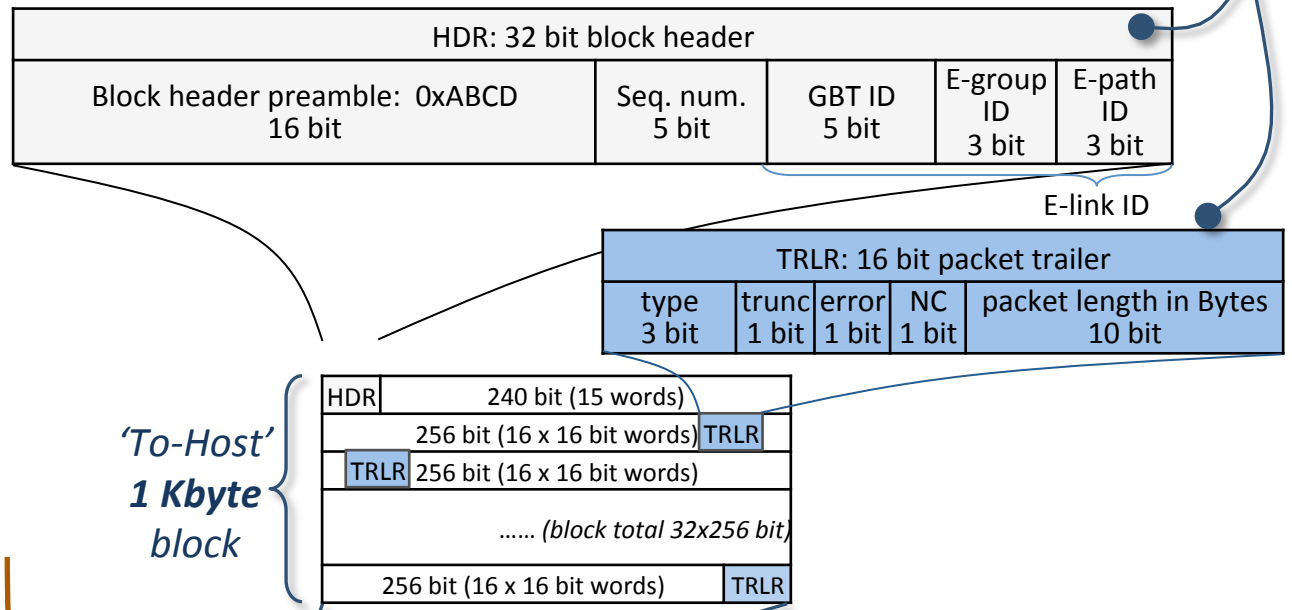


FELIX Flow: To-Host

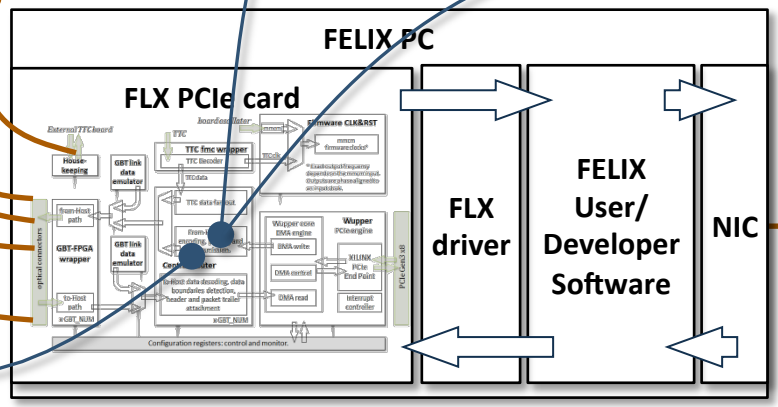
Attached by FLX card firmware, used by FLX software

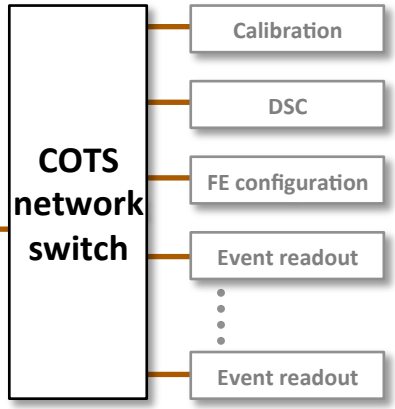
To-Host serial data stream





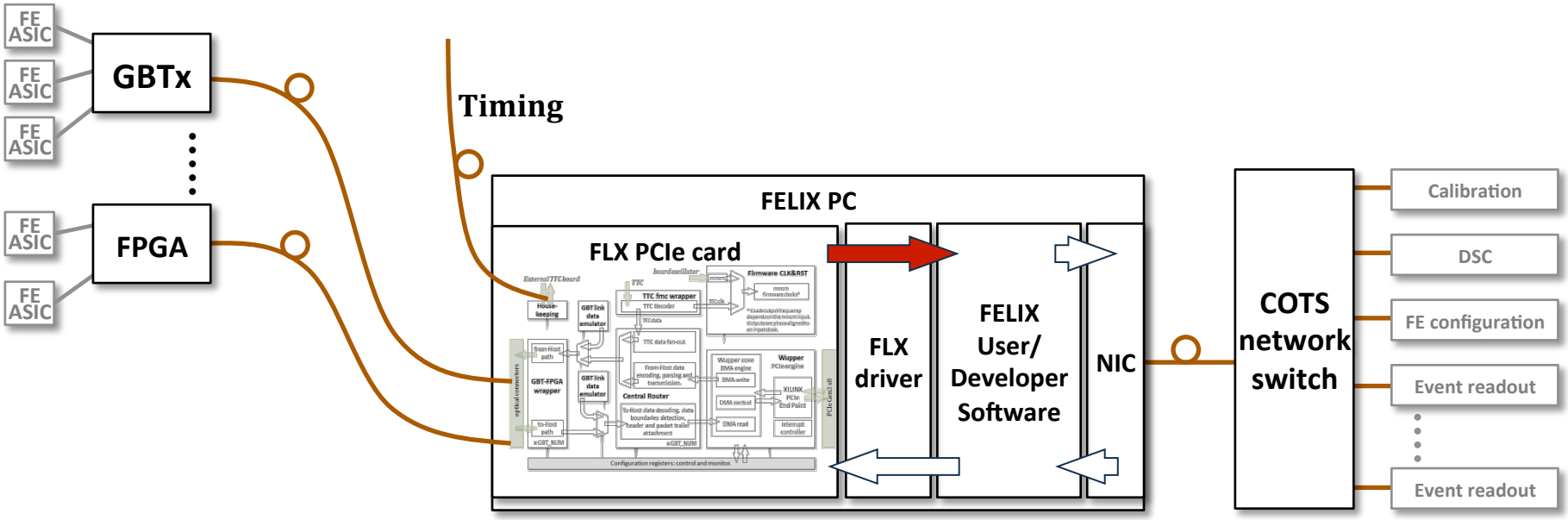
TTC fiber



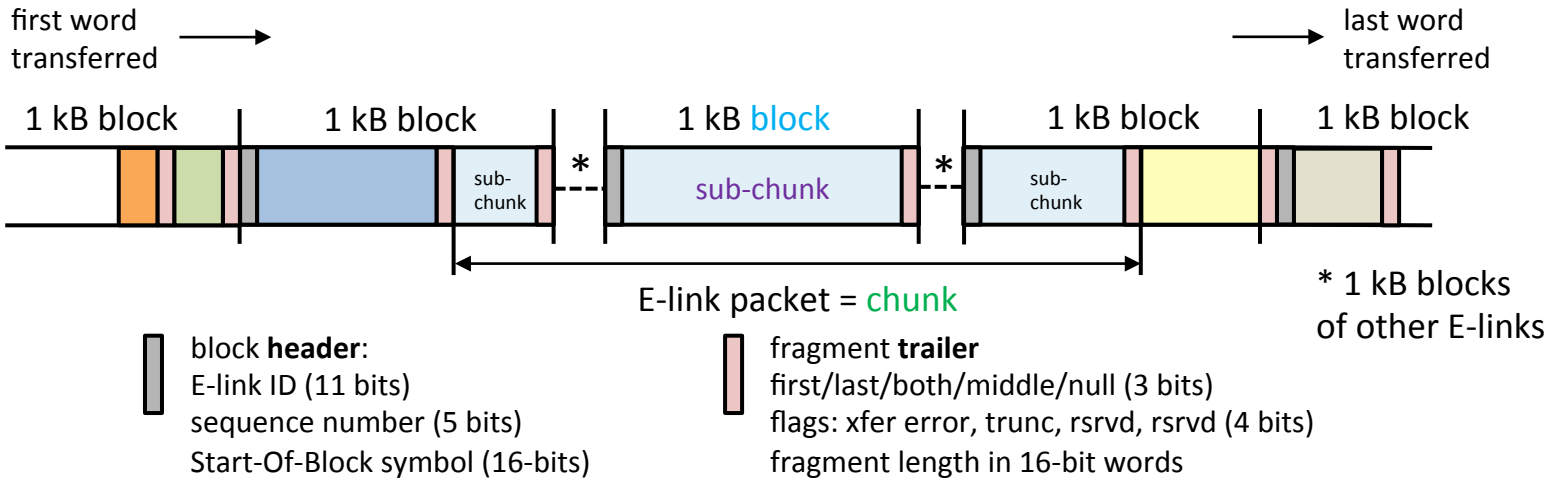


Routes and decodes data streams according to FELIX configuration

FELIX Flow: To-Host

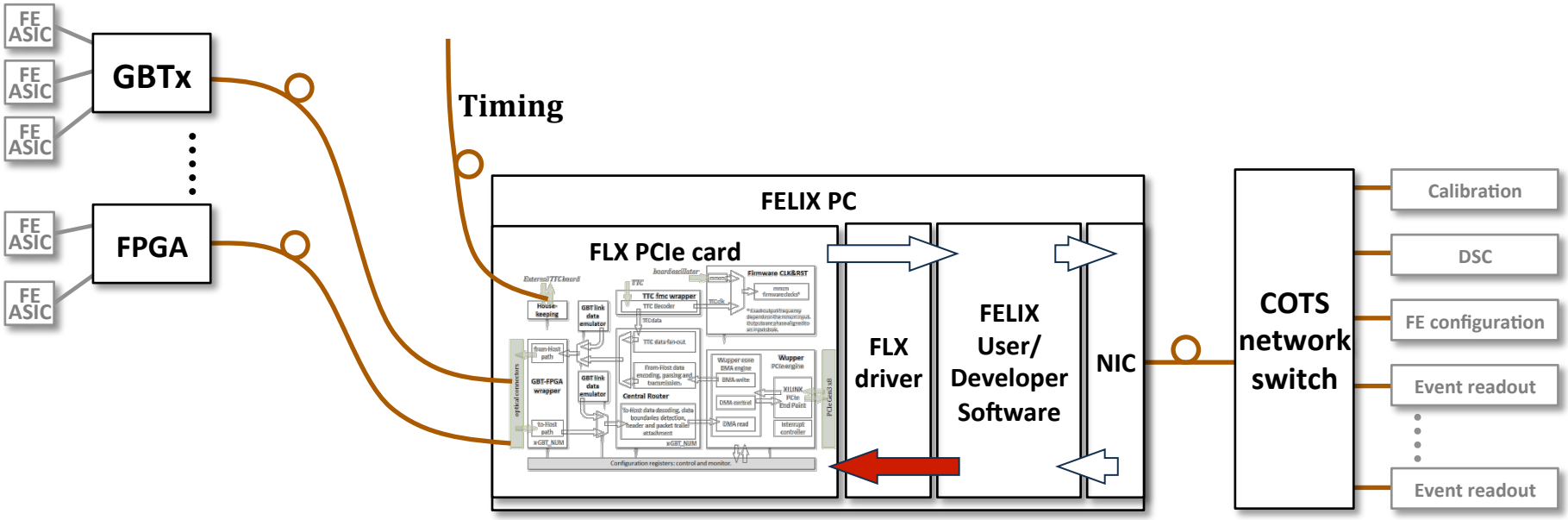


High throughput detector readout

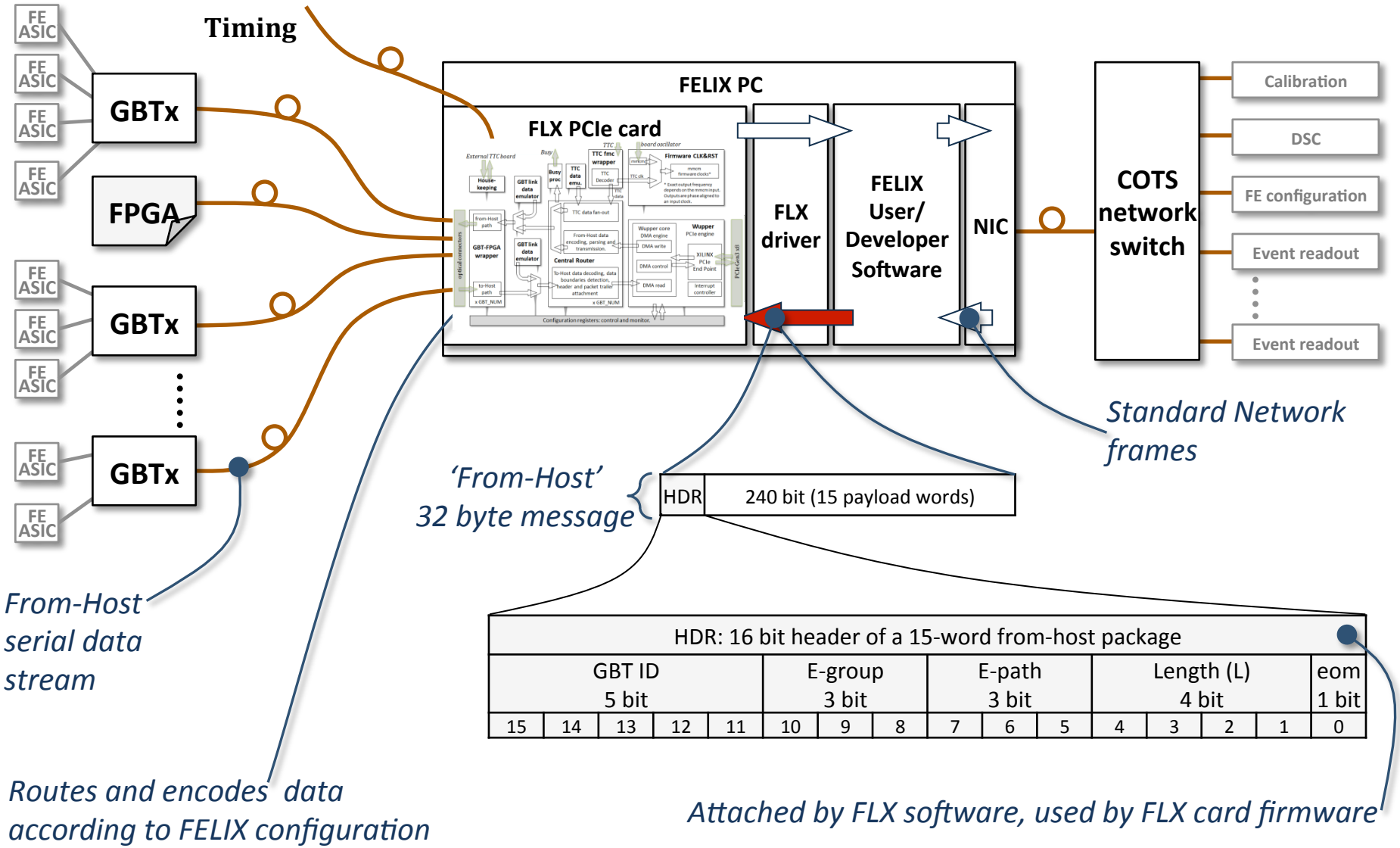


- For each E-link fixed size blocks (1 kByte) are filled with received data
- Each block has a 4-Byte **header**: E-link ID, sequence number, start of block symbol
- Data packets ("chunks") received can be of arbitrary length and are subdivided (typically after e.g. 8B/10B decoding) in sub-chunks as needed to fill the blocks
- Each sub-chunk has a **trailer** with information on its length and type
- In case of low data rates time-outs will cause incompletely filled blocks to be padded and sent (the last sub-chunk in this block is then of type "null")
- Blocks are transferred using continuous DMA (Direct Memory Access) into a large (e.g. 4 GByte) circular buffer in host PC memory
- The buffer consists of contiguous memory allocated by a dedicated driver
- The DMA is controlled with two pointers, a write pointer maintained by the DMA controller in the FPGA, and a read pointer maintained by the FELIX application

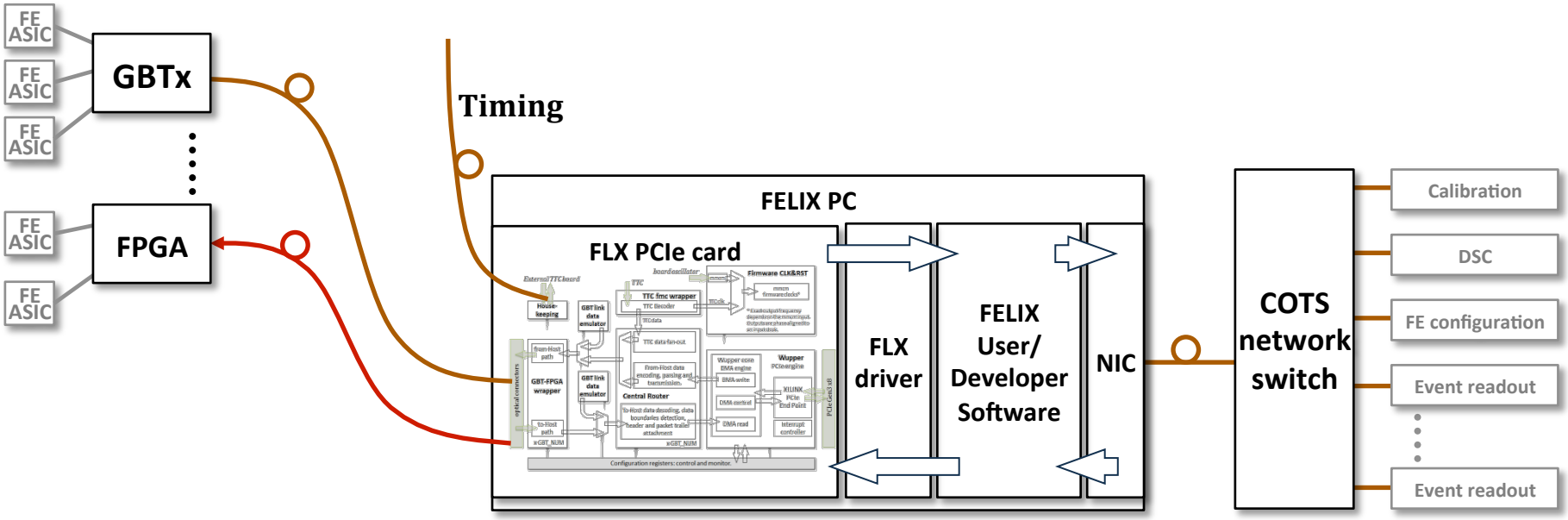
FELIX Flow: From-Host



FELIX Data Flow: From-Host

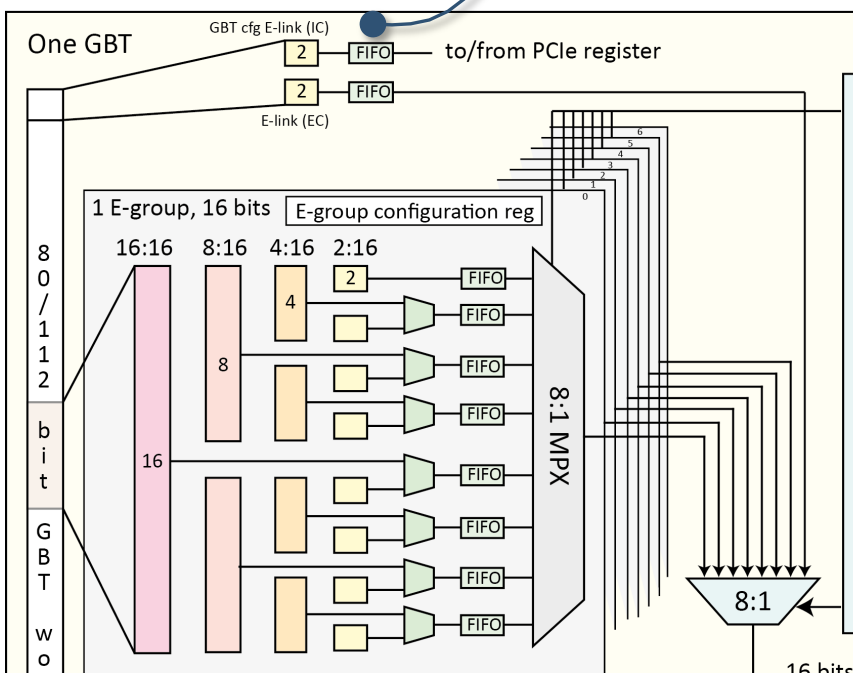


FELIX Flow: To-FrontEnd



The GBT protocol (simplified)

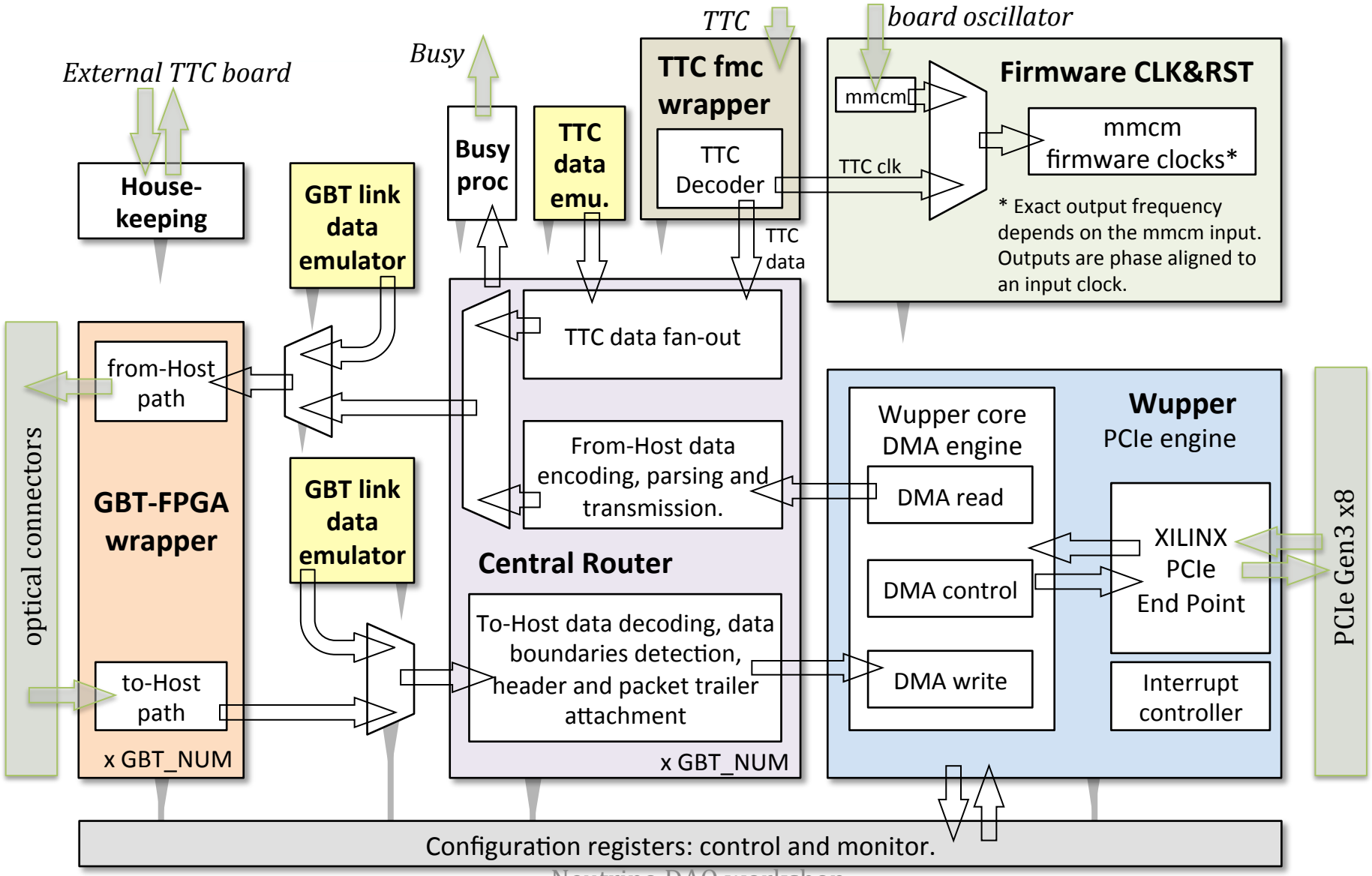
- The *GBT protocol* is a data serializier (with line encoding)
- The *GBT frame* is 120 bits @ 40 MHz
  - 4 bits are reserved for a header (data / no data)
  - **4 bits are always available for the To FrontEnd direction**



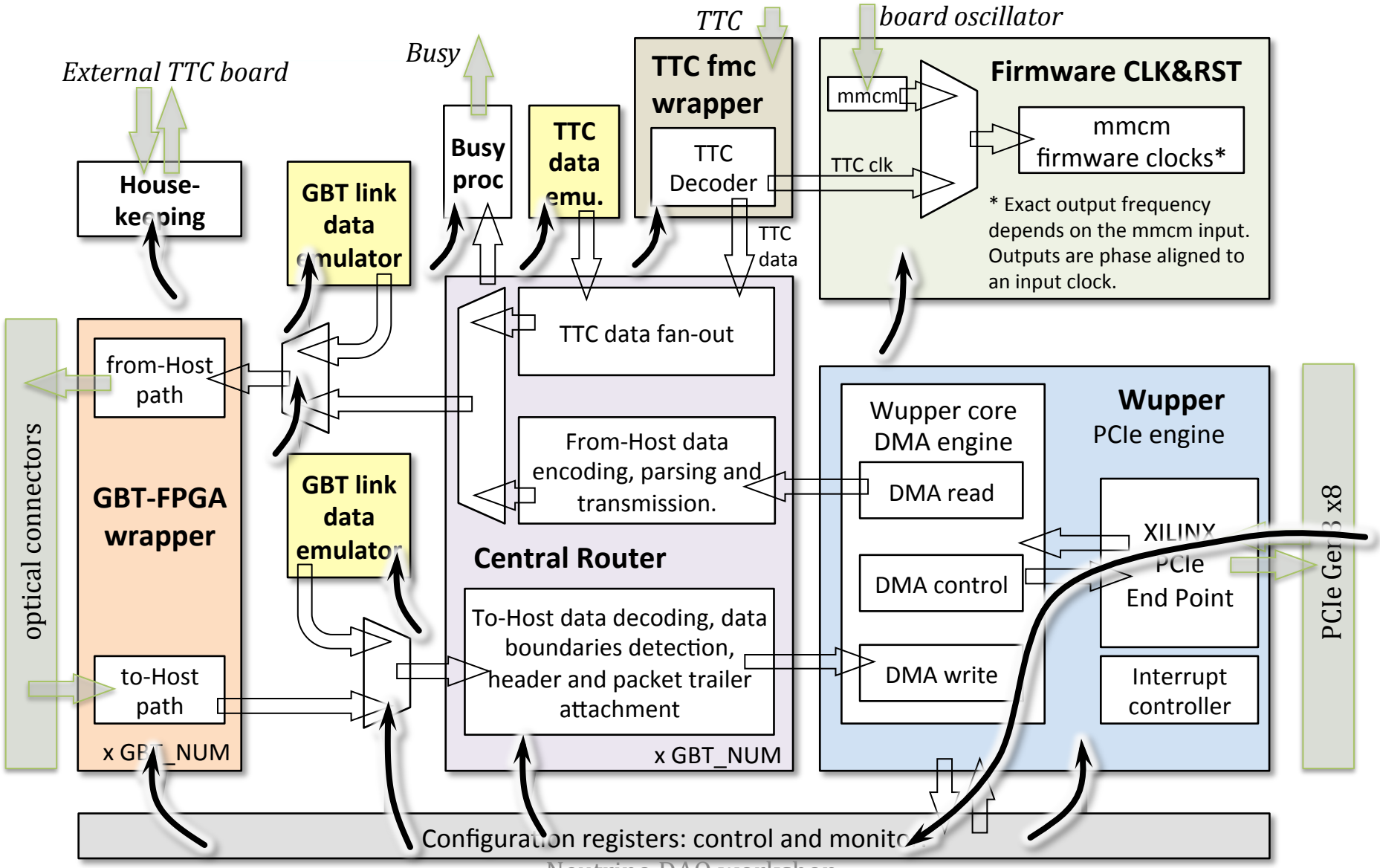
- *IC (Internal Control)* bits are used to configure the GBTx chip itself
- *EC (External Control)* bits are used to configure devices external to the GBTx

FELIX… halftime

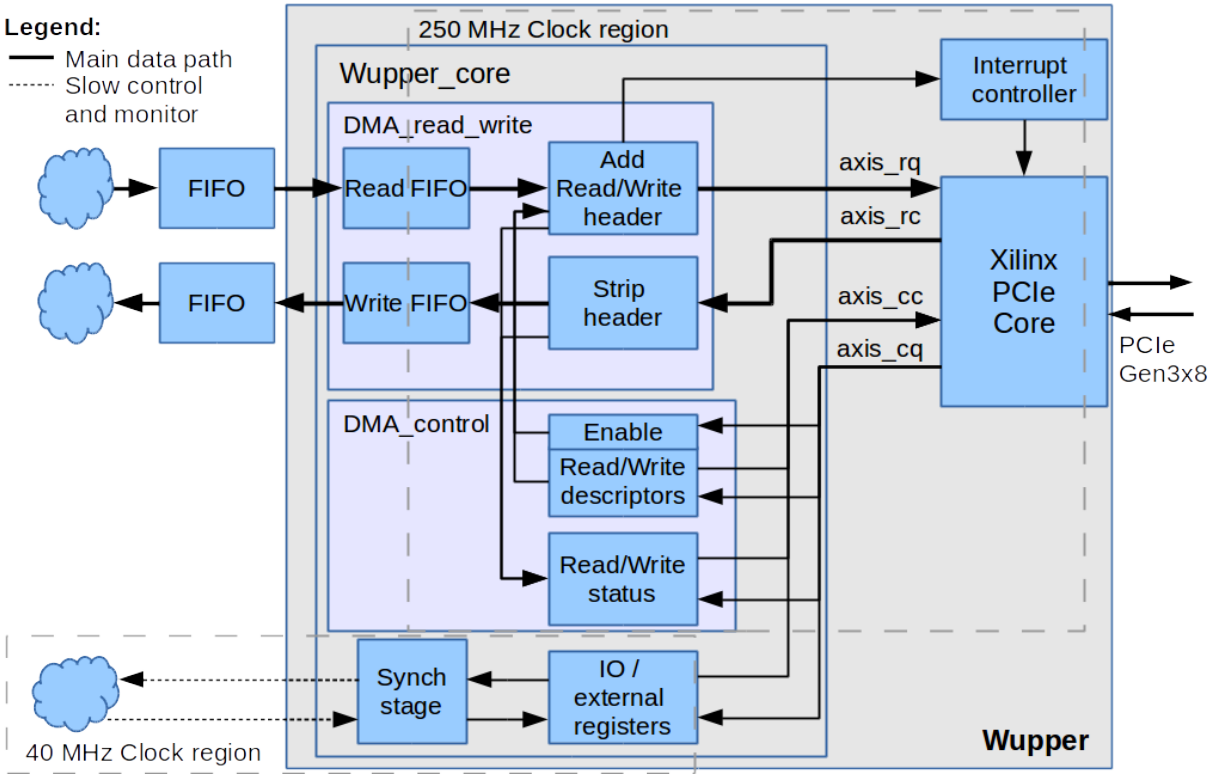
FLX PCIe card firmware: zoom-in



FLX PCIe card firmware: configuration

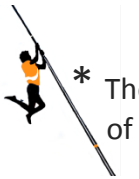


Wupper*: PCIe Engine for FELIX



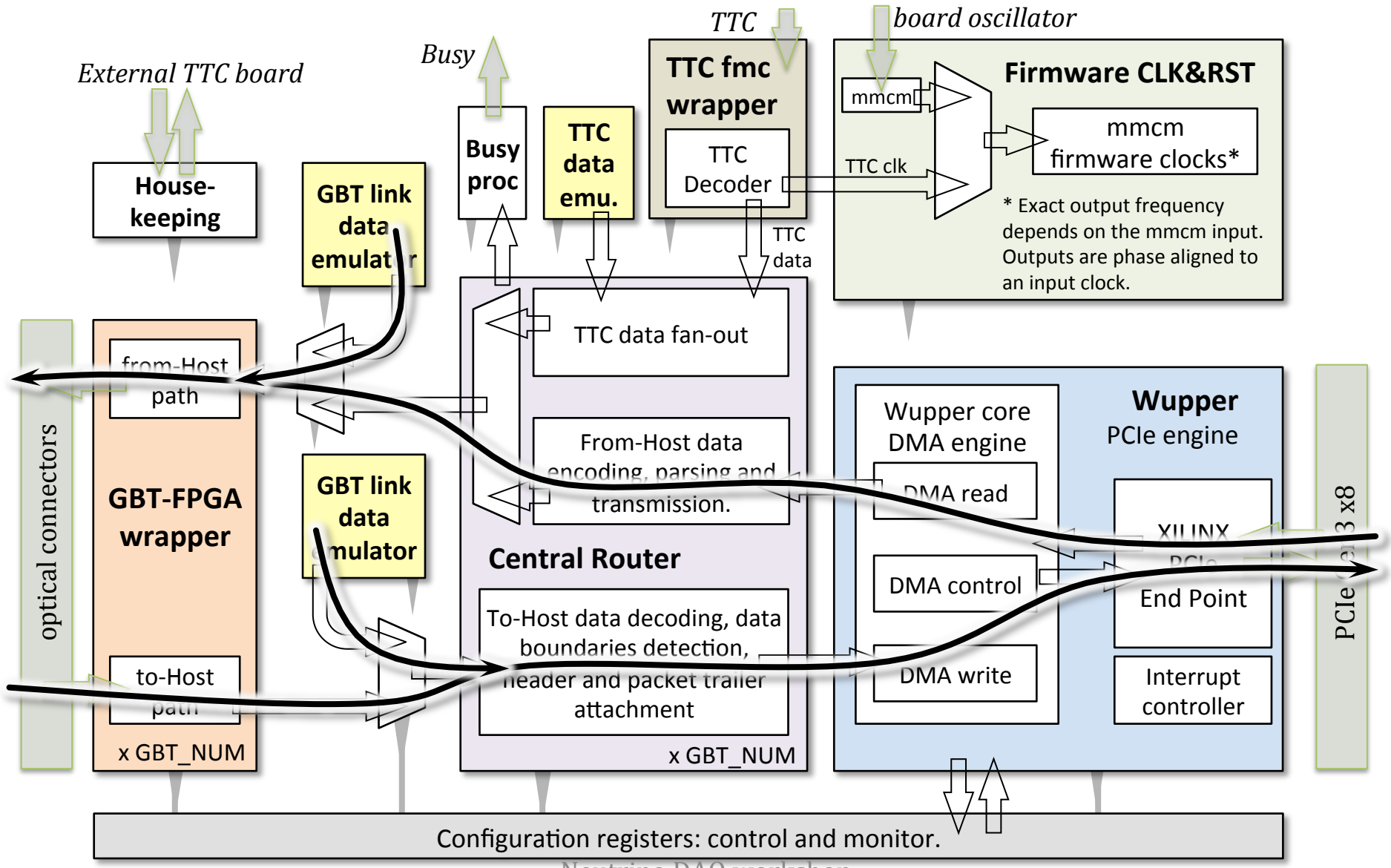
- Developed for use in FELIX
- Published as OpenSource (LGPL) on OpenCores
  http://opencores.org/project,virtex7_pcie_dma

- PCIe Engine with DMA interface to the Xilinx Virtex-7 PCIe Gen3 Integrated Block for PCI Express (PG023)
- Xilinx AXI (ARM AMBA) Stream Interface (UG761)
- MSI-X compatible interrupt controller
- Applications access the engine via simple FIFOs
- Register map for programmed I/O synchronized to a lower clock speed

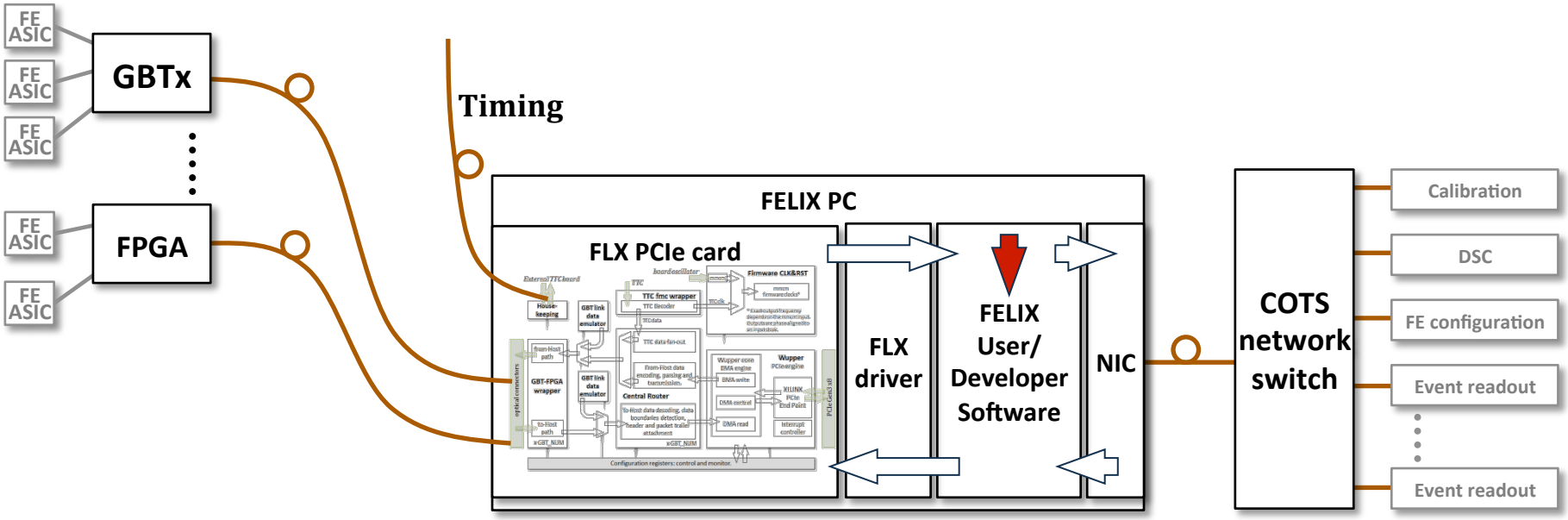


* The person performing the act of bongelwuppen, the version from the Dutch province of Groningen of the "famous" Frisian sport Fierljeppen (canal pole vaulting) https://www.youtube.com/watch?v=YP32iWoqjnQ

FLX PCIe card firmware: data path



FELIX Flow: Software



FELIX software: control, monitor

FELIX low-level tools: *FLXtools*

FELIX firmware configuration through register map	→ configure the FLX card via the internal registers mapped into the host memory.
FELIX 'housekeeping' functions	→ Control FLX external devices (e.g. clock chips) via I2C and SPI.
System test and debugging	→ Performance testing, data flow control and verification.
Complete user toolset	→ collection of tools to control, monitor, configure different FLX cards.

```
daqmustud@gimone:$ ./flip-info
General information
------------------
Board ID:         8261718
Card ID:          VC-709
FW version date: 26/8/15 17:18
SVN version:      1966

Interrupts, descriptors & channels
----------------------------------
Number of interrupts:  8
Number of descriptors: 8
Number of channels:    4

Internal PLL Lock : Yes
CDCE Lock         : Yes
CXP1 @ 240 MHz    : NO !!
CXP2 @ 240 MHz    : NO !!

FMC ADN TTC Status: ON
```

various card info (FLXtools)

```
$ flx-dump-blocks -n 100 -f output.blocks
$ flx-config
$ flx-dma-test
$ flx-eeprom
$ flx-i2c
$ flx-info
$ flx-init
$ flx-irq-test
$ flx-spi
$ flx-reset
$ flx-throughput
```

FLXtools set

```
daqmustud@gimone:$ ./flip-i2c list -d 1
Card model HTG 710
Switch I2C address: 0x70

List of available devices:
Device              Model                    Switch port      Address
=============================================================================
CLOCK_RAM           ICS8N4Q001L IDT              0:0             0x6e
CLOCK_SYS           ICS8N4Q001L IDT              1:0             0x6e
CLOCK_CXP1          IDT 8N3Q001                  2:0             0x6e
CLOCK_CXP2          IDT 8N3Q001                  3:0             0x6e
FMC_ADN             ADN2814 (on TTCfx FMC)       4:0             0x40
FMC_TEMP_SENSOR     TC74 (on CRORC TEST FMC)     4:0             0x4a
CXP1_TX             AFBR-83PDZ
CXP1_RX             AFBR-83PDZ
CXP2_TX             AFBR-83PDZ
CXP2_RX             AFBR-83PDZ
DDR3-1              SRAM-MT16JTF25664HZ
```

HW peripheral monitor (FLXtools)

```
daqmustud@gimone:$ ./flip-info GBT -d 1

GBT CHANNEL ALIGNMENT STATUS
            0     1     2     3     4     5     6     7
          -----------------------------------------------
Aligned |  Yes   Yes   Yes   Yes   Yes   Yes   Yes   Yes
```

various card statuses (FLXtools)

FELIX software: support

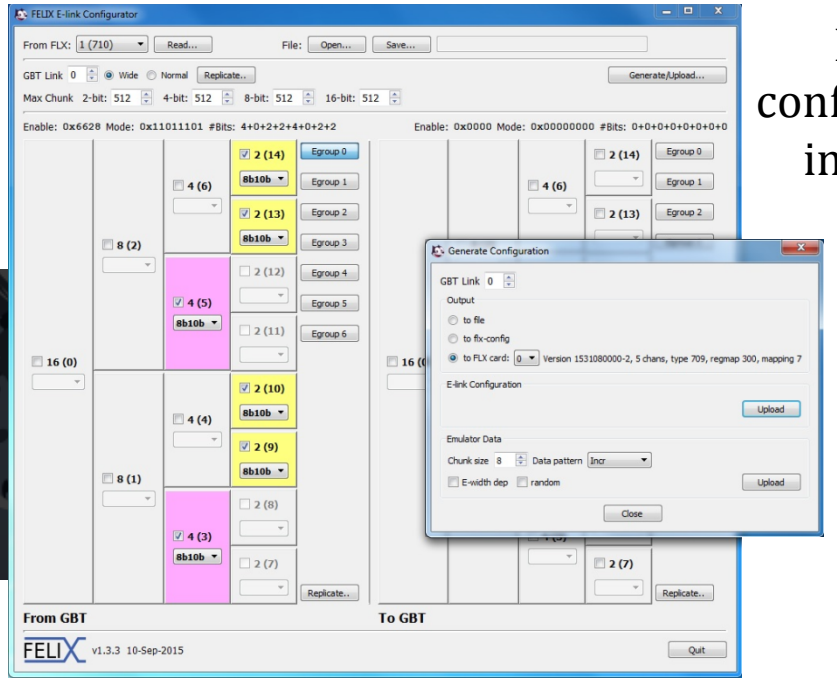
FELIX support software *(WupperCodeGen, fel, FELIX E-link configurator)*

Automatic generation of a register map	→ Automation software (Jinja based) to handle register map consistently between firmware (HDL), software (C headers), and documentation (Latex) and beyond (OKS, ...).
Firmware development tools	→ Configuration of the internal data emulators to verify the functionality 'off-line'.
Firmware test and debugging	→ Communication with FLX card, data read and analysis.

```
daqmustud@gimone:$ ./fel -u 2048 -f 64 -t 8 -s
FelixCardBuffer 1

BufSize 4194304
2 8 0.000068
   0 a0  1 cd ab aa  0 89  0  0 bb aa  4  0  1
  16  4  5  6  7  8  9  a  b  c  d  e  f 10 11
  32 14 15 16 17 18 19 1a 1b 1c 1d 1e 1f 20 21
  48 24 25 26 27 28 29 2a 2b 2c 2d 2e 2f 30 31
```

loopback via GBT links of data from internal generator ('fel' tool output)



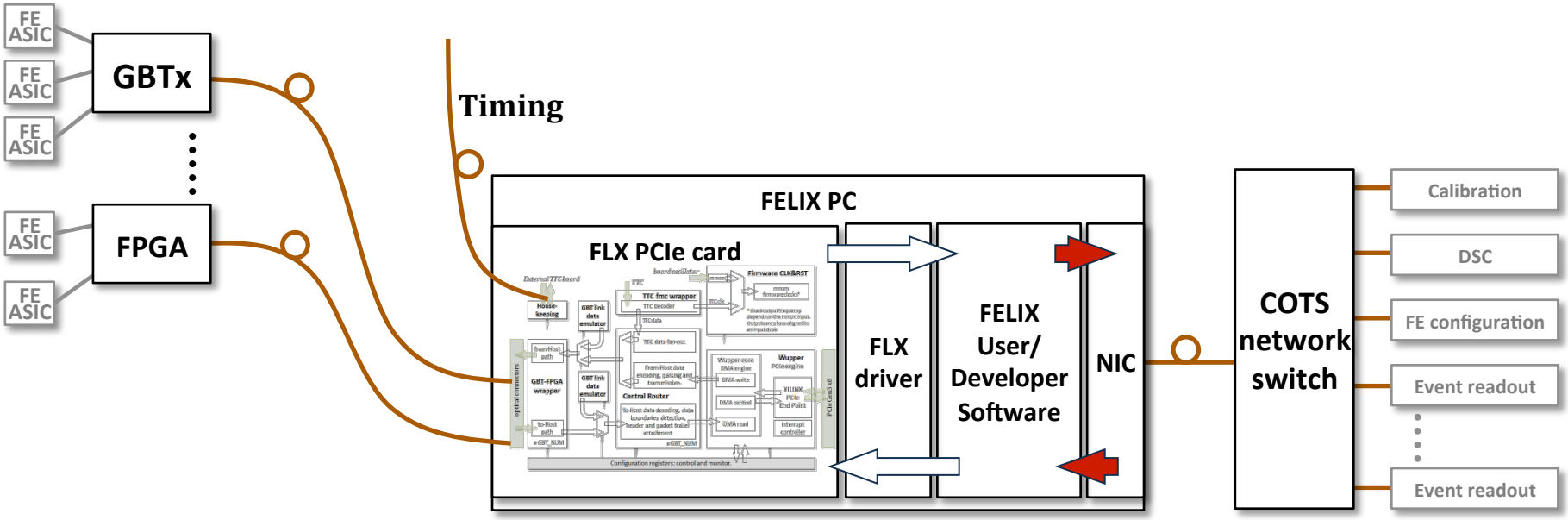
FELIX configuration interface

FELIX software: data path

FELIX application: *FELIX Core Application*

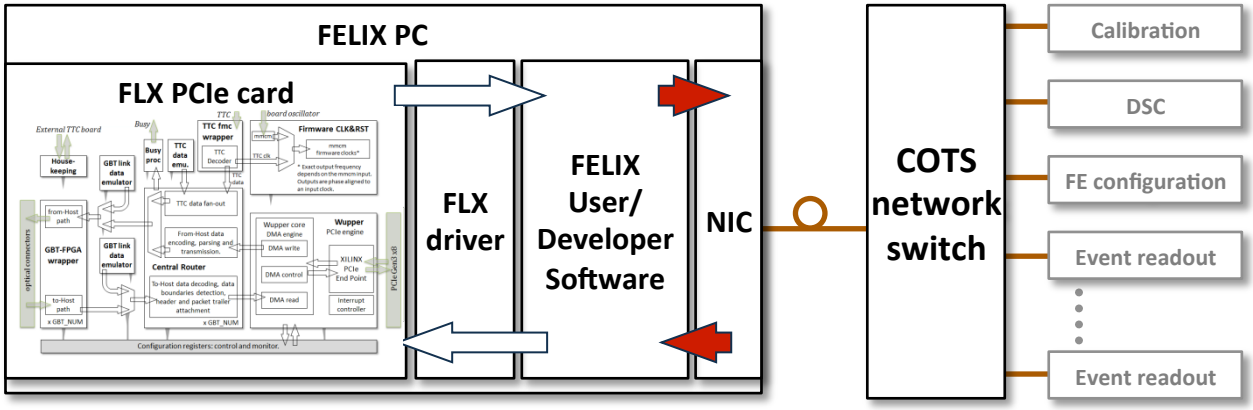
Read blocks from file descriptors	→ DMA transfers and reads 1KByte blocks of data, encoded for the transfer over PCIe. Every block contains accumulated data from one E-link.
Block decoding	→ Decode into variable sized packets for transmission over network.
Statistics	→ Count processed blocks, transfer rates, etc.
Extract Meta Information	→ Meta-information, for example event ID, is extracted and matched against a routing table.
Route	→ Route.
Load Balancing, Error Handling	→ Distribute load among multiple systems, handle automatic failover in case of system failures.
Send to network	→ Asynchronous message service 'netio' for data exchange with network hosts.

FELIX Flow: Networking



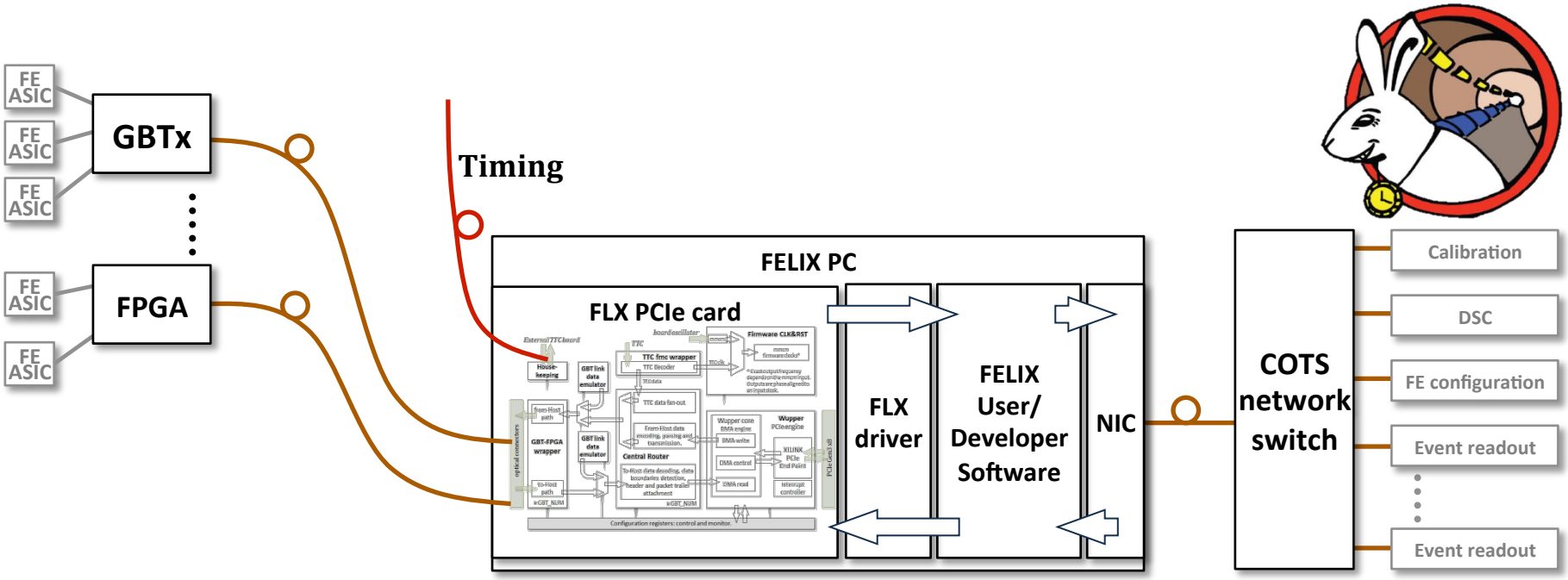
FELIX software: Networking

Network-side packet handling



- Networking is implemented in a FELIX library called "netio", which has different backends to support, Ethernet, InfiniBand and potentially more in the future.
- For Ethernet, FELIX uses TCP for communication with network endpoints.
- Using TCP has the advantage of guaranteeing no packets loss.
  InfiniBand RDMA connections guarantee reliable delivery.
- Note that although TCP sockets are bi-directional, GBT streams are not;
  a TCP socket is associated with only one direction of a GBT transfer.
- Streams may be cloned to several network end-points.
  For clones marked with QoS as less-than-best effort, some packets may not be sent, i.e. “sampling”.
- Broadcasts / multicasts to E-links are supported.

FELIX Flow: Timing



White Rabbit provides **sub-nanosecond accuracy** and picoseconds precision of **synchronization** for large distributed systems. It also allows for deterministic and reliable data delivery.

White Rabbit allows you to **precision time-tag measured data** and lets you trigger data taking in large installations while at the same time using the same **network to transmit data**.

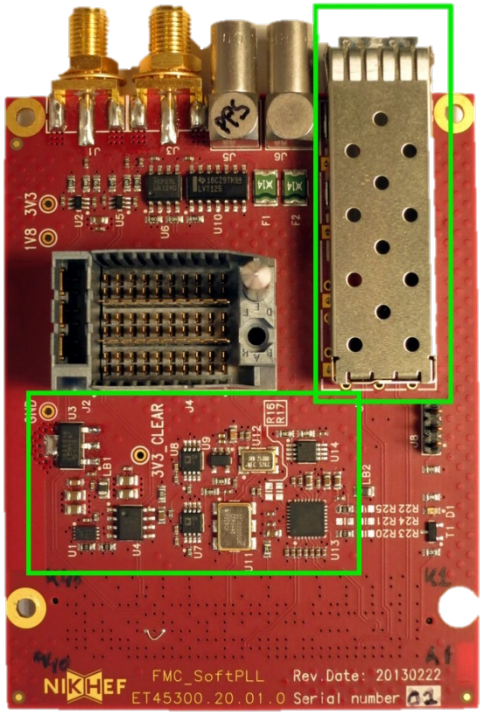
http://www.ohwr.org/projects/white-rabbit/wiki

http://www.ohwr.org/projects/white-rabbit/wiki/WRUsers

FELIX hardware: White Rabbit integration

RabbitFX: a white Rabbit Mezzanine for *FELIX*

TTCfx



→ For the FELIX development in ATLAS a TTC (Timing, Trigger and Control) FMC mezzanine was developed by the collaboration in order to interface commercial (development) cards to the TTC system

→ Nikhef has also expertise in the area of the White Rabbit timing protocol

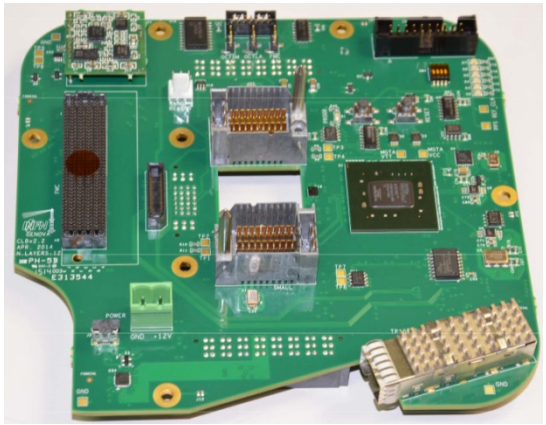
→ Already developed a simple FMC for another (Neutrino) experiment demonstration

→ Develop a new small FMC equipped with the hardware necessary to make a platform “white rabbit ready”

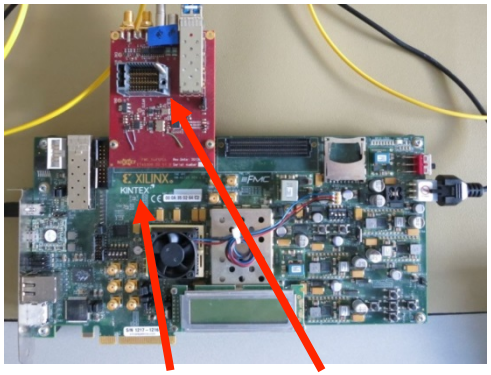
→ SFP input (optional)

→ Basically a DAC (Digital To Analog Converter) and a VCXO (Voltage Controlled Xtal Oscillator

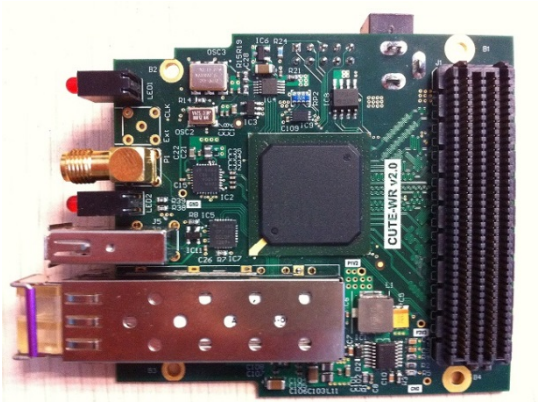
Other White Rabbit nodes examples



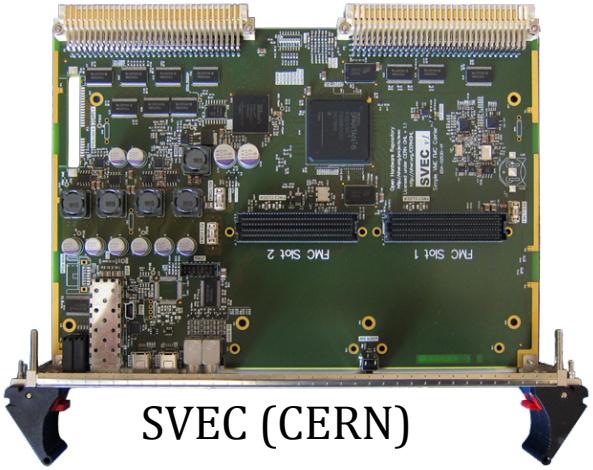
Central Logic Board (KM3NeT)



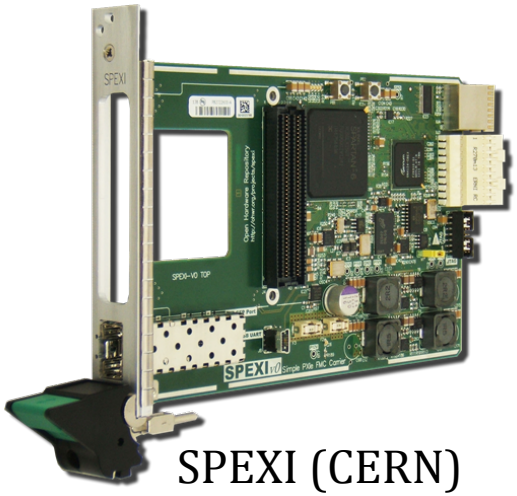
KC705+SoftPLL
(KM3NeT test)



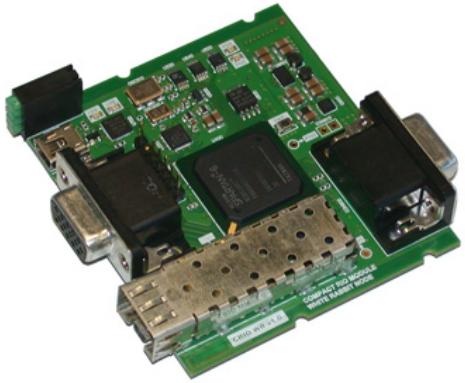
CUTE-WR (LHAASO)



SVEC (CERN)

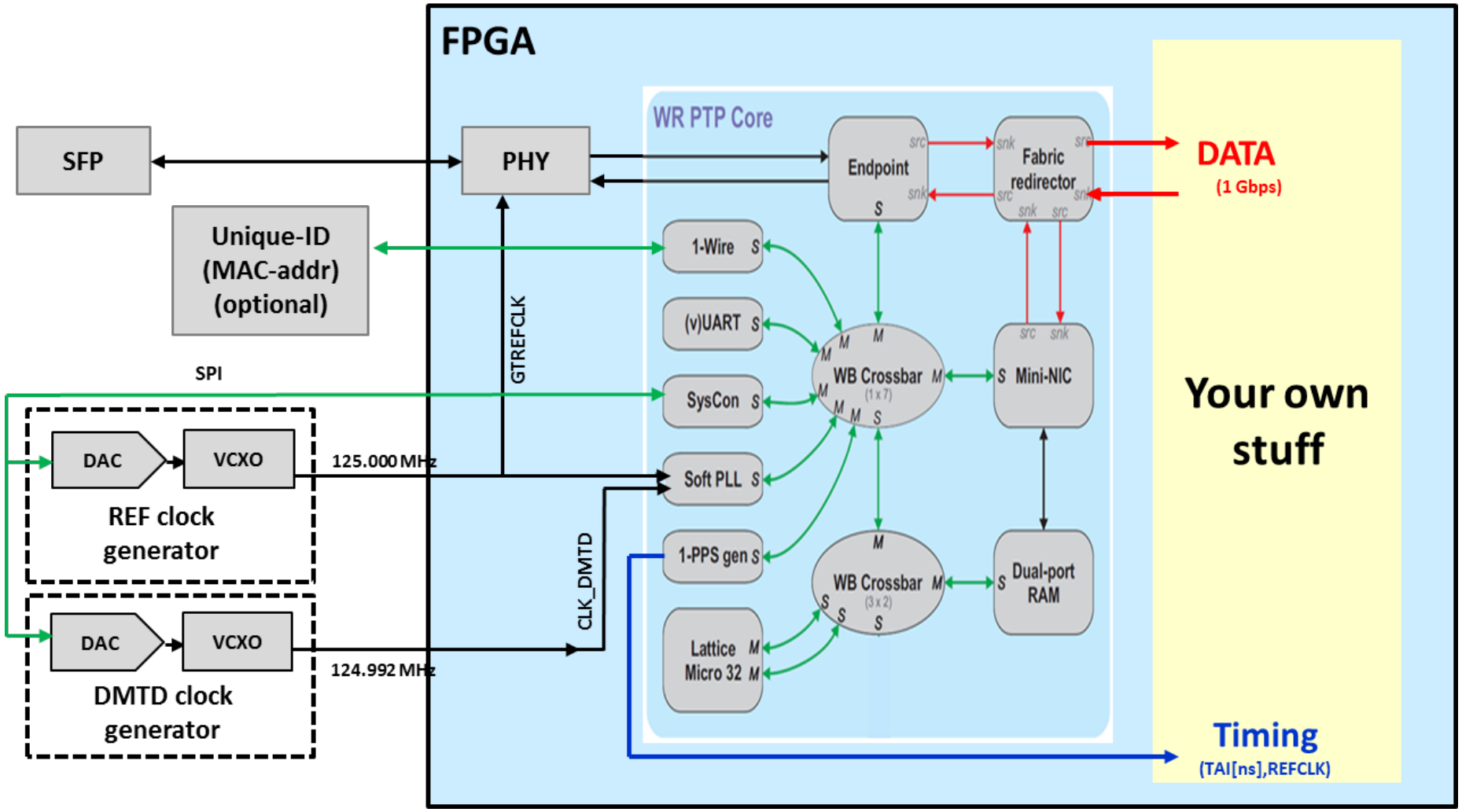


SPEXI (CERN)



CRIO-WR (CERN)

FELIX firmware: White Rabbit core



http://www.ohwr.org/projects/white-rabbit/wiki/WRReferenceDesign

the White Rabbit collaboration

Picture taken @ 4th WR workshop (2011), GSI Darmstadt

Picture taken @ 9th WR workshop (2016) Nikhef Amsterdam

ARKUS

KM3NeT
Opens a new window on our universe

cta
cherenkov telescope array

MIKES
METROLOGIA

LHAASO
大型高海拔空气簇射观测站

UNIVERSITY OF AMSTERDAM

VU VRIJE UNIVERSITEIT AMSTERDAM

DESY
TUNKA

VSL

FELIX development status

- FLX-709 (Xilinx VC-709) and its software reached a development status sufficient to be distributed to ATLAS Sub-Detectors Front End developers.
- Tests with the BNL FLX-711 successfully accomplished, a second revision with minor bug fixes is planned for the coming months.
- FLXtools for control and monitor of FELIX are in the advanced development stage.
- FELIX application is the development stage. Simple data transfers over the network are possible.
- Networking layer: Initial work has been done and simple communication works. Current focus is on improving interface, performance, and looking into features like fault-tolerance.
- Ongoing effort to increase overall system reliability.
- Ongoing effort to increase the number of input channels supported.
- Ongoing effort to implement more functionalities: control of GBTx ASIC via its IC port, HDLC encoding of E-link data to/from the SCA ASIC (for DCS), routing of data From-Host.
- Final Design Review planned for October 2016.
- FELIX is not (yet) a “product”!

FELIX reference documents

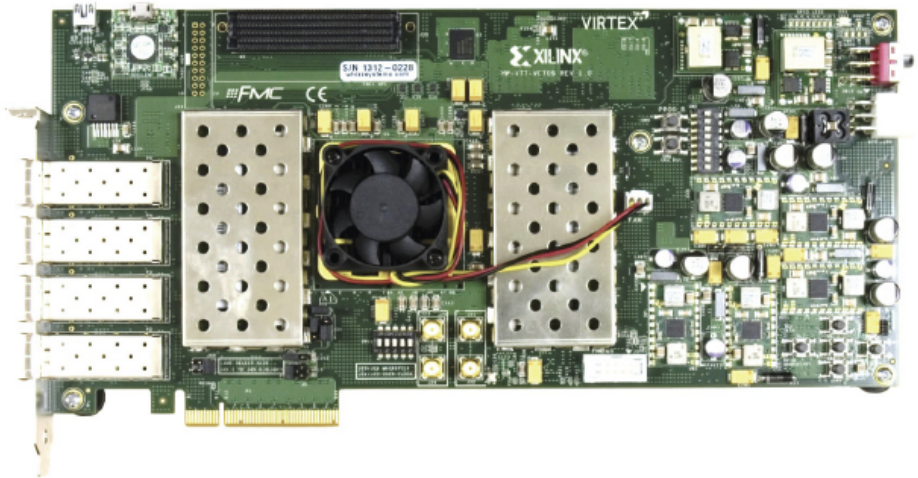
- FELIX Preliminary Design Review: https://indico.cern.ch/event/395378/
- DEBS2015 paper: http://cds.cern.ch/record/2014753
- TWEPP2015 paper [*JINST 11 C01055]*: http://iopscience.iop.org/article/10.1088/1748-0221/11/01/C01055/pdf
- http://www.ohwr.org/projects/white-rabbit/wiki
- http://www.ohwr.org/projects/white-rabbit/wiki/WRUsers

backup slides

FELIX development cards: FLX

FLX-710 (FELIX)

- HiTech Global HTG-710
- Virtex-7 X690T
- PCIe Gen 3 x 8 lanes
- 2x12 bidir CXP connectors
- FMC connector



FLX-709 (MiniFELIX)

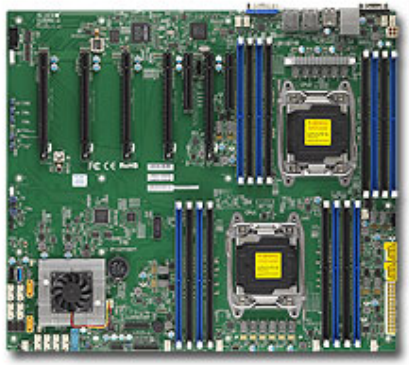
- Subset of the full FELIX functionality, intended for FE development support
- Xilinx VC-709
- Virtex-7 X690T
- PCIe Gen 3 x 8 lanes
- 4 SFP+ connectors, *card comes with optical transceivers*
- FMC connector

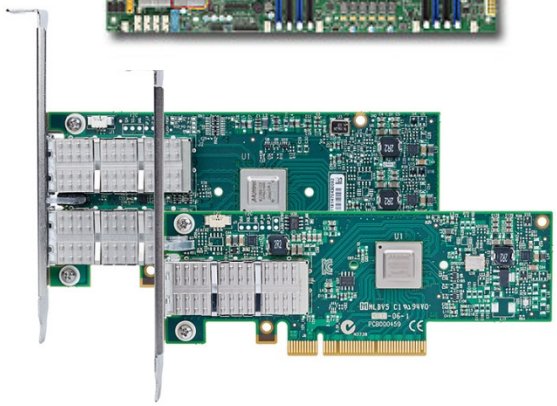
TTCfx

- Custom FMC accepting TTC input
- Outputs TTC clock and CH A-B info
- V1: ADN2814 + CDCE62005
- V2: ADN2814 + Si5338

FELIX PC: motherboard and NIC

OS: SLC6





Supermicro motherboards, e.g.:

SuperMicro X9SRL-F (Nikfef)

- 1x Ivy Bridge CPU, 6 cores
- 6x PCIe Gen-3 slots
- 16 GB DDR3 Memory

http://www.supermicro.com/products/motherboard/Xeon/C600/X9SRL-F.cfm

SuperMicro X10DRG-Q (CERN)

- 2x Haswell CPU, up to 10 cores
- 6x PCIe Gen-3 slots
- 64 GB DDR4 Memory

http://supermicro.com/products/motherboard/Xeon/C600/X10DRG-Q.cfm

Mellanox ConnectX-3 VPI

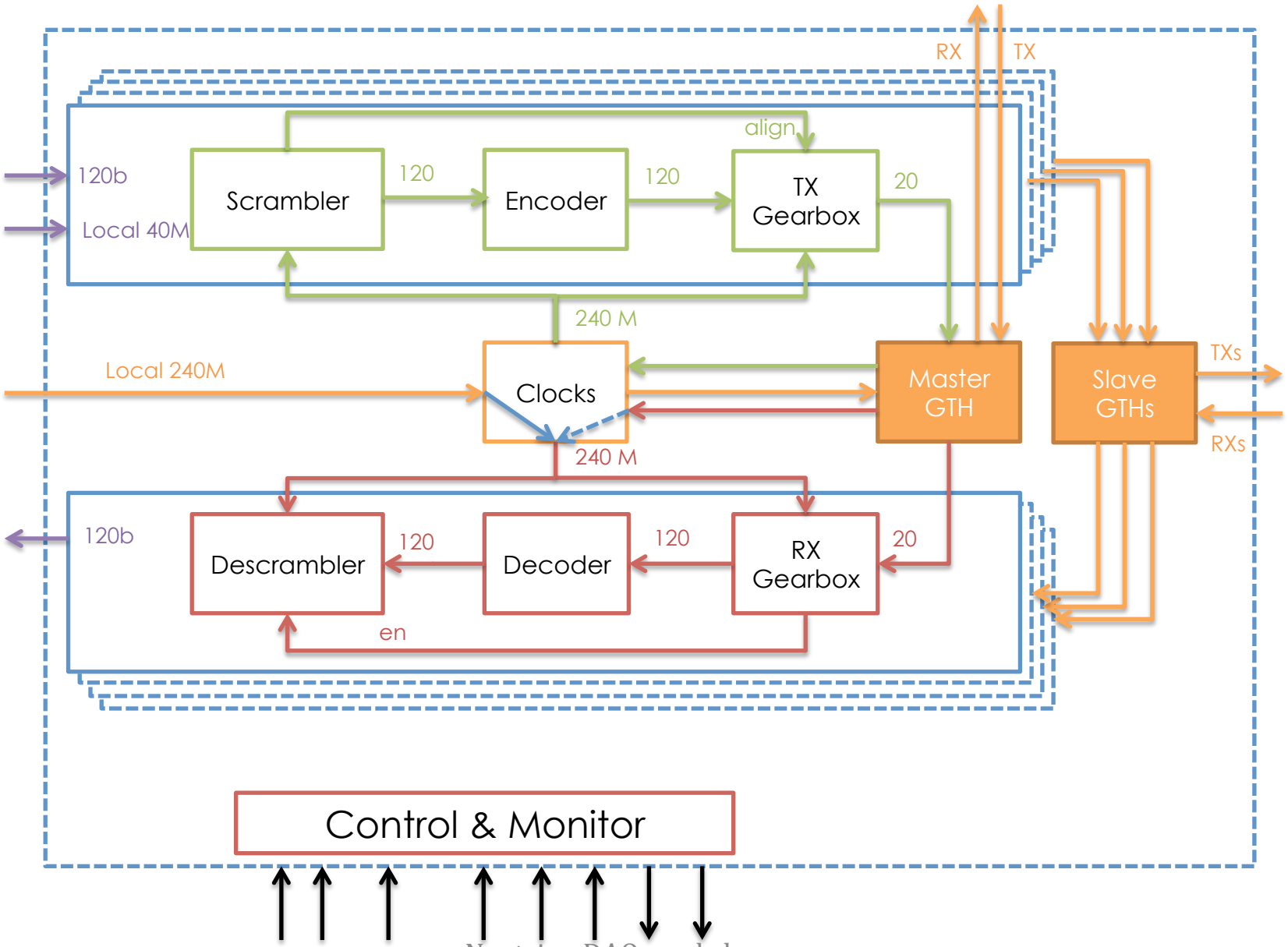
- FDR/QDR Infiniband
- 2x10/40 GbE

http://www.mellanox.com/page/products_dyn?product_family=119&mtag=connectx_3_vpi

FELIX To-Host data boundary format

FELIX Chunk trailer format					04-Jul-14											
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Type	**Type**			**mod1**	**mod0**											
first	1	1	0	*rsrvd*	*rsrvd*	length: 2^11 16-bit words → 2^(11+1) bytes = 4KB validity refers to this chunk only, i.e. CRC error, or bad 8b/10b encoding…										
middle	1	0	0	*rsrvd*	*rsrvd*											
last, even	1	0	1	Valid=1	Trunc=1											
both, even	1	1	1													
last, odd	0	0	1													
both, odd	0	1	1													
control	0	0	0	0	null=0	length: 2^11 16-bit words, excl this word; length=0 means only this word										
				0	err=1	error flags, e.g. GBT errors, loss of link sync, etc.										
				1	*rsrvd=0*											
				1	*rsrvd=1*											
reserved	0	1	0													

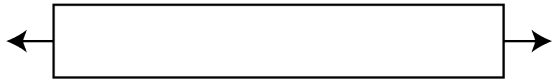
FELIX GBT-FPGA



FELIX E-link data packet format examples

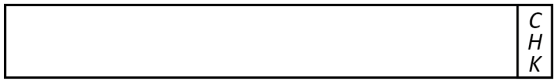
Single stream E-link:
Data can be routed to only a single end-point.

stream



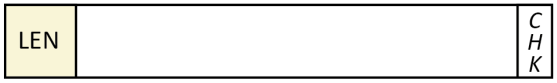
No packet boundaries, i.e. a TCP stream.
The single end-point must parse out packets.

fixed length



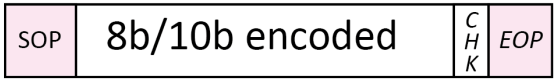
NOT recommended.
Packet boundaries lost if bits lost on an E-link.
Fixed length packets whose framing is periodically aligned by strings of zeroes will also be implemented.

variable length



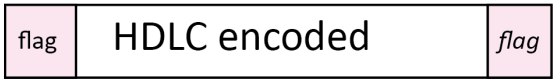
NOT recommended.
Packet boundaries lost if bits lost on an E-link.
CHK: an optional check-sum

framed



Guarantees packet boundaries, e.g. event boundaries.
8b/10b comma, SOP and EOP symbols are not forwarded.
CHK: an optional check-sum. EOP is optional.

framed HDLC



Guarantees packet boundaries, e.g. event boundaries.
Data outside frames are not forwarded.
Used by Slow Control Adapter ASIC.

Full mode: 8b/10b, either with or without stream-id

LL_SingleStreamE-link_V01

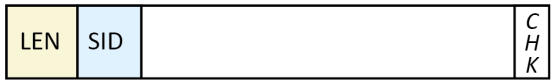
FELIX E-link data packet format examples

Multiple stream E-link:
Packets with different Stream IDs can be routed to different end-points.

fixed length

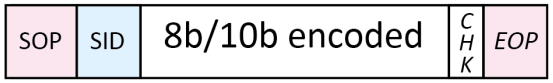
NOT recommended.
Packet boundaries lost if bits lost on an E-link.
Fixed length packets whose framing is periodically aligned by strings of zeroes will also be implemented.

variable length



NOT recommended.
Packet boundaries lost if bits lost on an E-link.
CHK: an optional check-sum

framed



Guarantees packet boundaries, e.g. event boundaries.
8b/10b comma, SOP and EOP symbols are not forwarded.
The stream-ID is 8b/10b encoded.
CHK: an optional check-sum. EOP is optional.

Full mode: 8b/10b, either with or without stream-id

LL_MultiStreamE-link_V01

FELIX flx-tools

Collection of tools to control, monitor, configure different FLX cards.

- flx-config
- flx-dma-test
- flx-dump-blocks
- flx-eeprom
- flx-i2c
- flx-info
- flx-init
- flx-irq-test
- flx-spi
- flx-reset
- flx-throughput

flx-config: access to configuration registers of the card:

```
$ flx-config list
$ flx-config set GBT_EMU_ENABLE=1
$ flx-config store my_config.dat
$ flx-config load my_config.dat
```

flx-init: Initializes an FLX device. No output = everything is fine.

flx-info: Display various status information of an FLX device

```
$ flx-info GBT
$ flx-info CXP
```

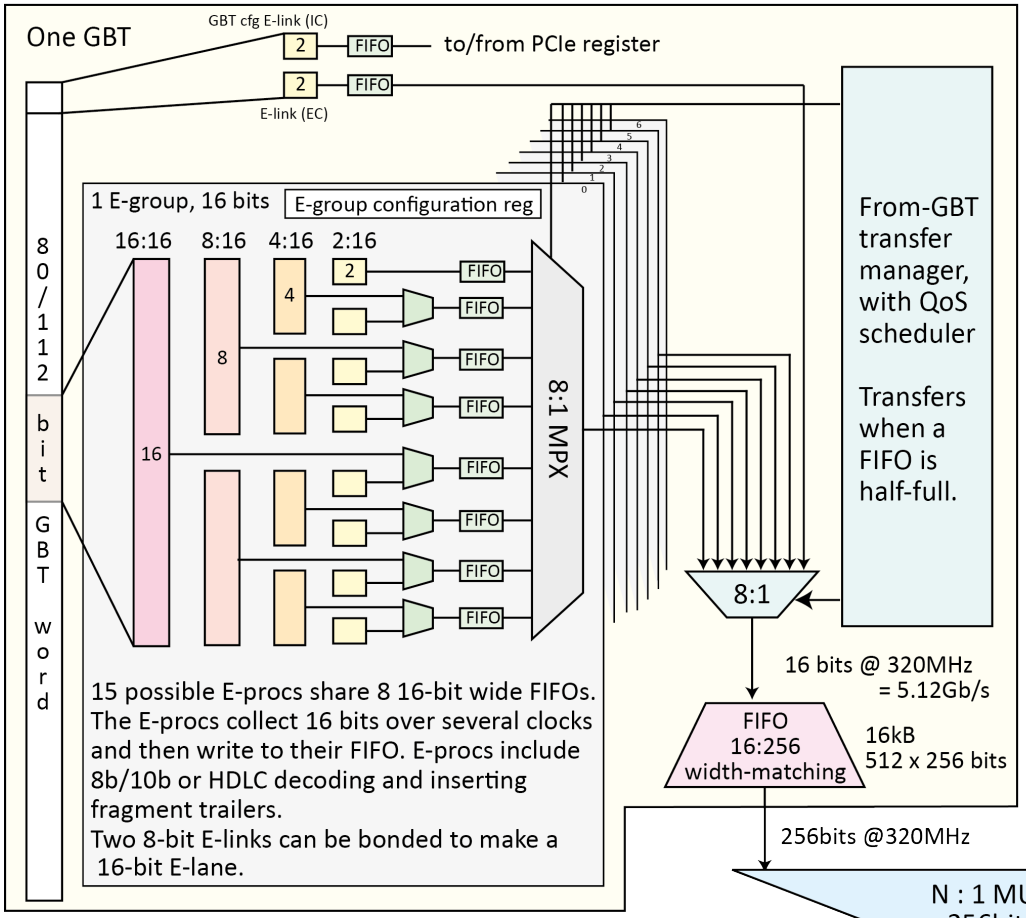
flx-dump-blocks: Dump blocks from FLX into a file for analysis. Example: Dump 100 1kB-blocks

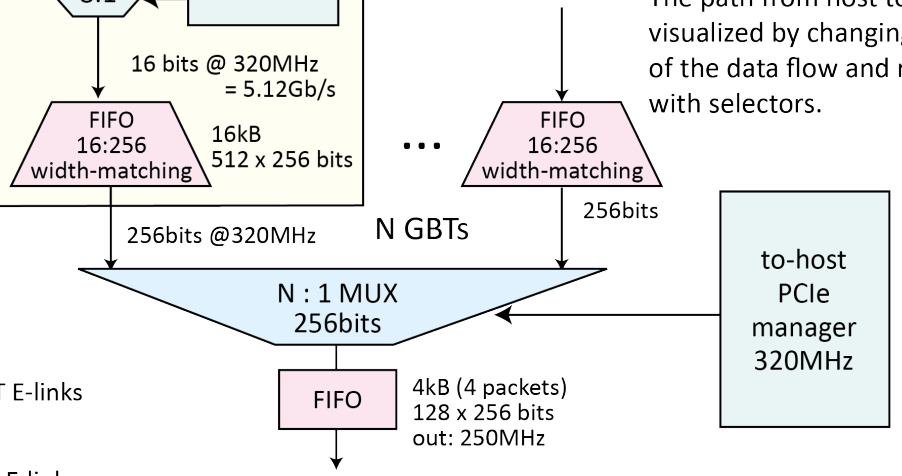
```
$ flx-dump-blocks –n 100 –f output.blocks
```

See '–h' for full list of options.

```
$ flx-reset –h
```

FELIX FPGA: internal data multiplexing to PCI





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

There is one output manager for each GBT.

The output transfer manager chooses one E-link at a time and transfers its data to the output FIFO in fixed length blocks. Each block is pushed via PCIe to host memory reserved for each E-link. The fragments of each stream packet are concatenated, if necessary, in the host and output to the LAN.

For E-lanes wider than 16 bits, 16-bits are taken from the top of each 16-bit FIFO comprising the E-lane in round robin order without any intervening data and sent to the output FIFO.

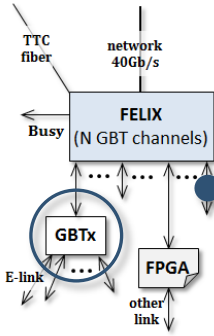
The path from host to GBT can be visualized by changing the direction of the data flow and replacing muxes with selectors.

1 GBT:

Uplink: 80 bits normal, 112 bits wide, mode
7 16-bit E-groups, 8 FIFOs each, + 2x2-bit DCS, GBT E-links
max: 116 bits @40 MHz= 4.64Gb/s
Down link: 80 bits normal, 48 bits wide, mode
5 16-bit E-groups, 8FIFOs each, + 2x2-bit DCS, GBT E-links
max: 84 bits @40 MHz= 3.36Gb/s
Total of (7+5)x8 + 4 = 100 FIFOs per GBT

LL_FELIX_internal_V13

FELIX Operation modes implementation



- → GBT Normal mode (with Forward Error Correction, 3.2Gb/s)
- → GBT wide mode (112 bit, 4.48Gb/s)
- → "Full mode" (9.6GB/s)
- → TTC distribution

GBT mode, 3.2 Gb/s (80 bit @ 40MHz), 5 bidirectional E-groups

- each E-group has up to 8 E-links
- each E-group rate is up to 320 Mb/s (8bit @ 40MHz)
- available E-link rates: 80, 160, 320 Mb/s (640 Mb/s)

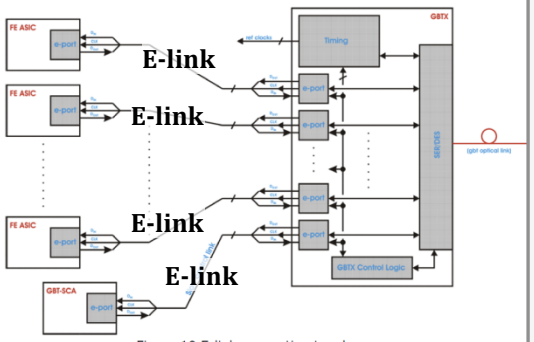
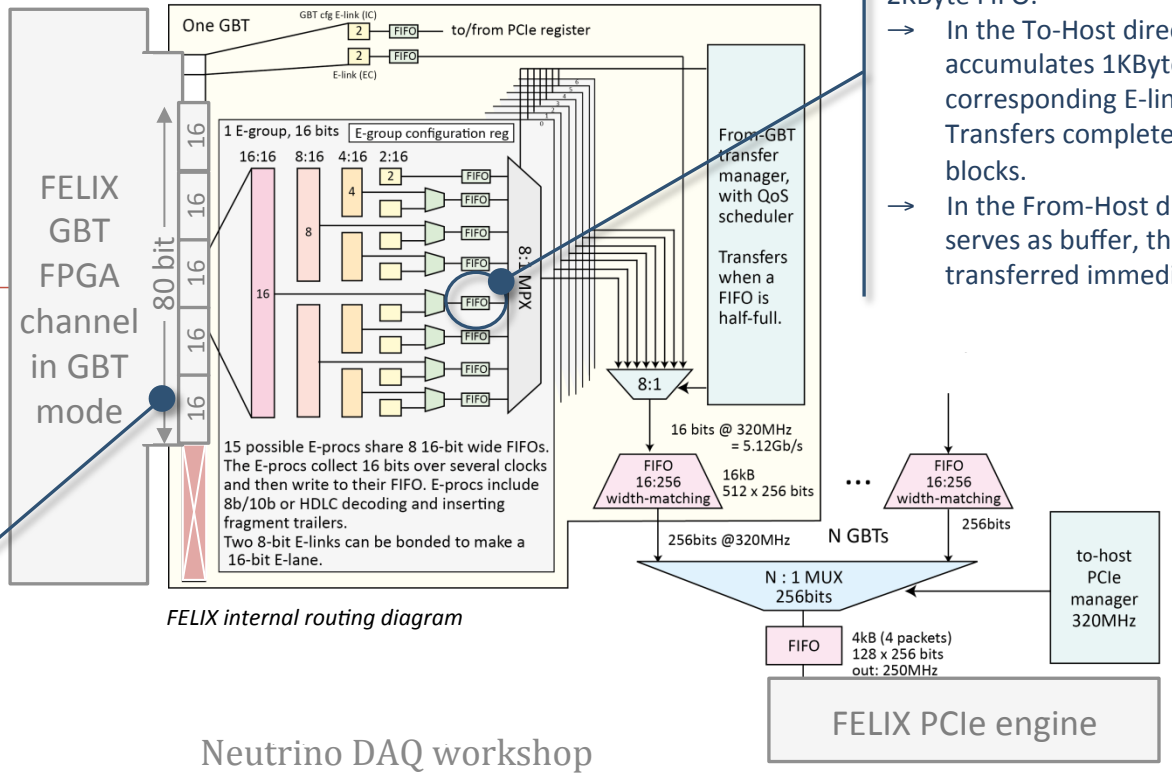


Figure 10 E-link connection topology.
From GBTx Manual

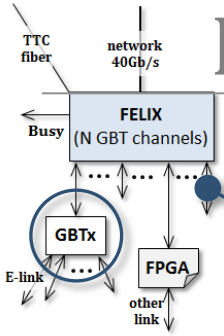
80 bit @ 40MHz



FELIX internal routing diagram

2KByte FIFO.
- → In the To-Host direction accumulates 1KByte blocks of corresponding E-link data. Transfers complete 1KByte blocks.
- → In the From-Host direction serves as buffer, the data is transferred immediately.

FELIX Operation modes implementation



- → GBT Normal mode (with Forward Error Correction, 3.2Gb/s)
- → **GBT wide mode (112 bit, 4.48Gb/s)**
- → "Full mode" (9.6GB/s)
- → TTC distribution

GBT wide mode, 4.48 Gb/s (112 bit @ 40MHz), 7 to-Host, 3 From-Host E-groups

- each E-group has up to 8 E-links
- each E-group rate is up to 320 Mb/s (8bit @ 40MHz)
- available E-link rates: 80, 160, 320 Mb/s (640 Mb/s)

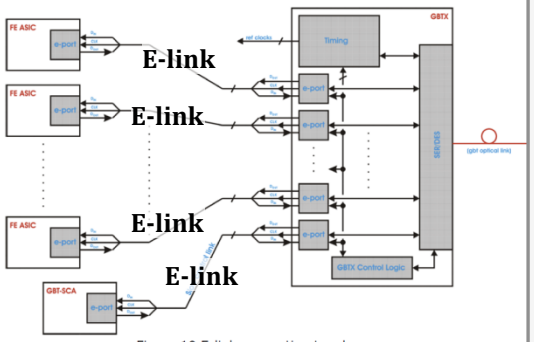
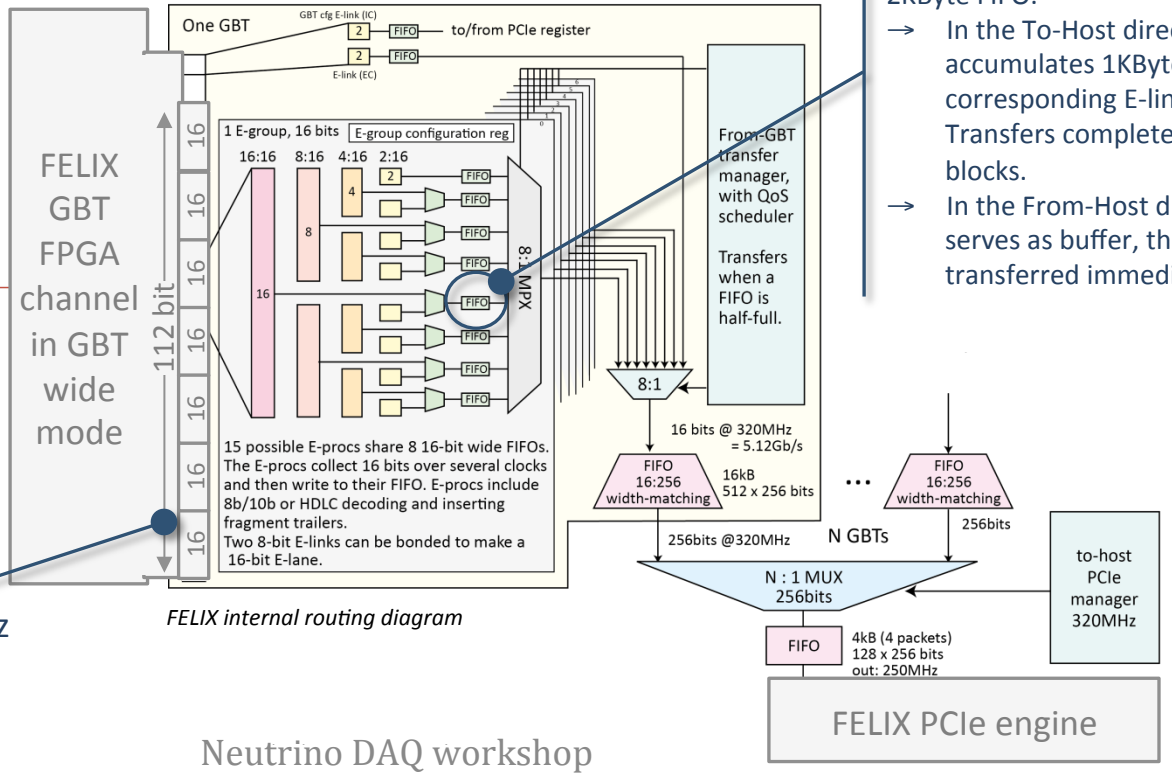


Figure 10 E-link connection topology.

From GBTx Manual

112 bit @ 40MHz

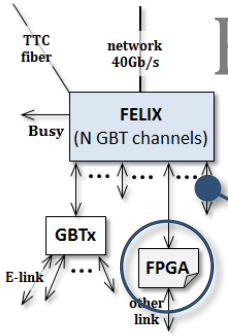


FELIX internal routing diagram

2KByte FIFO.

- → In the To-Host direction accumulates 1KByte blocks of corresponding E-link data. Transfers complete 1KByte blocks.
- → In the From-Host direction serves as buffer, the data is transferred immediately.

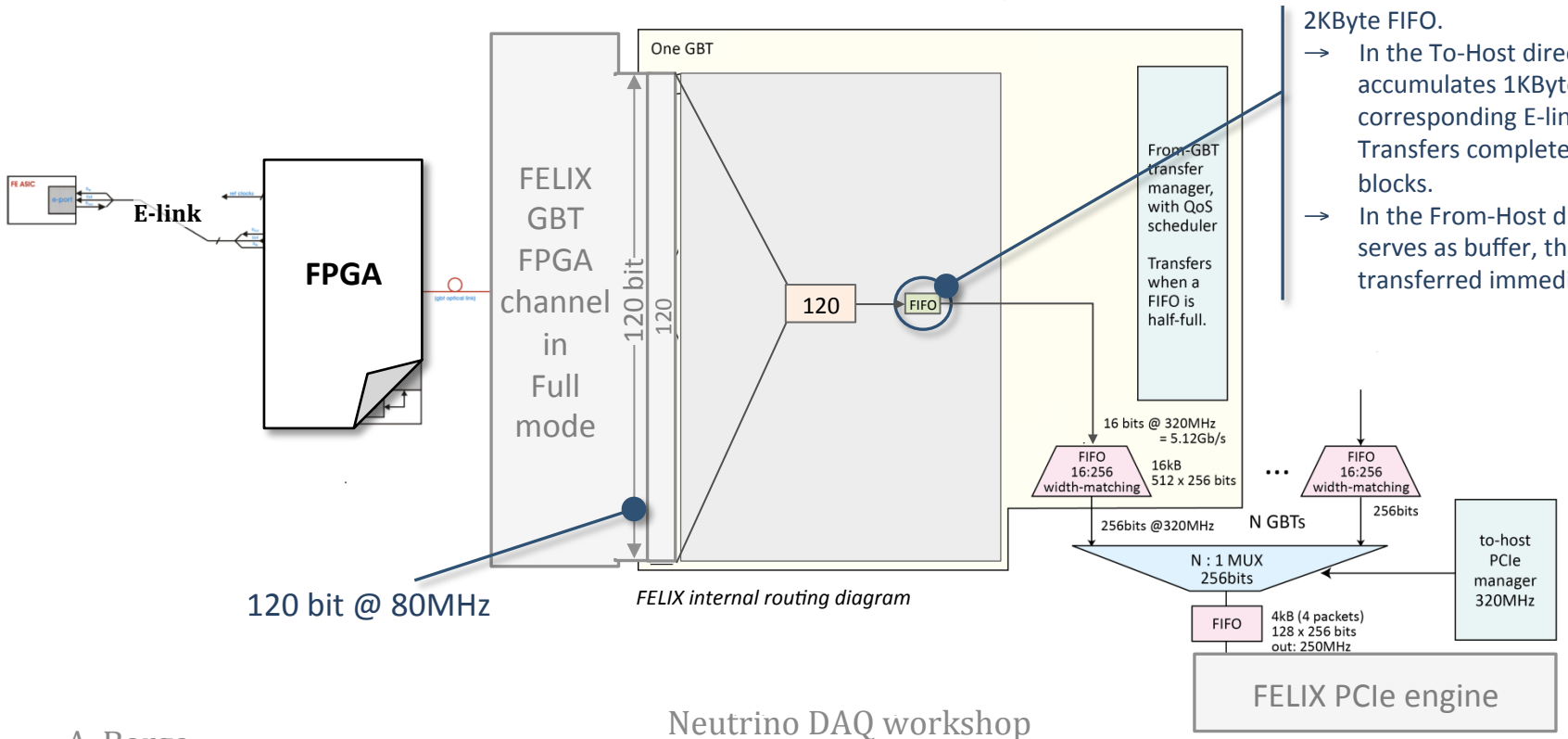
FELIX Operation modes implementation



→ GBT Normal mode (with Forward Error Correction, 3.2Gb/s)
→ GBT wide mode (112 bit, 4.48Gb/s)
→ "Full mode" (9.6GB/s)
→ TTC distribution

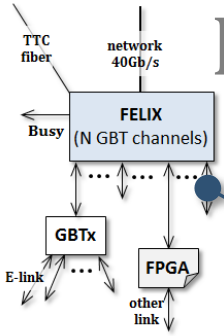
To-Host Full mode, ~9.6 Gb/s (under discussion), From-Host: standard GBT links

- single E-link at full link bandwidth, can use 8b/10b encoding
- supports BUSY-ON and BUSY-OFF symbols (K-characters)
- From-Host "IC" link is used for flow control, XON and XOFF



FELIX internal routing diagram

FELIX Operation modes implementation



→ GBT Normal mode (with Forward Error Correction, 3.2Gb/s)
→ GBT wide mode (112 bit, 4.48Gb/s)
→ "Full mode" (9.6GB/s)
→ TTC distribution

TTC mode (FEC), 3.2 Gb/s (80 bit @ 40MHz), 5 From-Host E-groups

- each E-group has up to 8 E-links
- each E-group rate is up to 320 Mb/s (8bit @ 40MHz)
- available E-link rates: 80, 160, 320 Mb/s

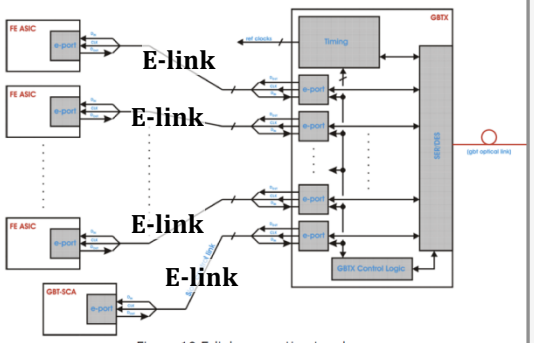
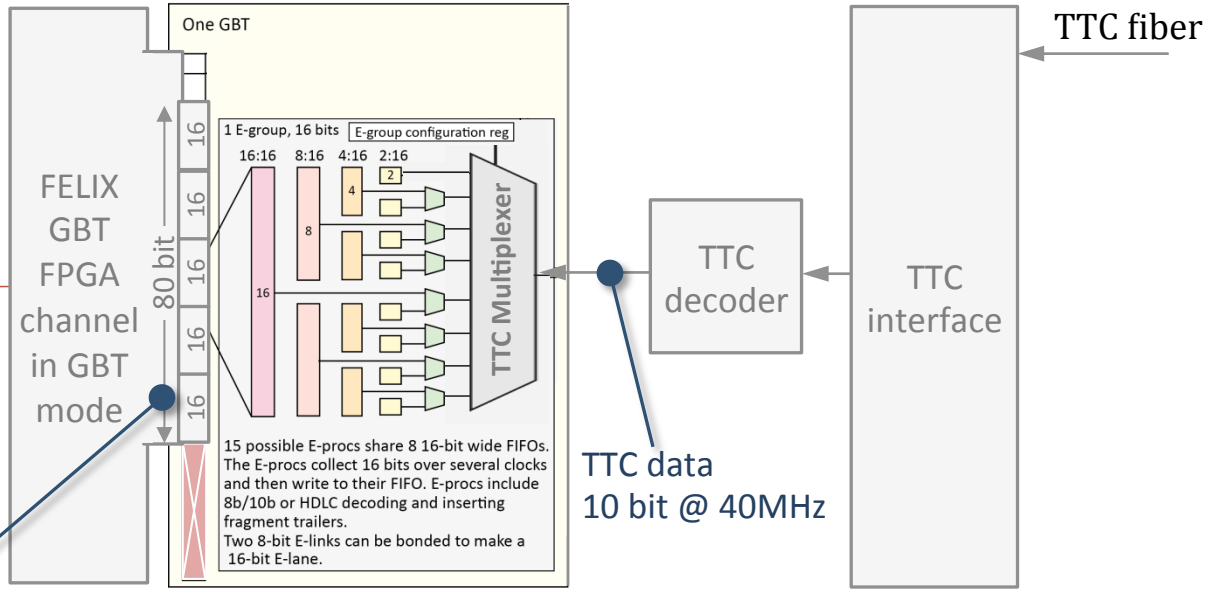


Figure 10 E-link connection topology.
From GBTx Manual



FELIX internal routing diagram

Digital Dual Mixer Time Difference (DDMTD)