Timing and Fast Control of the Upgraded LHCb Readout System

LHCb Upgrade Electronics Review

5 December 2012

Federico Alessio
Richard Jacobsson

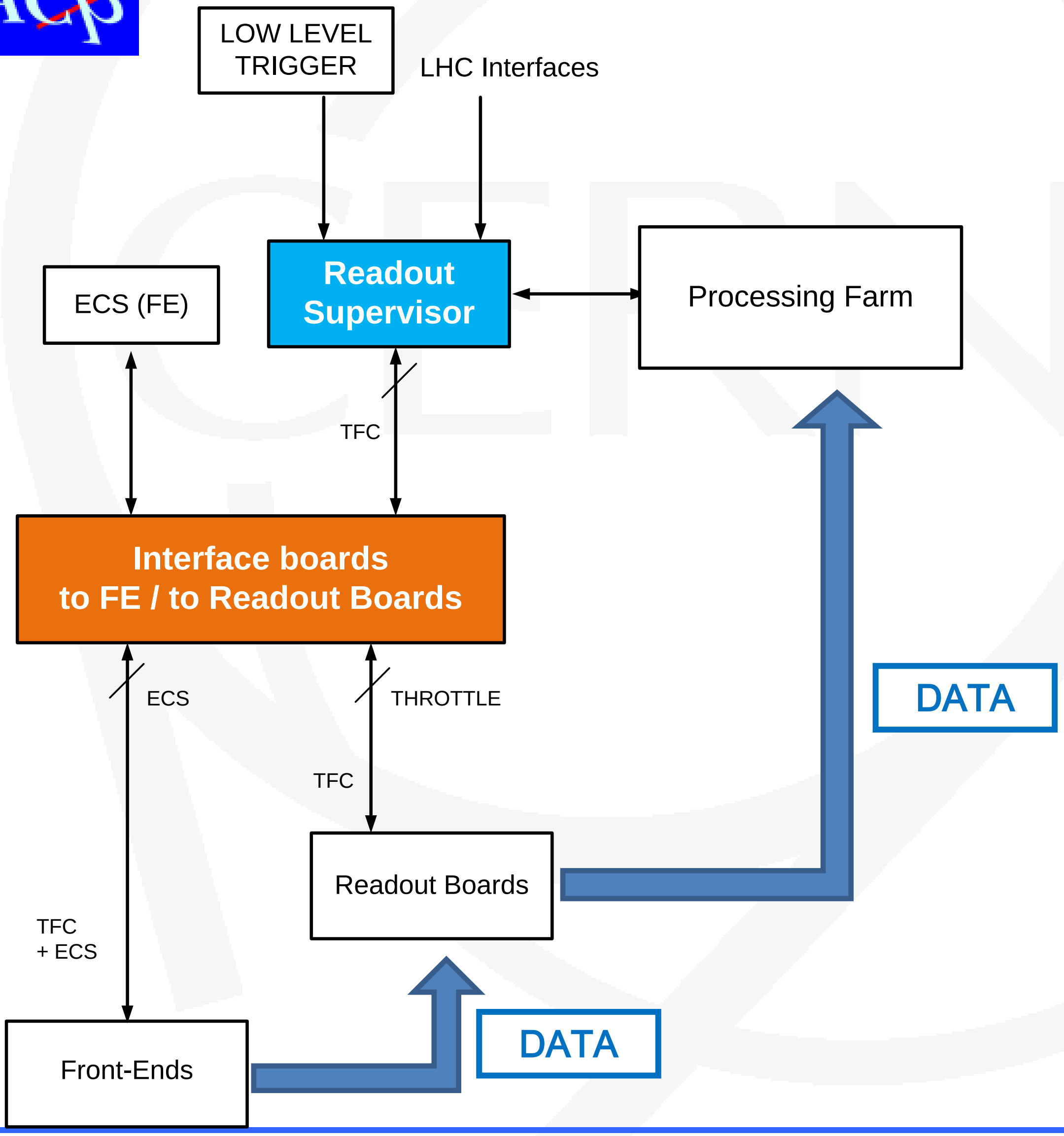
Outline

1. System and functional requirements
2. The new S-TFC system within the upgraded readout architecture
   - ✓ System-level specifications
   - ✓ Concept of readout slice
   - ✓ FPGAs implementation
3. Clock distribution and phase/latency control
4. TFC protocols to TELL40 and FE
5. TFC readout control sequences
   - ✓ commands and resets
6. GBT implementation guidelines at the FE
7. Slow Control to FE
8. Mapping hardware on ATCA
9. Running a hybrid system
10. Simulation framework
11. Plans for test-bench and LS1

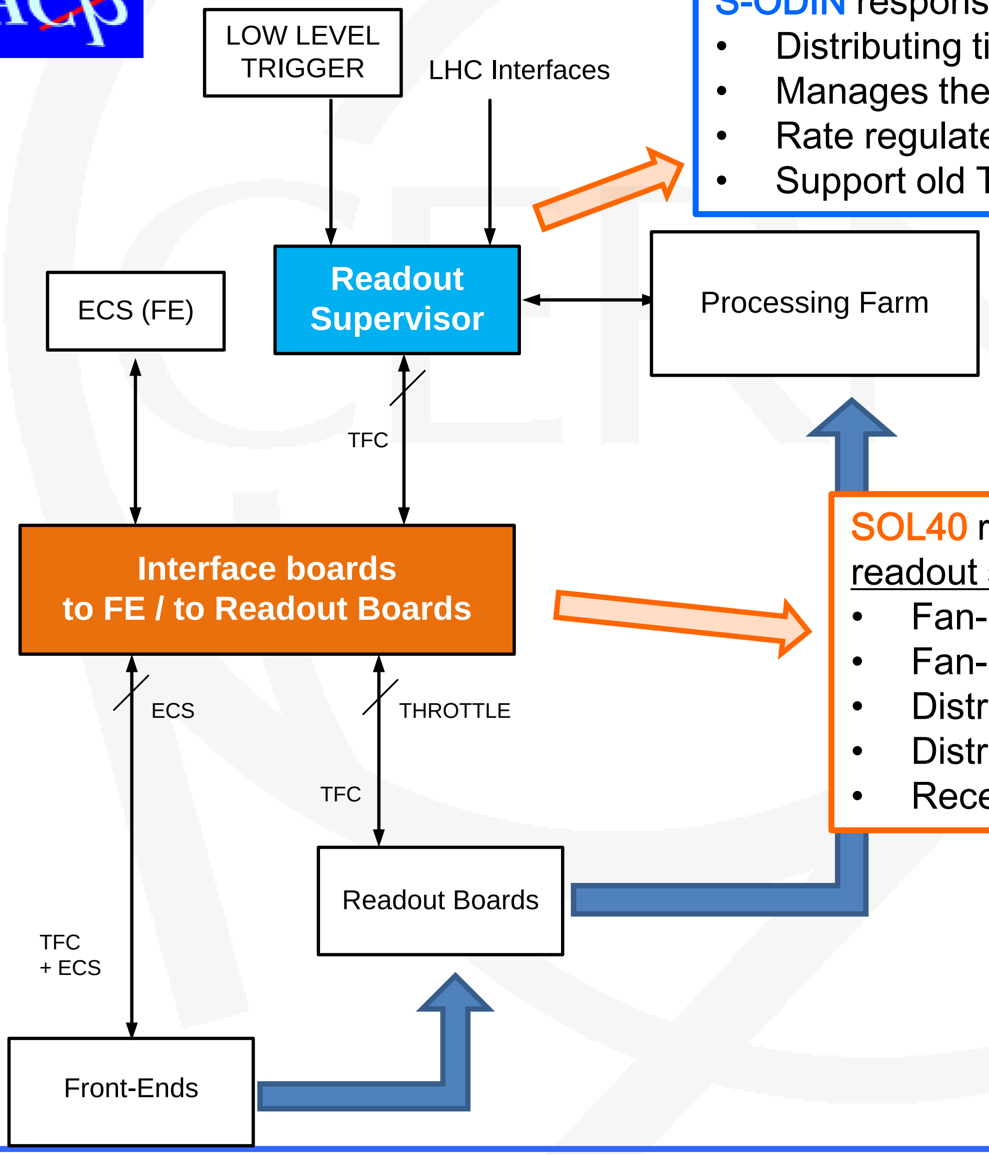
System and functional requirements

1. Bidirectional communication network
2. Full control of clock jitter, phase and latency
   - ✓ At the FE, but also at TELL40 and between S-TFC boards
3. Interfaces with the LHC
4. Events rate control (throttle, deadtime)
5. Low-Level-Trigger connection
6. Sub-detectors calibration triggers
7. Accounting and monitoring of various readout parameters (not covered here)
   - ✓ Rates, triggers, luminosity, deadtime…
8. Partitioning to allow running with any ensemble and parallel partitions (not covered here)
9. Destination control for the event packets (not covered here)
10. S-ODIN data bank with information about accepted events (not covered here)
11. Support for old TTC-based distribution system
12. Test-bench support and system development

S-TFC at a glance



S-TFC at a glance



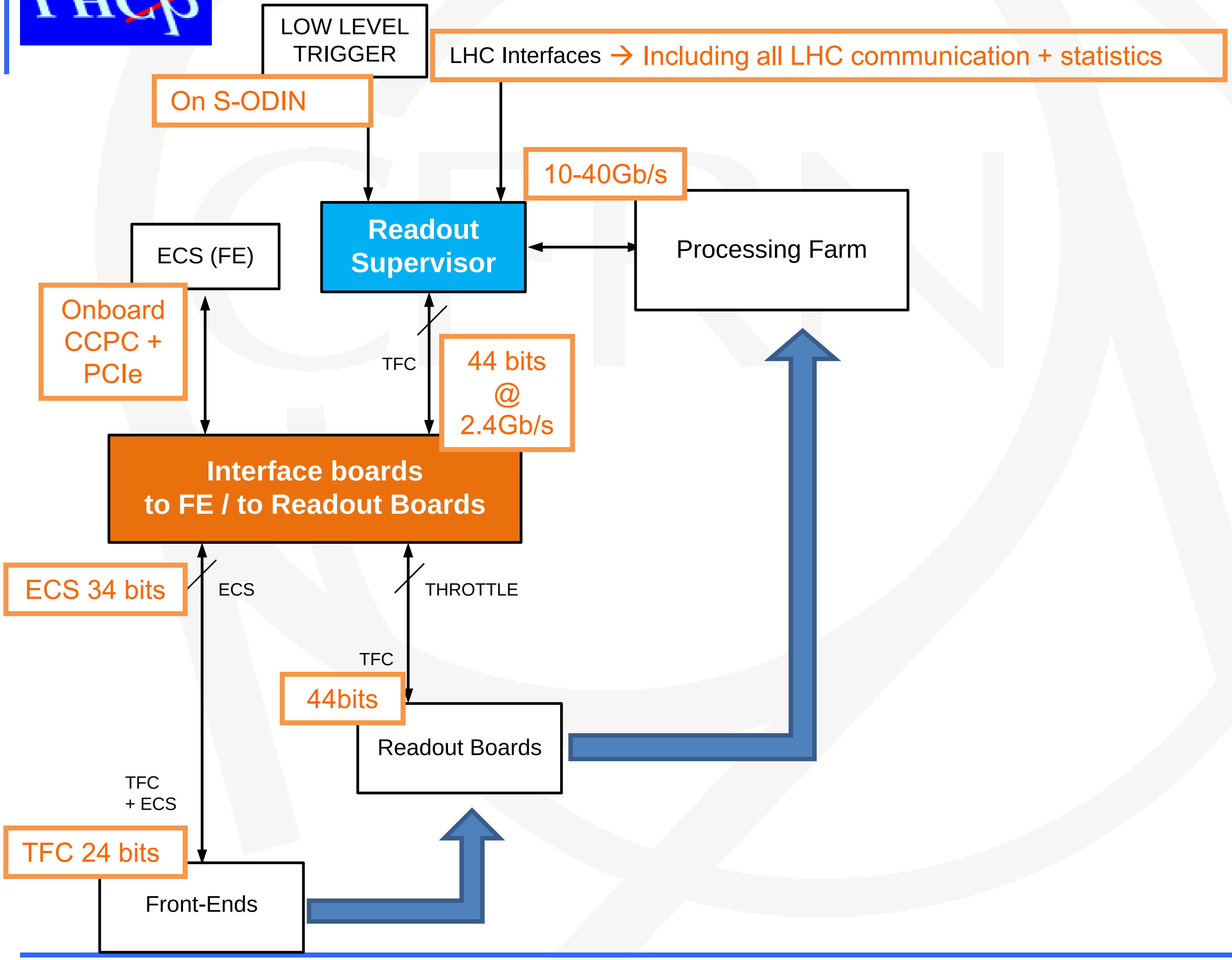
S-ODIN responsible for controlling upgraded readout system

- Distributing timing and synchronous commands
- Manages the dispatching of events to the EFF
- Rate regulates the system
- Support old TTC system: *hybrid system!*

SOL40 responsible for interfacing FE+TELL40 readout slice to S-ODIN

- Fan-out TFC information to TELL40
- Fan-in THROTTLE information from TELL40
- Distributes TFC information to FE
- Distributes ECS configuration data to FE
- Receives ECS monitoring data from FE

S-TFC at a glance



LOW LEVEL TRIGGER

LHC Interfaces → Including all LHC communication + statistics

On S-ODIN

10-40Gb/s

ECS (FE)

Readout Supervisor

Processing Farm

Onboard CCPC + PCIe

TFC

44 bits @ 2.4Gb/s

Interface boards to FE / to Readout Boards

ECS 34 bits

ECS

THROTTLE

TFC

44bits

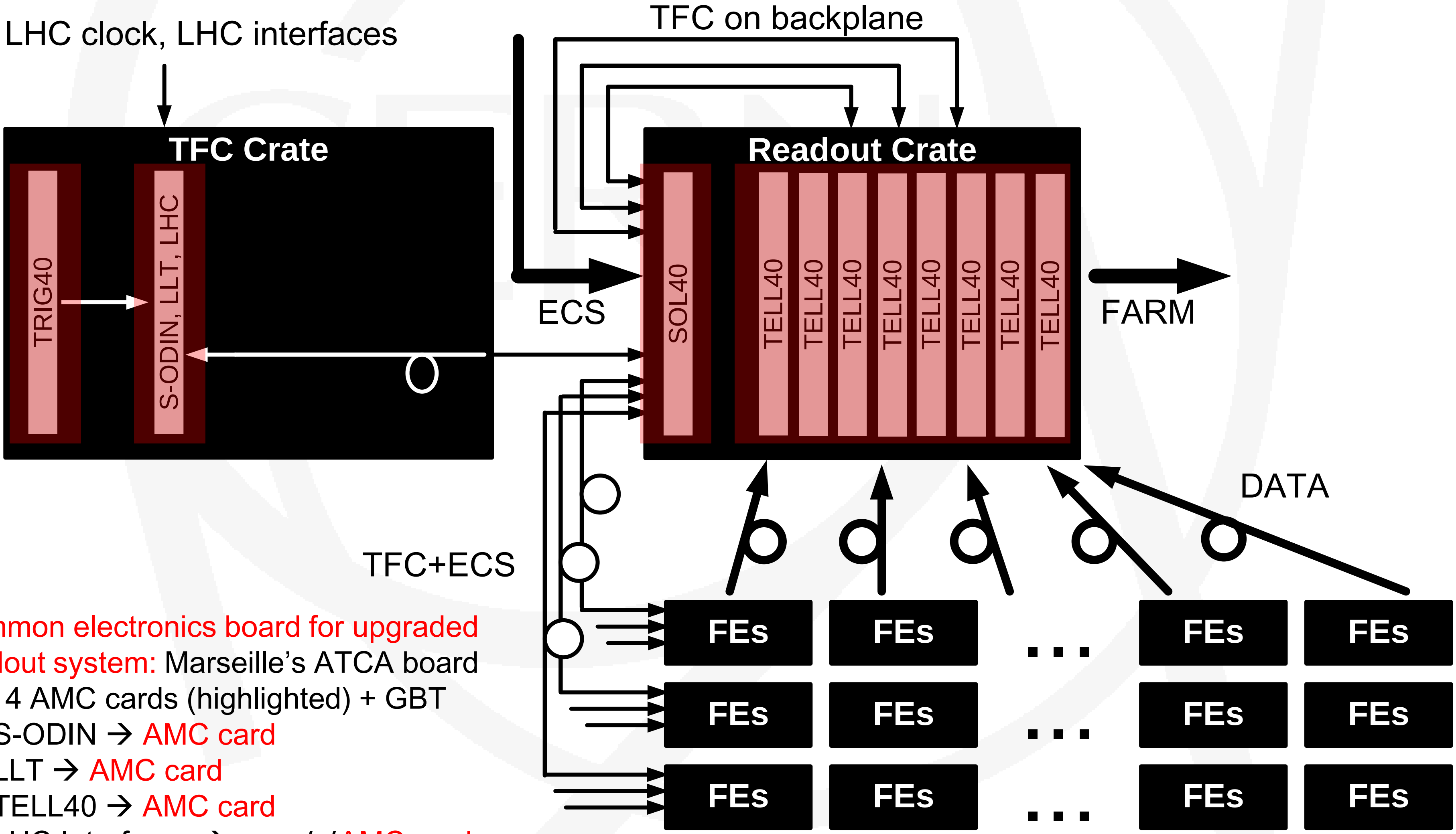
Readout Boards

TFC + ECS

TFC 24 bits

Front-Ends

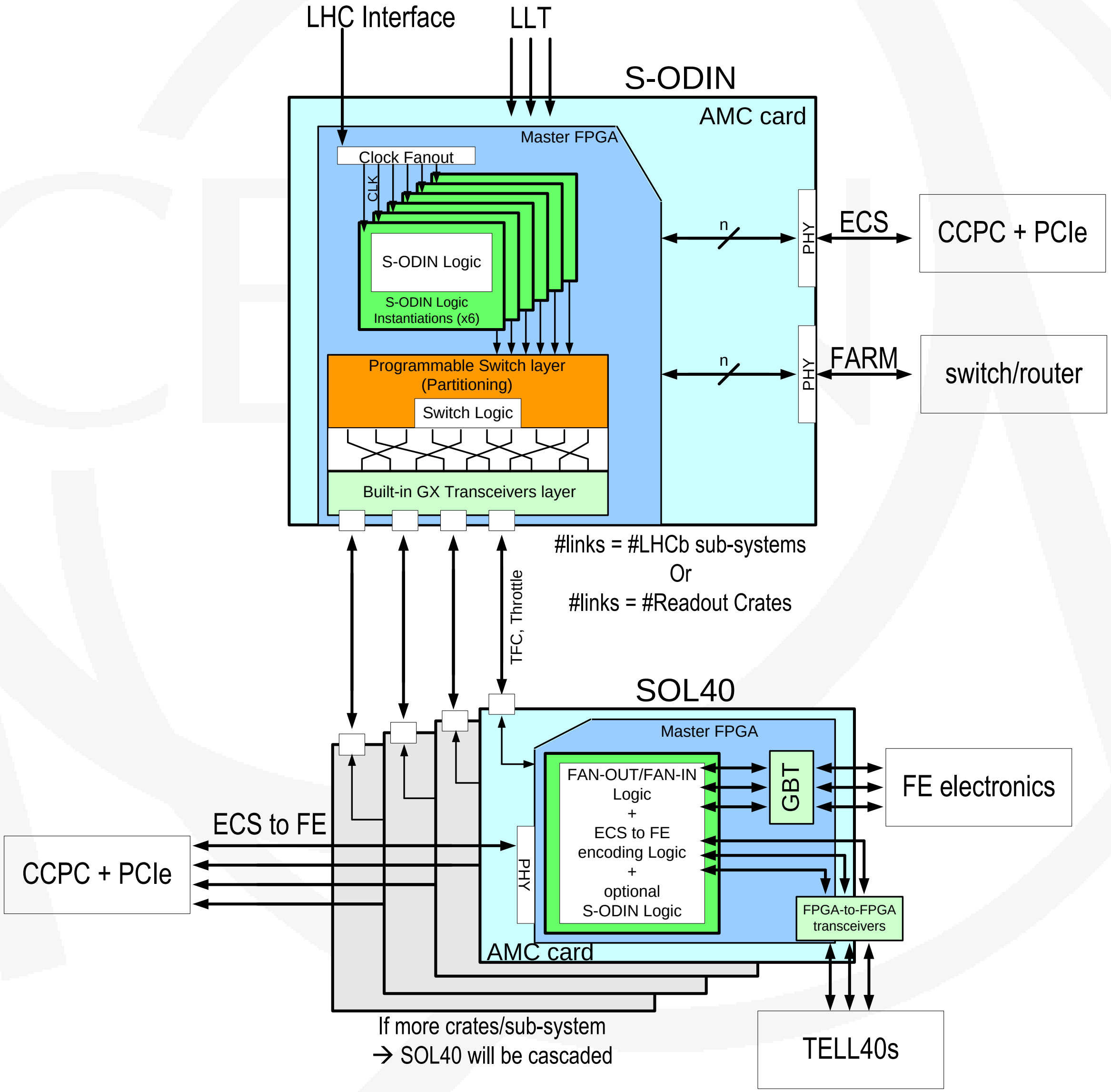
The upgraded physical readout slice



Common electronics board for upgraded readout system: Marseille's ATCA board with 4 AMC cards (highlighted) + GBT

- S-ODIN → AMC card
- LLT → AMC card
- TELL40 → AMC card
- LHC Interfaces → *special* AMC card

S-TFC concept



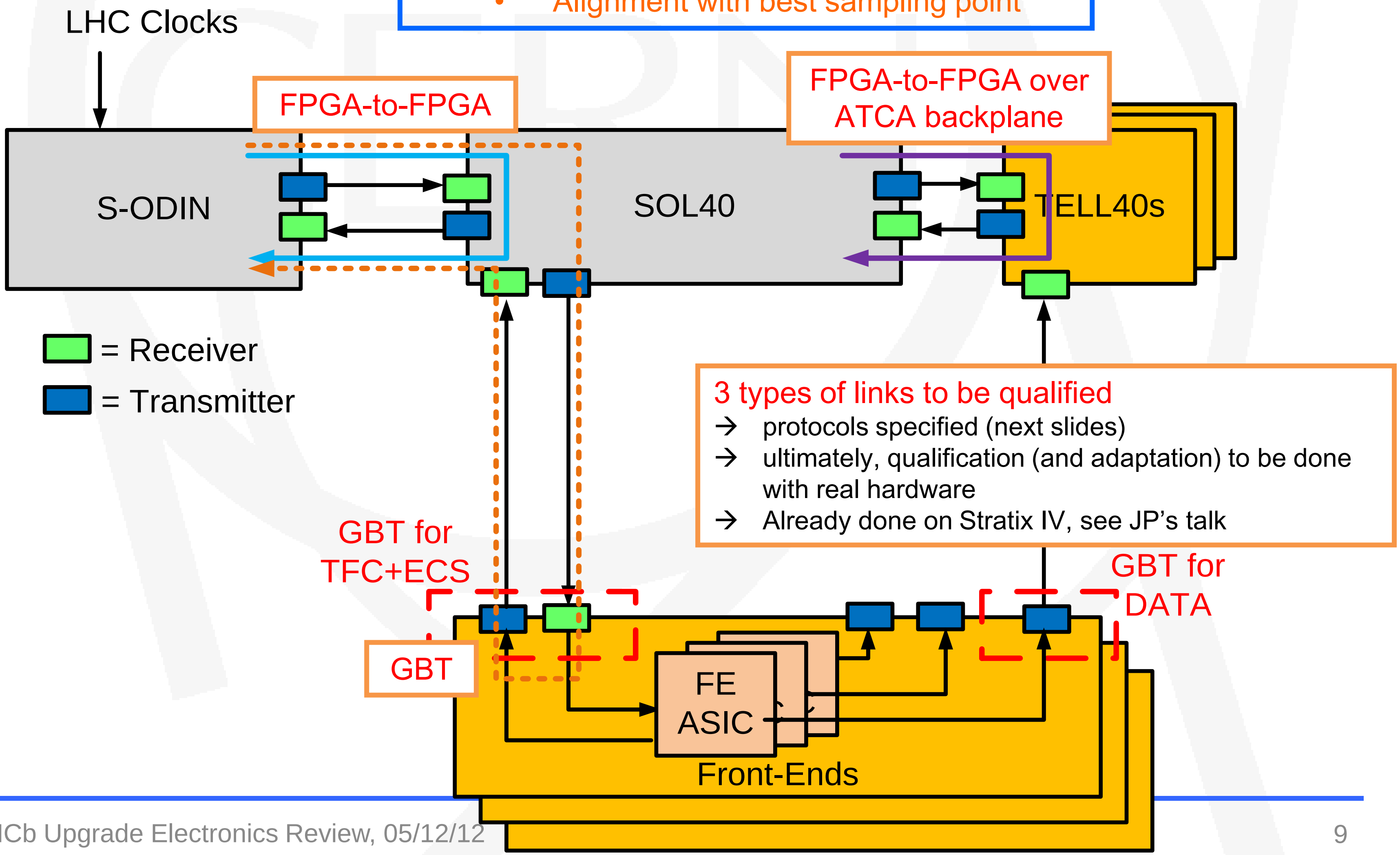
#links = #LHCb sub-systems
Or
#links = #Readout Crates

If more crates/sub-system
→ SOL40 will be cascaded

Clock distribution and phase/latency control

✓ LATENCY
- Alignment with Bunch crossing (BXID)

✓ FINE PHASE
- Alignment with best sampling point



🟩 = Receiver

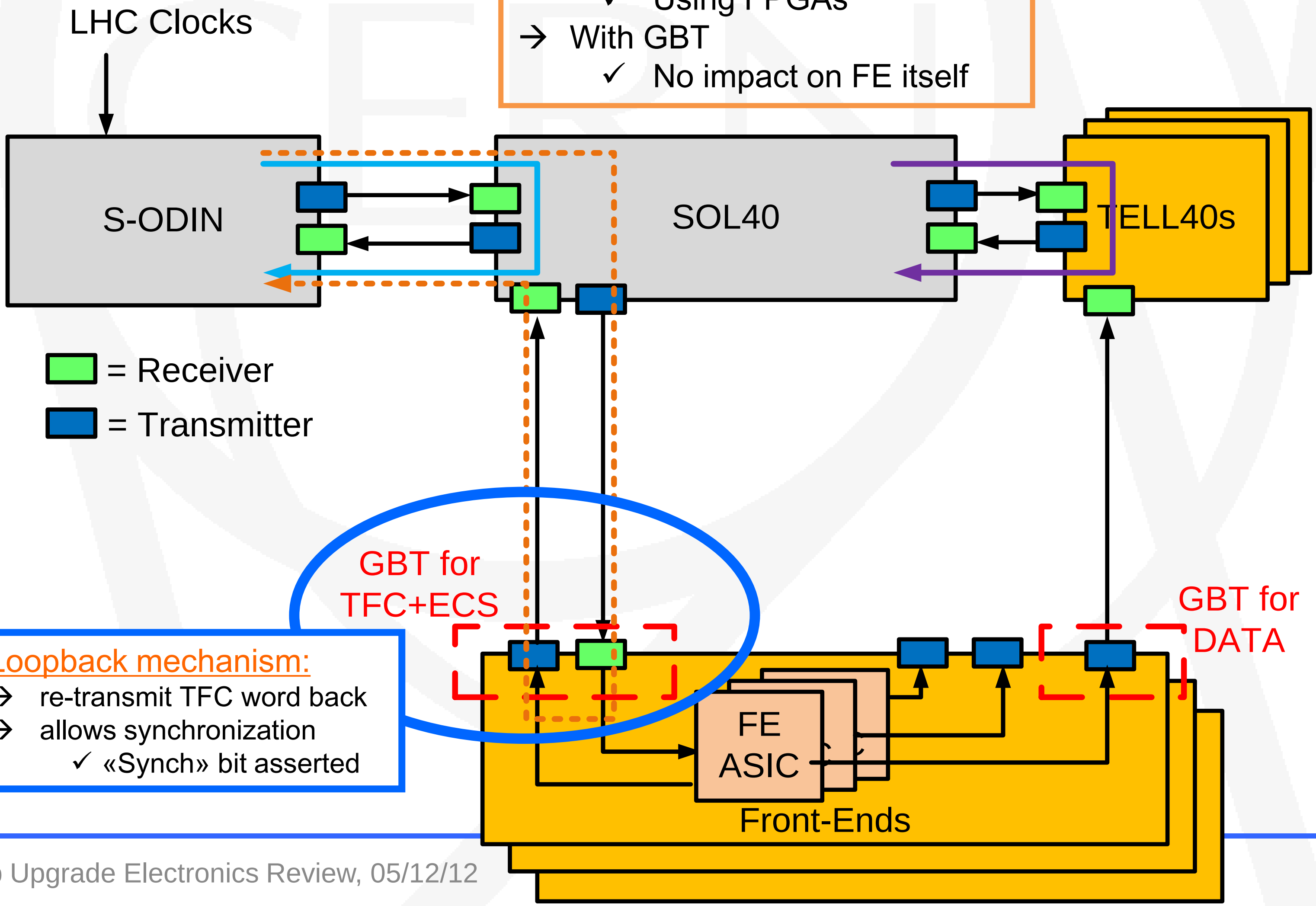
🟦 = Transmitter

3 types of links to be qualified
→ protocols specified (next slides)
→ ultimately, qualification (and adaptation) to be done with real hardware
→ Already done on Stratix IV, see JP’s talk

Resynchronization mechanism

Some resynchronization mechanisms envisaged:

- → Within TFC boards
  - ✓ Using FPGAs
- → With GBT
  - ✓ No impact on FE itself



🟩 = Receiver

🟦 = Transmitter

Loopback mechanism:

- → re-transmit TFC word back
- → allows synchronization
  - ✓ «Synch» bit asserted

TFC protocol to TELL40

TFC decoding block for the TELL40 provided: VHDL entity with inputs/outputs

TFC word to TELL40 via SOL40:

- ✓ 44 bits sent every 40 MHz = ~1.8 Gb/s (on backplane)
- ✓ all commands are associated to BXID in TFC word
- ✓ initial idea to use half GBT protocol, 8b/10b is straightforward solution
- ✓ extension possible, protocol easily modifiable
  - → (could be extended between S-ODIN and SOL40 to distribute more information)

43 .. 32	31 .. 16	15 .. 13	12..10	9	8
BXID(11..0)	**MEP Dest(15..0)**	**Trigger Type(2..0)**	**Calibration Type(2..0)**	**Synch**	**Snapshot**

7	6	5	4	3	2	1	0
Trigger	**BX Veto**	**NZS Mode**	**Header Only**	**BE Reset**	**FE Reset**	**EID Reset**	**BXID Reset**

Constant latency after S-ODIN

THROTTLE information from each TELL40 to SOL40:

- ✓ 1 bit for each AMC board (4 bits from each ATCA card) + BXID for which the throttle was set
  - → 16 bits in 8b/10b encoder

TFC protocol to FE

Decoding of TFC commands/reset must be implemented in FE

TFC word on downlink to FE via SOL40 embedded in GBT word:

- ✓ 24 bits in each GBT frame every 40 MHz = ~1 Gb/s
- ✓ all commands associated to BXID in TFC word

23 .. 12	11..10	9	8	7 .. 5	4	3	2	1	0
BXID(11..0)	**Reserve**	**Synch**	**Snapshot**	**Calibration Type(2..0)**	**BX Veto**	**NZS Mode**	**Header Only**	**FE Reset**	**BXID Reset**

Strict requirement for FE

- ✓ configurable delays for each TFC command
  - GBT does not support individual delays for each line
  - need for «local» pipelining: detector delays+cables+operational logic (i.e. laser pulse?)

TFC word will arrive before the actual event takes place

- ✓ allow use of commands/resets in FE for particular BXID
- ✓ accounting of delays in S-ODIN: for now, 16 clock cycles earlier + time to receive
  - aligned to the «furthest» FE (simulation, then in situ calibration!)

TFC protocol to FE has implications for GBT configuration (later)

- → remember: ECS to FE embedded in GBT protocol

TFC readout control sequences (to FE)

"BXID" and "BXID Reset"

- Every TFC word carries a BXID for synchronicity checks of the system
- A BXID Reset is sent at every turn of the LHC (orbit pulse)
  - ✓ Should only reset the internal bunch counter of the FE

"FE RESETS"

- Bit set for one clock cycle in TFC word
- Reset of FE operational logic for data processing, formatting, transmission…
  - ✓ Should not touch the internal bunch counter
  - ✓ FE electronics should be back asap:
    - → S-ODIN will ensure no data is being accepted during the FE reset process
    - → wait to the slowest by setting Header Only bit, i.e. no data is recorded at FE
  - ✓ Reset TELL40 data input logic: the same bit is sent to TELL40 for same BXID

"HEADER ONLY"

- Idling the system: only header (or few bits) in data word if this bit is set
  - ✓ Multiple purposes: set it during reset sequence, during NZS transmission, during TAE mode…

"BX VETO"

- Based excusively on filling scheme, processing of that particular event is inhibited
  - ✓ Only header (or few bits) in data word if this bit is set
  - ✓ Allows "recuperating" buffer space in a LHC-synchronous way if taken into account

TFC readout control sequences (to FE)

"CALIBRATION TYPE" COMMAND

- Used to take data with special trigger pulses (periodic, calibration)
  - ✓ Dedicated 4 bits: i.e. 4 different calibration commands possible
  - ✓ Dynamic association to be used for calibration and monitoring
    - Absolute need of delays to account for each individual delay in the detectors
  - ✓ S-ODIN overrides LLT decision at TELL40
    - Periodic or calibration higher priority (but lower rate, max 11.245kHz each!)

"NZS MODE"

- Read out (all) FE channels non-zero suppressed
  - ✓ Packing of full set of bits in many consecutive GBT frames:
  - ✓ Needs buffering at FE (could be a different buffer wrt to data buffer)
- Possible to have also multi-NZS readout: *consecutive NZS events or timing events*
  - ✓ S-ODIN will take care of setting Header Only bit for a defined set of clock cycles later to allow recuperating buffer space (programmable as well, to the slowest of the detector)

"SYNC"

- FE should send fixed 'synch' pattern, configured via ECS
- Pointers of write and read of FE buffers are zeroed.
  - → When deasserted, header shall be in a known position within the GBT frame.
  - → Asserted for more than 2 clock cycles to avoid accidental matches

"SNAPSHOT"

- Read out all status and counter registers in a "latched" way
  - ✓ Latch monitoring registers on snapshot bit, which is set periodically (programmable) and also single shot
  - ✓ When snapshot bit is received, send all data via ECS field in TFC on best effort

TFC readout control sequences (to TELL40)

“EID Reset”

- Reset the EventID counter
  - ✓ EventID assigned to each and every accepted event by S-ODIN
  - ✓ Usually at every Run change or when BE Reset is asserted

“BE Reset”

- Bit set for one clock cycle in TFC word
- Reset of BE operational logic for data processing, formatting, reception/transmission…
  - ✓ Should not touch the internal bunch counter
  - ✓ BE electronics should be back asap:
    - → S-ODIN will ensure no data is being accepted during the BE reset process (header only to FE)
    - → wait to the slowest by setting Header Only bit, i.e. no data is recorded

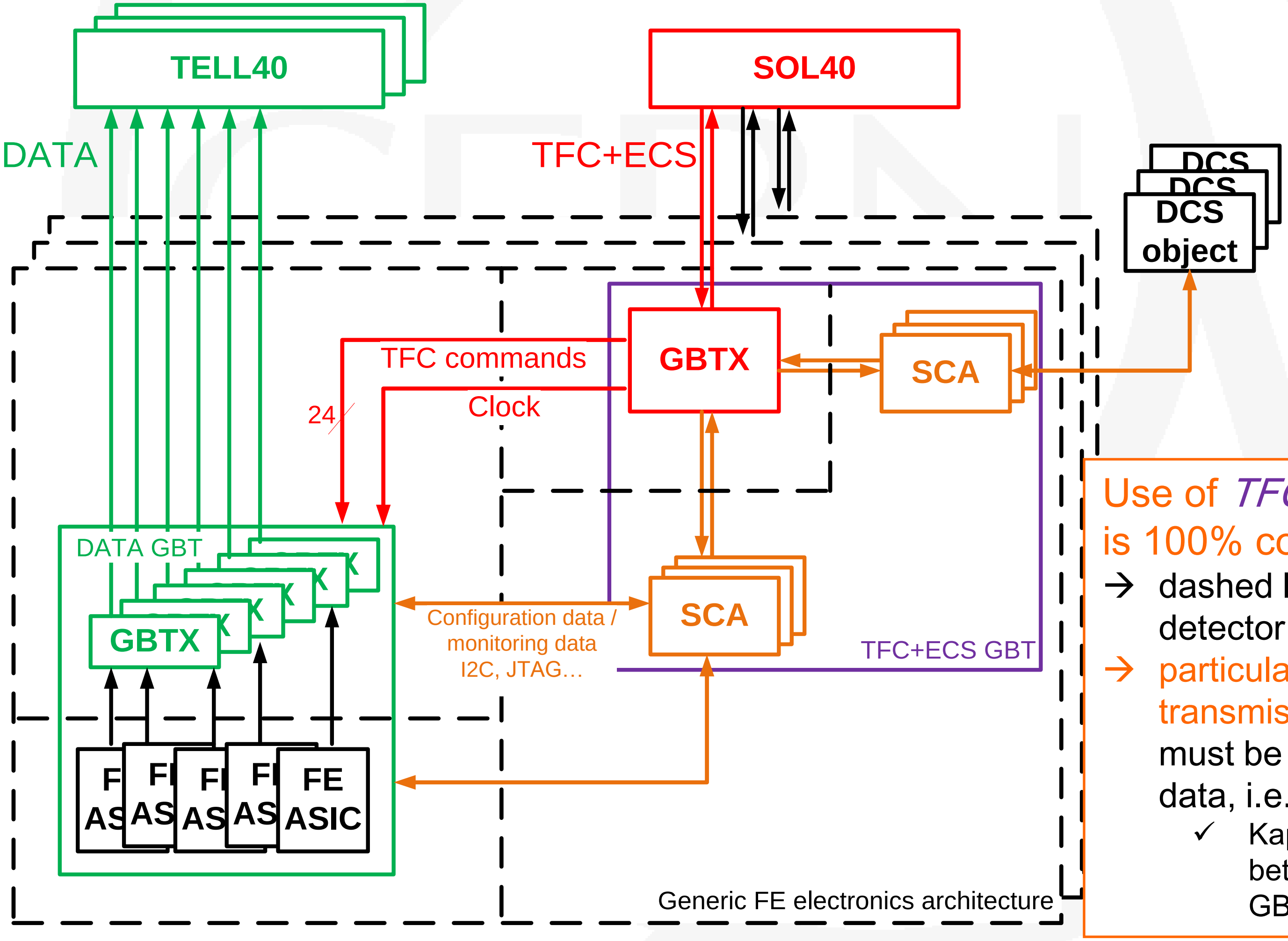
“MEP DESTINATION”

- IP address to which a Multi-Packet Event should be sent
  - ✓ Transmitted by S-ODIN on credit-based scheme, i.e. events sent only to available processing nodes
  - ✓ Transmitted to every TELL40, centrally managed. Packing factor is programmable, based on bandwidth.

“TRIGGER AND TRIGGER TYPE”

- Bit set when event is accepted
  - ✓ Associated with a type which defines its origin (physics, beamgas, calibration, periodic…)
    - Type based on a *priority scheme*: centrally managed by S-ODIN if two triggers are concurrent
  - ✓ Trigger always transmitted at a fixed latency wrt to BXID. Latency to be defined based on simulation

How to decode TFC in FE chips?



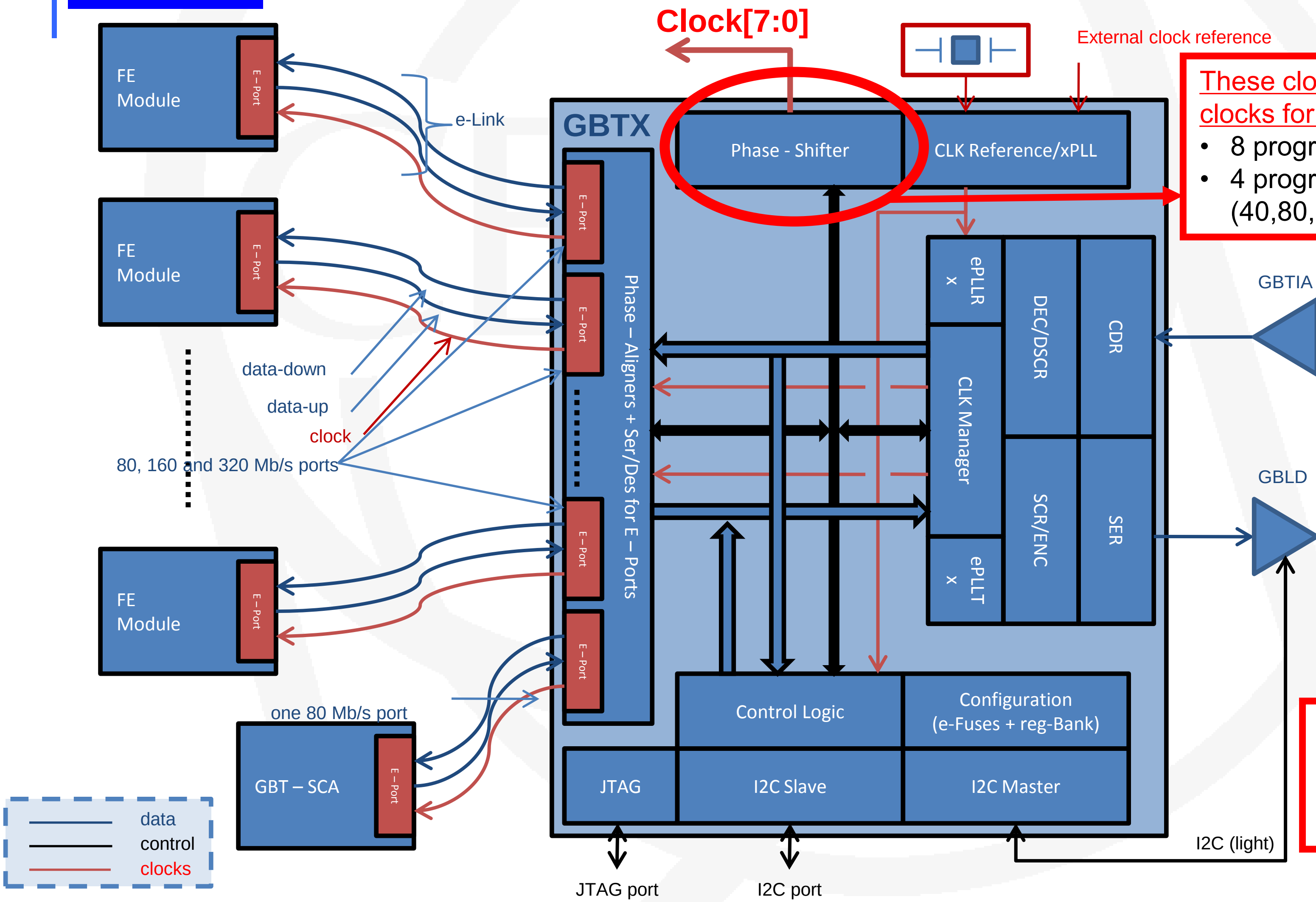
Use of *TFC+ECS GBTs* in FE is 100% common

→ dashed lines indicate the detector specific interface parts

→ particular care in the clock transmission: the TFC clock must be used by FE to transmit data, i.e. low jitter! (next slide)

- ✓ Kapton cable, crate, copper between FE ASICs and GBTX

Timing at the FE via GBT



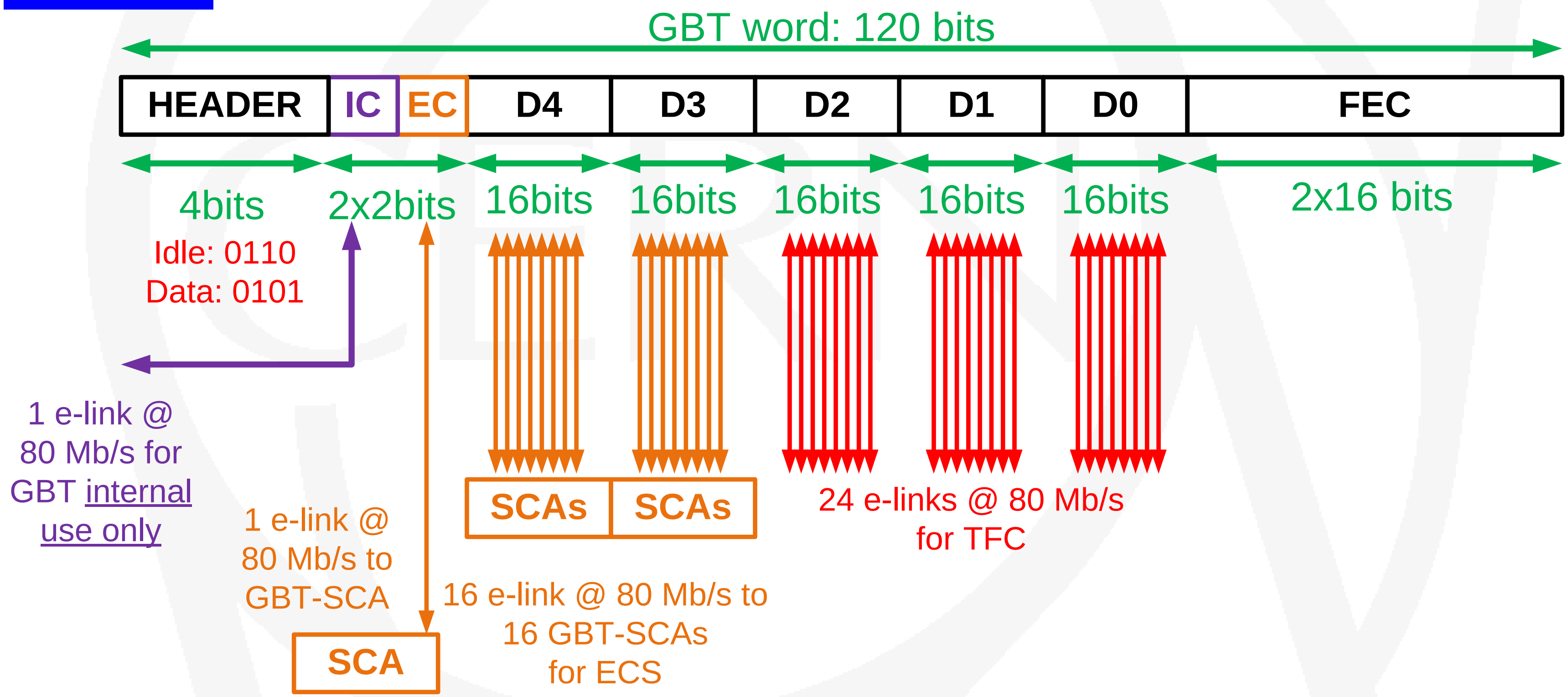
These clocks should be the main clocks for the FE

- 8 programmable phases
- 4 programmable frequencies (40,80,160,320 MHz)

Used to:

- sample TFC bits
- drive *Data GBTs*
- drive FE processes

The TFC+ECS protocol to GBT FE

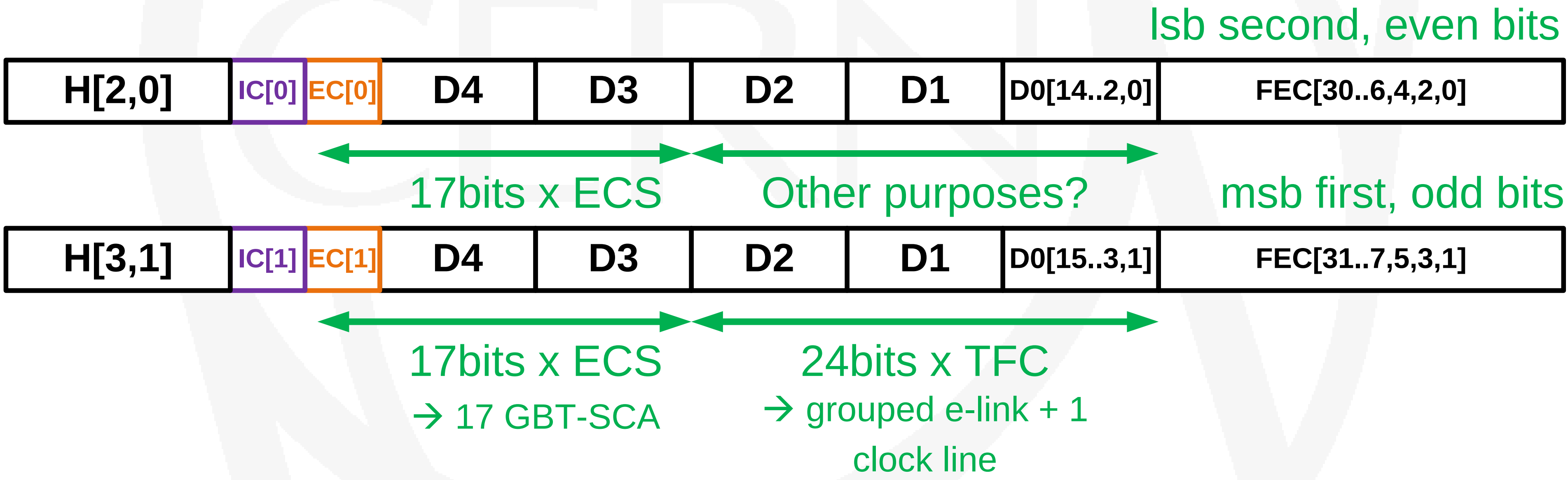

GBT word: 120 bits
HEADER
IC
EC
D4
D3
D2
D1
D0
FEC
4bits
2x2bits
16bits
16bits
16bits
16bits
16bits
2x16 bits
Idle: 0110
Data: 0101
1 e-link @ 80 Mb/s for GBT internal use only
1 e-link @ 80 Mb/s to GBT-SCA
SCA
SCAs
SCAs
16 e-link @ 80 Mb/s to 16 GBT-SCAs for ECS
24 e-links @ 80 Mb/s for TFC

→ TFC protocol has direct implications in the way in which GBT should be configured in FE

- 24 e-links @ 80 Mb/s dedicated to TFC word:
  - ✓ use 80 MHz phase shifter clock to sample TFC parallel word
- TFC bits are packed in GBT frame so that they all come out on the same clock edge
  - ✓ We can repeat the TFC bits also on consecutive 80 MHz clock edge if needed

→ Leftover 17 e-links dedicated to GBT-SCAs for ECS configuring and monitoring

TFC decoding at FE GBT



lsb second, even bits

H[2,0]	IC[0]	EC[0]	D4	D3	D2	D1	D0[14..2,0]	FEC[30..6,4,2,0]

17bits x ECS

Other purposes?

msb first, odd bits

H[3,1]	IC[1]	EC[1]	D4	D3	D2	D1	D0[15..3,1]	FEC[31..7,5,3,1]

17bits x ECS

→ 17 GBT-SCA

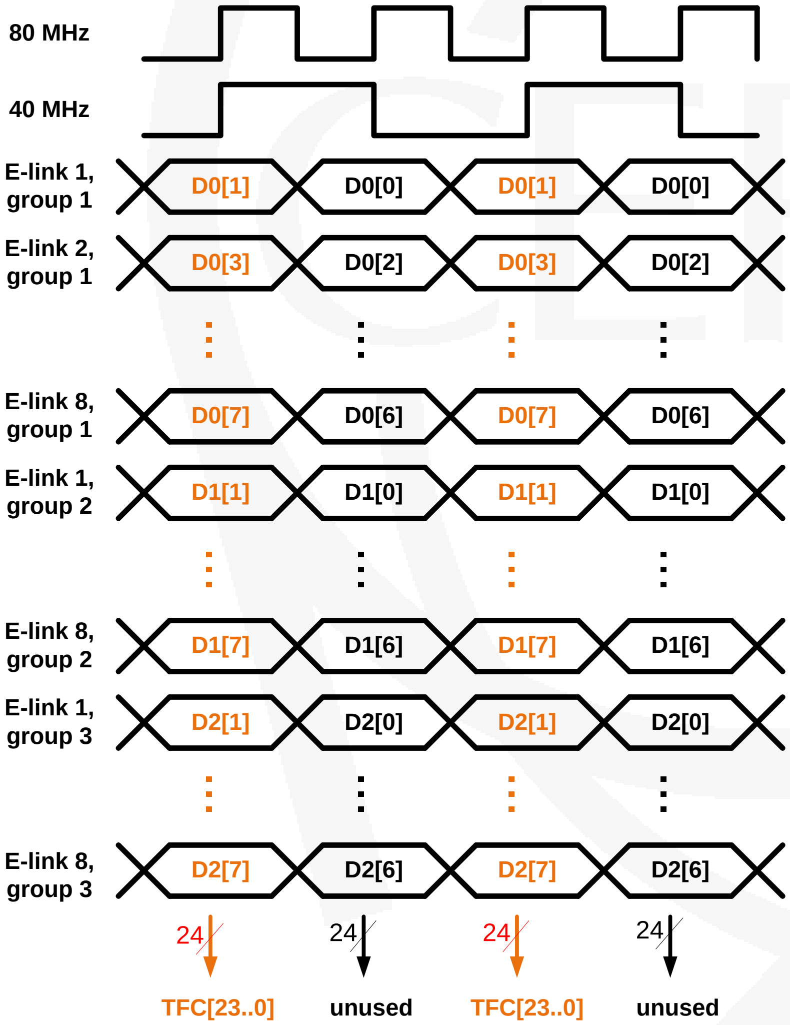
24bits x TFC

→ grouped e-link + 1 clock line

In simple words:

- Odd bits of GBT protocol on rising edge of 40 MHz clock (first, msb),
- Even bits of GBT protocol on falling edge of 40 MHz clock (second, lsb)

TFC decoding at FE after GBT



This is crucial for FE

→ we can already specify where each TFC bit will come out on the GBT chip

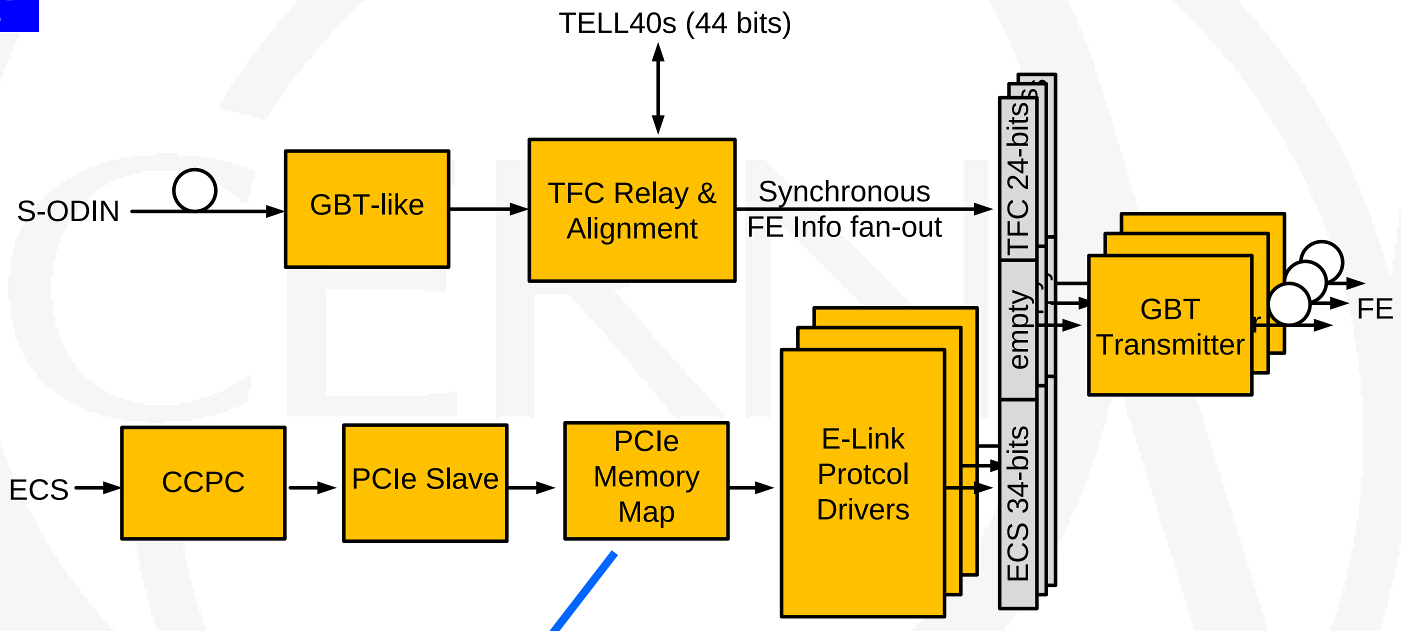
→ this is the only way in which FE designers still have minimal freedom with GBT chip

- ✓ if TFC info was packed to come out on only 12 e-links (first odd then even), then decoding in FE ASIC would be mandatory!
- ✓ which would mean that the GBT bus would have to go to each FE ASIC for decoding of TFC command

→ there is also the idea to repeat the TFC bits on even and odd bits in TFC protocol

→ Or maybe in the future to use them for other purposes

Slow Control (ECS) to FE



Memory Map with internal addressing scheme for GBT-SCA chips + FE chips addressing, e-link addressing and bus type: *content of memory loaded from ECS*

Protocol drivers build GBT-SCA packets with addressing scheme and bus type for associated GBT-SCA user busses to selected FE chip

→ *Each block will build one of the GBT-SCA supported protocols*

→ *Selected based on FE implementation*

Considerations on ECS to FE

- Many FE chips can be connected to same GBT-SCA
  - ✓ Provided we have a proper addressing scheme at the FE side (and its mapping)
- Or (viceversa) many GBT-SCAs can be connected to the same FE chip
  - ✓ If needed to have more than 80 Mb/s for each FE chip ECS…

To make this work, the only thing we need is to be able to:

.. drive the right FE chip at the right address ..

.. with the right GBT-SCA ..

.. with the right protocol for the chosen bus ..

→ SOL40 will do that in the most flexible way possible

✓ driving the GBT-SCA through the downlink

→ ECS (Control PC) must provide to the firmware:

✓ the mapping of the FEs

✓ the mapping of GBT-SCAs

✓ the selected protocols

Firmware

Software

→ See Clara's talk

Considerations on ECS to FE

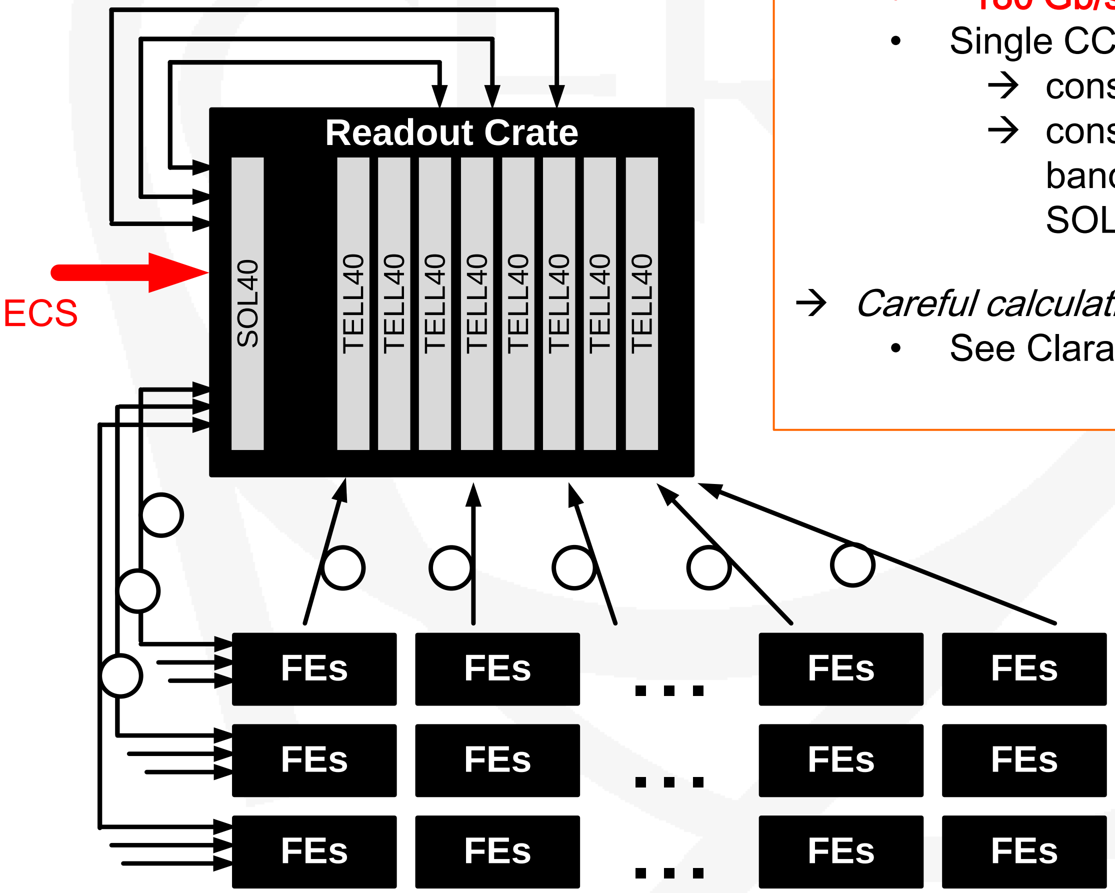
SOL40 has the ECS load of an entire FE cluster for configuring and monitoring

→ 34bits @ 40 MHz = **1.36Gb/s** on single GBT link

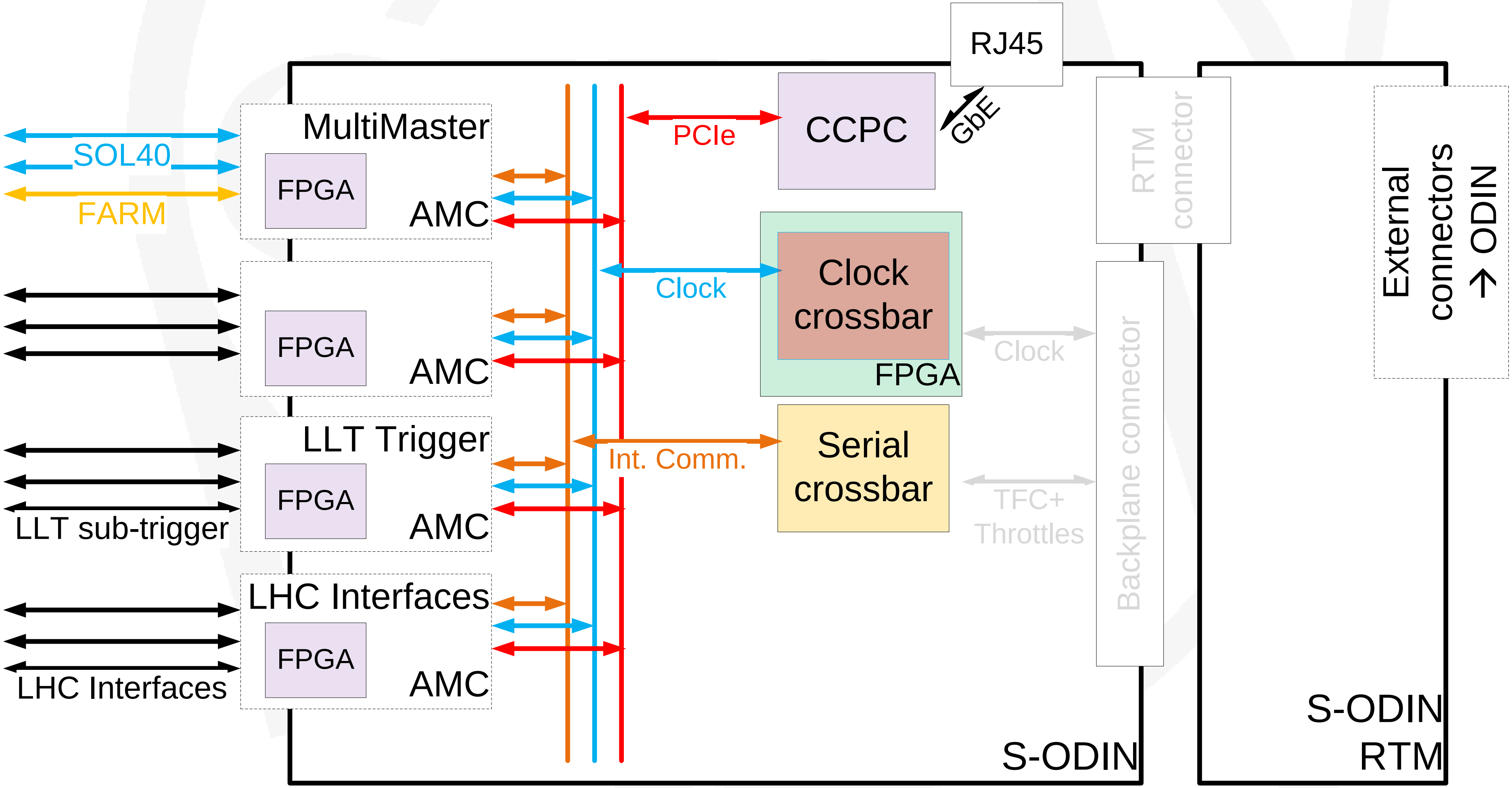
- **~180 Gb/s** for full SOL40 (132 links)
- Single CCPC might become bottleneck
  - → consider overhead of protocol
  - → consider to put 2 CCPCs or extend ECS bandwidth with RTM or even cascade SOL40 boards

→ *Careful calculations to be done by FE people:*

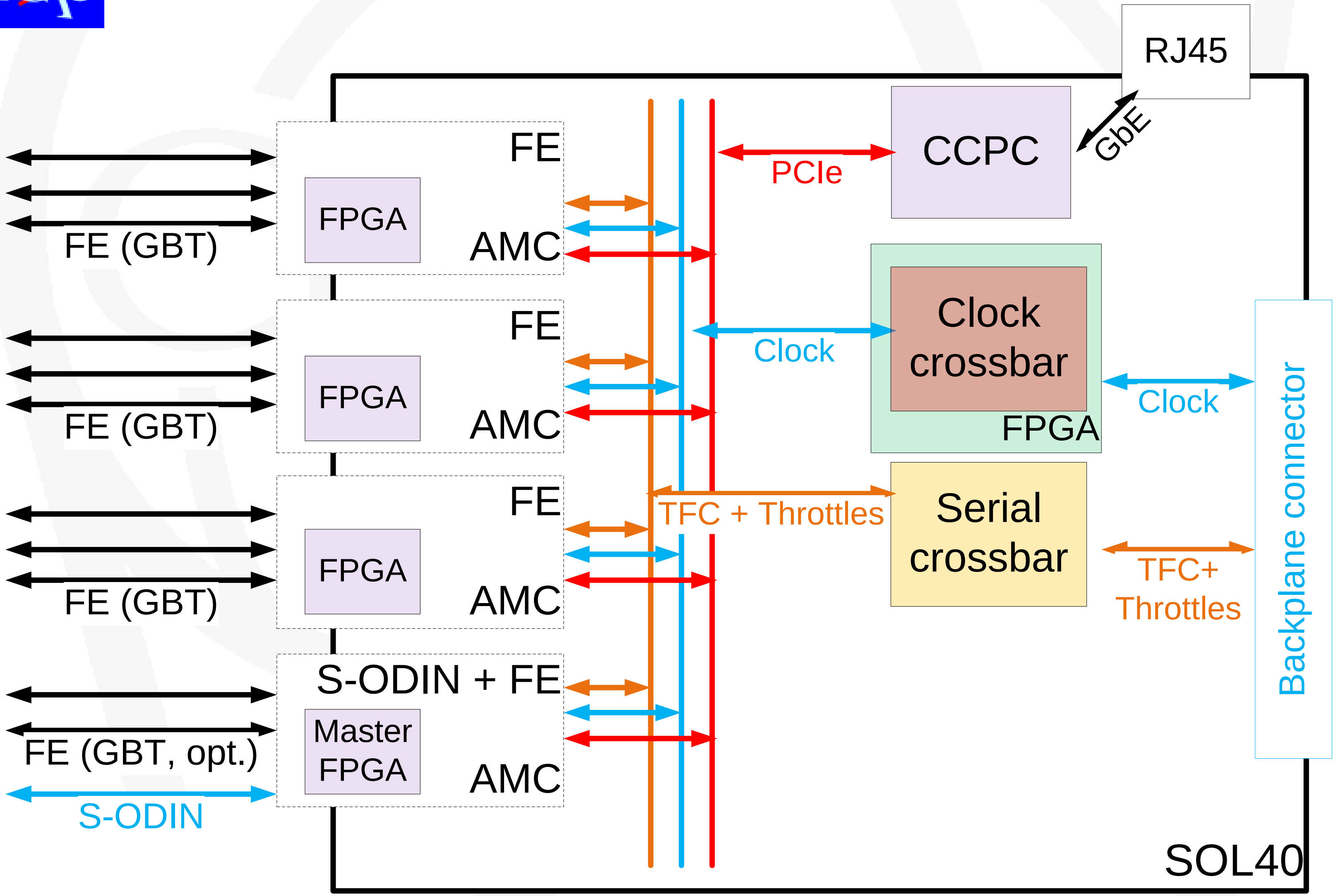
- See Clara's talk



S-ODIN on ATCA



SOL40 on ATCA



Hybrid system

Suggested the idea of a *hybrid system*

reminder: current L0 electronics relying on TTC protocol

→ part of the system runs with old TTC system

→ part of the system runs with the new architecture

How?

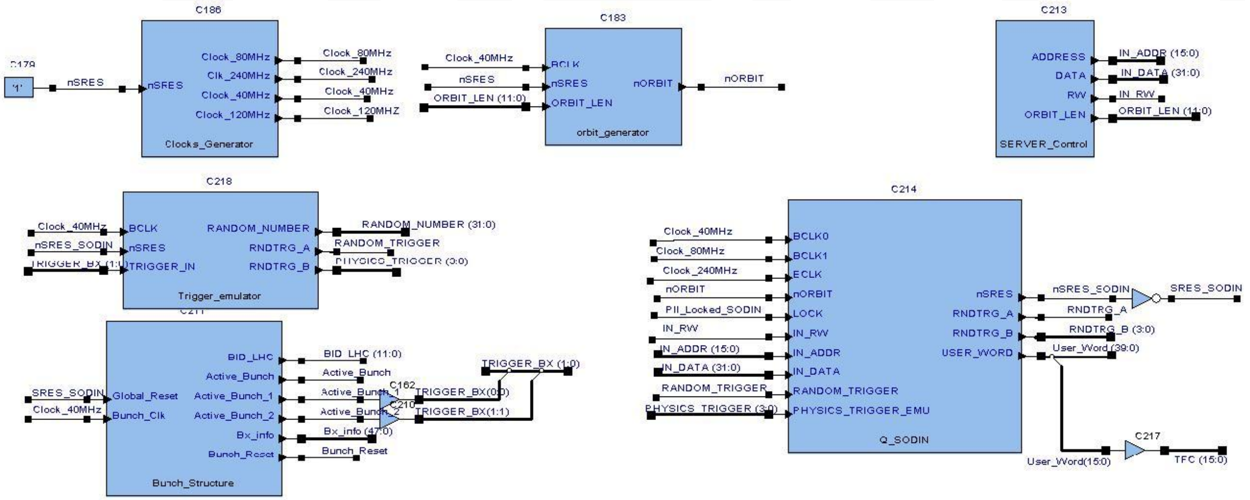
1. Need connection between S-ODIN and ODIN (bidirectional)
   → ODIN has LVDS or ECL inputs/outputs: use dedicated RTM board
2. In an early commissioning phase ODIN is the master, S-ODIN is the slave
   → S-ODIN task would be to distribute new commands to new FE, to new TELL40s, and run processes in parallel
   → ODIN tasks are the ones today + S-ODIN controls the upgraded part
   - ✓ In this configuration, upgraded slice will run at 40 MHz, but positive triggers will come only at maximum current 1.1MHz limit…
     - Great test-bench for development + tests + apprenticeship…
     - Possibility to install an upgraded detector after LS1 as test-bench or even improve LHCb physics programme
3. In the final system, S-ODIN is the master, ODIN is the slave
   → ODIN task is only to interface the L0 electronics path to S-ODIN and to provide clock resets on old TTC protocol

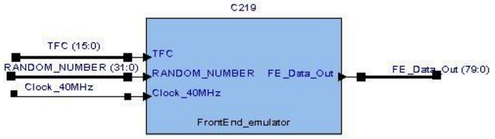
Clock-level simulation framework

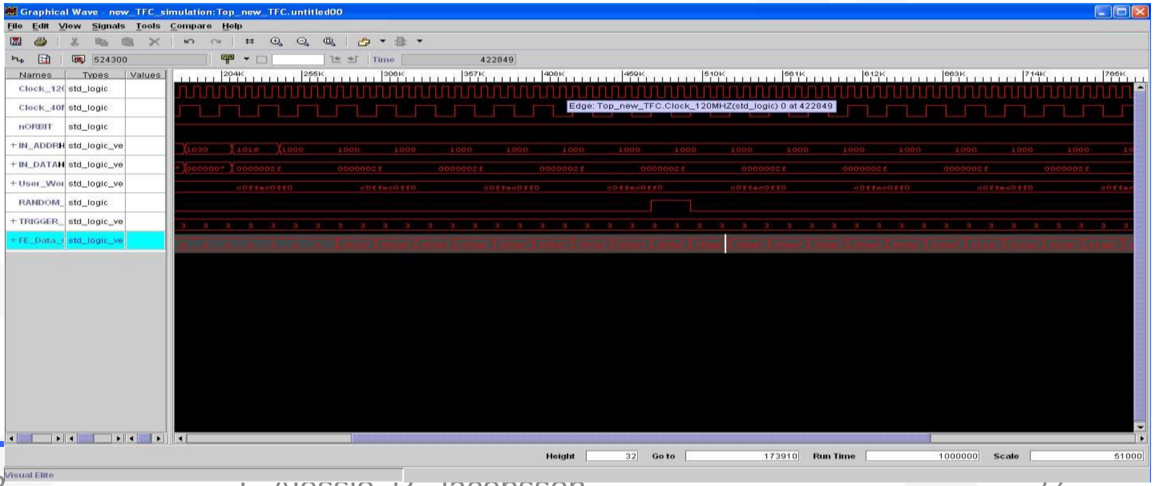
Complete clock-level simulation framework for the development of S-ODIN and SOL40 fw

- Based on Mentor Graphics VisualElite, but easily portable to QSYS
- Already developed in 2009 (see backup slides for some examples)

→ *Pluggable FE emulators*
→ *Based on configurable parameters*
→ *Study occupancies, readout logic*







Planned activities during LS1 & conclusions

New system-level S-TFC specs have been refined and will be extended in detail during LS1

→ LHCb-PUB-2012-001

→ External revision beginning of December as final kick-off

→ *First version* of S-ODIN and SOL40 *firmware* by March/April 2013 to be used with lab setup from Marseille

Defined upgraded readout control specifications for Front-End and Back-End

→ LHCb-PUB-2012-017, discussion with sub-detectors ongoing

→ Development of a clock-level simulation framework:

- S-ODIN and SOL40 firmware development started and ongoing
- plug-in FE emulators to simulate in proper environment and simulate buffer occupancies, DSPs…
- might help while GBT chips are not available

First version of the system will be interfaced to current ODIN in the pit

→ Development of an ATCA RTM board to be interfaced with Marseille's ATCA card and to connect to current ODIN + to upgraded LHC interfaces

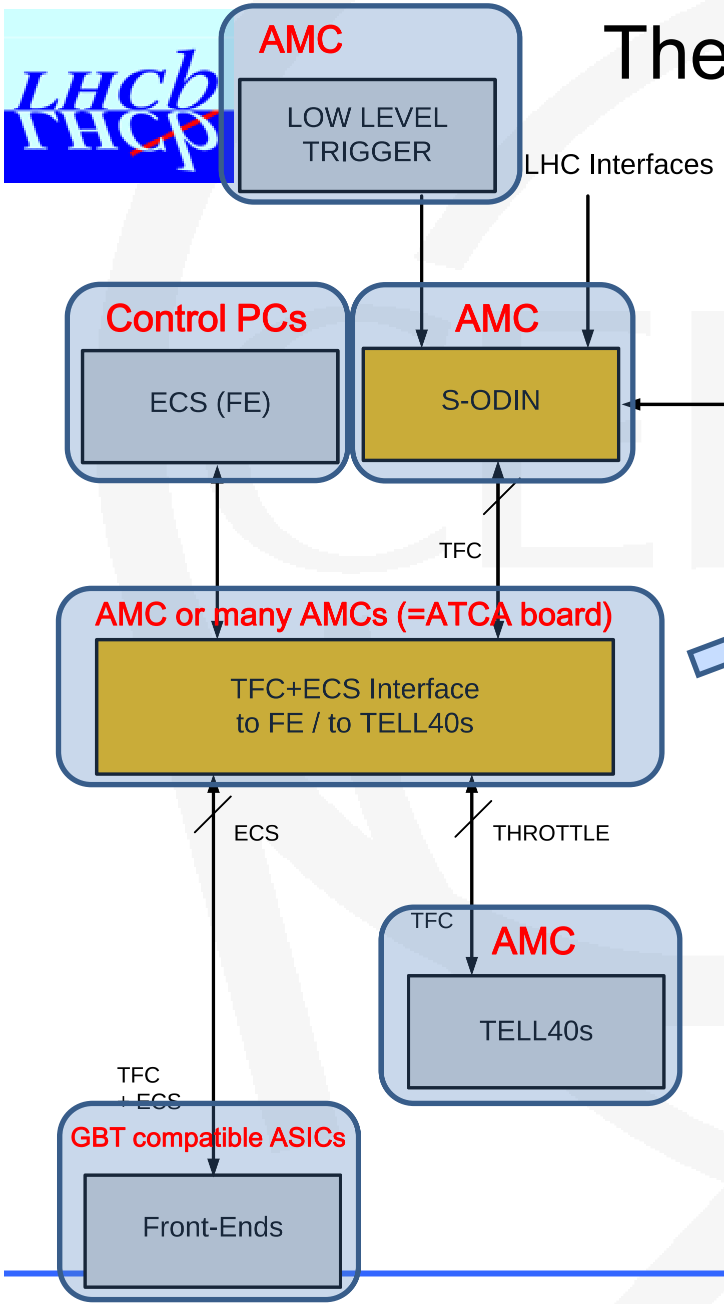
→ scaled commissioning of new system

→ scaled commissioning of new sub-detectors and readout architecture

- This could allow reading out an upgraded sub-detector in parallel *(fear not!)*

Backups

The physical S-TFC at a glance

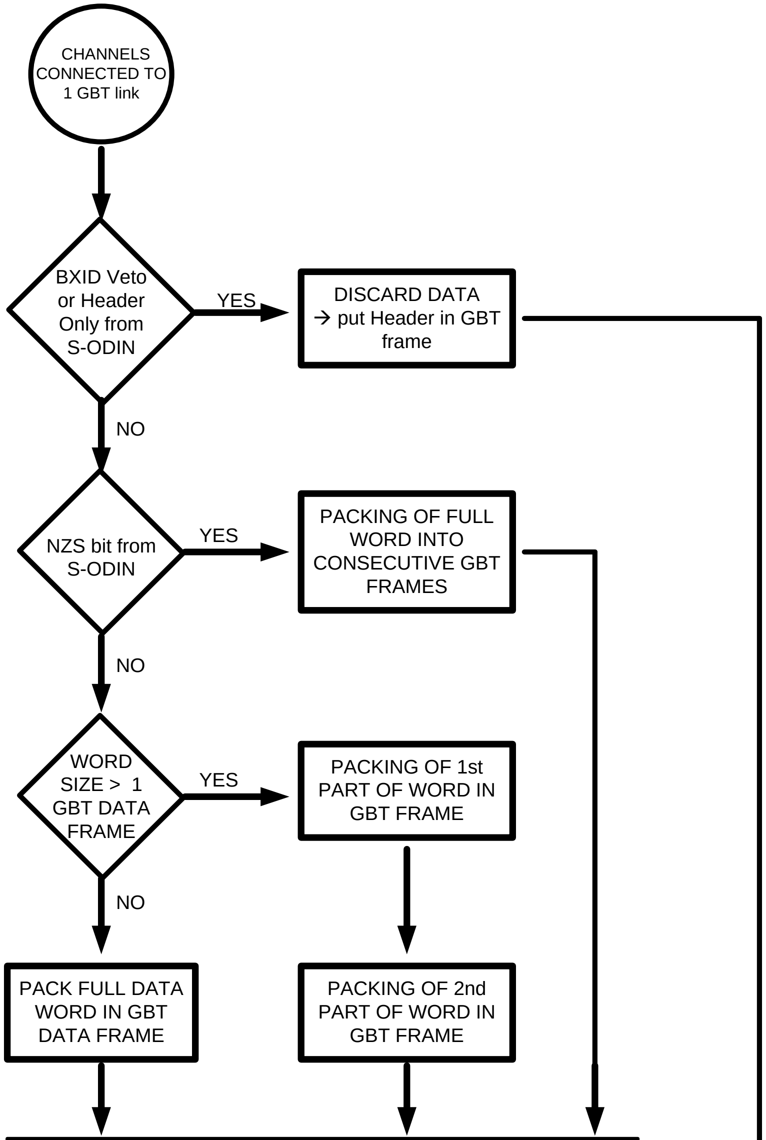


Cascading possible using a dedicated bidirectional link

- Account for latency at each cascaded stage

✓ Physically mapped on ATCA
→ Using Marseille's boards
✓ Including GBT for FE

Example of readout mechanism


CHANNELS CONNECTED TO 1 GBT link
BXID Veto or Header Only from S-ODIN
YES
DISCARD DATA → put Header in GBT frame
NO
NZS bit from S-ODIN
YES
PACKING OF FULL WORD INTO CONSECUTIVE GBT FRAMES
NO
WORD SIZE > 1 GBT DATA FRAME
YES
PACKING OF 1st PART OF WORD IN GBT FRAME
NO
PACK FULL DATA WORD IN GBT DATA FRAME
PACKING OF 2nd PART OF WORD IN GBT FRAME

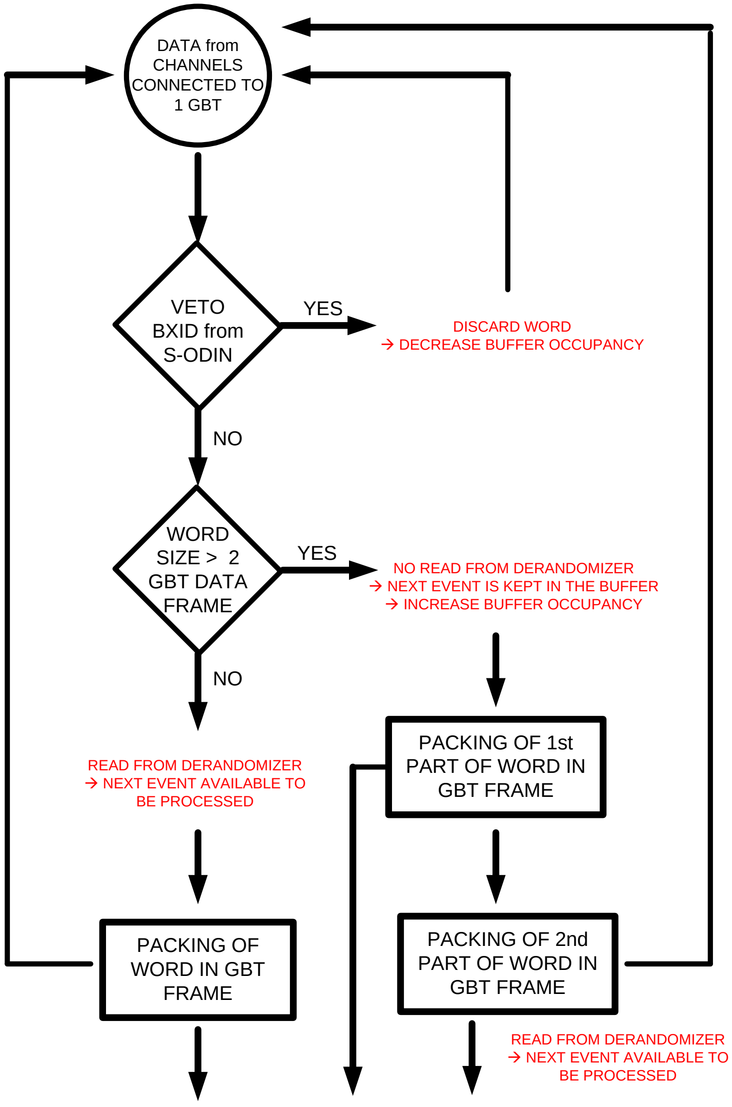
GBT WORD 1	H	EV01 H	DATA WORD of EV 01		
GBT WORD 2	H	DATA WORD of EV 01	EV H 02	DATA WORD...	
GBT WORD 3	H	...OF EV 02	EV H 03	NZS DATA WORD of EV 03	
GBT WORD 4	H	NZS DATA WORD of EV 03			
GBT WORD 5	H	NZS DATA WORD of EV 03	EV H 04	EV H 05	...

GBT FRAME – 84 BITS (example of consecutive words)

This scheme implies that the GBT bandwidth is exploited to its maximum:

- → Pack of events consecutively independently from GBT boundaries
- → Needs buffering to keep events stored waiting to be sent off
- → Header needs more information than just BXID
  - ✓ length of word, type of word…
- → Needs a well-defined truncation scheme and boundaries

Use of VETO BXID in FE



DATA from CHANNELS CONNECTED TO 1 GBT

VETO BXID from S-ODIN

YES

DISCARD WORD
→ DECREASE BUFFER OCCUPANCY

NO

WORD SIZE > 2 GBT DATA FRAME

YES

NO READ FROM DERANDOMIZER
→ NEXT EVENT IS KEPT IN THE BUFFER
→ INCREASE BUFFER OCCUPANCY

NO

READ FROM DERANDOMIZER
→ NEXT EVENT AVAILABLE TO BE PROCESSED

PACKING OF 1st PART OF WORD IN GBT FRAME

PACKING OF WORD IN GBT FRAME

PACKING OF 2nd PART OF WORD IN GBT FRAME

READ FROM DERANDOMIZER
→ NEXT EVENT AVAILABLE TO BE PROCESSED

GBT FRAME – 84 BITS (example of consecutive words)

GBT WORD 1	H	EV H 01	DATA WORD of EV 01		
GBT WORD 2	H	DATA WORD of EV 01	EV H 02	EMPTY	
GBT WORD 3	H	EV H 03	DATA WORD of EV 03		
GBT WORD 4	H	DATA WORD of EV 03	EV H 04	DATA WORD of EV 04	
GBT WORD 5	H	DATA WORD of EV 04	EMPTY		

1- if word size > 2 GBT frame
→ buffer occupancy increase
→ store event, without reading

2- if word size < 2 GBT frame
→ buffer occupancy is steady
→ read event, store event

3- if VETO BXID = ‘1’
→ buffer occupancy decrease
→ discard event

→ Build GBT words consecutively
→ Truncation when buffer full and can’t store events anymore
→ Which depth??
→ Simulation

«BX VETO»

@ 40 MHz

S-ODIN vetoes the readout of an event

↓

Based on filling scheme

→ Used to control the rate of readout while < 30MHz

→ INDEPENDENT FROM LLT DECISION!

↓

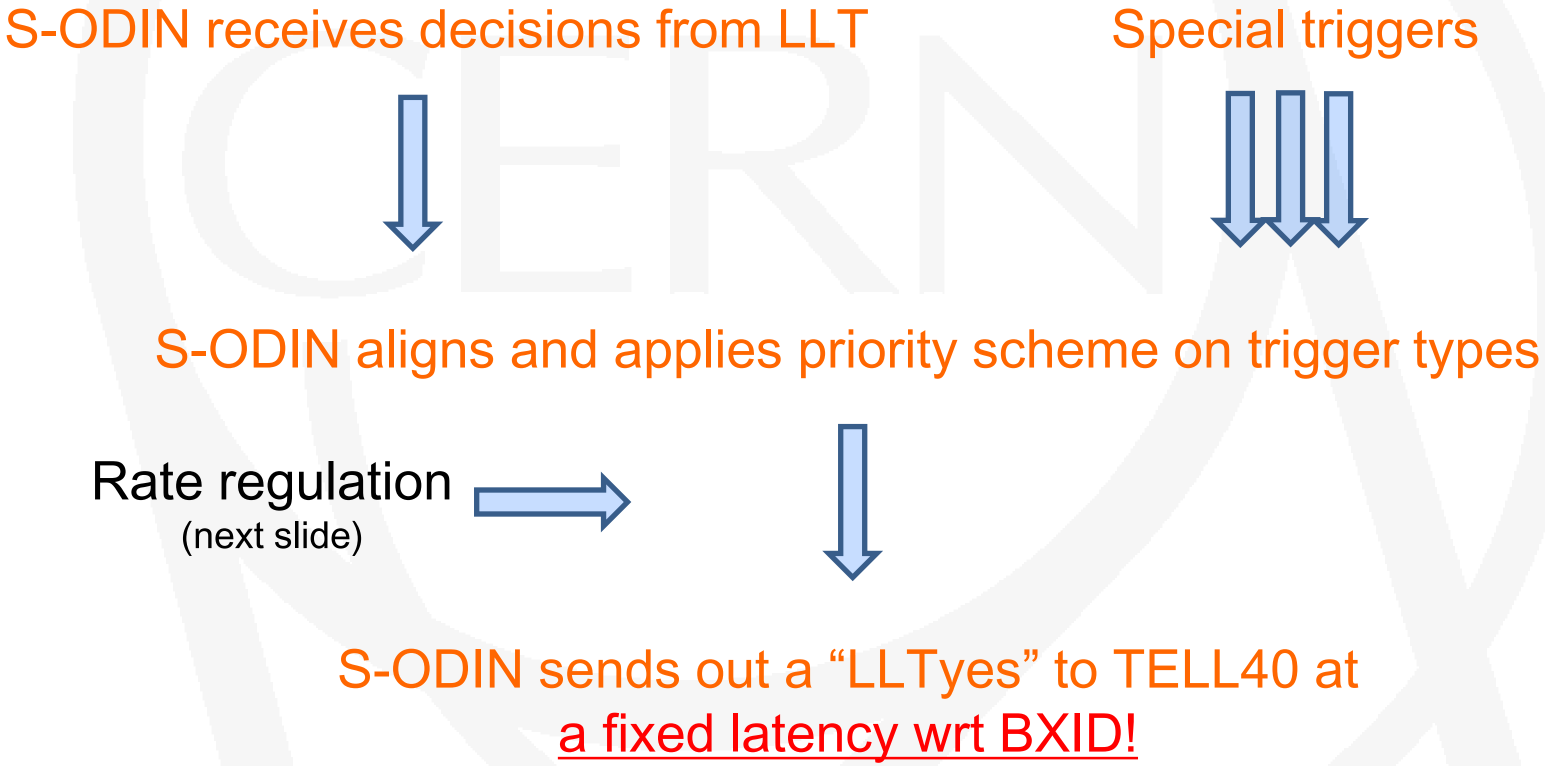
FE can use this info to recuperate time for processing events

→ Only header for vetoed events

→ Flexible packing of data into GBT frame

Sending a «LLTyes»

@ 40 MHz



S-ODIN receives decisions from LLT

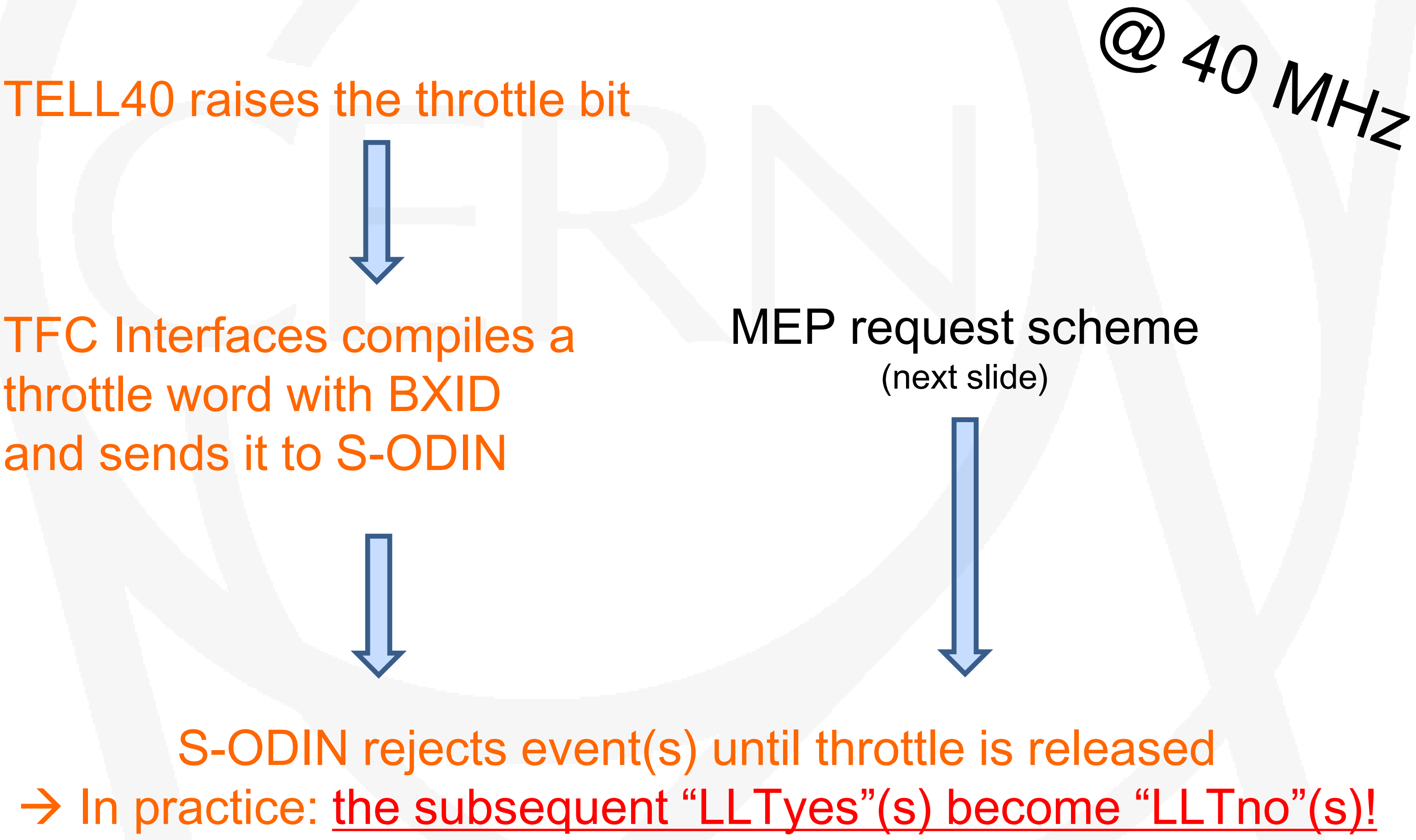
Special triggers

S-ODIN aligns and applies priority scheme on trigger types

Rate regulation
(next slide)

S-ODIN sends out a “LLTyes” to TELL40 at
a fixed latency wrt BXID!

Rate regulation



@ 40 MHz

TELL40 raises the throttle bit

TFC Interfaces compiles a throttle word with BXID and sends it to S-ODIN

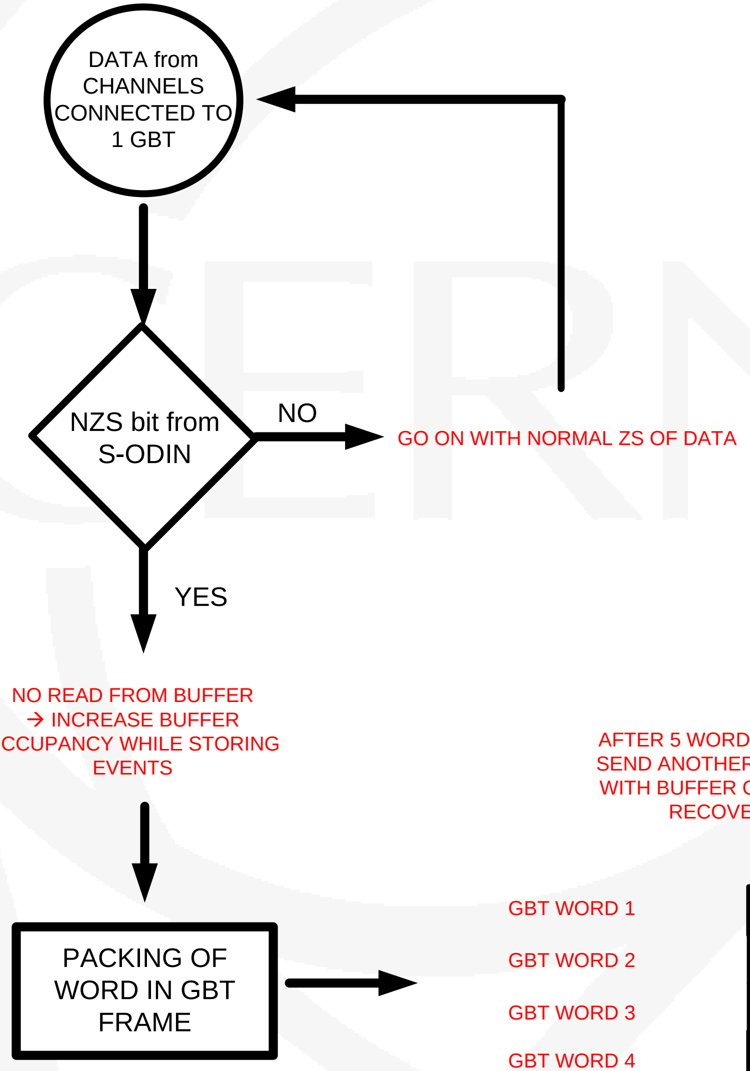
MEP request scheme

(next slide)

S-ODIN rejects event(s) until throttle is released

→ In practice: the subsequent “LLTyes”(s) become “LLTno”(s)!

Use of NZS bit



DATA from CHANNELS CONNECTED TO 1 GBT

NZS bit from S-ODIN

NO

GO ON WITH NORMAL ZS OF DATA

YES

NO READ FROM BUFFER → INCREASE BUFFER OCCUPANCY WHILE STORING EVENTS

PACKING OF WORD IN GBT FRAME

AFTER 5 WORDS READY TO SEND ANOTHER NZS EVENT WITH BUFFER OCCUPANCY RECOVERED!

GBT FRAME – 84 BITS (example of consecutive words)

GBT WORD 1	H	EV H 01	NZS DATA WORD of EV 01		
GBT WORD 2	H	NZS DATA WORD of EV 01			
GBT WORD 3	H	NZS DATA WORD of EV 01			
GBT WORD 4	H	NZS DATA WORD of EV 01	EV H 02		
GBT WORD 5	H	EV H 03	EV H 04	EV H 05	EMPTY

Event Destination and Farm Load Control

The current system operates in a powerful mixture of *push and pull protocol* controlled by ODIN (current RS) :

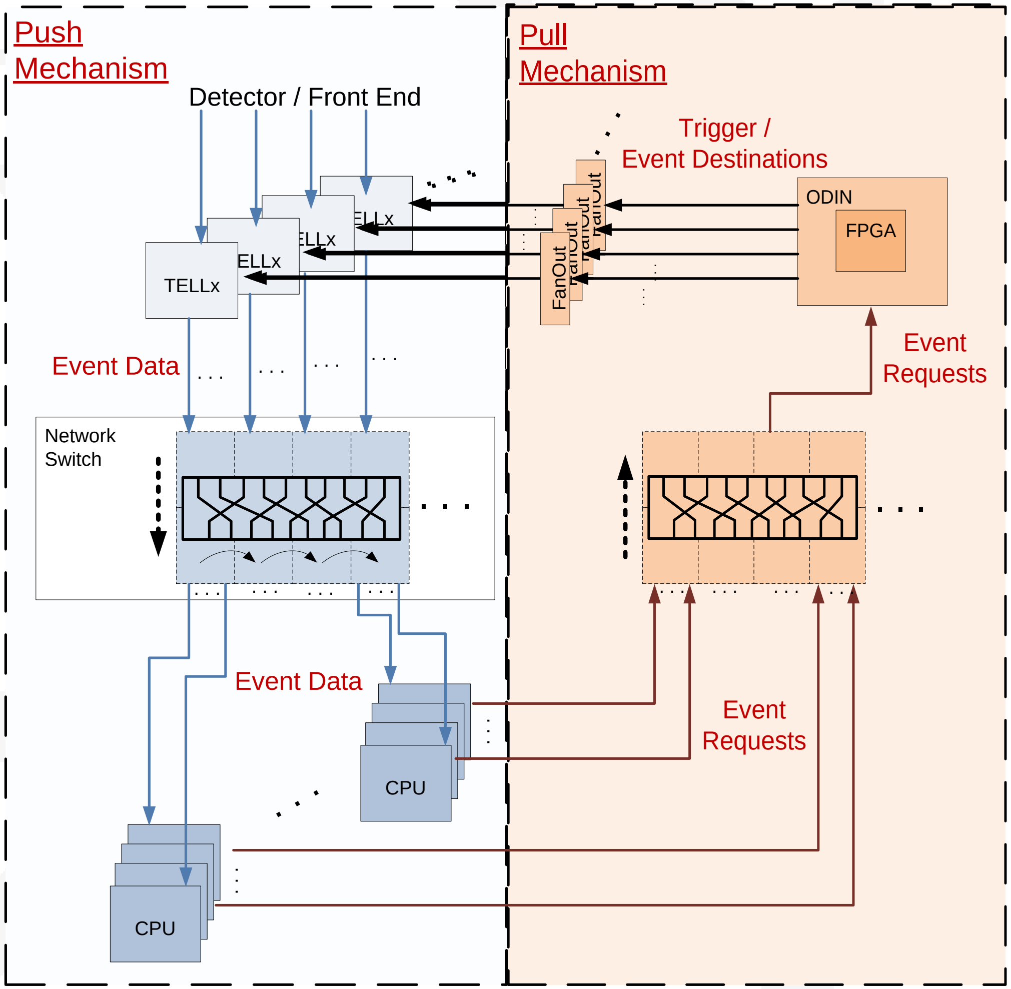
→ Asynchronous pull mechanism

→ “Push” driven by trigger type and destination command

→ Note: LHCb-PUB-2011-023

→ 4 years faultless operation

Similar proposal for upgrade



Event Destination and Farm Load Control

Central FPGA based implementation

- Extreme reliability, flexibility, speed, controllable latencies

→ Central event packing control

- Different trigger types and destination types
- Variable MEP packing factor

→ Dynamic rate regulation as function of farm rate capacity

- Accounts for statistical variations in processing time

→ Dynamic handling of farm nodes in-flight

- Processing blockages, failures, interventions
- All impacts on rate capacity handled automatically
- As soon as nodes are recovered, included automatically in-flight by event request mechanism

→ Minimal event loss and accurate dead-time counting

Contrary to conventional pull scheme, this is robust against event request packet losses

Event Destination and Farm Load Control

Buffer requirement trivia

- Readout boards: ~1.5 MEPs per link
- Network: Some naïve assumptions
  - Rate: 30 MHz
  - MEP packing factor 10 → 3 MHz MEPs and 3 MHz MEP Requests

    → Current ODIN can handle 1.8 MHz of MEP Requests (ODIN <-> FARM is 1 GbE…)
  - Event size 100 kB → 1 MB / MEP
  - Farm nodes 5000 → 600 MEPs/node/s → 1.7ms / MEP
  - Switch subunit sharing resources: 50 links / subunit → 100 subunit

  → 30 kHz of MEPs per switch subunit

  → Every 1.7ms, 50 MEPs to juggle with → <buffer> = O("50 MB")

  → True need of memory depends on statistical variation of HLT processing time and "local farm derandomizer"
- Farm nodes: few MEPs in local derandomizing buffer

In our view, this looks like a straight-forward implementation…

S-ODIN data bank

S-ODIN transmits a data bank for each accepted event in a MEP

→ Run number, event identifier, orbit number, bunch identifier, UTC time, event type, trigger mask, bunch crossing information

+

S-ODIN data bank and LLT data bank is merged

(reminder: LLT is in same board as new S-ODIN)

→ Info about timestamp, trigger type, bxid, trigger decision…

- Mostly like now

→ Will need at least 10GbE connection directly to FARM

- what about 40GbE…? ☺
- has to allow bandwidth partitioning as well
  - In fact «several» 10GbE (n*10GbE…),

→ reduced bank size for local tests

- No LLT for instance

Partitioning

Partitioning is assured by having:

✓ Many instances of S-ODIN codes inside main FPGA

✓ Switching is done inside main FPGA

- Simply assure that TFC information are sent to right output

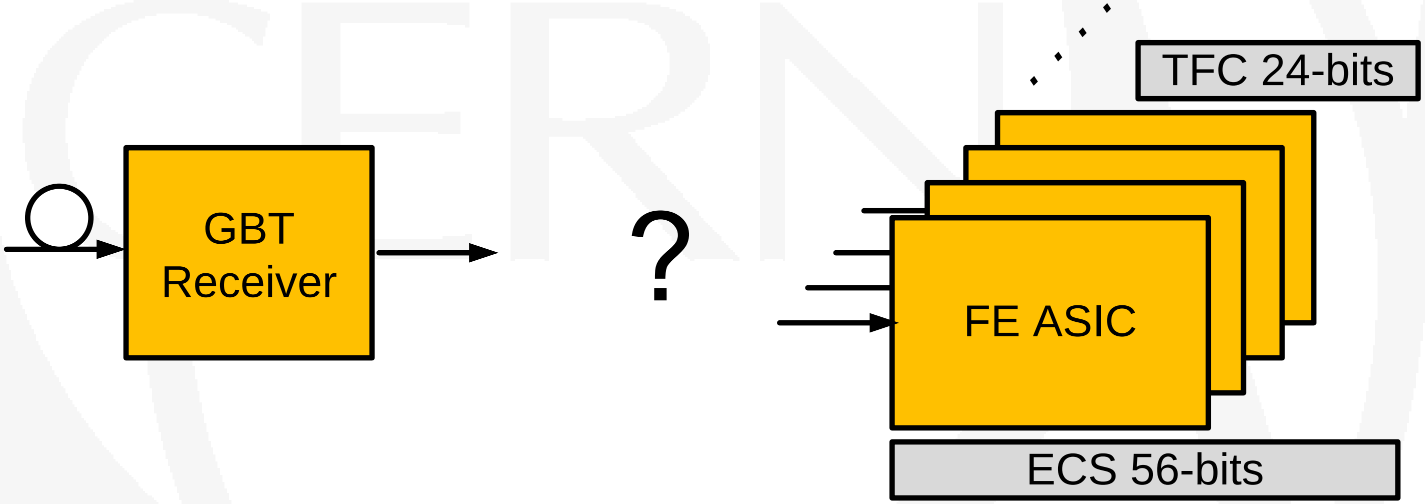
✓ TFC+ECSInterface «interfaces» S-ODIN with partitioned FE+TELL40 slice(s)

- Logical distribution of TFC+ECSInterfaces in TELL40s crates

- Important to respect the «logical concept» of partitioning
  → Should not span over different sub-systems with same TFC+ECSInterface…
  → Need at least one dedicated TFC+ECSInterface for each sub-system

TFC+ECS to FE

Remember: load on TFC+ECSInterface CCPC to configure entire FE slice!!

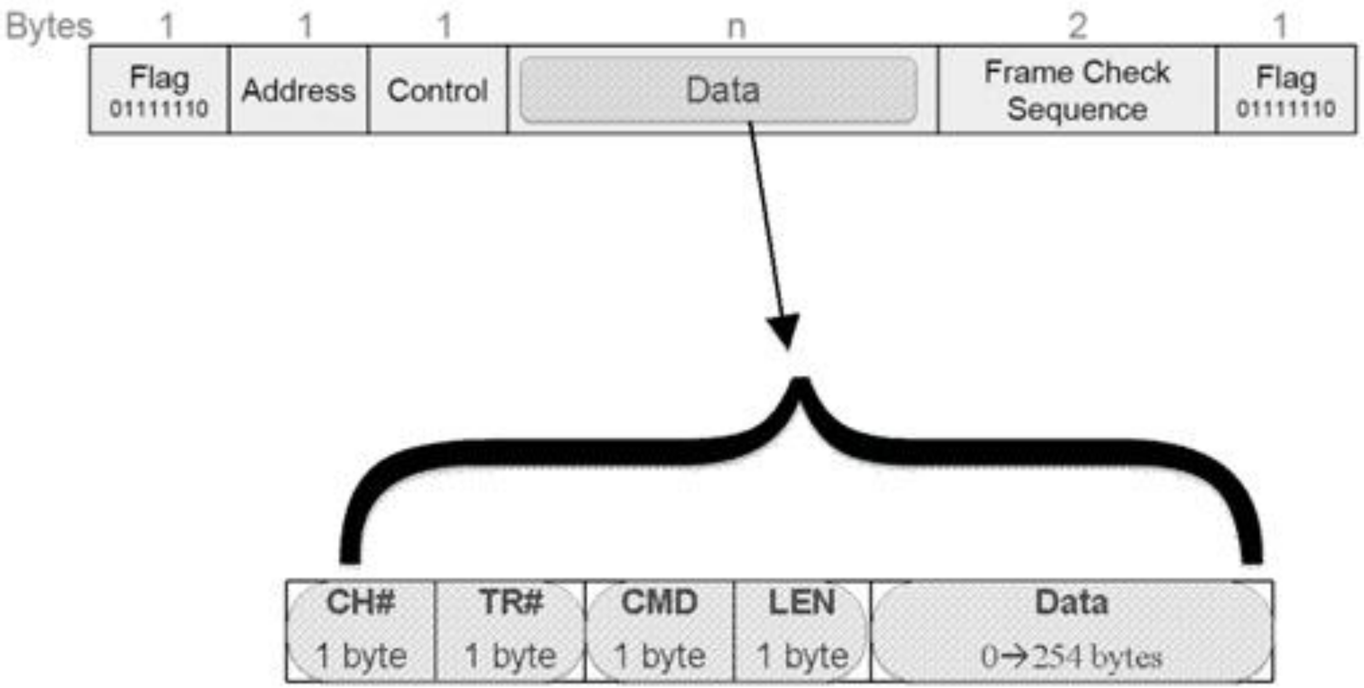


Need to strip out TFC information and ECS information from GBT word

TFC is the same for all FE channels → *would need only some kind of fan-out*

ECS needs addressing scheme based on GBT-SCA
→ *many SCA chips connected to a single GBT frame!*

Extra slide - GBT-SCA Packet



Bytes	1	1	1	n	2	1
	Flag 01111110	Address	Control	Data	Frame Check Sequence	Flag 01111110

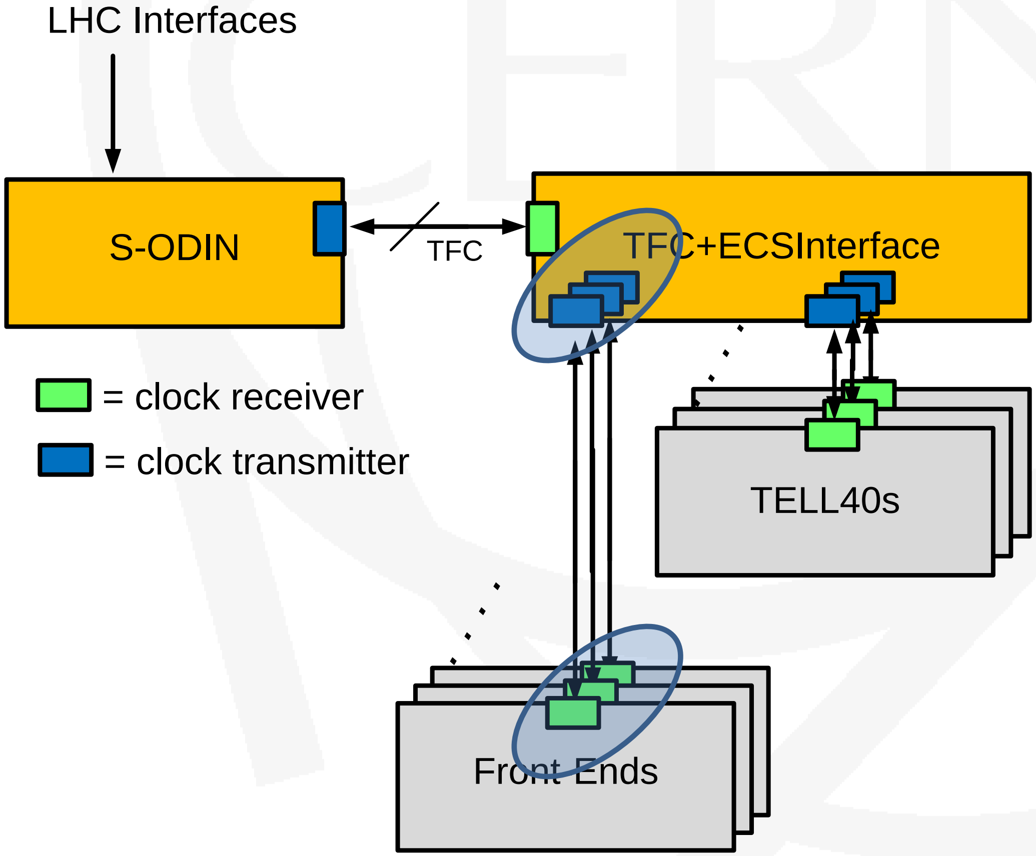
CH#	TR#	CMD	LEN	Data
1 byte	1 byte	1 byte	1 byte	0→254 bytes

Easy: the GBT-SCA has everything set up (on paper…)

→ Only thing is to build proper protocol

→ FE designer selects the bus + addressing scheme, we built a generic entity in TFC+ECSInterface to drive it.

Clock distribution and phase/latency control



1. at the FE: GBT

→ Does the job for us

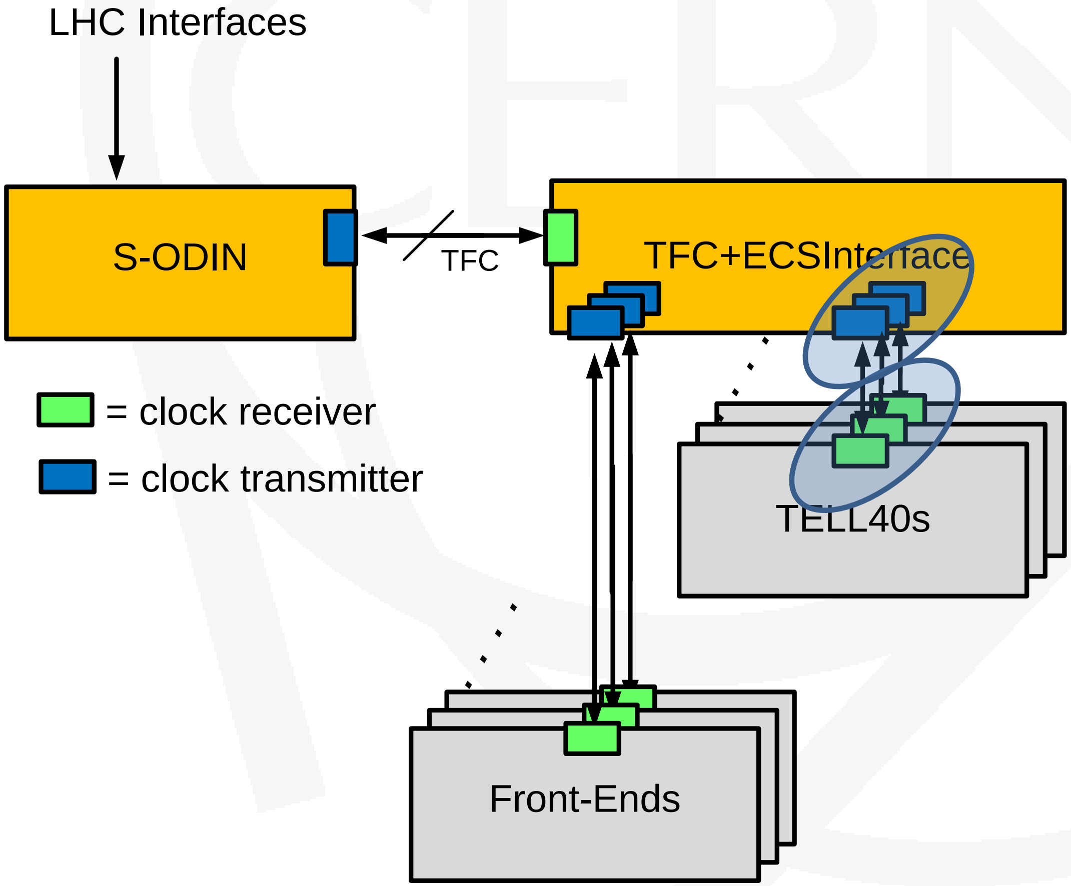
→ control of fine phase + latency at the FE + minimize jitter

→ No problem in ECS monitoring

- Simply decoding GBT protocol in TFC+ECSInterface FPGA

  → No need of fine phase or latency control for ECS.

Clock distribution and phase/latency control

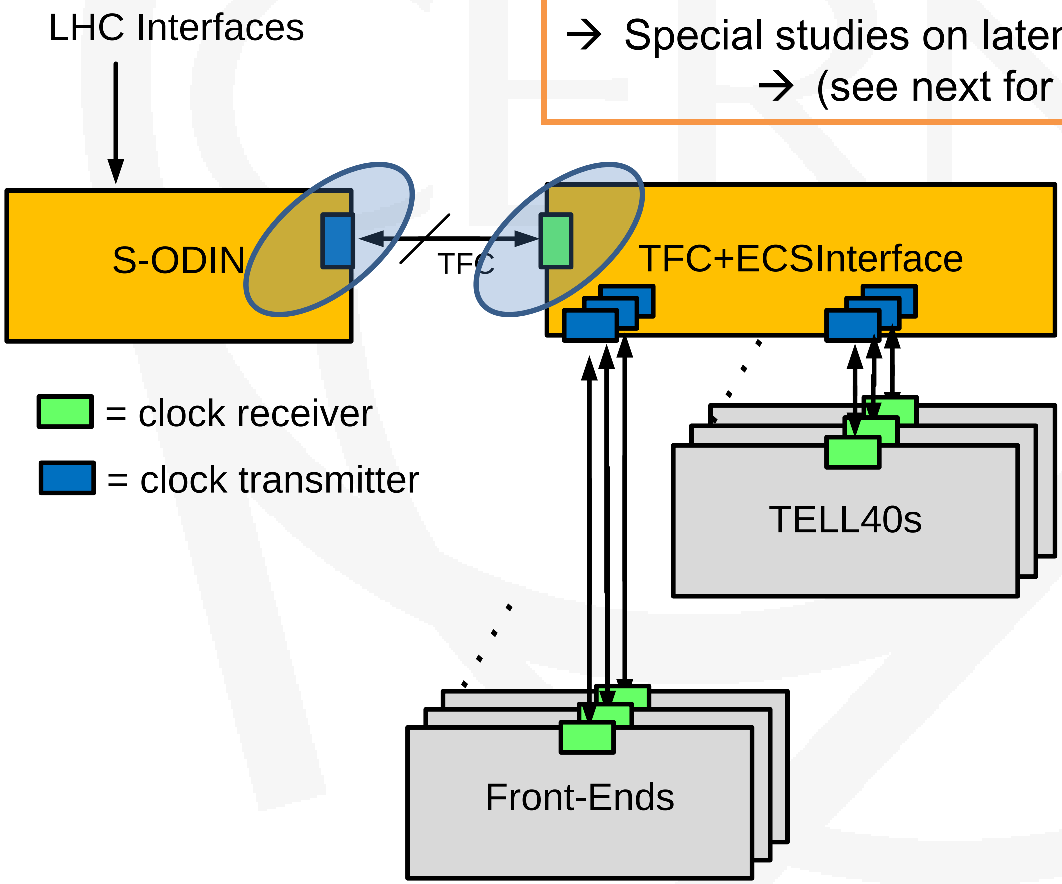


2. ATCA backplane

→ Does the job for us

- → control of latency
- → jitter and fine phase less of an issue
- → Effectively is an FPGA-to-FPGA link on backplane dedicated lines
- → To be checked: jitter on backplane!

Clock distribution and phase/latency control



🟩 = clock receiver

🟦 = clock transmitter

3. FPGA to FPGA transceivers

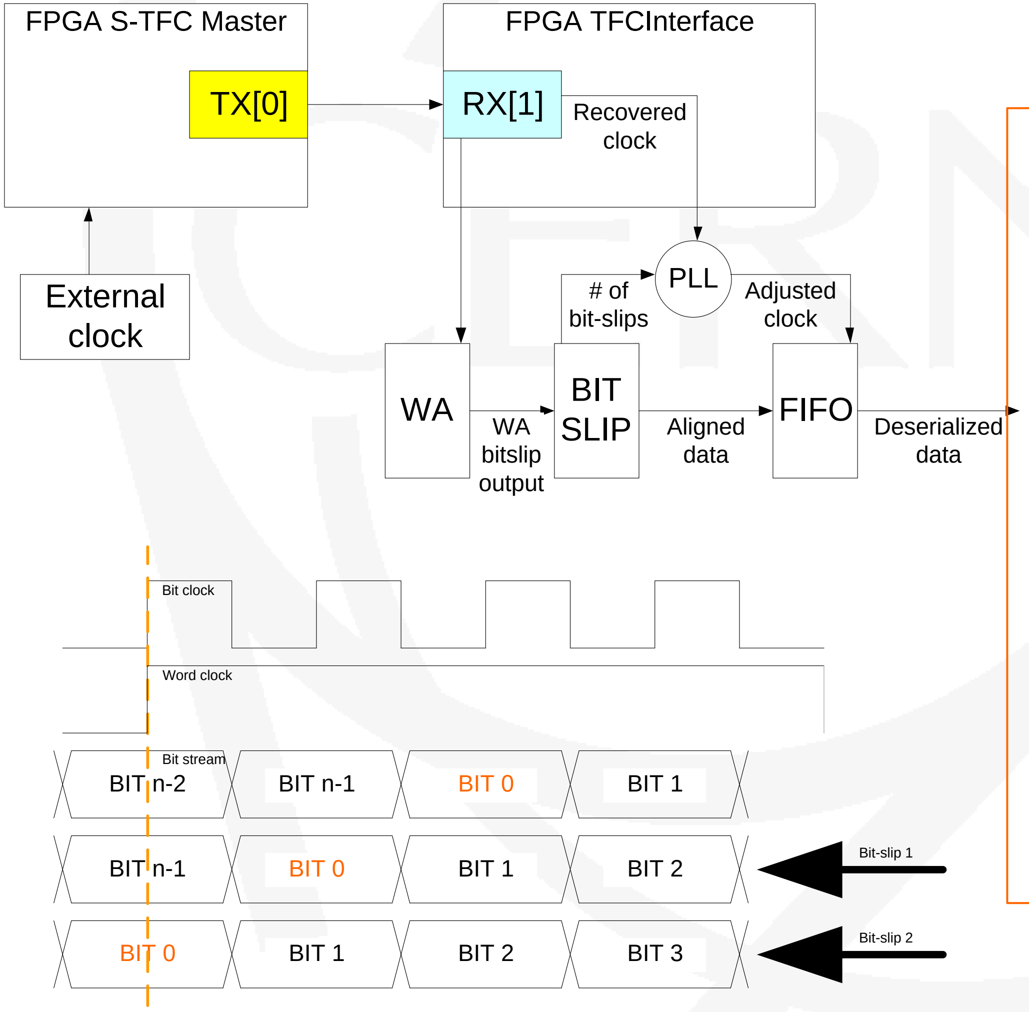
→ Special studies on latency and phase alignment

→ (see next for preliminary tests)

→ control of fine phase and latency

→ minimize jitter

Latency and phase control



Investigate with ALTERA and Jean-Pierre two years ago

- → Trick is to use the bitslip management already in FPGA
  - ✓ Issue: need 8b/10b encoding
- → Set a reference value
- → Bitslip the word by the difference between the measured value and the reference value to re-align the words

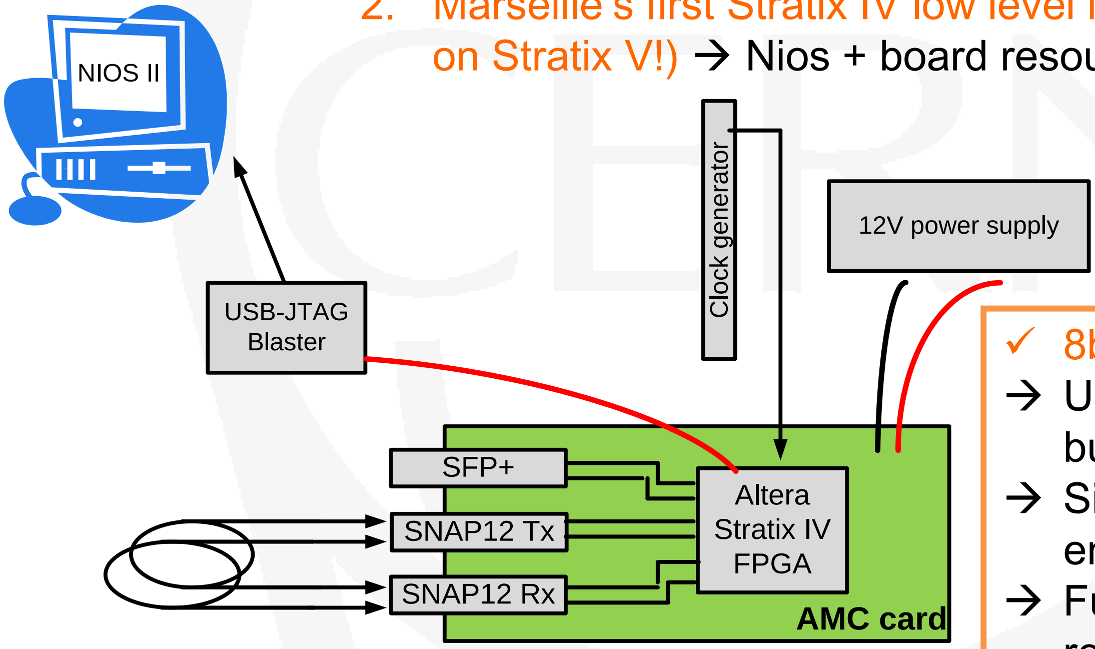
First implementation seems to work

- → Stress tests various clock loss/recovering situations
- → Estimation of maximum delay in recovering

Clock distribution and phase/latency control

First preliminary tests on phase/latency control using:

1. Marseille's first AMC prototype with ALTERA Stratix IV
2. Marseille's first Stratix IV low level interfaces (will need re-validation on Stratix V!) → Nios + board resources interfaces + thanks to Marseille team!



✓ 8b/10b protocol: no problem

→ Using «word alignment» from Altera GX buffer + «deterministic latency»

→ Simply add Ctrl word for the 8b/10b encoder: 2bits more

→ Full reproducibility upon power-up and resets and reconfiguration

✓ FPGA-to-FPGA GBT protocol: ok, but needs special frame alignment

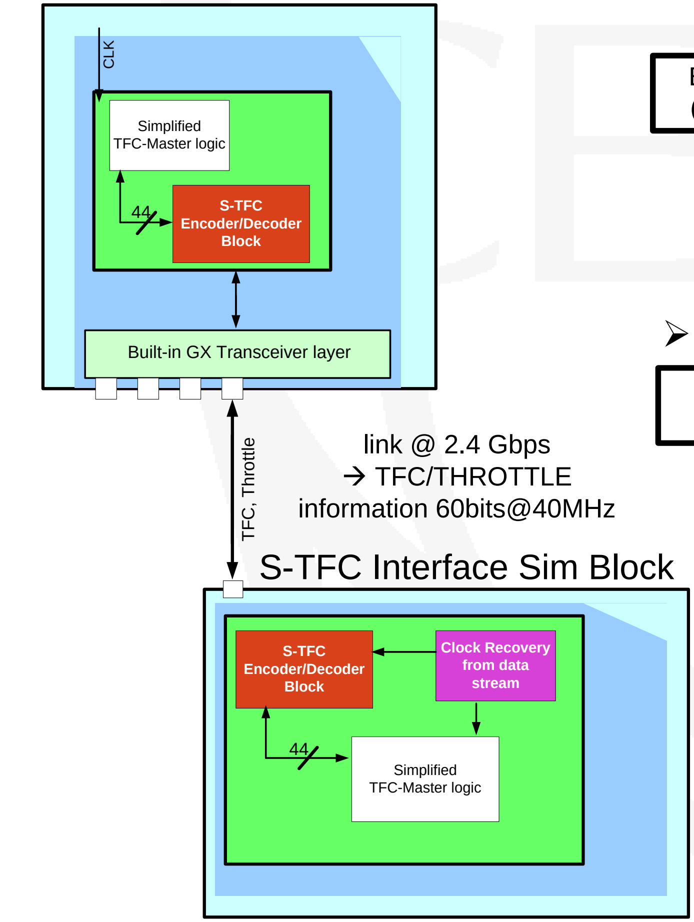
→ No deterministic latency if no special words are sent!

→ Needs a special word (10 bits minimum) at power-up/after reset/after reconfiguration for the GX buffer to realign to the beginning of the frame + «deterministic latency»

- First preliminary tests were ok, ***but needs more validation under stress tests!***

1. Simulating S-TFC links

S-TFC Master Simulation Block



link @ 2.4 Gbps
→ TFC/THROTTLE
information 60bits@40MHz

S-TFC Interface Sim Block

S-ODIN ←→ SOL40 link **preliminary** protocol

- ➢ TFC control fully synchronous 60bits@40MHz → 2.4 Gb/s (max 75 bits@ 40 MHz → 3.0 Gb/s)

EVENT ID (4-12 bits)	TFC information (40-32 bits)	ReedSolomon-FEC (16 bits)

1. Reed Solomon-encoding used on TFC links for maximum reliability (header ~16 bits) (ref. CERN-GBT)
2. Asynchronous readout → TFC info must carry Event ID

- ➢ Throttle(“trigger”) protocol

EVENT ID (4-12 bits)	THROTTLE information (20 bits)	OTHERS	ReedSolomon-FEC (16 bits)

1. Must be synchronous and carry Event ID
   → Protocol will require alignment similar to TFC protocol

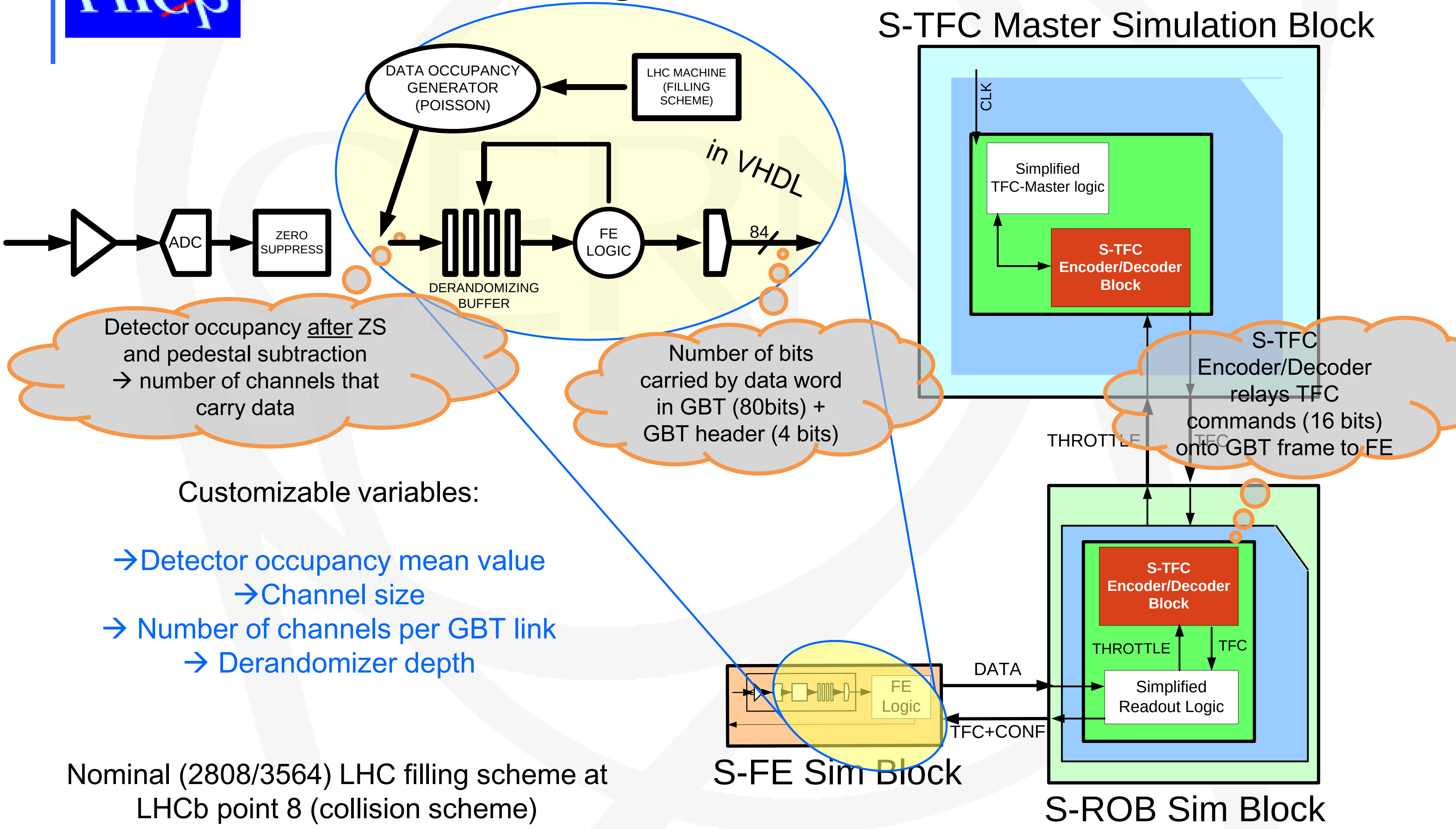
The links are successfully simulated

→ The clock recovery from data stream needs additional firmware logic (thanks for disclosed info from Altera) in order to control the latency and the phase of the clock w.r.t. the data stream

→ Plan to test the link with a (real) board developed in Marseille

N.B. in the full simulation framework, the links are not simulated for “time consuming” reasons but emulated

2. Simulating the RO architecture


S-TFC Master Simulation Block
DATA OCCUPANCY GENERATOR (POISSON)
LHC MACHINE (FILLING SCHEME)
in VHDL
ADC
ZERO SUPPRESS
DERANDOMIZING BUFFER
FE LOGIC
84
CLK
Simplified TFC-Master logic
S-TFC Encoder/Decoder Block
THROTTLE
TFC
S-TFC Encoder/Decoder Block
THROTTLE
TFC
Simplified Readout Logic
DATA
TFC+CONF
FE Logic
S-FE Sim Block
S-ROB Sim Block

Detector occupancy after ZS and pedestal subtraction → number of channels that carry data

Number of bits carried by data word in GBT (80bits) + GBT header (4 bits)

S-TFC Encoder/Decoder relays TFC commands (16 bits) onto GBT frame to FE

Customizable variables:

→Detector occupancy mean value

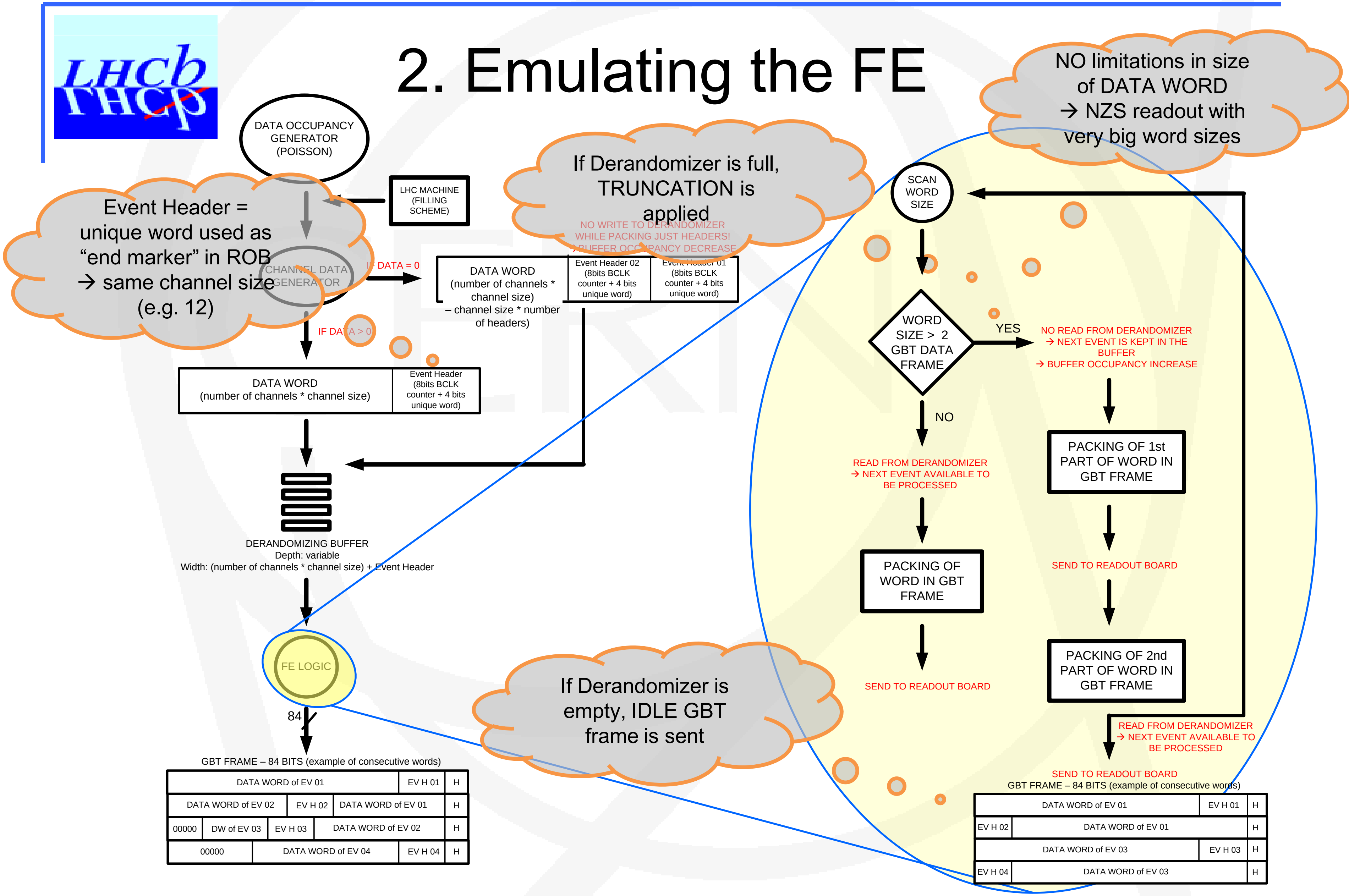
→Channel size

→ Number of channels per GBT link

→ Derandomizer depth

Nominal (2808/3564) LHC filling scheme at LHCb point 8 (collision scheme)

2. Emulating the FE



NO limitations in size of DATA WORD
→ NZS readout with very big word sizes
DATA OCCUPANCY GENERATOR (POISSON)
LHC MACHINE (FILLING SCHEME)
Event Header = unique word used as “end marker” in ROB
→ same channel size (e.g. 12)
CHANNEL DATA GENERATOR
IF DATA = 0
IF DATA > 0
If Derandomizer is full, TRUNCATION is applied
NO WRITE TO DERANDOMIZER WHILE PACKING JUST HEADERS!
→BUFFER OCCUPANCY DECREASE
DATA WORD (number of channels * channel size) – channel size * number of headers)
Event Header 02 (8bits BCLK counter + 4 bits unique word)
Event Header 01 (8bits BCLK counter + 4 bits unique word)
DATA WORD (number of channels * channel size)
Event Header (8bits BCLK counter + 4 bits unique word)
DERANDOMIZING BUFFER
Depth: variable
Width: (number of channels * channel size) + Event Header
FE LOGIC
84
If Derandomizer is empty, IDLE GBT frame is sent
SCAN WORD SIZE
WORD SIZE > 2 GBT DATA FRAME
YES
NO READ FROM DERANDOMIZER
→ NEXT EVENT IS KEPT IN THE BUFFER
→ BUFFER OCCUPANCY INCREASE
PACKING OF 1st PART OF WORD IN GBT FRAME
SEND TO READOUT BOARD
PACKING OF 2nd PART OF WORD IN GBT FRAME
READ FROM DERANDOMIZER
→ NEXT EVENT AVAILABLE TO BE PROCESSED
SEND TO READOUT BOARD
NO
READ FROM DERANDOMIZER
→ NEXT EVENT AVAILABLE TO BE PROCESSED
PACKING OF WORD IN GBT FRAME
SEND TO READOUT BOARD

GBT FRAME – 84 BITS (example of consecutive words)

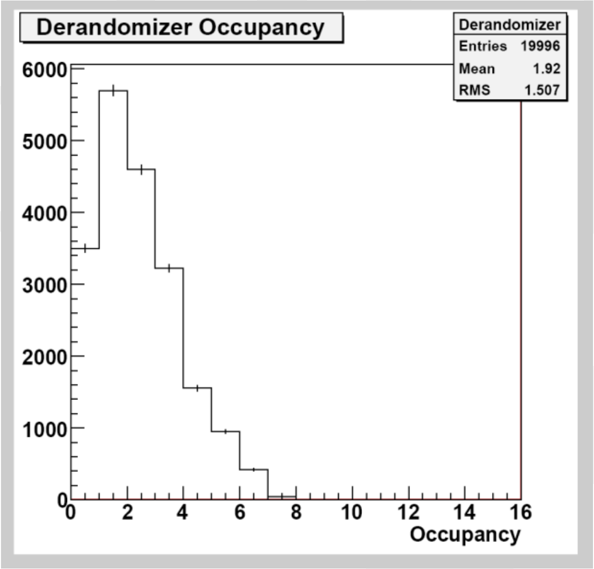
DATA WORD of EV 01	EV H 01	H		
DATA WORD of EV 02	EV H 02	DATA WORD of EV 01	H	
00000	DW of EV 03	EV H 03	DATA WORD of EV 02	H
00000	DATA WORD of EV 04	EV H 04	H	

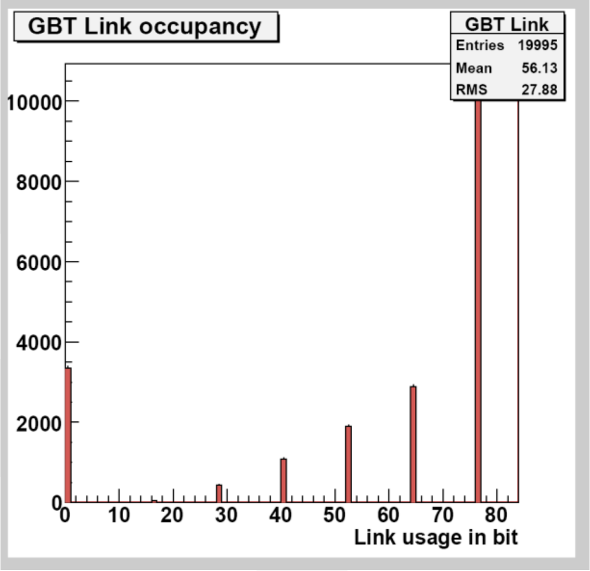
GBT FRAME – 84 BITS (example of consecutive words)

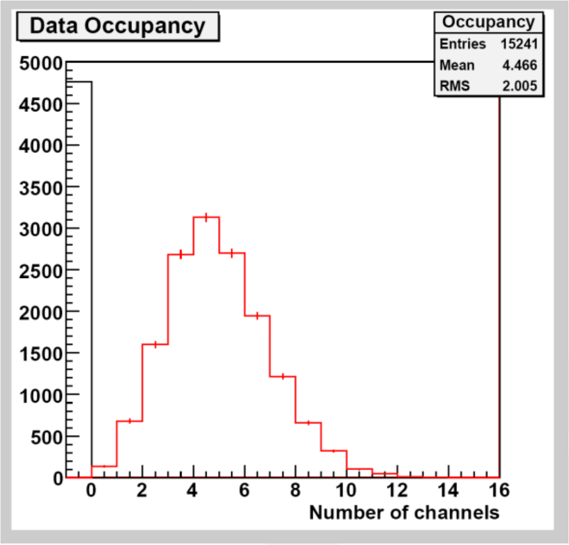
DATA WORD of EV 01	EV H 01	H
EV H 02	DATA WORD of EV 01	H
DATA WORD of EV 03	EV H 03	H
EV H 04	DATA WORD of EV 03	H

2. Results (Example 1)

Variables applied:
<Detector occupancy> = 30%
#channels / GBT link = 15
→ ~ 4.5 channels / GBT after ZS
Derandomizer depth = 16
Channel size = 12
(e.g. ADDR = 4bits + ADC_DATA = 8bits)



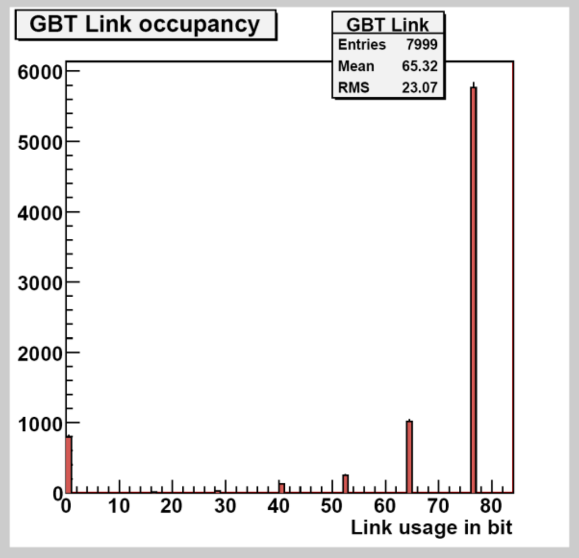


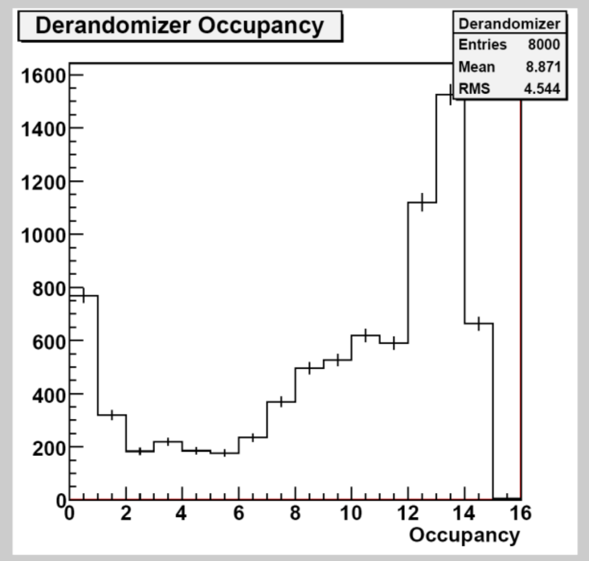


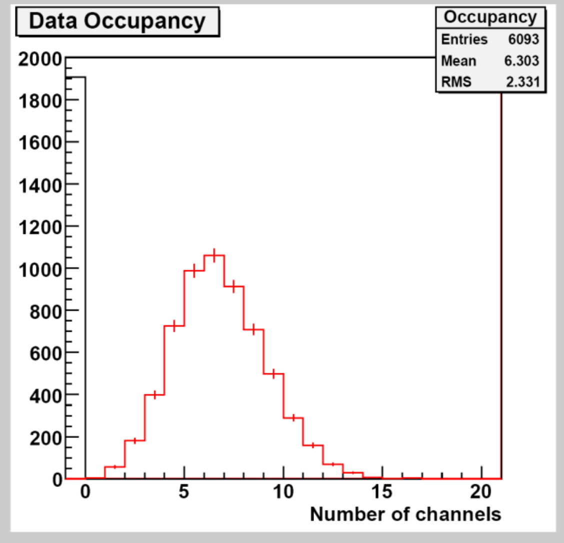
→ No truncation occurred: system undercommitted (overdesigned)
→ ~145kbits of channel data sent through one GBT link over full LHC turn: 48.3% of GBT link bandwidth + 14% Event Header + 4.8% GBT header (unavoidable)

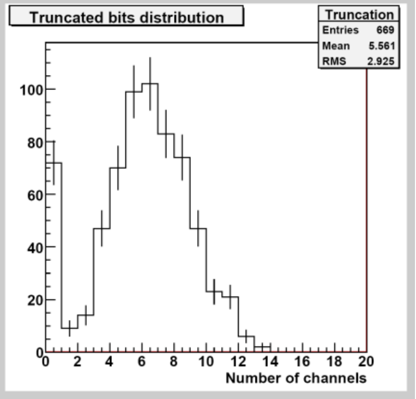
2. Results (Example 2)

Variables applied:
<Detector occupancy> = 30%
#channels / GBT link = 21
→ ~ 6.3 channels / GBT after ZS
Derandomizer depth = 16
Channel size = 12
(e.g. ADDR = 5bits + ADC_DATA = 7bits)









→Truncation occurred: "raw truncation" = 10.5%
"effective truncation" = 9.5%

→ Size of truncated events follows occupancy PDF, no bias!

→ 184~kbits of data sent through one GBT link over full LHC turn: 61.6% of GBT link bandwidth + 14% Event header + 4.8% of GBT header (unavoidable)

2. Results (Example 3)

Variables applied:

<Detector occupancy> = 30%

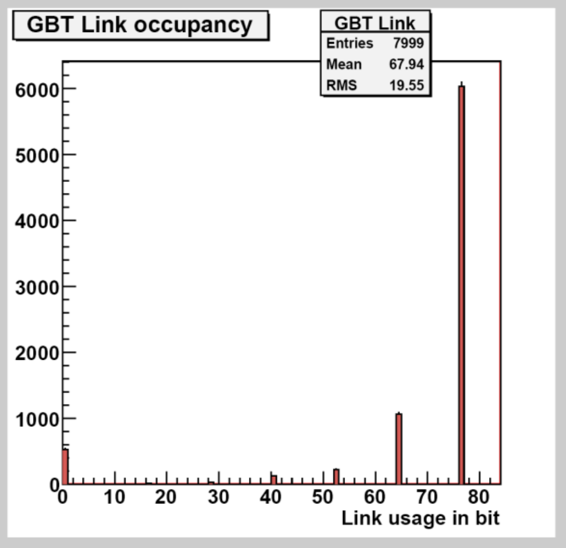
#channels / GBT link = 21

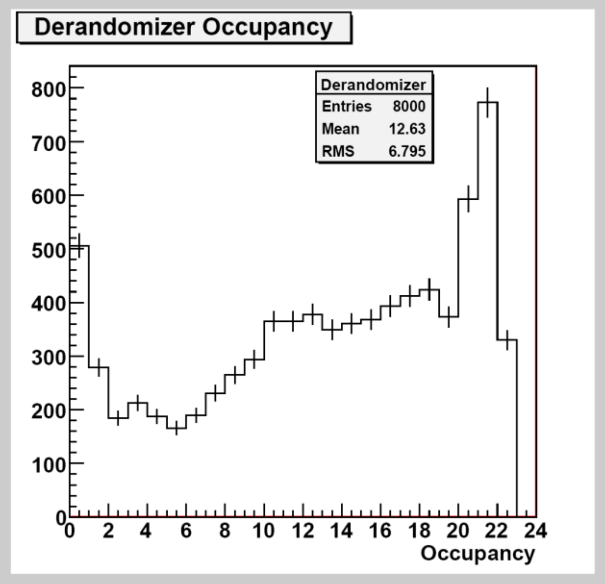
→ ~ 6.3 channels / GBT after ZS

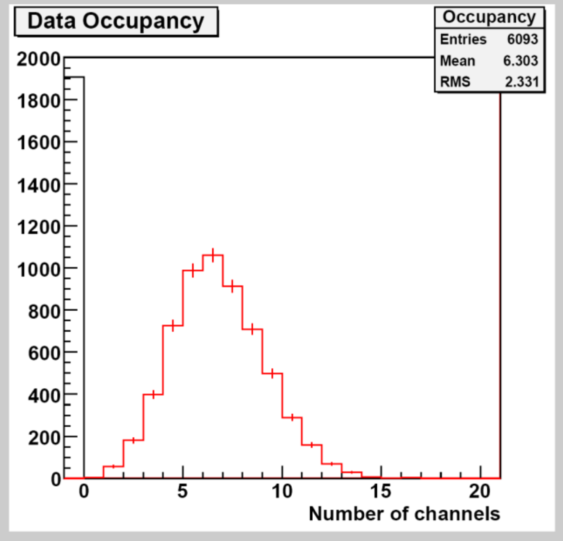
Derandomizer depth = 24

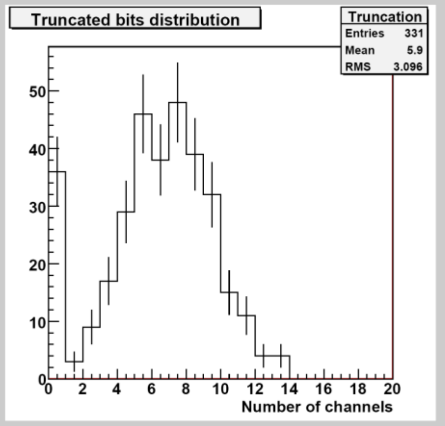
Channel size = 12

(e.g. ADDR = 5bits + ADC_DATA = 7bits)









→Truncation occurred: "raw truncation" = 5.4%

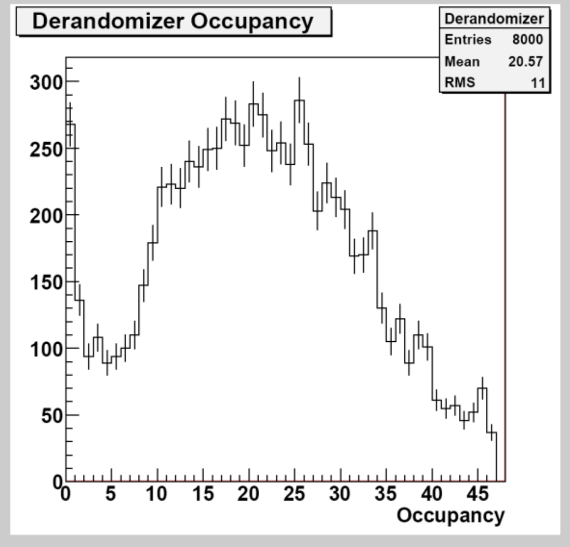
"effective truncation" = 4.7%

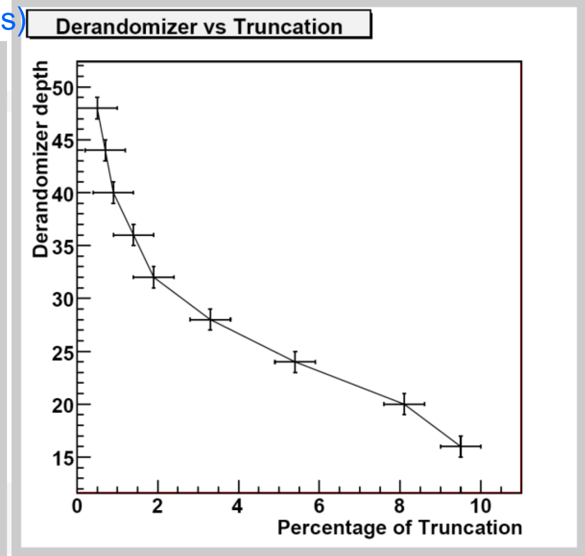
→ Size of truncated events follows occupancy PDF, no bias!

→ ~193kbits of data sent through one GBT link over full LHC turn: 64.4% of GBT link bandwidth + 14% Event header + 4.8% GBT header (unavoidable)

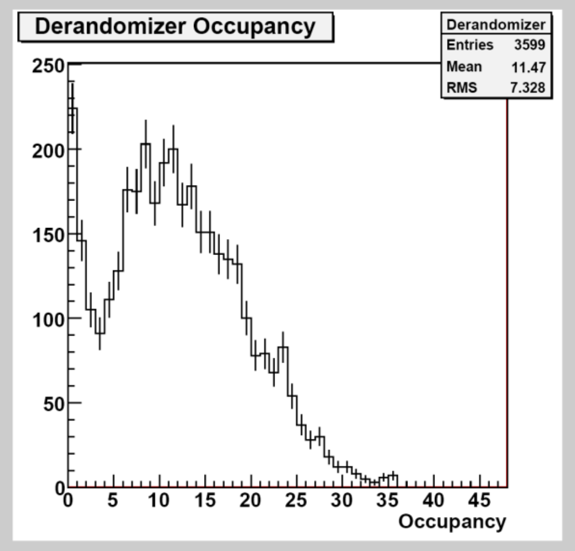
2. Results (Other examples)

Variables applied:
<Detector occupancy> = 30%
#channels / GBT link = 21
→ ~ 6.3 channels / GBT after ZS
Derandomizer depth = 48
Channel size = 12
(e.g. ADDR = 5bits + ADC_DATA = 7bits)





Variables applied:
<Detector occupancy> = 30%
#channels / GBT link = 20
→ ~ 6.3 channels / GBT after ZS
Derandomizer depth = 48
Channel size = 12
(e.g. ADDR = 5bits + ADC_DATA = 7bits)



→NO Truncation occurred

→Truncation occurred: “raw truncation” = 0.5%
“effective truncation” = 0.4%

→ ~200kbits of data sent through one GBT link over full LHC turn: 68.0% of GBT link bandwidth + 14% Event header + 4.8% GBT header (unavoidable)

Lab work and planned tests

Some lab work has already started (in parallel to current operations):

1. Set up a S-TFC test stand at the pit ✓ Done!

2. First prototype of Marseille's AMC board ✓ Done!
   - First version of firmware with GBT protocol in it (developed in Marseille)
   - First version of control software in NIOS (developed in Marseille)
     - ➢ One word of acknowledgment for the work done by Jean-Pierre and Frederic Hachon in Marseille

3. Get acquainted with the hardware ✓ Done!

4. Send first trigger words ✓ Done!
   - 8b/10b protocol: using ALTERA features
   - GBT protocol: not possible to use ALTERA features

5. Stress tests to validate latency control
6. Stress tests to validate phase control

✓ Done (but will need revalidation on Stratix V + more tests!)

7. Implement first version of S-ODIN firmware with functionalities presented here
   - Including first version of SOL40
8. Develop first single-AMC test stand board with firmware + software to control the hardware
   - Need dedicated simulation framework ✗
     → Already developed in 2009 (see https://indico.cern.ch/conferenceDisplay.py?confId=65307)
     → Needs mostly adaptations …
   - Work to control (software) the single AMC card done using ALTERA tools