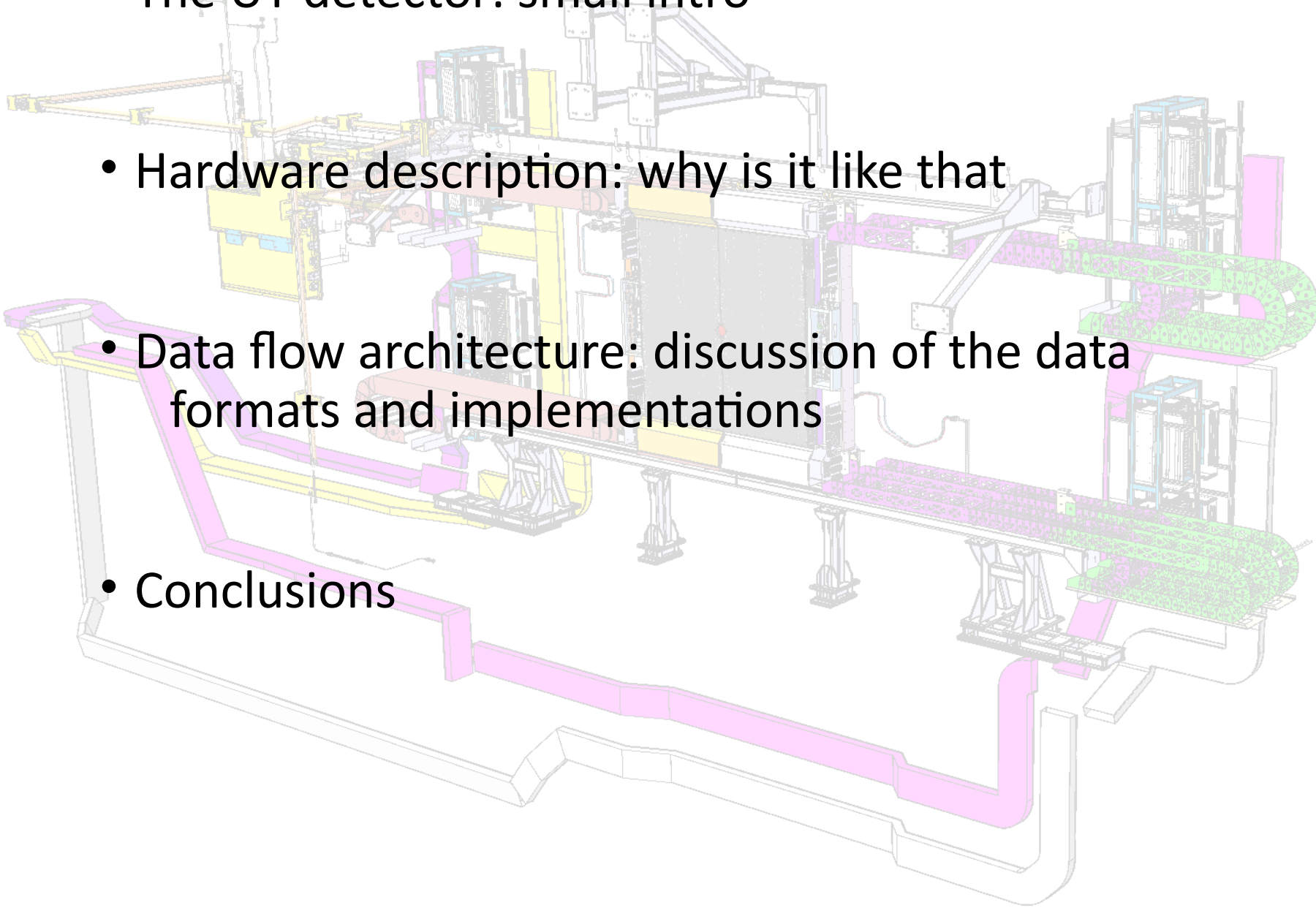
The LHCb Upstream Tracker data flow architecture

Carlos Abellan on behalf of the LHCb UT collaboration

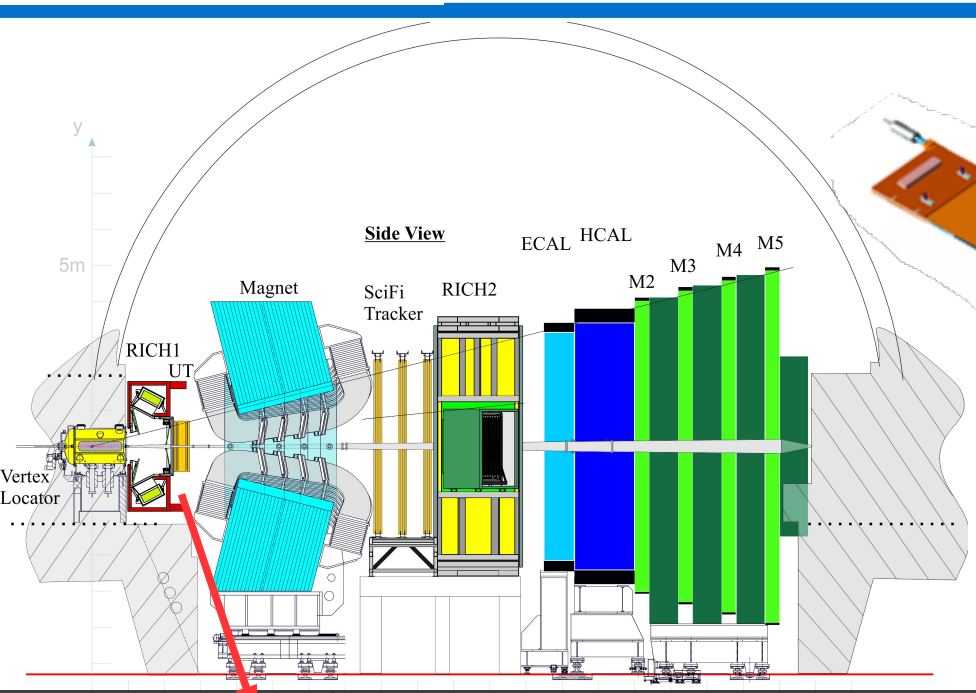
Waterloo, 26th May 2021

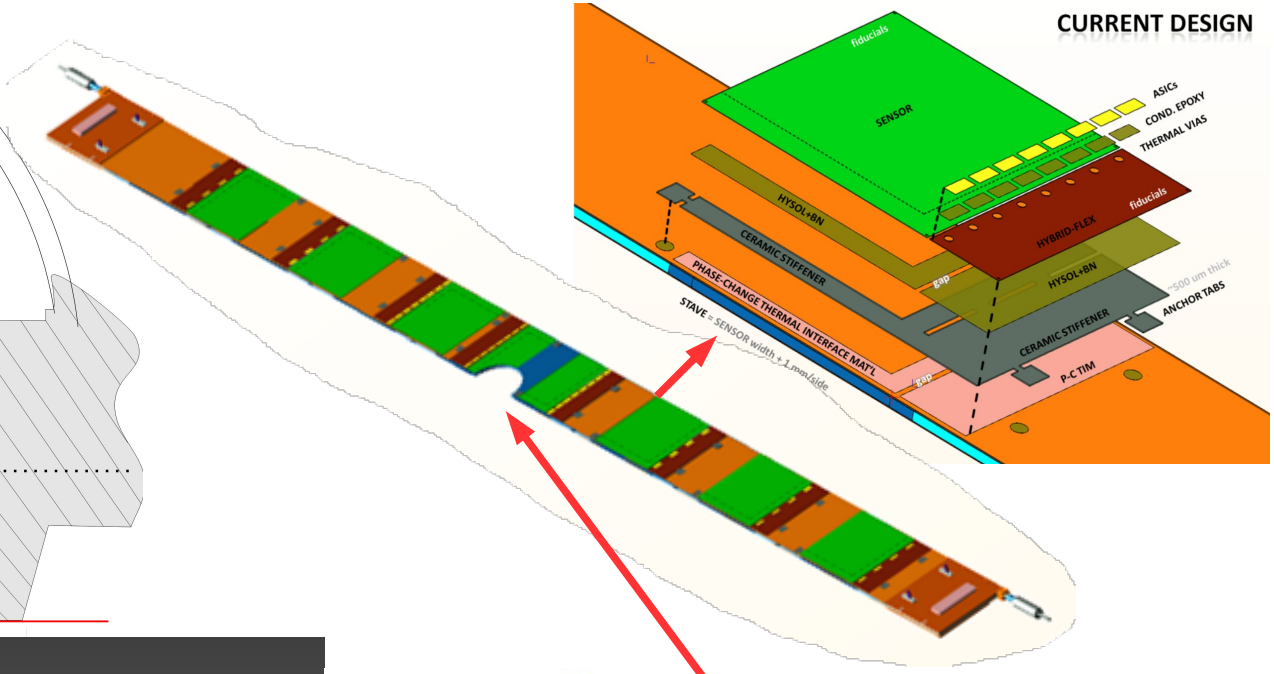
Presentation Outline

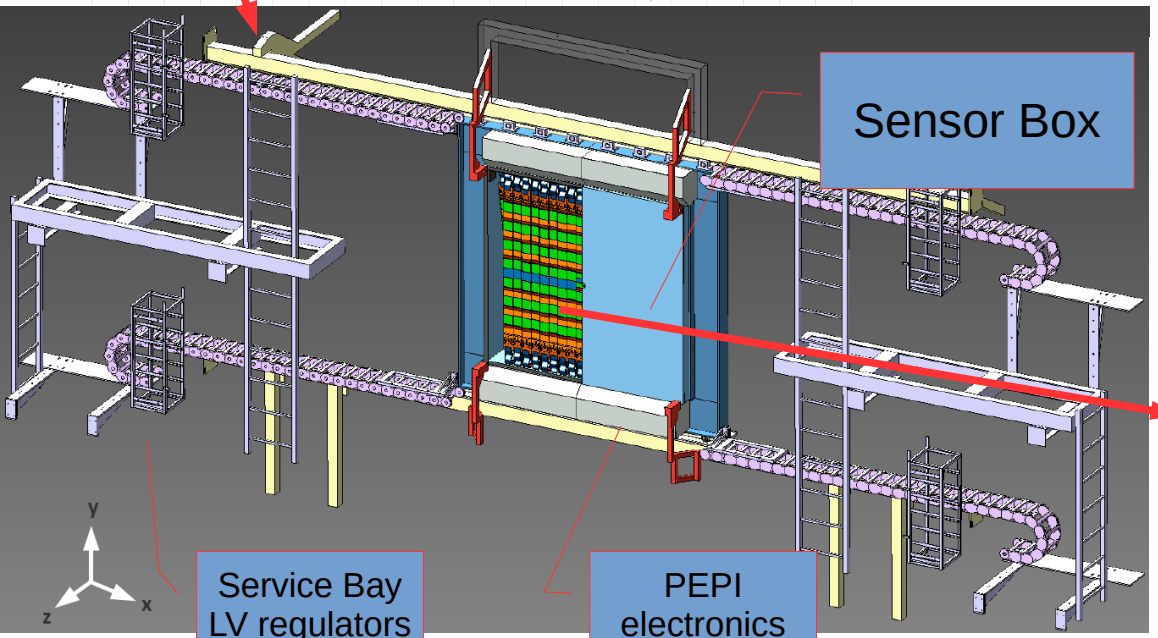
- The UT detector: small intro
- Hardware description: why is it like that
- Data flow architecture: discussion of the data formats and implementations
- Conclusions

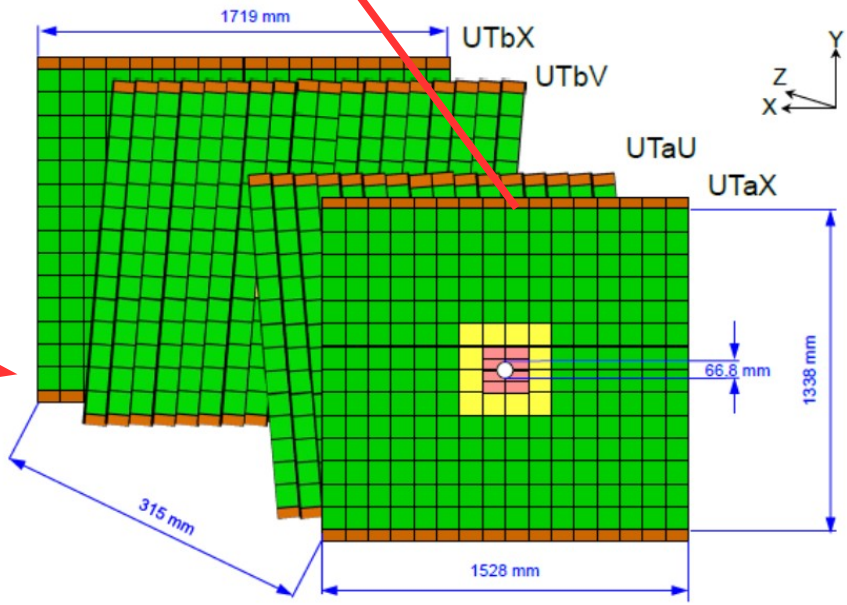


The UT detector





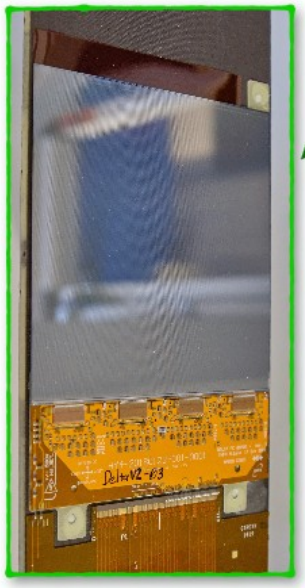


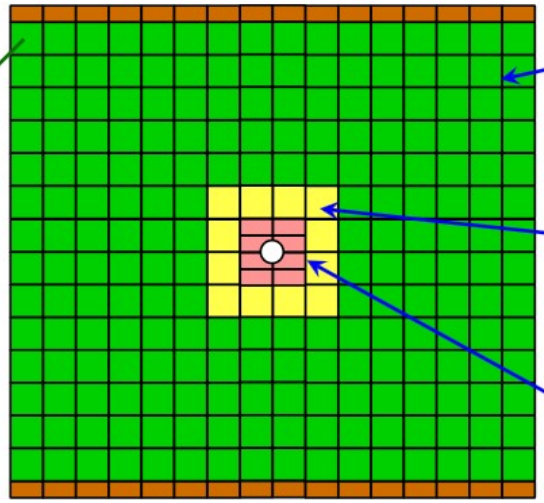


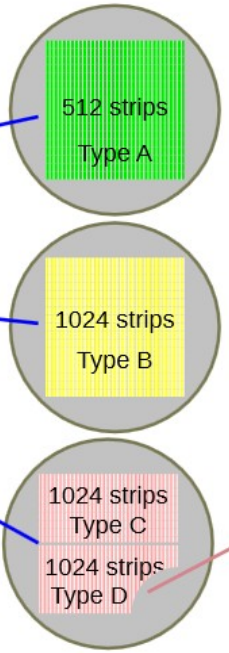
The UT detector

- Silicon Sensors

Sensor	Type	Pitch	Length	Strips	# sensors
A	p-in-n	187.5 μm	99.5 mm	512	888
B	n-in-p	93.5 μm	99.5 mm	1024	48
C	n-in-p	93.5 μm	50 mm	1024	16
D	n-in-p	93.5 μm	50 mm	1024	16



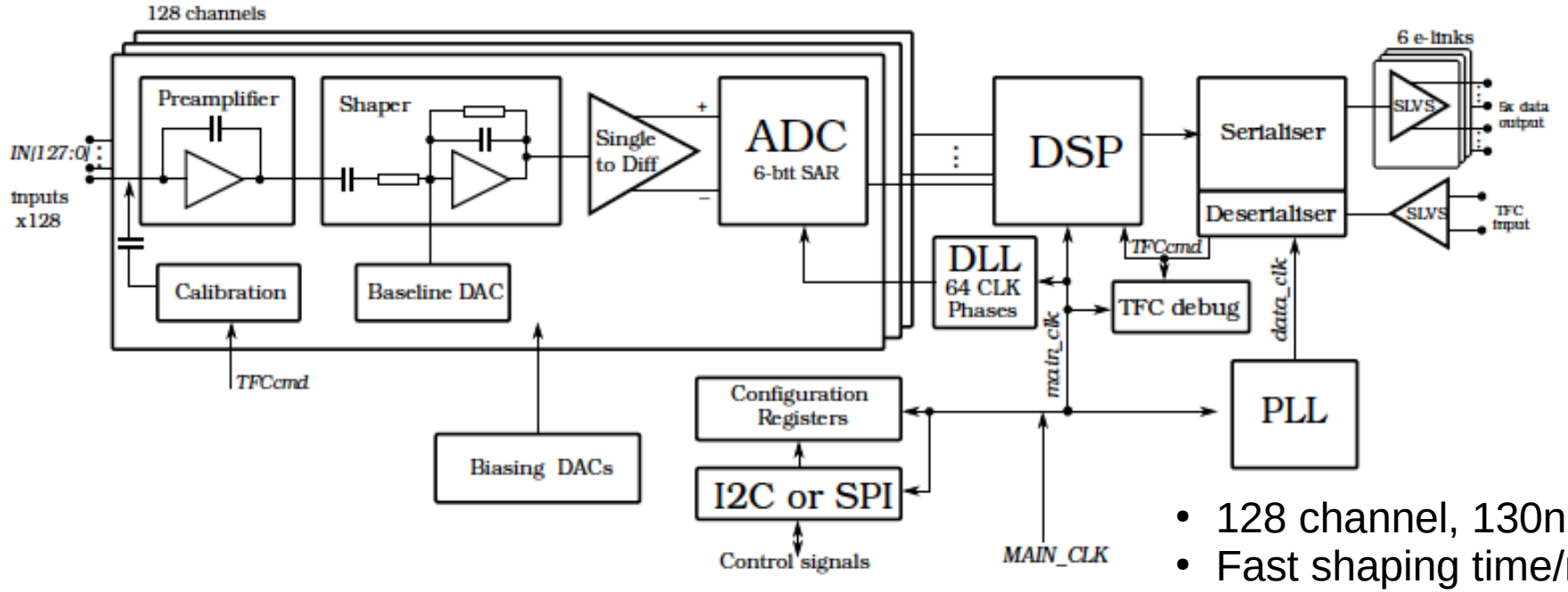


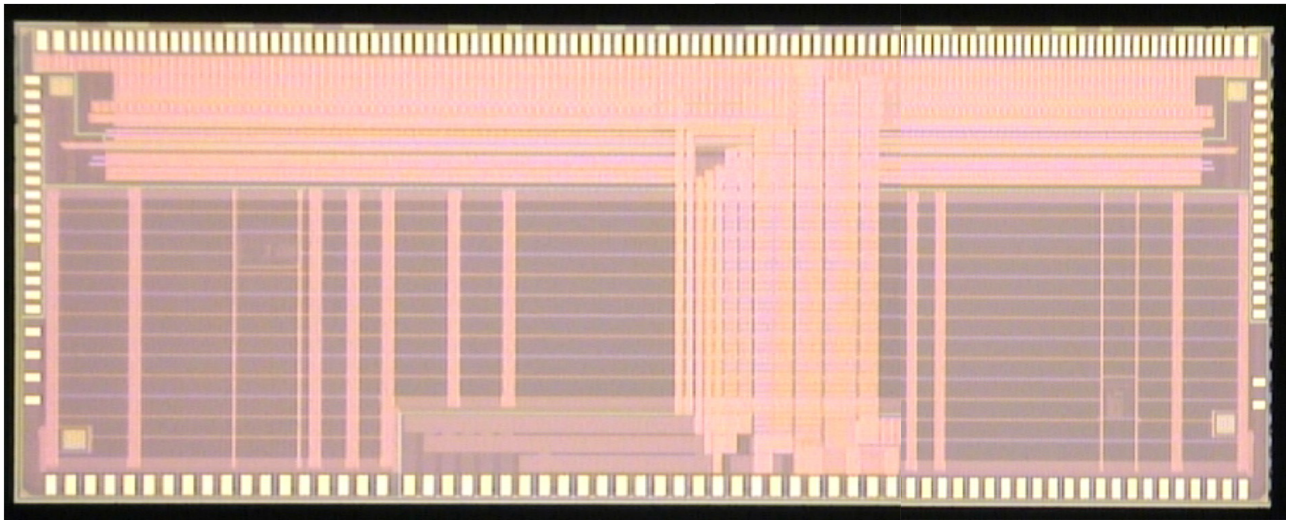


- D type sensors:
  - Circular cut-out to maximize acceptance next to beam pipe
- A type sensors:
  - 187.5um pitch
  - Built-in pitch adapters (190um to 80um)
- Top-side biasing via wire bonds rather than conductive glue to backplane

The SALT ASIC

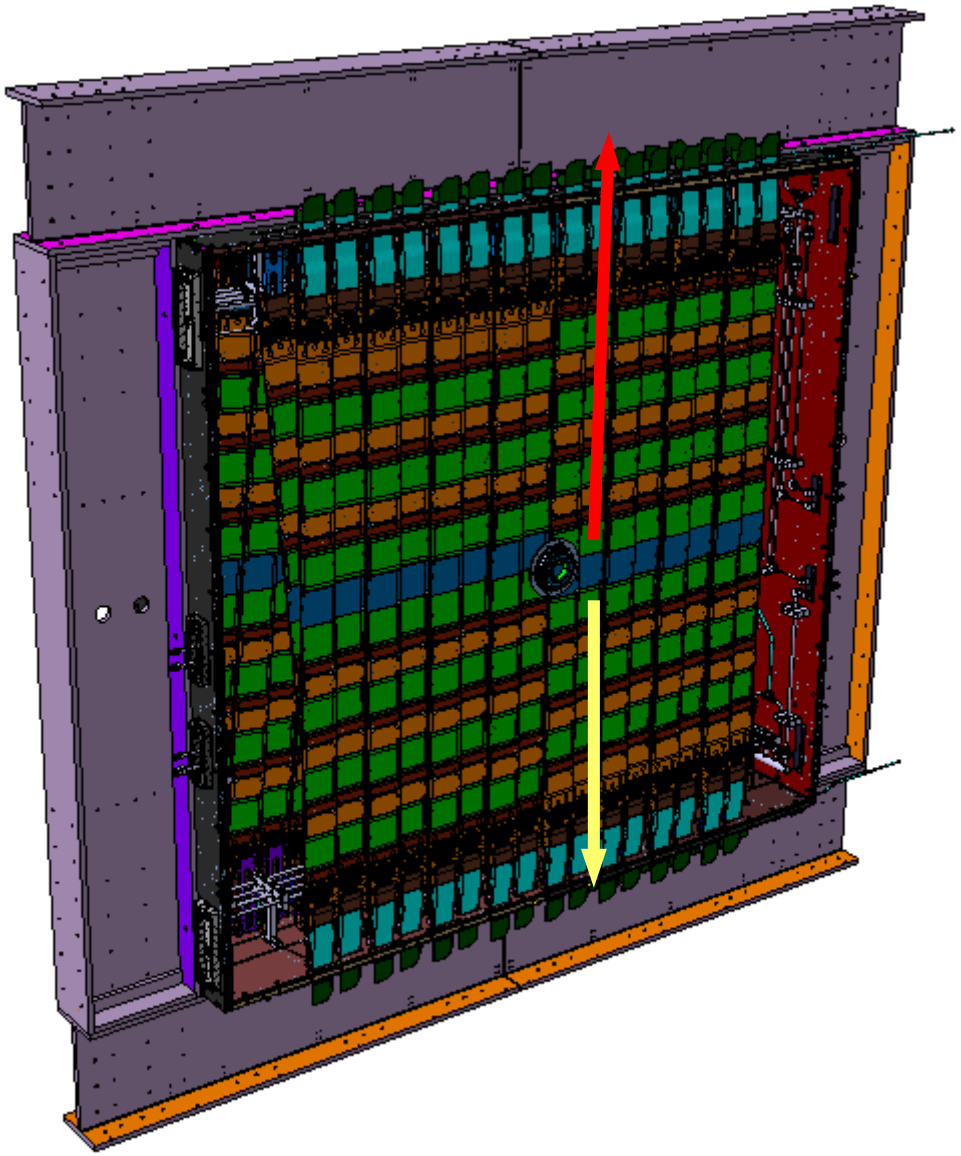
- SALT128 ASIC prototype done and tested





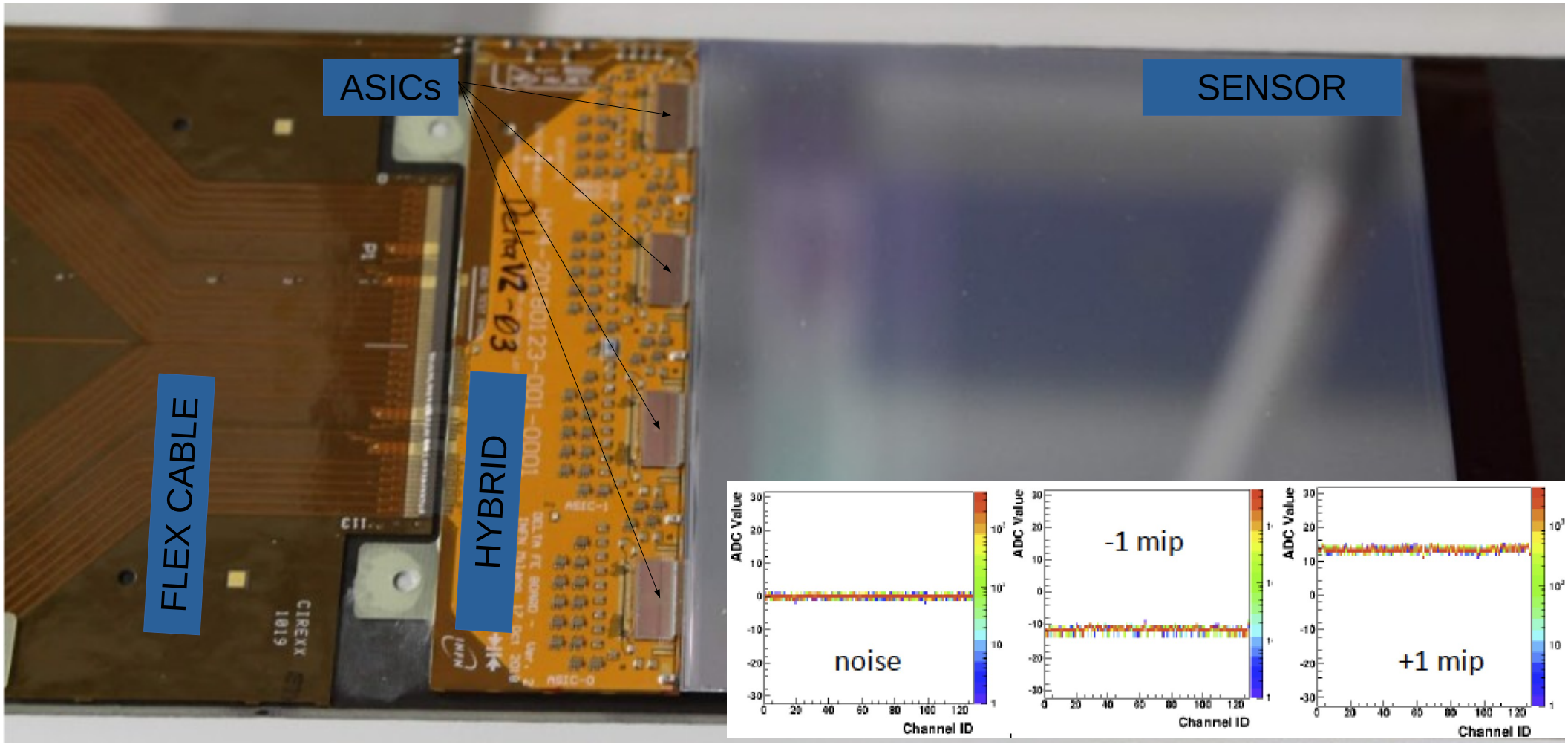
- 128 channel, 130nm TSMC ASIC
- Fast shaping time/return to baseline
- 6 bit embedded ADC
- DSP:
  - Pedestal subtraction
  - Zero suppression
  - e-link data formating
- Outputs arrays of Channel-Value pairs.
- SLVS e-links (typically x3, but also x4 and x5 are used)

Hardware description



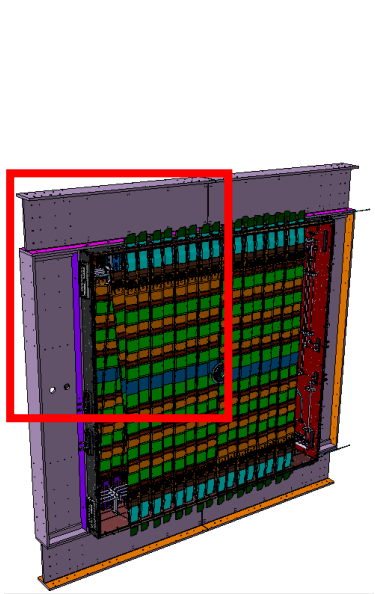
Hardware description

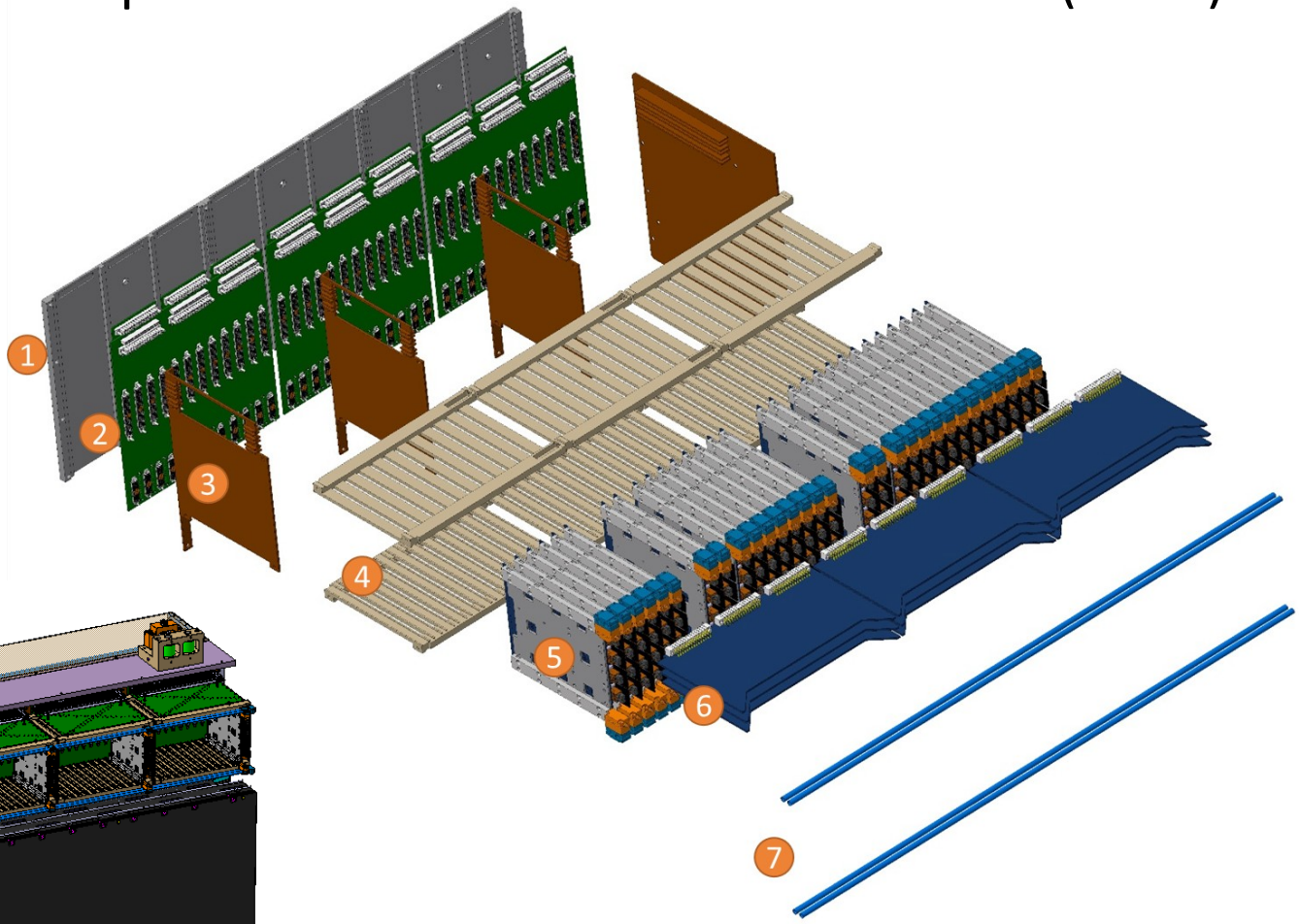
- The sensor is wire bonded to the ASIC
- The ASIC does most of the work: analog processing, digitization, zero suppression, MCM suppression, serialization, …
- Data (Channel-Value pairs)leaves the system in digital form, elink packets.

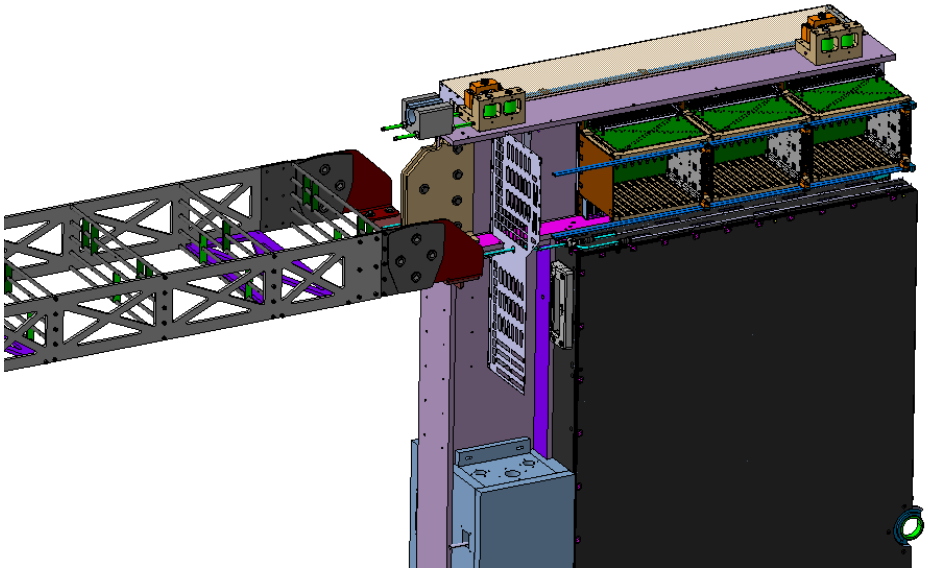


Hardware description

- The PEPI area contains backplanes and data concentrator boards (DCBs)

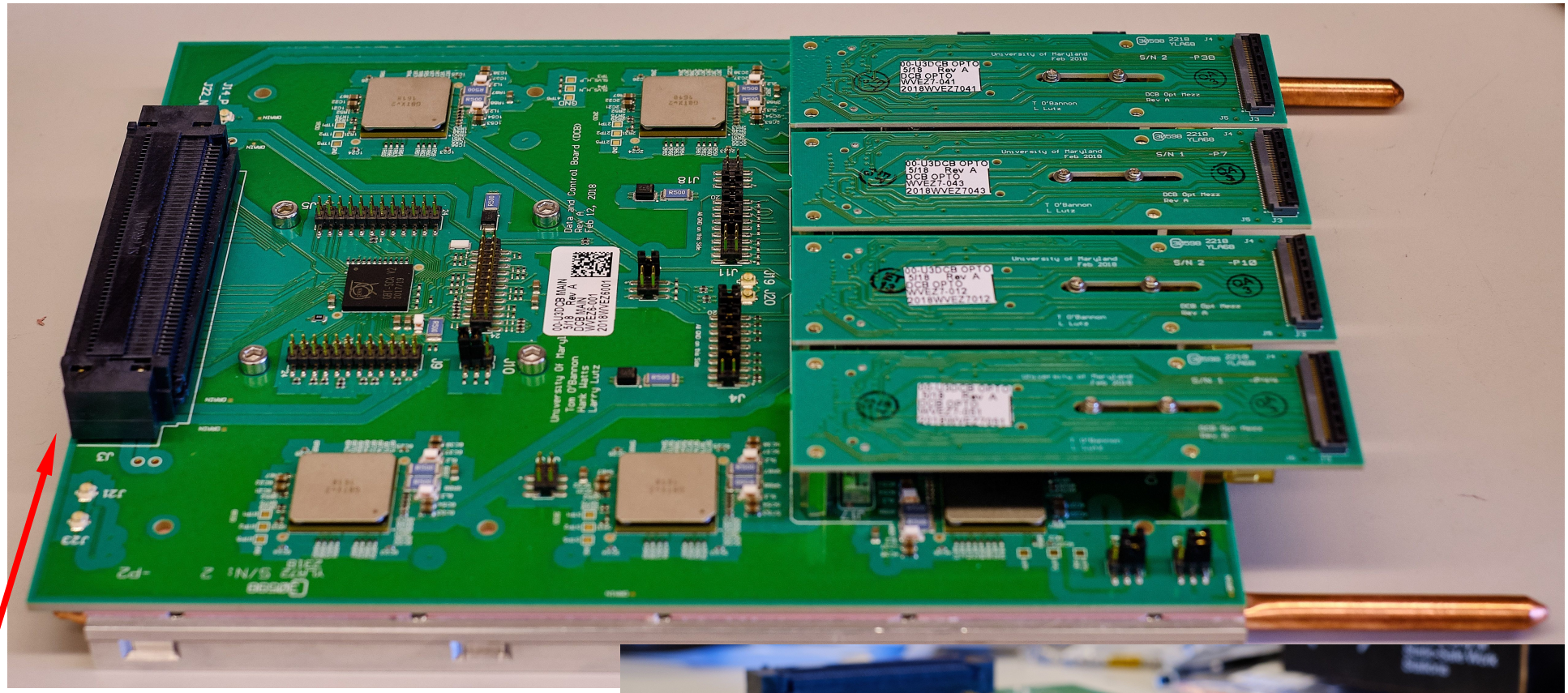




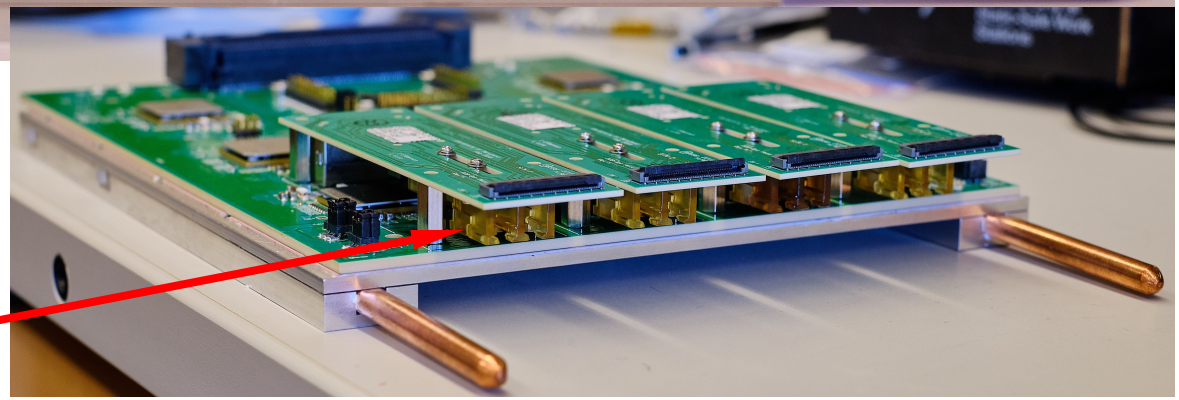


Hardware description

- DCBs provide upstream and downstream data and control links



Electrical Interface (backplane)



Optical Interface (to surface)

Hardware description

- Communication with surface is mainly optical, using PCIe40 boards (TELL40 and SOL40)

Hardware description

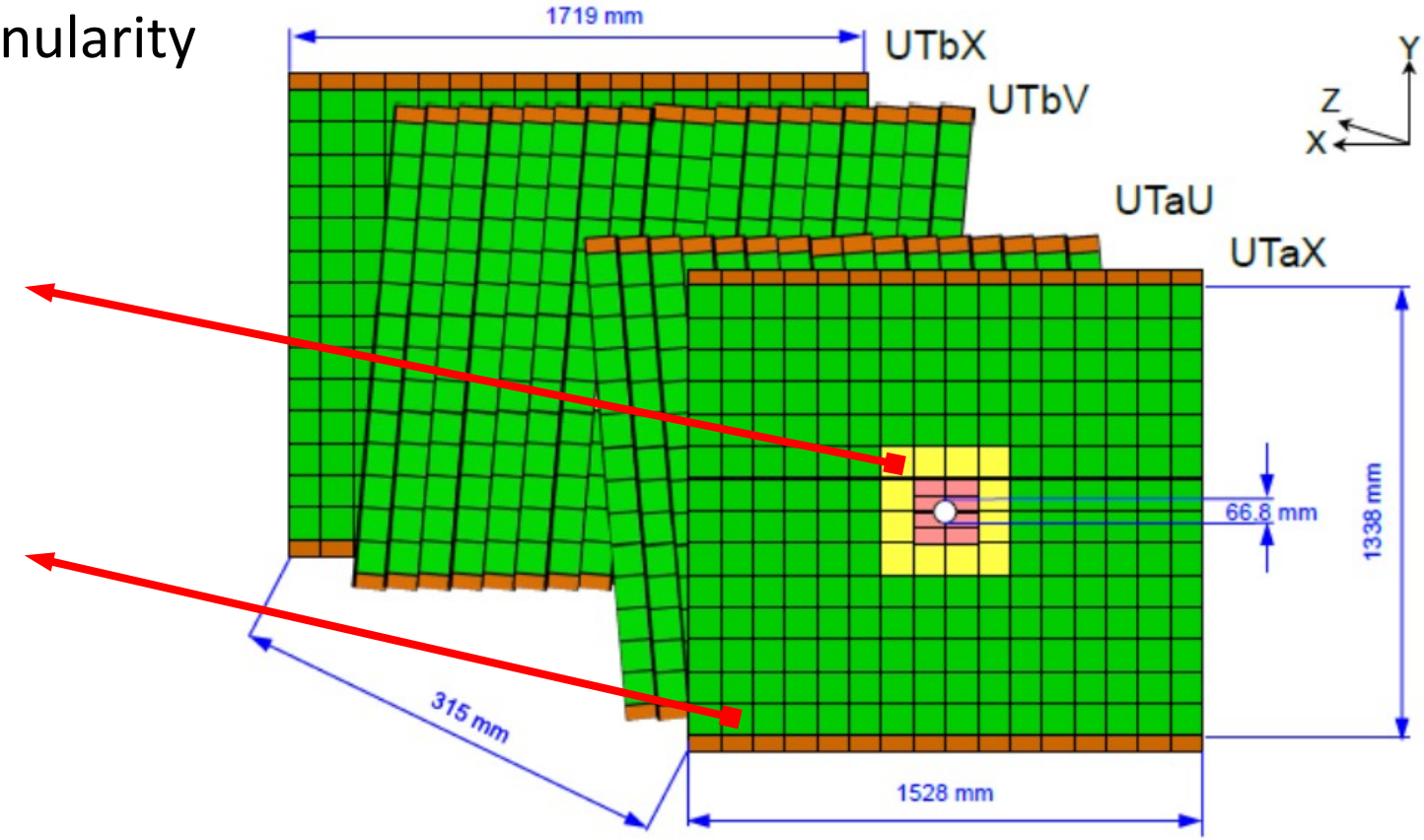
- Restrictions and irregularities:
  - Material budget: minimal marterial in acceptance are
    - **Minimal traces**
  - **Different occupancy depending on position**
  - Different granularity

Many hits

4 or 5 elinks

Few hits

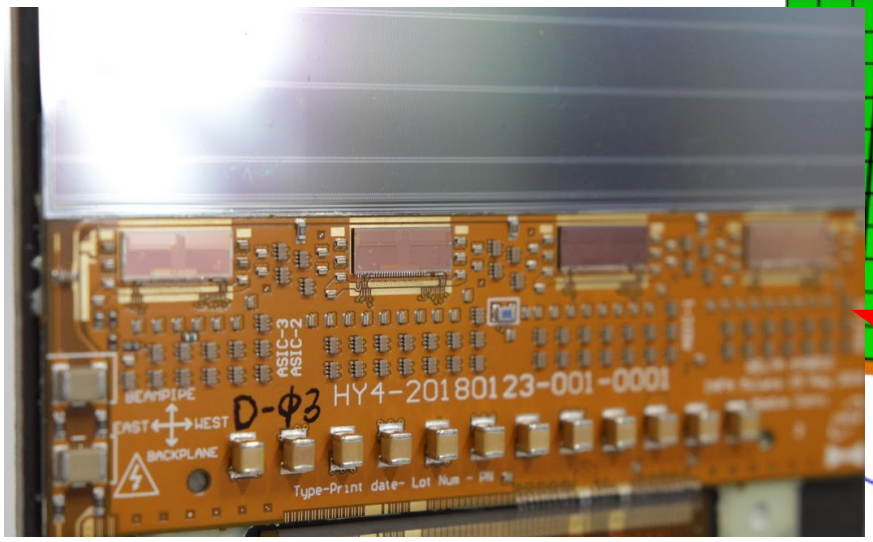
3 elinks

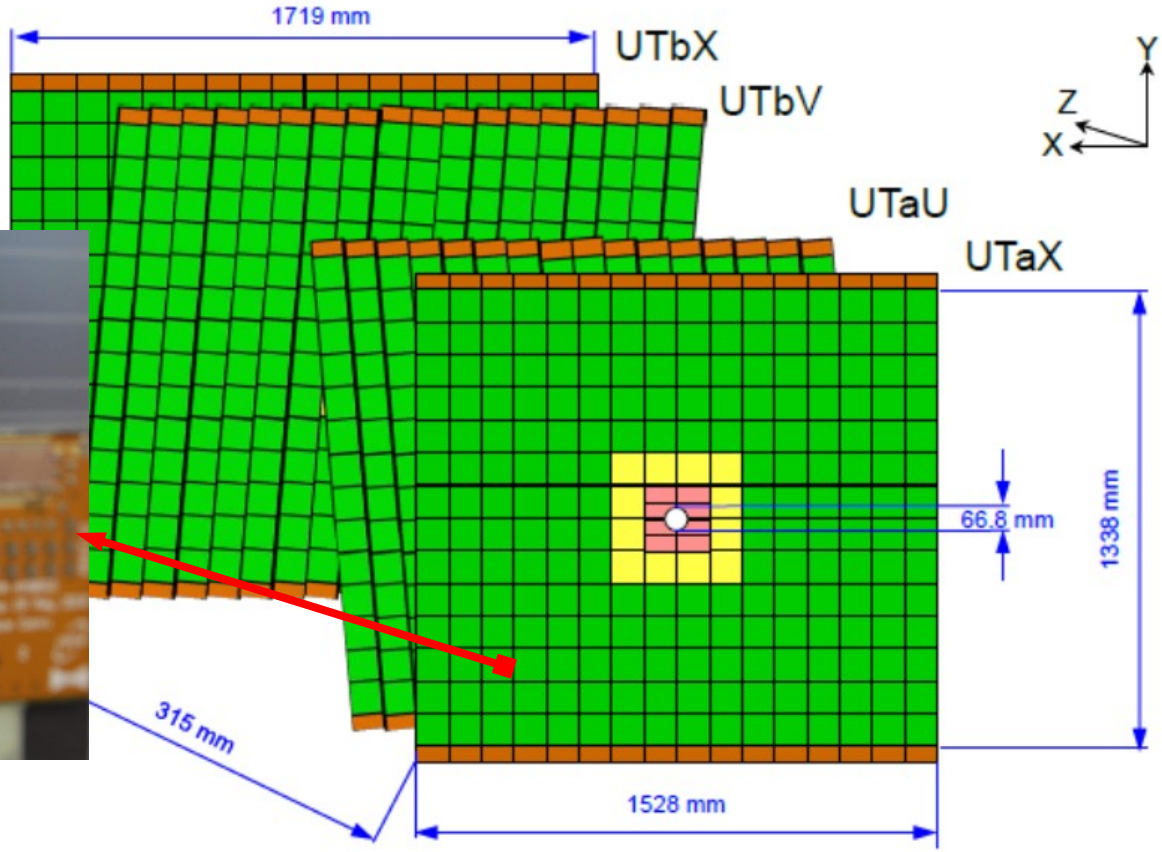


Hardware description

- Restrictions and irregularities:
  - Material budget: minimal metal in acceptance are
    - Minimal traces
  - Different occupancy depending on position
  - **Different granularity**

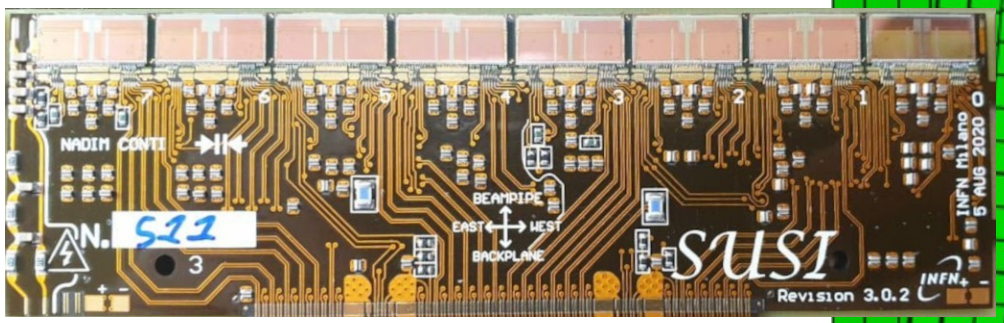
512 strips => 4 ASICs



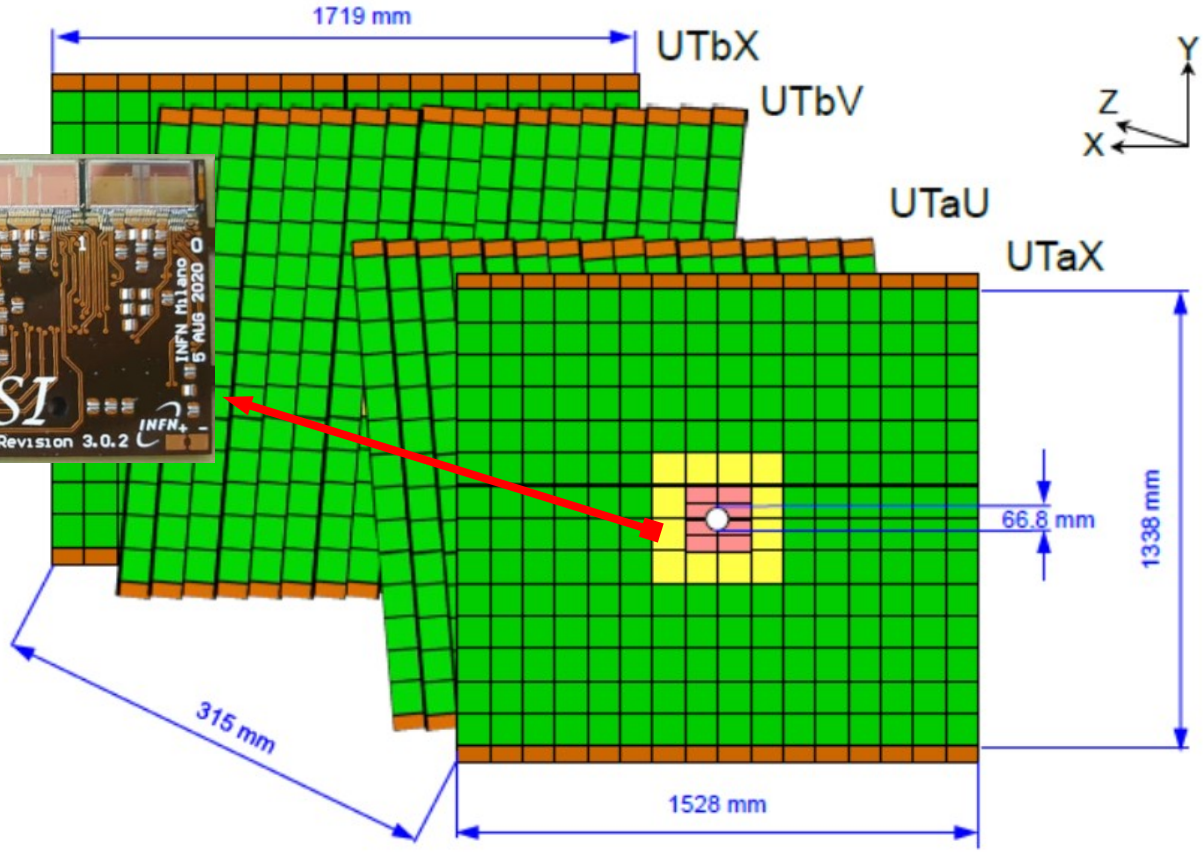


Hardware description

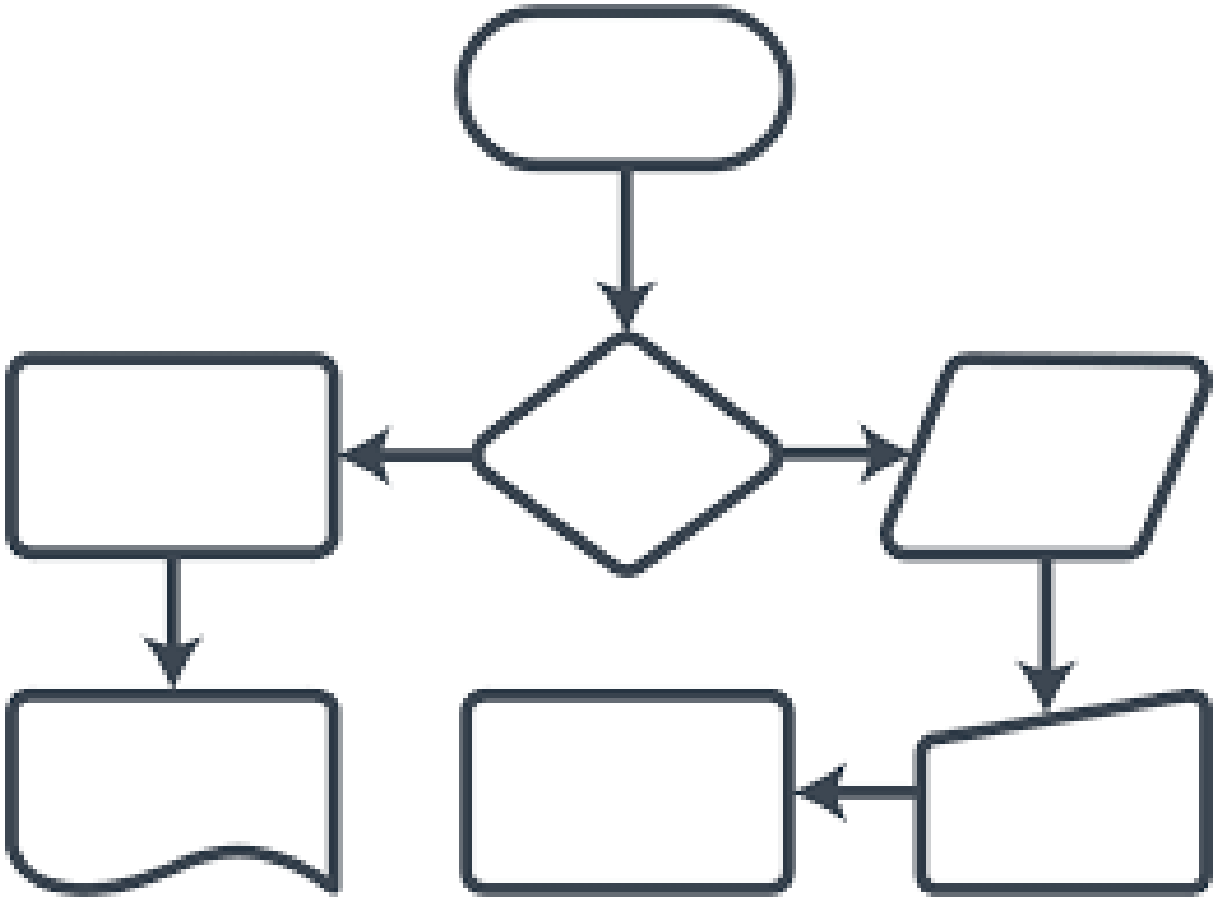
- Restrictions and irregularities:
  - Material budget: minimal metal in acceptance are
    - Minimal traces
  - Different occupancy depending on position
  - **Different granularity**



1024 strips => 8 ASICs

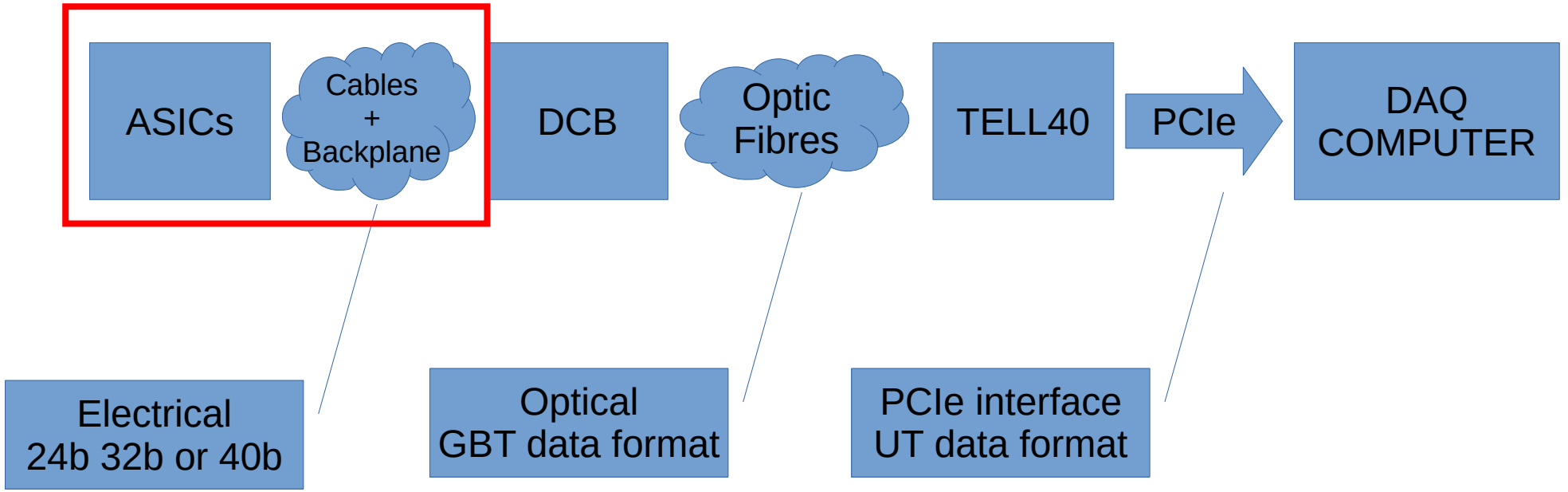


Data Flow Architecture



Data Flow Architecture

- Description of the different data formats



Data Flow Architecture

- ASICs send data in a common format:

Normal Packet:

4 bits			6 bits	12 bits	12 bits		
BXID	*	0	*nHits*	Hit 0	Hit 1	...	Hit *nHits*-1

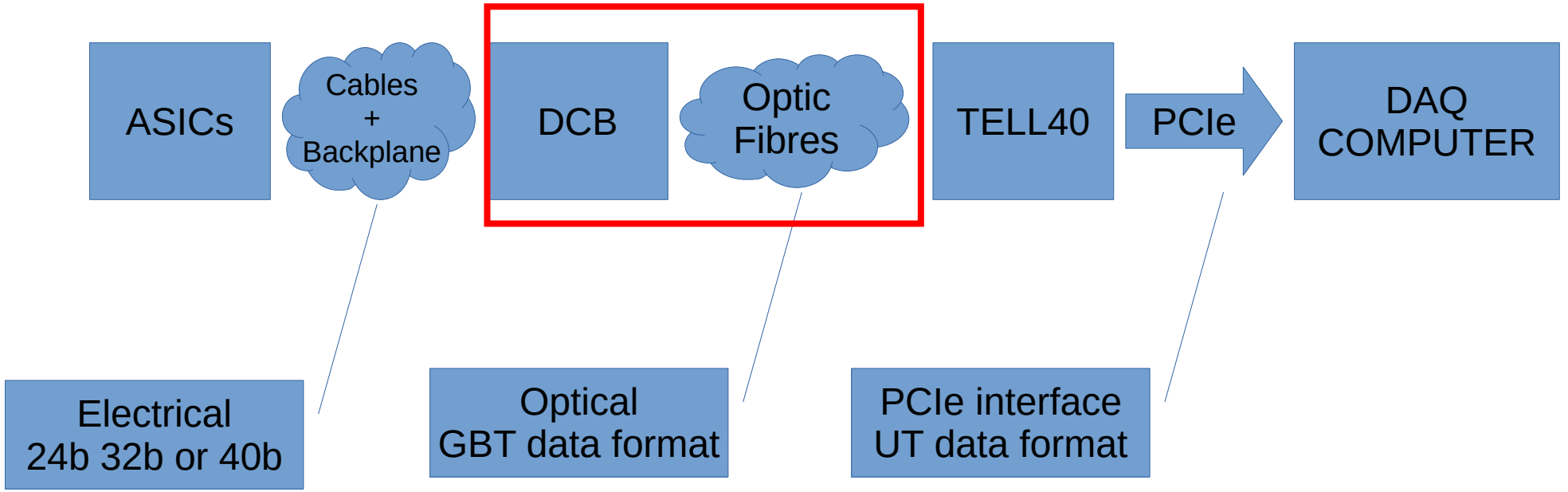
Hit:

7 bits	5 bits
channel number	channel value after DSP

Packet name	Header (12-bit)				Data	Comment
	BXID	Parity	Flag	Length		
	4 bits	1 bit	1 bit	6 bit	n·12 bits	
Idle	0000	1	1	'b11_0000	—	no enough data
BxVeto	*bxid_cnt[3:0]*	*	1	'b01_0001	—	BxVeto in TFCcmd
HeaderOnly	*bxid_cnt[3:0]*	*	1	'b01_0010	—	HeaderOnly in TFCcmd
BusyEvent	*bxid_cnt[3:0]*	*	1	'b01_0011	—	*nHits*> 63
BufferFull	*bxid_cnt[3:0]*	*	1	'b01_0100	—	no space in memory
BufferFullN	*bxid_cnt[3:0]*	*	1	'b01_0101	—	no space in memory
NZS	*bxid_cnt[3:0]*	*	1	'b00_0110	Values	NZS in TFCcmd
Normal	*bxid_cnt[3:0]*	*	0	*nHits*	Hits	Normal event
Sync	*bxid_cnt[11:0]*				*sync_pattern*	Synch in TFCcmd

Data Flow Architecture

- Description of the different data formats



Data Flow Architecture

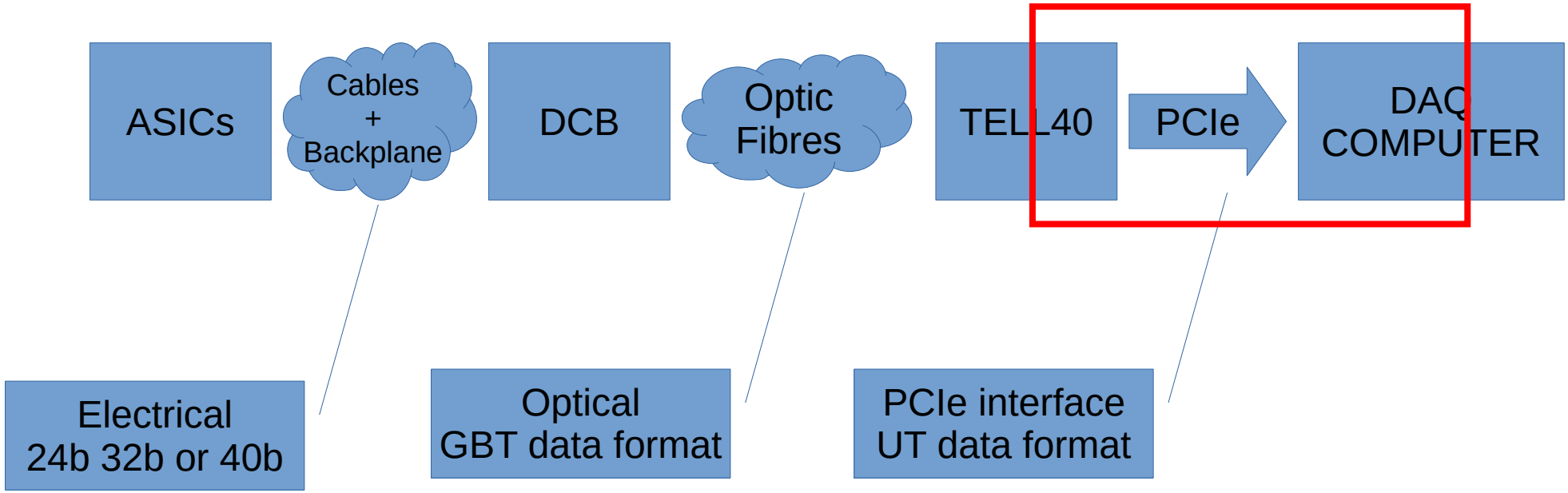
- A mix of practical reasons lead the UT to use several GBT data formats
- The GBT frame is 112b wide, the UT format, being 24b, 32b or 40b, leaves some unused space.
- Some flex cables are using 40 bit configurations to transmit 24b or 32b, cost reasons

		8b	8b	8b	8b	8b	8b	8b	8b	8b	8b	8b	8b	8b	8b
	Byte number	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GBT Frame Type	4 ASIC x 3 elink			24 bits			24 bits			24 bits			24 bits		
	2 ASIC x 3 elink			24 bits					24 bits						
	2 ASIC x 4 elink			32 bits					32 bits						
	2 ASIC x 5 elink			40 bits					40 bits						

- This arrangement is done between the flex cables and PEPI backplanes. Depending on the region the UT uses one format or another.

Data Flow Architecture

- Description of the different data formats



Data Flow Architecture

- A common output data format
- Desirable features:
  - Easy to implement in an FPGA
  - Low resources
  - Easy / fast to decode in a CPU/GPU
    - Aligned to 8b
    - Multithread friendly
  - Fit in the available PCIe bandwidth
  - Cover all corner cases gracefully
    - What if one ASIC stops working?
    - What if one ASIC goes to error?
    - What if one ASIC misbehaves?
    - Try NOT to lose too much data (i.e. 1 ASIC failing shouldn't interfere with the information of the other 47!)

Data Flow Architecture

- A common output data format
- Aligning data to 8b:
  - ASIC data is 7b+5b=12b, we add 4b padding?
  - We could find a use for those 4b…
  - Add ASIC_ID and we have a strip ID valid within a sensor

Normal Packet:

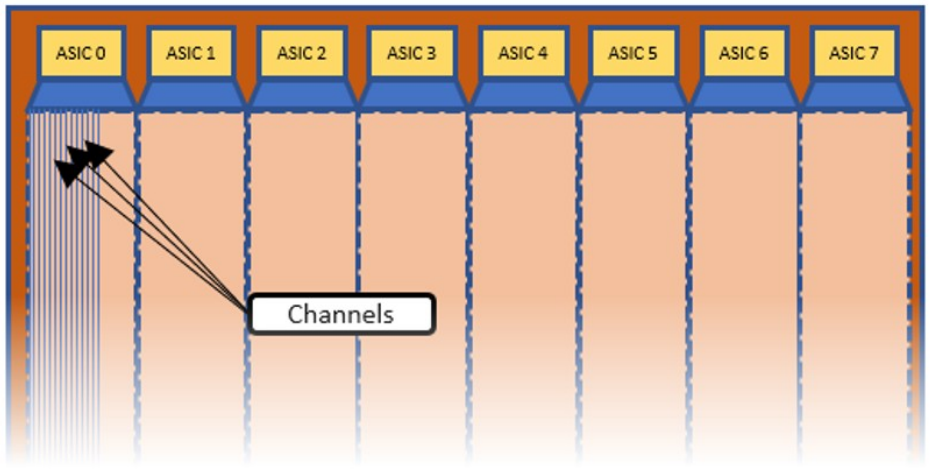
4 bits			6 bits	12 bits	12 bits		
BXID	*	0	*nHits*	Hit 0	Hit 1	…	Hit *nHits*-1

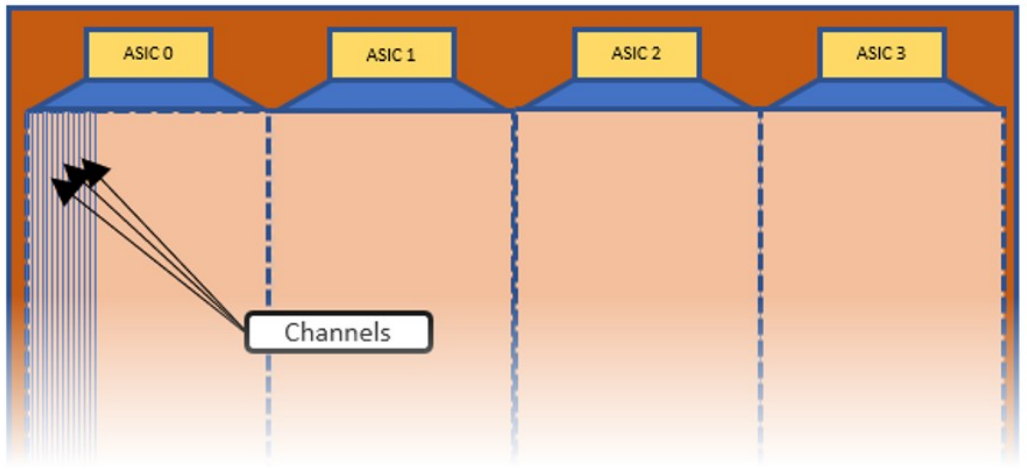
Hit:

7 bits	5 bits
channel number	channel value after DSP

Normal ASICxx Hit

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ASIC_ID [4b]				Chan Number [7b]							DSP Val. [5b]				
Local Strip ID [11b]											DSP Val. [5b]				



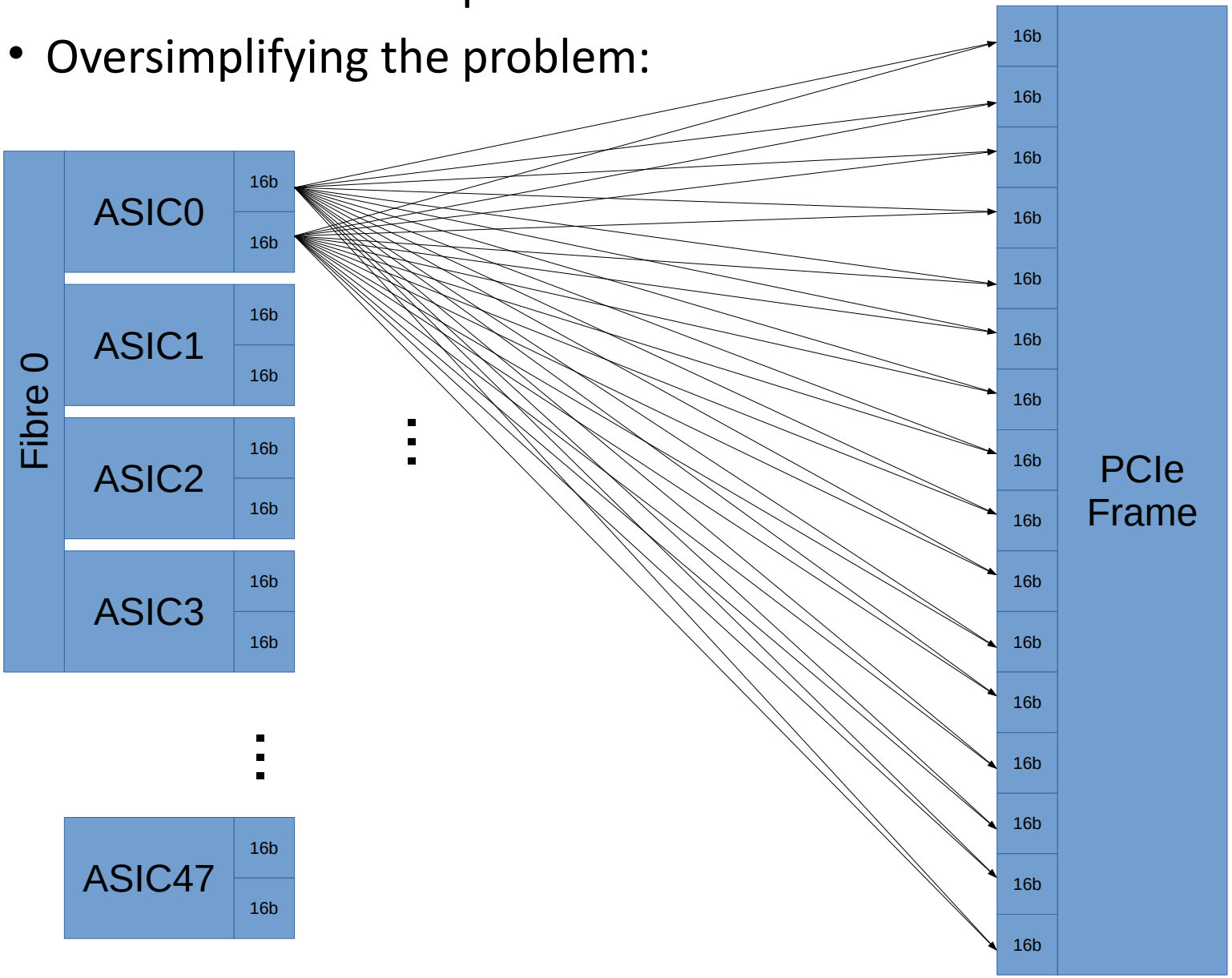


Data Flow Architecture

- Introduce the concept of *lanes*
- Oversimplifying the problem:
  - Each half TELL40 has 6 input fibres
    - With up to 4 ASICs each => 24 inputs total ASICs
    - Simplest case 3 elinks => 24b => 2x12b word => 2x16b after align
    - Total of 2 words x 24 ASICs = 48 words of 16b
  - Each half TELL40 has a 256b wide PCIe interface
    - 256b/16b = 16 words of 16b each interface
  - We would have to map 48 positions to 16 possible destinations each
    - Combinatorics is very large, hard to do, might require pipelining or complicated structures.

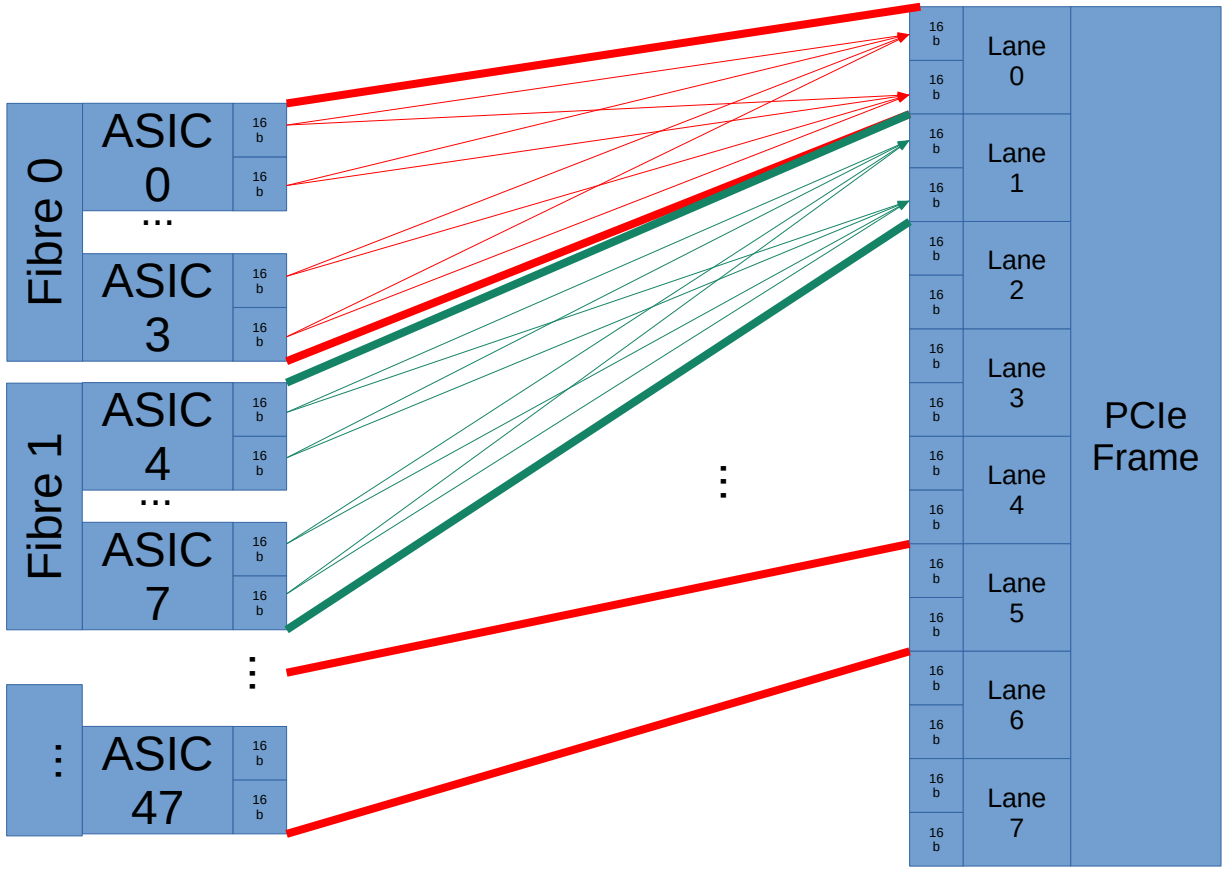
Data Flow Architecture

- Introduce the concept of *lanes*
- Oversimplifying the problem:



Data Flow Architecture

- A common output data format
- Introduce the concept of *lanes*.
- What if we simplify the combinatorics?
  - 1 fibre can only go to a small portion of the PCIe frame



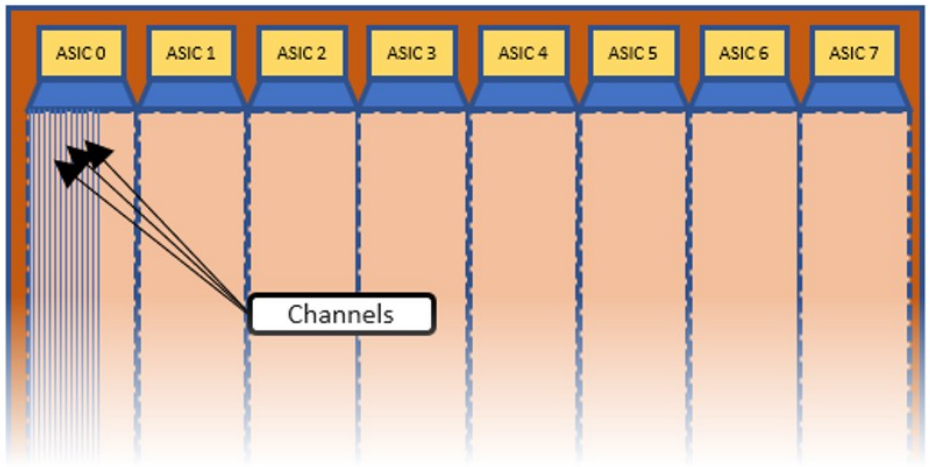
Data Flow Architecture

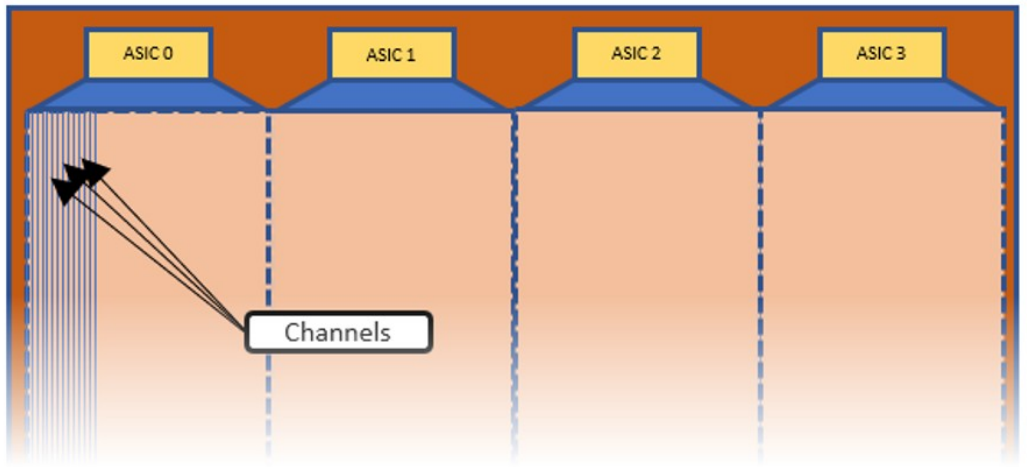
- Introduce the concept of *lanes*
- Lanes greatly reduce combinatorics:

Example of a normal event

Event Headers & Flags								LANE 5		LANE 4		LANE 3		LANE 2		LANE 1		LANE 0	
16b		16b		16b		16b		16b	16b	16b	16b	16b	16b	16b	16b	16b	16b	16b	16b
8b	8b	8b	8b	8b	8b	8b	8b	8b	8b	8b	8b	8b	8b	8b	8b	8b	8b	8b	8b	8b	8b	8b	8b	8b	8b	8b	8b	8b	8b	8b	8b
Event 0 Header								Hit1	Hit0		Hit0	Hit1	Hit0	Hit1	Hit0	Hit1	Hit0	Hit1	Hit0
													Hit2	Hit3	Hit2		Hit2		
															Hit4				

- Conceptually it makes sense too:
  - In most of the cases 1 lane = 4 ASICs = 1 sensor
  - Strip ID is within a sensor
  - Not all cases, in some cases 1 lane = ½ or ¼ of a sensor





Data Flow Architecture

- Introduce the concept of *lanes*
- Lanes greatly reduce combinatorics:

Example of a normal event

Event Headers & Flags				LANE 5		LANE 4		LANE 3		LANE 2		LANE 1		LANE 0	
16b	16b	16b	16b	16b	16b	16b	16b	16b	16b	16b	16b	16b	16b	16b	16b
8b 8b	8b 8b	8b 8b	8b 8b	8b 8b	8b 8b	8b 8b	8b 8b	8b 8b	8b 8b	8b 8b	8b 8b	8b 8b	8b 8b	8b 8b	8b 8b
Event 0 Header				Hit1	Hit0		Hit0	Hit1	Hit0	Hit1	Hit0	Hit1	Hit0	Hit1	Hit0
									Hit2	Hit3	Hit2		Hit2		
											Hit4				

- It is multithread friendly:
  - 6 lanes allow for up to 6 threads, lane positions are fixed, so every thread knows where to go to look for its data without colliding with other threads
- What is that header?

Event Header Format

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Hits in Lane 3 [8b]								Hits in Lane 2 [8b]								Hits in Lane 1 [8b]								Hits in Lane 0 [8b]							

63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
Event ID from TFC [8b]								RESERVED								Hits in Lane 5 [8b]								Hits in Lane 4 [8b]							

Data Flow Architecture

Example of a normal event

Event Headers & Flags								LANE 5				LANE 4				LANE 3				LANE 2				LANE 1				LANE 0			
16b		16b		16b		16b		16b		16b		16b		16b		16b		16b		16b		16b		16b		16b		16b		16b	
8b	8b	8b	8b	8b	8b	8b	8b	8b	8b	8b	8b	8b	8b	8b	8b	8b	8b	8b	8b	8b	8b	8b	8b	8b	8b	8b	8b	8b	8b	8b	8b
Event 0 Header								Hit1		Hit0				Hit0		Hit1		Hit0		Hit1		Hit0		Hit1		Hit0		Hit1		Hit0	
																		Hit2		Hit3		Hit2				Hit2					
																						Hit4									

- Corner cases:
  - ASICs flag uncommon states by a special header, when 1 or more flags are present we add the FLAGs HEADER, that is nothing else but all headers from all ASICs.

Example of an event with an irregularity in at least one ASIC Statiscitally improbable, expected < 1% of the times

Event Headers & Flags								LANE 5				LANE 4				LANE 3				LANE 2				LANE 1				LANE 0			
16b		16b		16b		16b		16b		16b		16b		16b		16b		16b		16b		16b		16b		16b		16b		16b	
8b	8b	8b	8b	8b	8b	8b	8b	8b	8b	8b	8b	8b	8b	8b	8b	8b	8b	8b	8b	8b	8b	8b	8b	8b	8b	8b	8b	8b	8b	8b	8b
Event 0 Header								Hit1		Hit0				Hit0		Hit1		Hit0		Hit1		Hit0		Hit1		Hit0		Hit1		Hit0	
FLAG HEADER																		Hit2		Hit3		Hit2				Hit2					
																						Hit4									

Flag Header Format: only present if one or more ASICs have special cases to report, FTYPE not normal, when present all channels report

64b							
8b	8b	8b	8b	8b	8b	8b	8b
ASIC3 Lane 1 Head	ASIC2 Lane 1 Head	ASIC1 Lane 1 Head	ASIC0 Lane 1 Head	ASIC3 Lane 0 Head	ASIC2 Lane 0 Head	ASIC1 Lane 0 Head	ASIC0 Lane 0 Head
Lane 3 Headers [32b]				Lane 2 Headers [32b]			
Lane 5 Headers [32b]				Lane 4 Headers [32b]			

Header (12 bits)				Data	Comment
BXID	Parity	Flag	Length		
4-bit	1-bit	1-bit	6-bit	12n-bit	
0000_b	1_b	1_b	$11\ 0000_b$	Not present	Idle packet (append if no enough data)
bxid	*		$01\ 0000_b$		CorruptedEvent
			$01\ 0001_b$		BXVeto
			$01\ 0010_b$		HeaderOnly
			$01\ 0011_b$		BusyEvent (nHits>63)
			$01\ 0100_b$		BufferFull
			$01\ 0101_b$		BufferFullNZS
			$00\ 0110_b$	data	NZS packet, true length is fixed in firmware
	*	0_b	nHits	data	Normal event (nHits≤63)
12-bit bxid				pattern	Synch packet, fill one whole frame

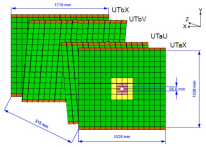
Data Flow Architecture

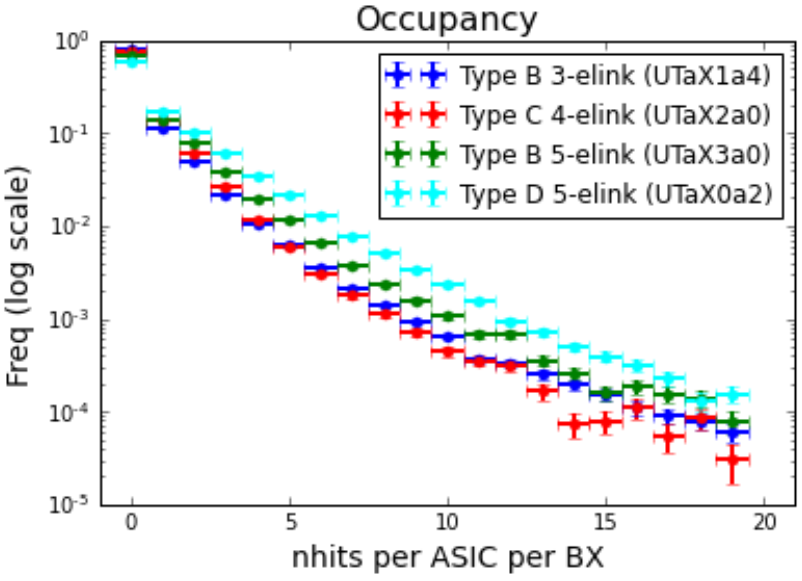
- But isn't that terribly inefficient?!

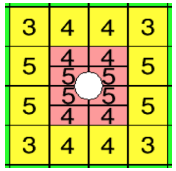
Example of a normal event

Event Headers & Flags						LANE 5		LANE 4		LANE 3		LANE 2		LANE 1		LANE 0	
16b		16b		16b		16b	16b	16b	16b	16b	16b	16b	16b	16b	16b	16b	16b
8b	8b	8b	8b	8b	8b	8b	8b	8b	8b	8b	8b	8b	8b	8b	8b	8b	8b
Event 0 Header						Hit1	Hit0		Hit0	Hit1	Hit0	Hit1	Hit0	Hit1	Hit0	Hit1	Hit0
											Hit2	Hit3	Hit2		Hit2		
													Hit4				

- It trades efficiency for other benefits. It needs to fit in our ~90Gbps PCIe available bandwidth.
- We have done simulations using the most realistic MC data we have



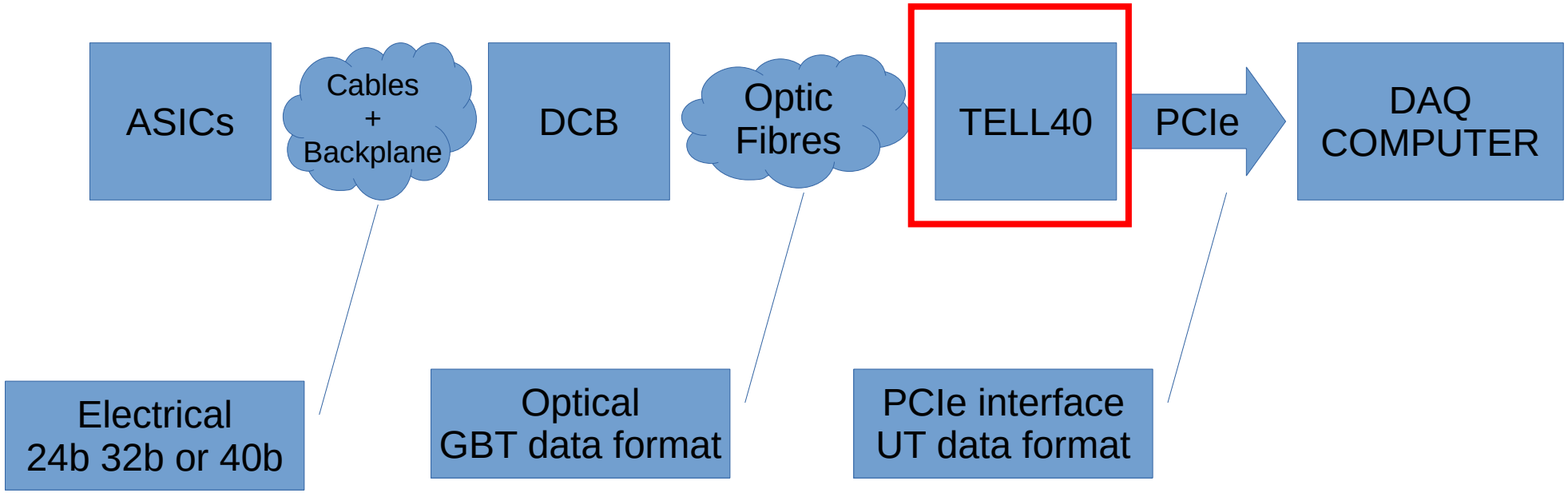




- Simulating the busiest TELL40s we have computed worst cases for all configurations (3,4,5 elinks)
- Our simulations do not go above 56.15Gbps
- True it could be more efficient, but it is very convenient

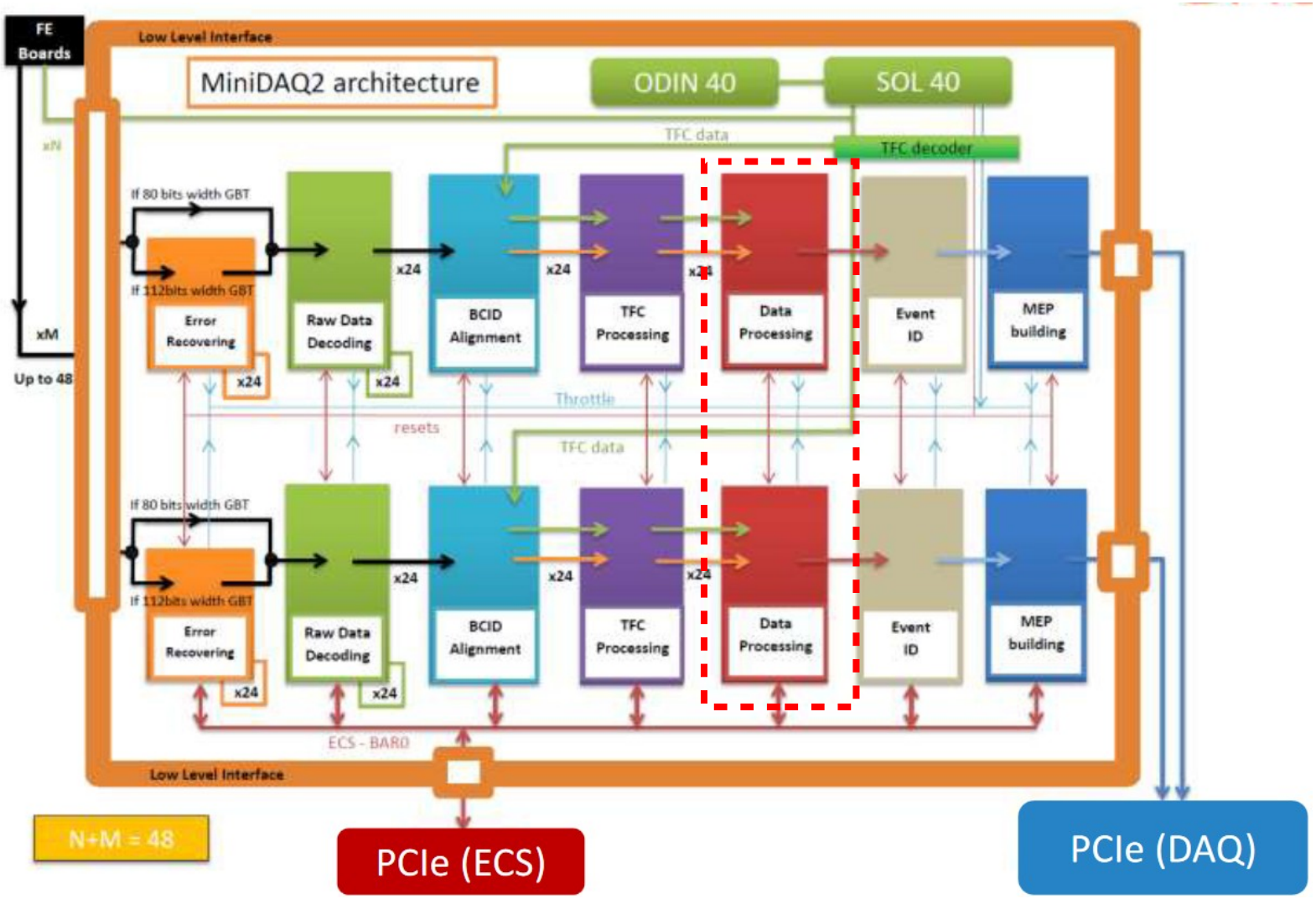
Data Flow Architecture

- Implementation



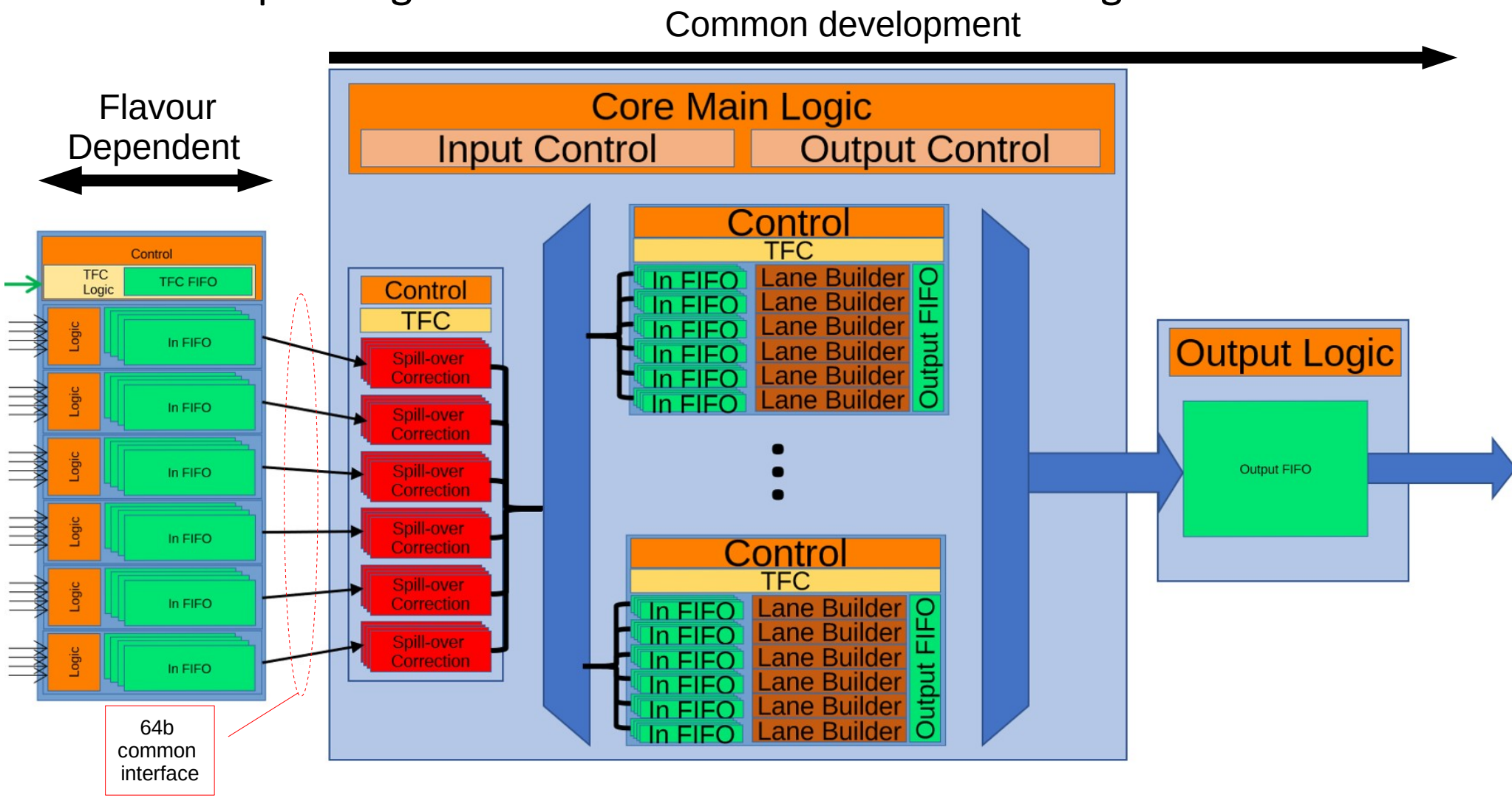
Data Flow Architecture

- FPGA implementation
  - LHCb provides a framework



Data Flow Architecture

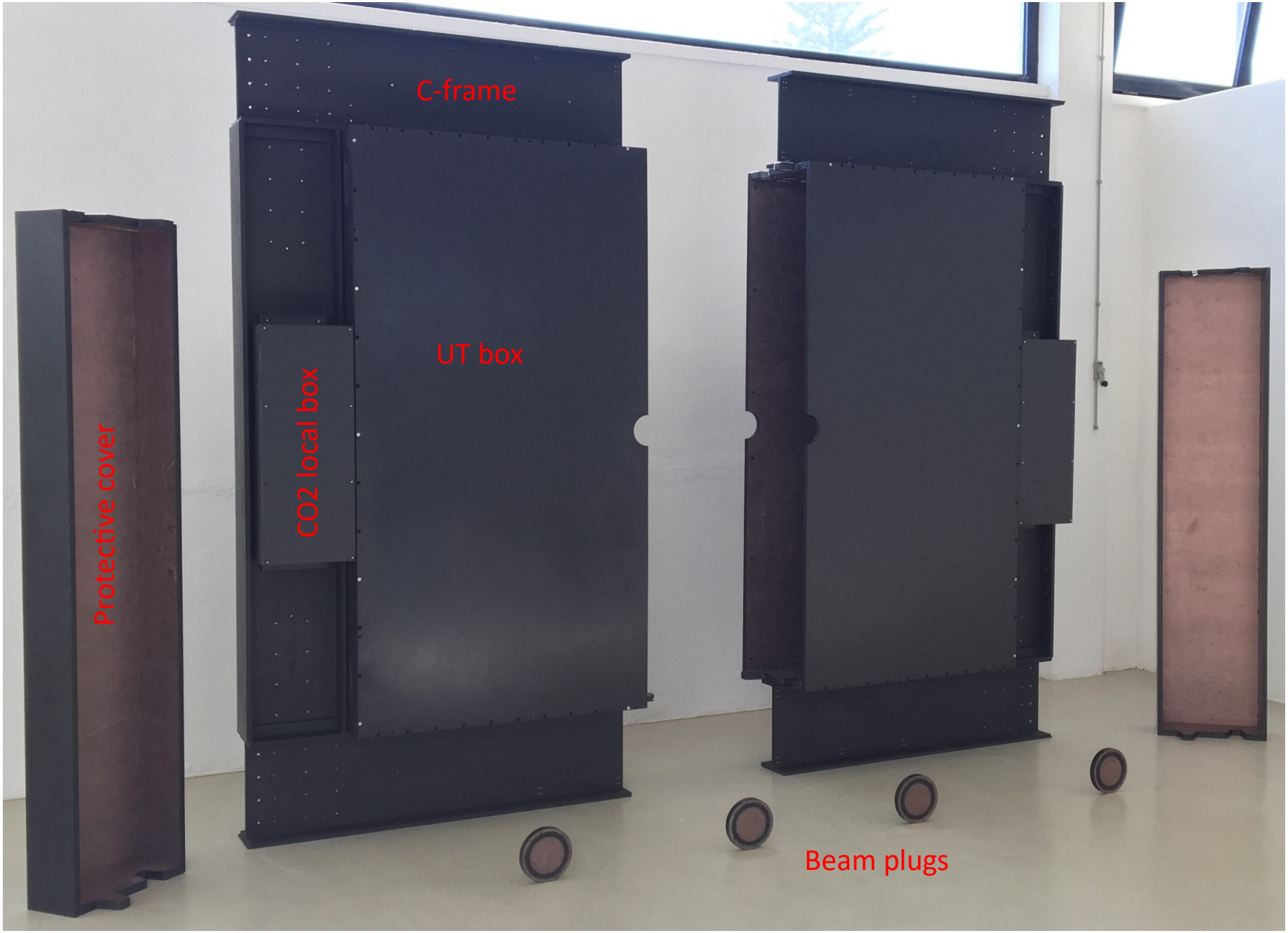
- FPGA implementation
  - LHCb provides a framework: data is received in a bus of 24b, 32b or 40b depending on what flavour the fibre is receiving.



Conclusions

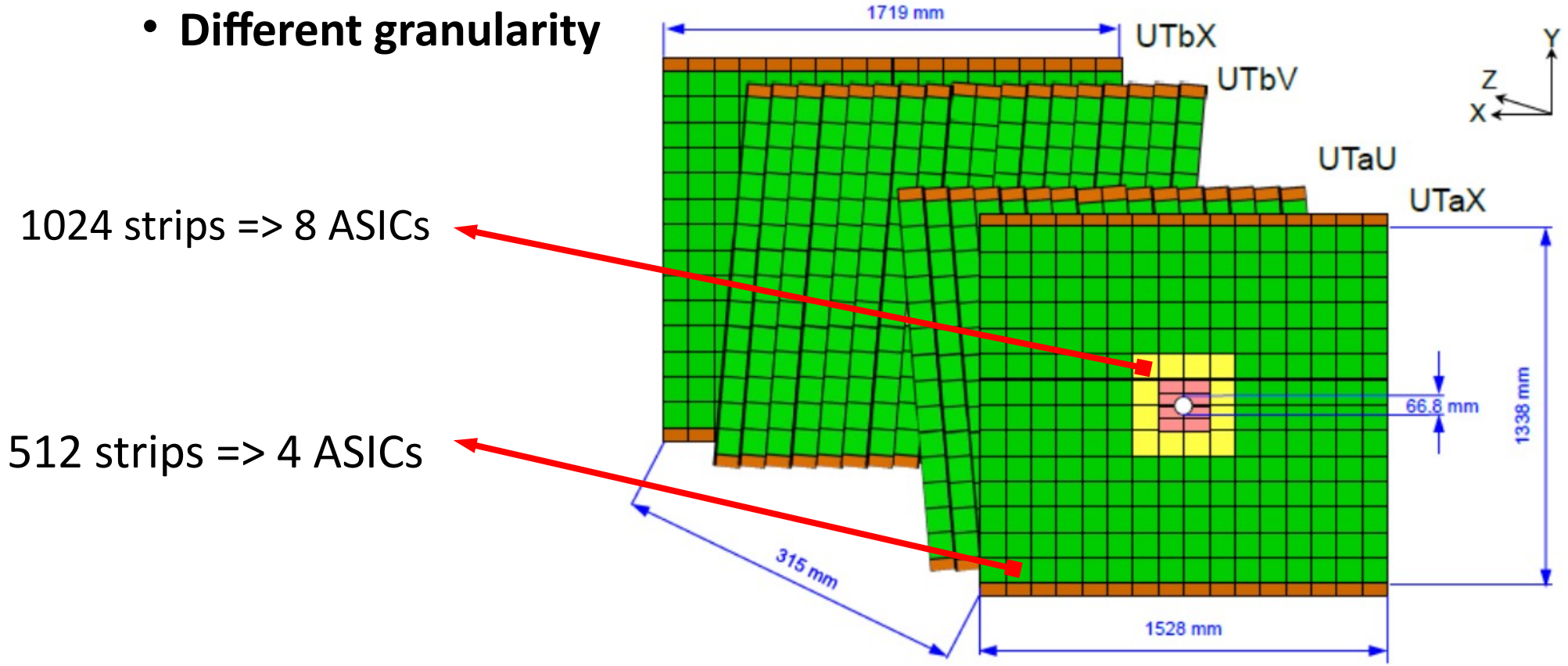
- Conclusions
  - The UT data flow architecture is presented
  - All data formats are defined
    - Several cases because of occupancy and geometry
    - All cases are covered: Different occupancies, different number of links, all possible states of all ASICs, etc…
    - The output format is easy to decode in a CPU
    - Information loss due to errors is minimized
    - Our simulations show that we have a good headroom in terms of bandwidth
    - The architecture is designed to hide all complexity from the DAQ and present a uniform data format
    - The design is intended to minimize the impact of having multiple cases to cover
  - Implementation is undergoing

Backup Slides



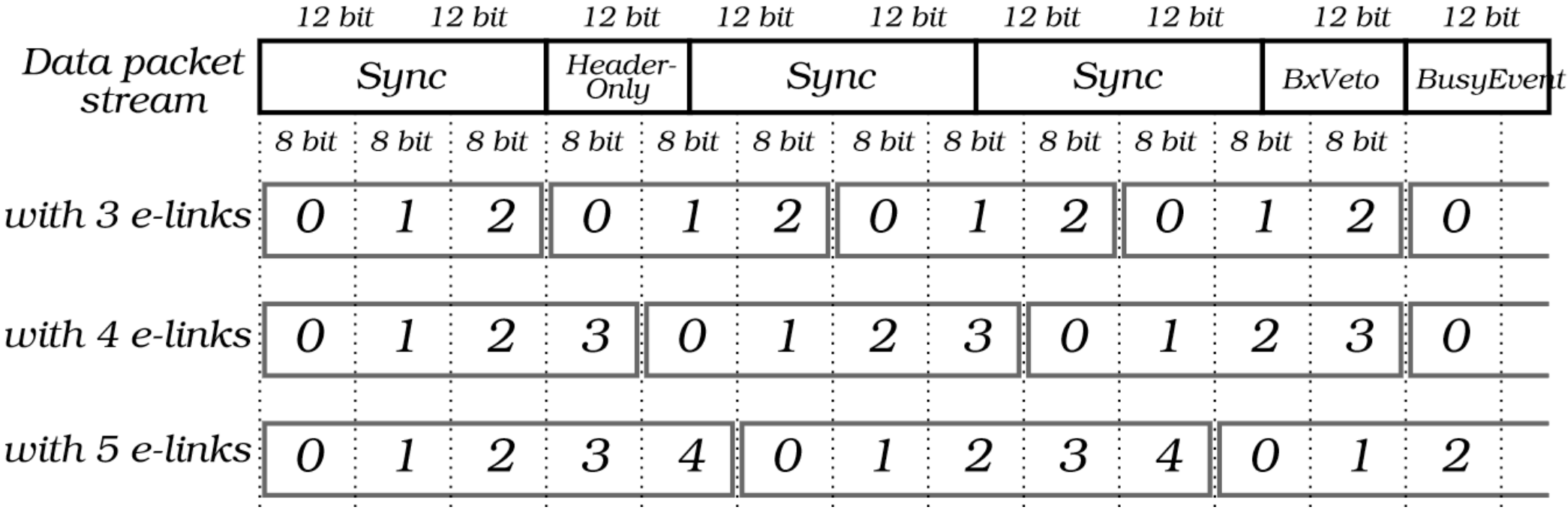
Hardware description

- Restrictions and irregularities:
  - Material budget: minimal metal in acceptance are
    - Minimal traces
  - Different occupancy depending on position
  - **Different granularity**



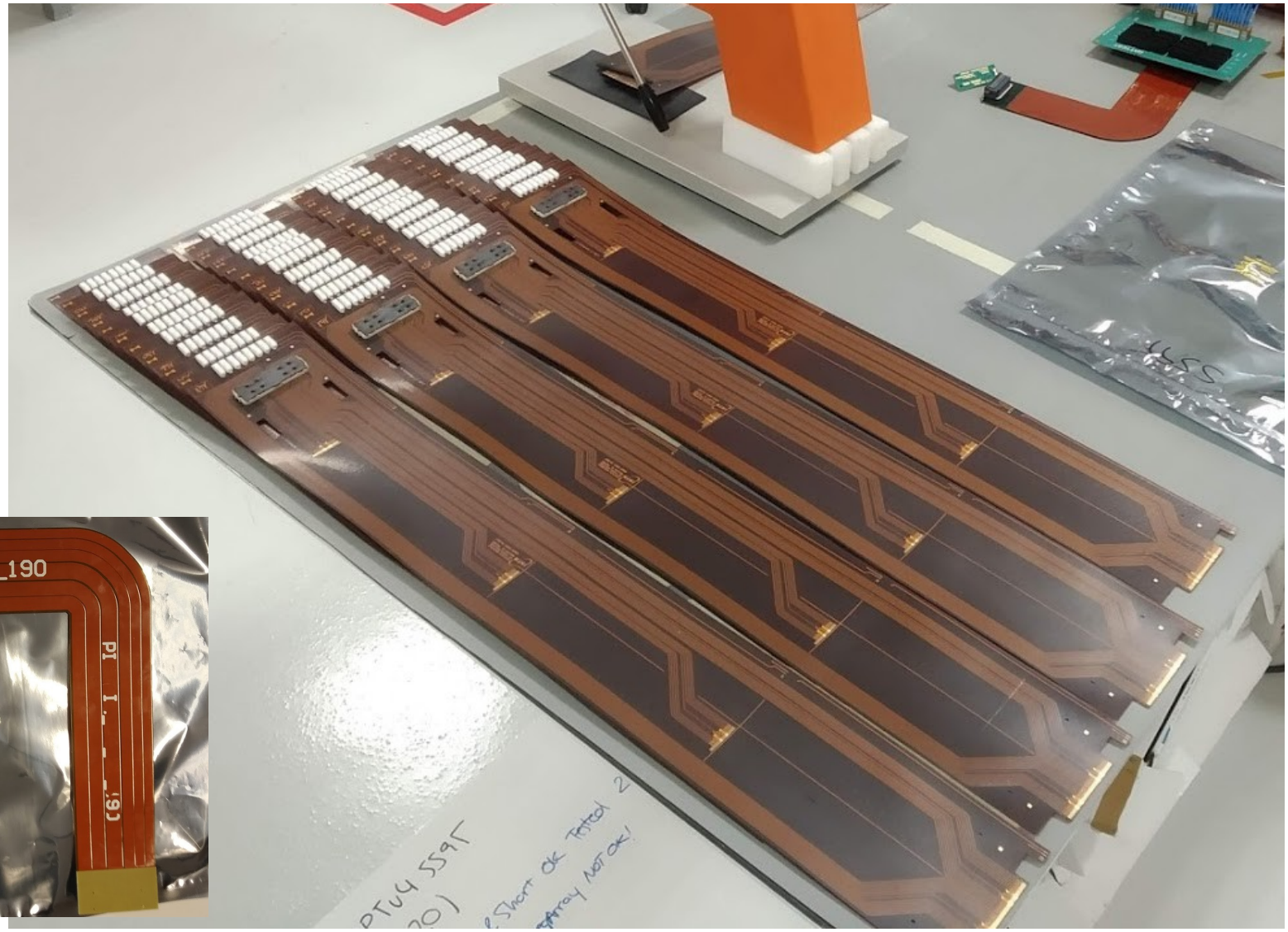
Data Flow Architecture

- ASICs send data in a common format. But depending on the number of elinks information is spread differently over the links:



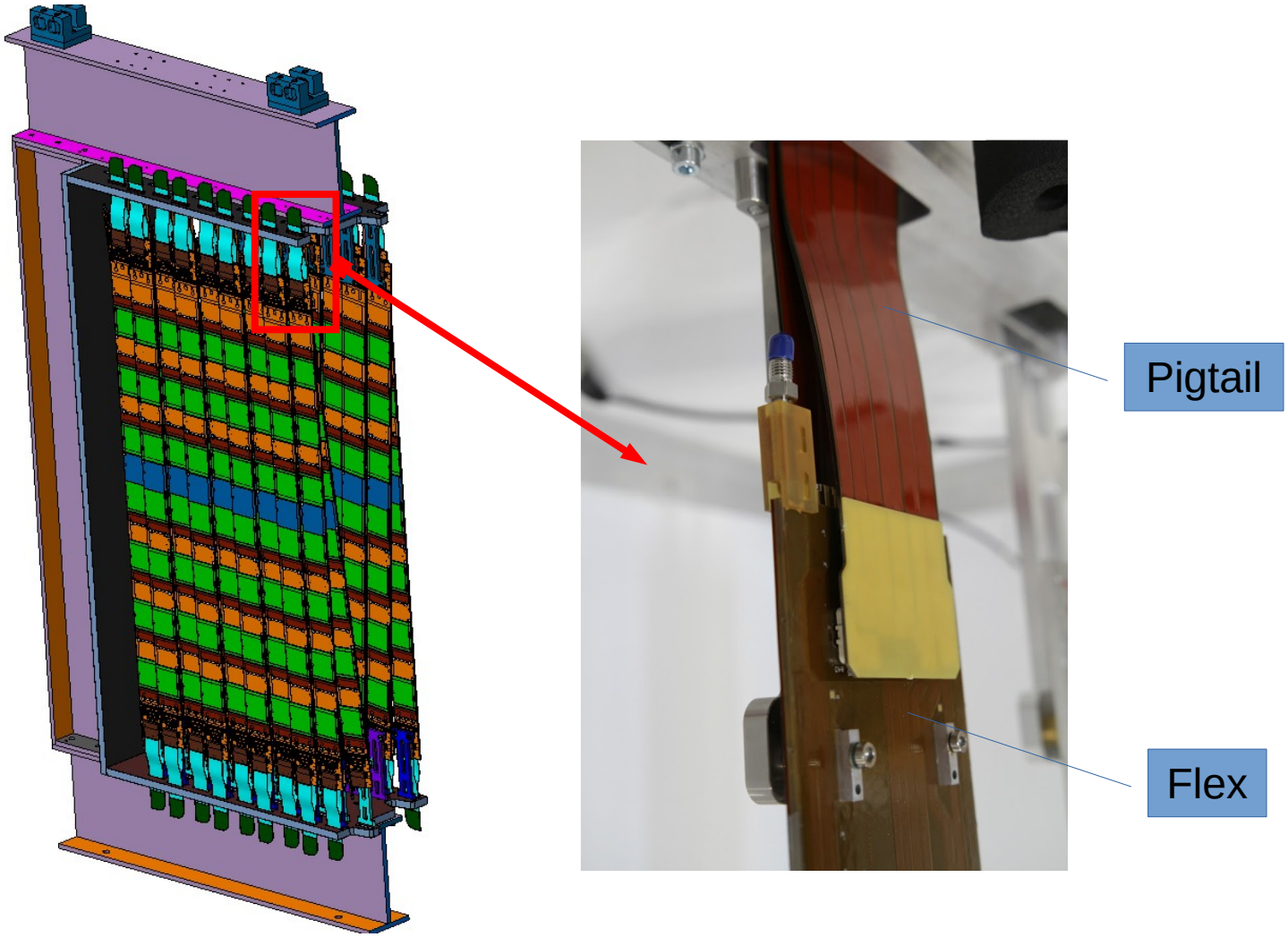
Hardware description

- Flex cables provide power to the chips and transmit signals to the readout electronics.



The data flow

- The flexes are connected to pigtail cables that cross the detector box boundaries and connect to the readout electronics



The data flow

- DCBs mainly host GBT chips and optical transceivers