OAK RIDGE National Laboratory

SVT Calibrations

J. Schambach

ORNL is managed by UT-Battelle LLC for the US Department of Energy

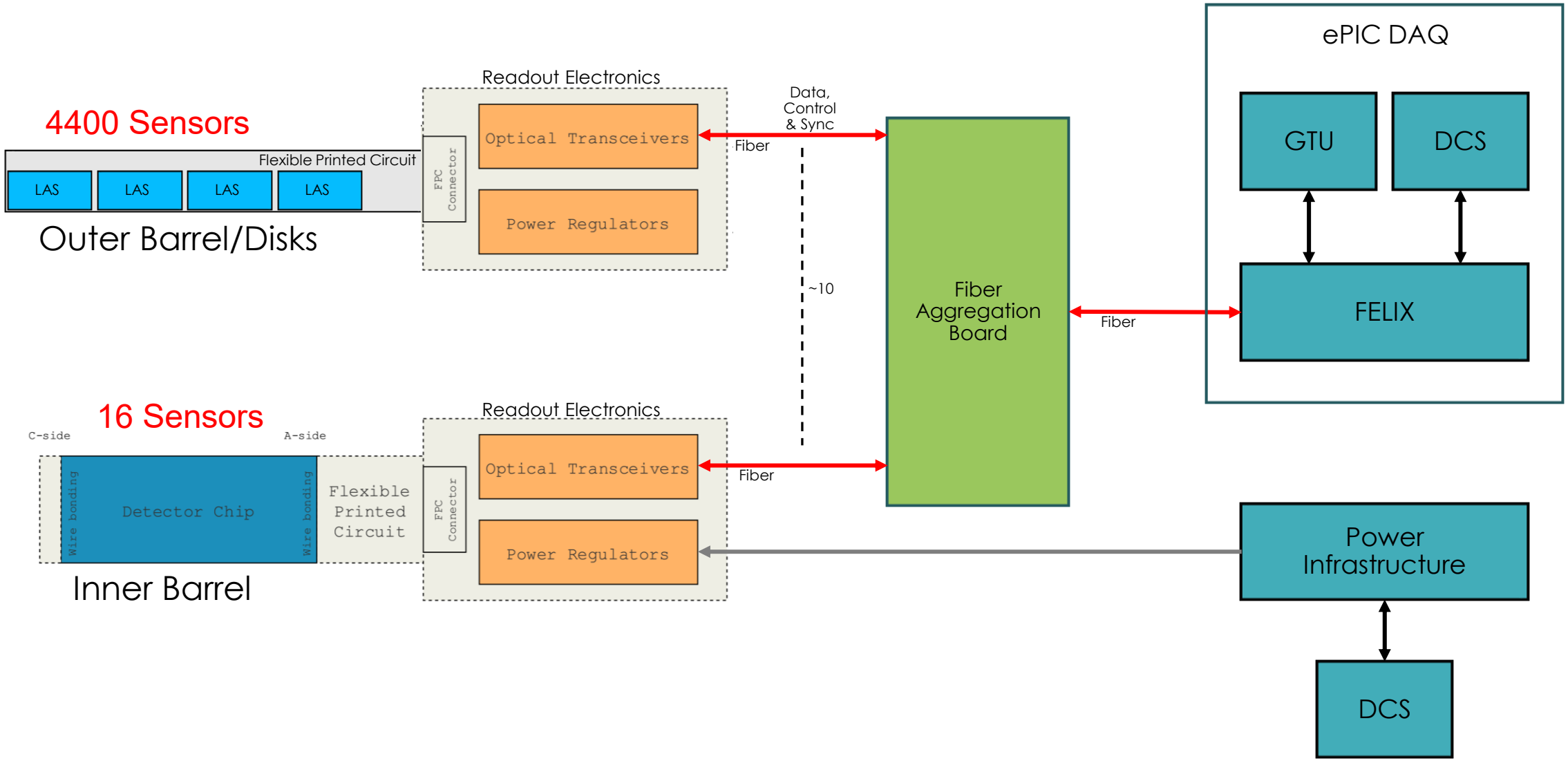
U.S. DEPARTMENT OF ENERGY

SVT Numbers

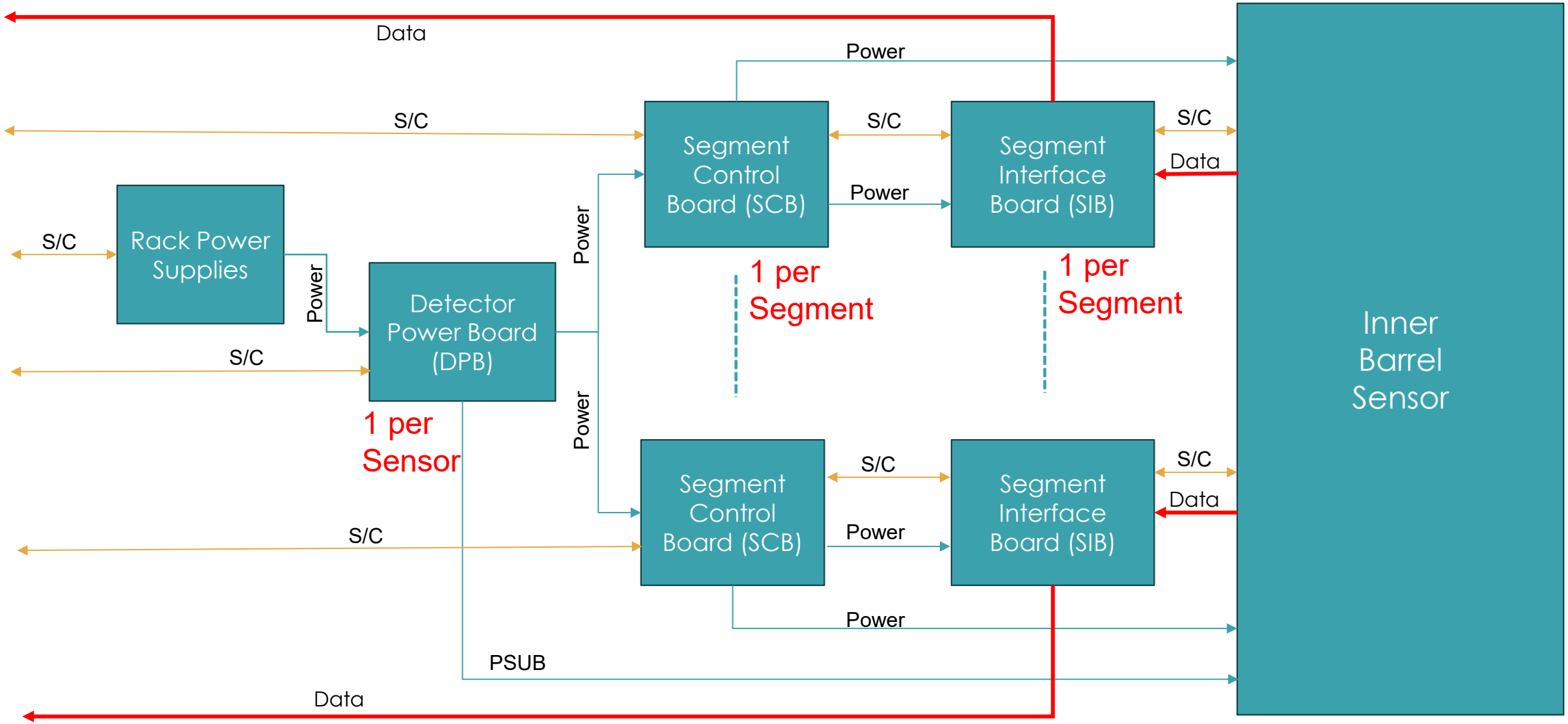
Barrel							10x275 GeV	
Layer Index	Area (mm^2)	# sensors	# pixels	# Data Fibers	# SC Fibers	Hit Rate/ms	Data Rate (Mbps)	Data Rate / Fiber (Mbps)
L0	58,811	4	119,688,192	96	6	1791732.00	218717.29	6075.48
L1	78,414	4	159,584,256	128	8	1095604.00	133740.72	2786.27
L2	196,035	8	398,960,640	320	20	600729.00	73331.18	611.09
L3	882,158	368	1,835,218,944	368	16	1645551.00	200872.92	545.85
L4	2,216,706	1120	4,654,540,800	1120	48	888591.00	108470.58	96.85
e-endcap								
Disk index								
ED0	176,710	96	398,960,640	96	8	538309.00	65711.55	684.50
ED1	536,815	334	1,388,050,560	334	16	976195.00	119164.43	356.78
ED2	553,632	334	1,388,050,560	334	16	940608.00	114820.31	343.77
ED3	552,835	334	1,388,050,560	334	16	618422.00	75490.97	226.02
ED4	551,127	334	1,388,050,560	334	16	97019.00	11843.14	35.46
h-endcap								
Disk index								
HD0	176,710	96	398,960,640	96	8	438216.00	53493.16	557.22
HD1	536,815	334	1,388,050,560	334	16	624534.00	76237.06	228.25
HD2	553,216	334	1,388,050,560	334	16	165565.00	20210.57	60.51
HD3	548,909	334	1,388,050,560	334	16	14334.00	1749.76	5.24
HD4	542,422	334	1,388,050,560	334	16	7064.00	862.30	2.58
TOTAL	**8,161,315**	**4448**	**19,070,318,592**	**4896**	**242**	**1.0442E+10**	**hits/s**	
						1244.8	**Gbps**	
			Noise Rate			1.00E-06	(1/pixel/10us)	
			Total Noise Pixels/s			1.91E+07	pixel/s	
			Total Noise Rate			**114**	**Gbps**	

~300 Mbps
average per link

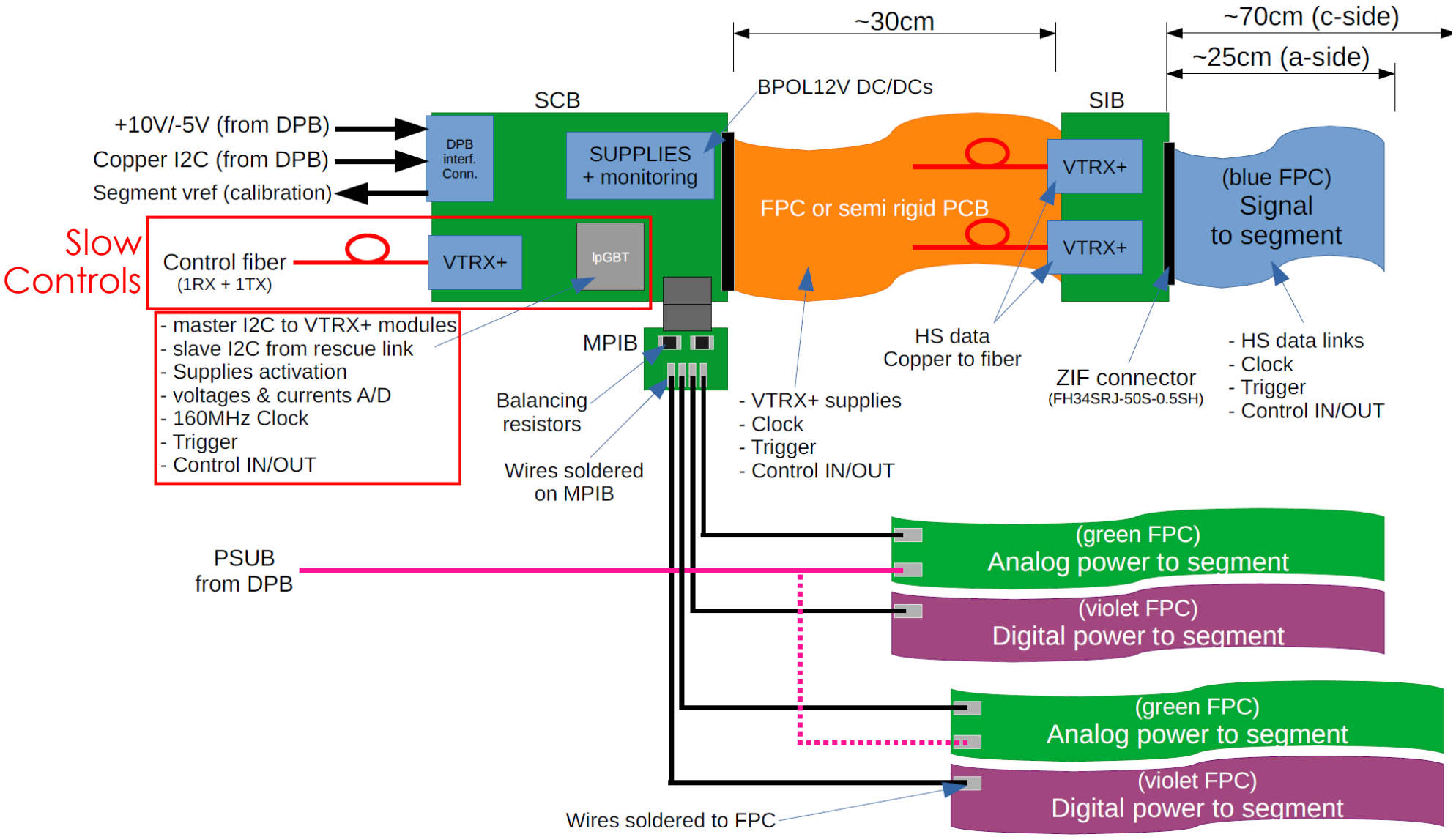
SVT Electronics – Simplified Overview



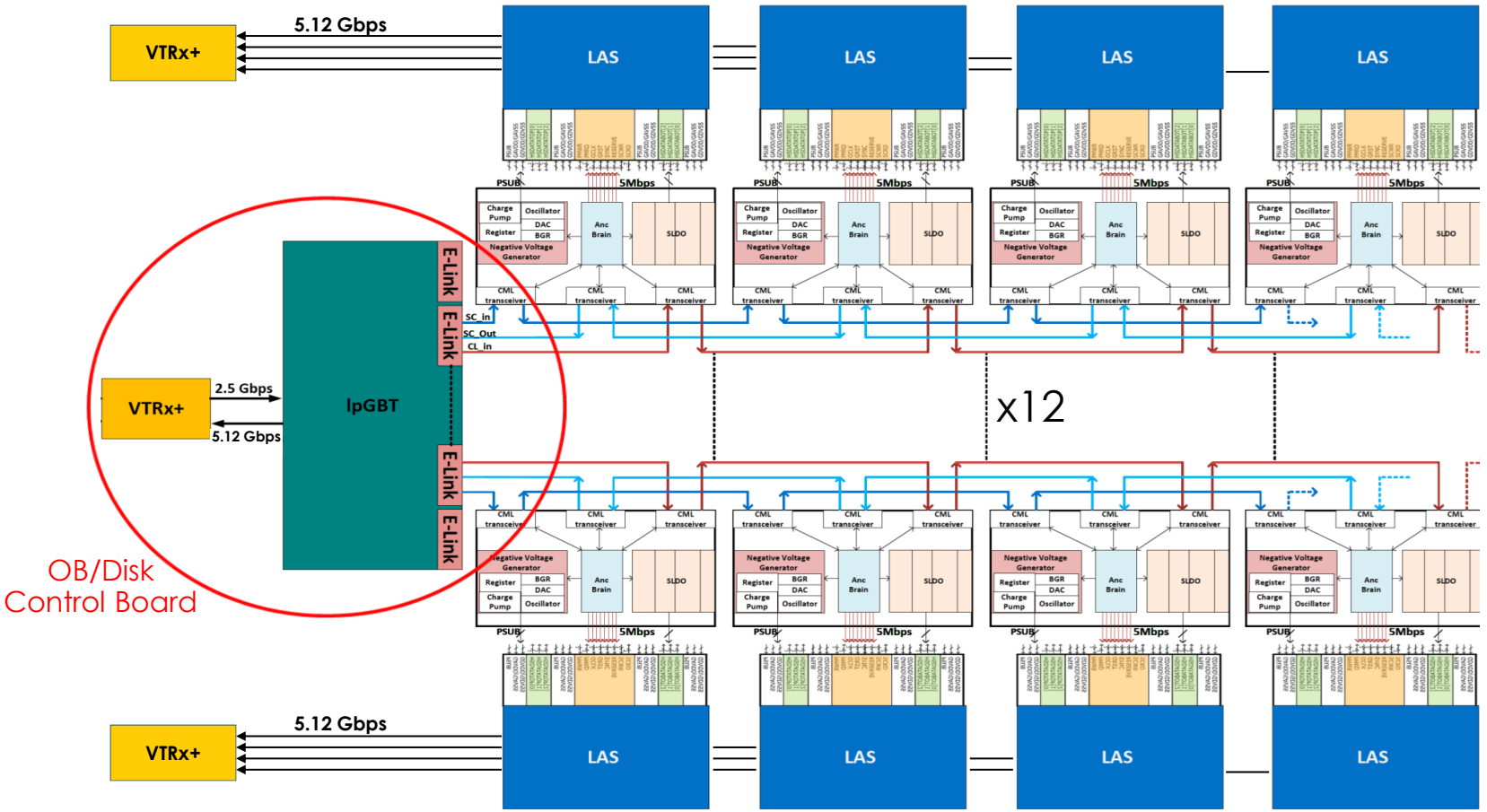
IB Readout, Power & Controls Scheme



Segment Readout Architecture



OB & Disk Readout Scheme



The Slow Controls Interface on the OB and disks is utilizing the "**ANCBrain**" of the **ANCASIC**, a custom ePIC ASIC developed jointly by **BNL**, **RAL**, and **LBNL.**

Multiple LAS sensors are daisy-chained to only one **eLink** on the **lpGBT**.

Multiple (~12) of these chains can be serviced by the provided **eLinks** (up and downlink) of just **one lpGBT**.

Summary of all calibration and monitoring scans for ITS2

(from the ALICE ITS Calibration Paper: arXiv:2510.27592v1)

Scan	Objective	Frequency	Description
Digital/Analog	Calibration	~1/year	Tagging of problematic pixels and pixel columns
VCASN	Calibration	~1/year	Course tuning of pixel thresholds
ITHR	Calibration	~1/year	Fine tuning of pixel thresholds
Short Threshold	Monitoring	~1/day	Measurement and monitoring of pixel thresholds
Full Threshold	Monitoring	~1/year	Measurement of thresholds used as reference
VRESETD	Monitoring	~1/year	Monitoring of VRESETD working point
Pulse Shape	Monitoring	~1/year	Probing of the pixel analog signal shape
Noise	Calibration	~1/year	Detection of noisy pixels

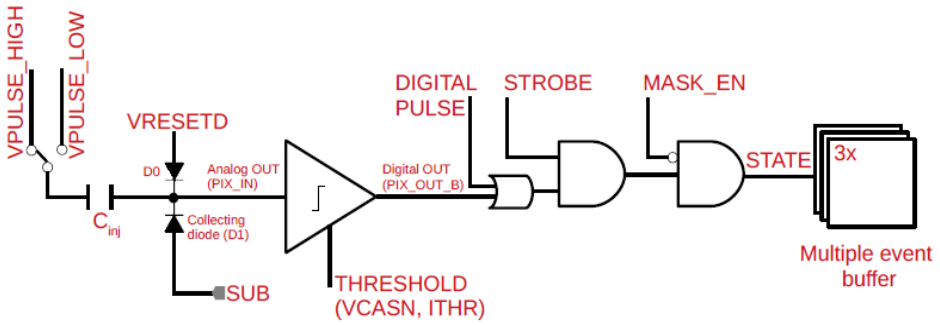


Figure 2: Simplified schematic view of the ALPIDE pixel cell.

The **MOSAIX** pixel cell is conceptually very similar to the (ITS2) **ALPIDE** pixel cell, so I am showing here the calibration scans used for ALPIDE as an example of similar scans necessary for MOSAIX.

All DACs controlling the various pixel circuitries are **global** to a pixel matrix, no individual pixel control is possible (other than the mask)

VCASN, ITHR, and Threshold scans

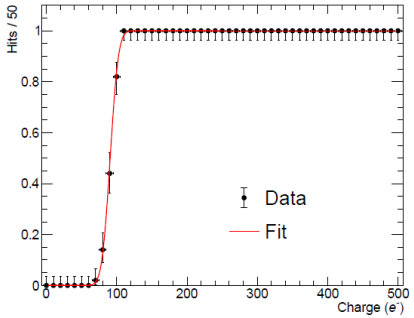


Figure 3: Number of hits normalized to the number of injections per charge point as a function of the injected charge (in electrons) for a pixel of ITS2 in a threshold scan. The errors are evaluated as Binomial errors based on the Clopper–Pearson [9] method with a confidence level of 68.27%, as provided by the ROOT `TEfficiency` class [10]. The red line represents a fit to the data performed with an error function. See text for more details.

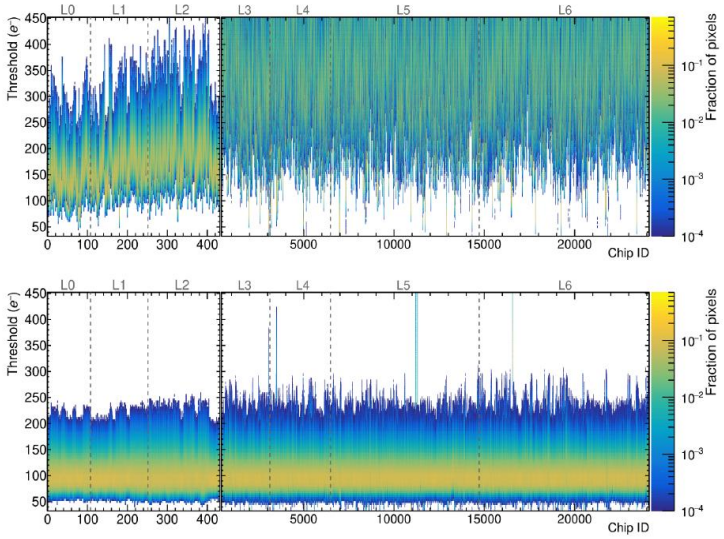


Figure 5: Pixel threshold distributions for every chip of ITS2 from two different full threshold scans, both recorded in June 2025. The top figure shows the untuned case where default settings for VCASN and ITHR (50 DACs) have been adopted. The bottom panel refers to the case where a threshold tuning to 100 e^- has been performed. The x axis of both plots is split into two parts: IB chips on the left, and OB ones on the right. The y axis maximum is set to 450 e^- since above this limit the threshold cannot be reliably extracted, given that the maximum injected change is 500 e^-. See text for more details on outlier chips.

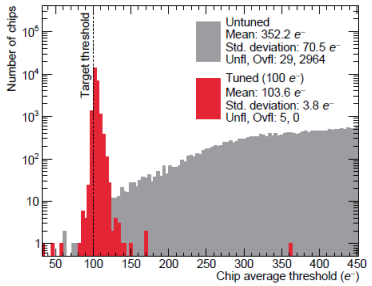


Figure 6: Distributions of the average chip thresholds in both the tuned and untuned cases. The vertical dashed line represents the target threshold of 100 e^-. The legend reports mean, standard deviation, and the number of entries in the underflow and overflow bins. See text for more details on outlier chips.

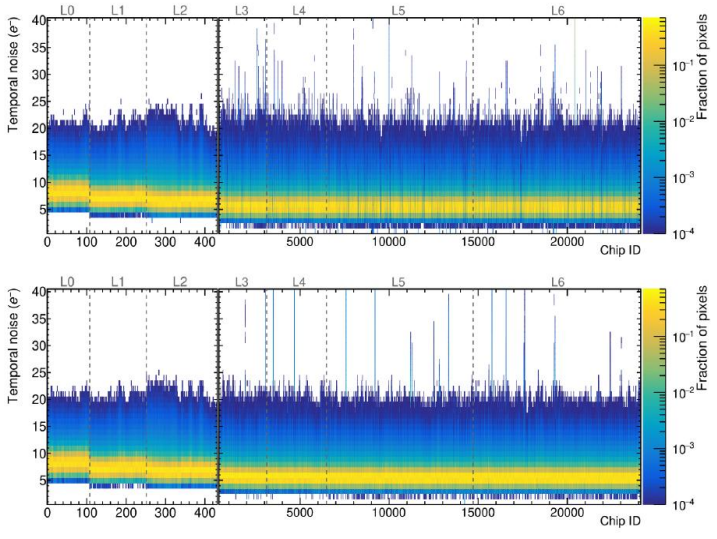


Figure 7: Pixel temporal noise distributions for every chip of ITS2 from two different full threshold scans, both recorded in June 2025. The top figure shows the untuned case where default settings for VCASN and ITHR (50 DACs) have been adopted. The bottom panel refers to the case where a threshold tuning to 100 e^- has been performed. The x axis of both plots is split into two parts: IB chips on the left (chip ID from 0 to 431) and OB ones on the right (chip ID from 432 to 24119). See text for more details on outlier chips.

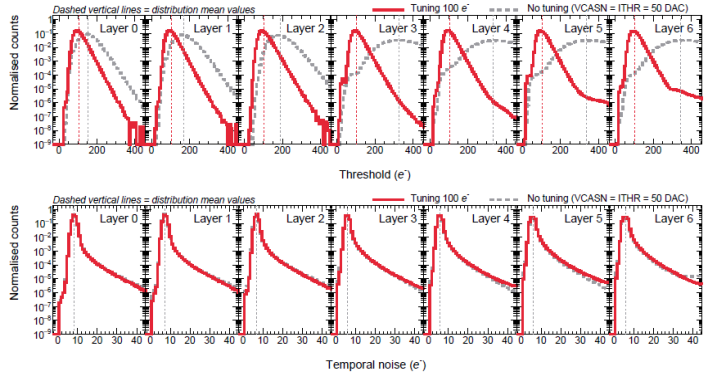


Figure 8: Distributions of the pixel threshold (top) and temporal noise (bottom) for each ITS2 layer from two different full threshold scans, both recorded in June 2025. The red distributions refer to the case with thresholds tuned to 100 e^- while the gray ones to the untuned case with the default settings for `ITHR` and `VCASN` (50 DAC for both). Distributions are normalized to their integral in order to have a direct comparison between the layers. The vertical dashed lines represent the average of the distributions.

Threshold Scan: “Bad” pixels

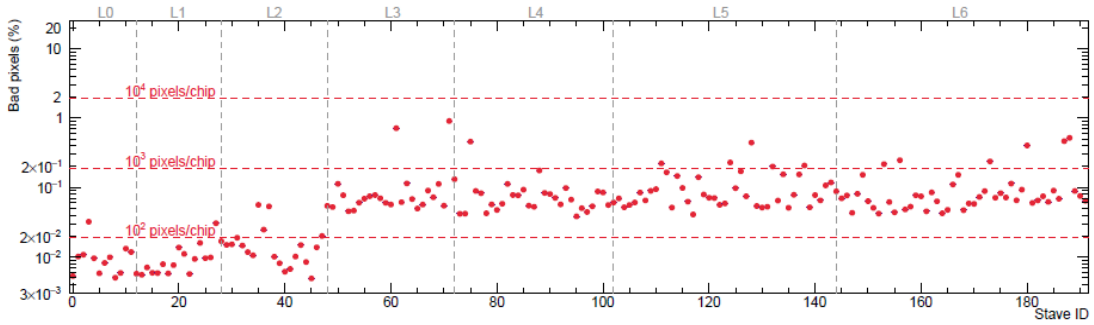


Figure 9: Percentage of bad pixels for every stave of ITS2 as extracted from a full threshold scan with thresholds tuned to 100 e^-. The fully non-working chips are excluded from the calculation. The number of chips per stave is 9, 112 and 196 for IB, ML, and OL staves, respectively. Vertical dashed lines separate the different layers, while the horizontal red lines are used as a reference to indicate the percentages corresponding to 100, 1000, and 10000 bad pixels per chip. See text for more details.

Reasons for “bad” pixels (bad S-Curve):

1. an excessive noise or a too-low threshold leading to hits independent of the thresholds;
2. a too-high threshold preventing the pixel from firing with 100% probability within the charge range;
3. a malfunction in the charge injection circuit, preventing the pixel from being stimulated;
4. a malfunction in the pixel or its respective readout circuitry.

Noise Scans: Fake Hit Rate (FHR) runs

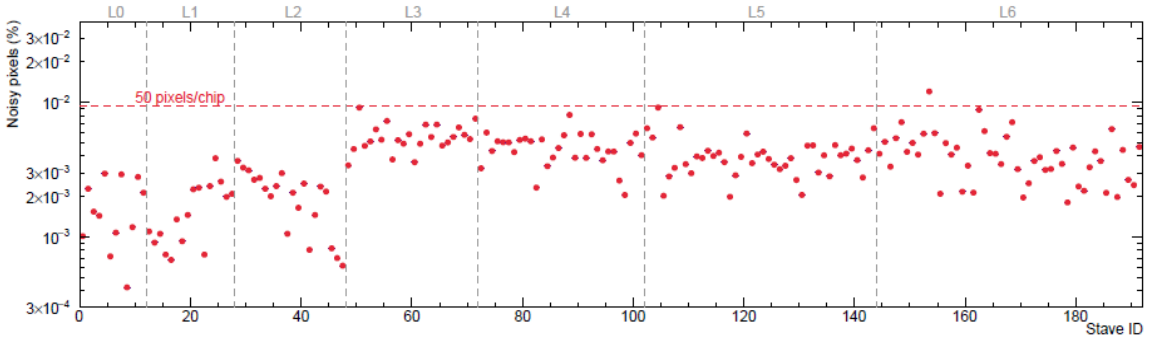


Figure 10: Percentage of noisy pixels for every detector stave in a noise scan performed in September 2023 after a threshold tuning to 100 e^-. The number of chips per stave is 9, 112 and 196 for IB, ML, and OL staves, respectively. Vertical dashed lines separate the different layers, while the horizontal red line is used as a reference to indicate the percentage corresponding to 50 noisy pixels per chip.

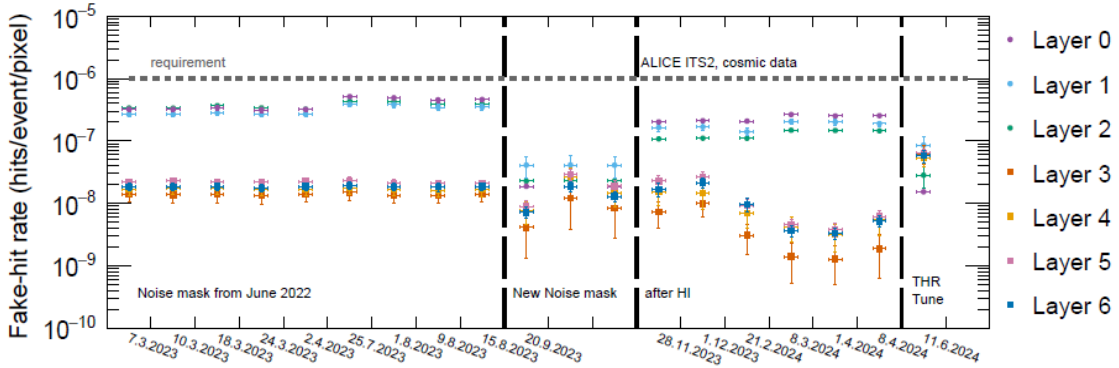


Figure 18: Trend of the fake-hit rate averaged for every detector layer. Cosmic runs in different periods are used to determine the fake-hit rate. The error bars represent half of the difference between the maximum and minimum fake-hit rate of the chips in a given layer. See the text for more details on the four different periods separated by vertical dashed lines.

VRESETD and Pulse Shape Scans

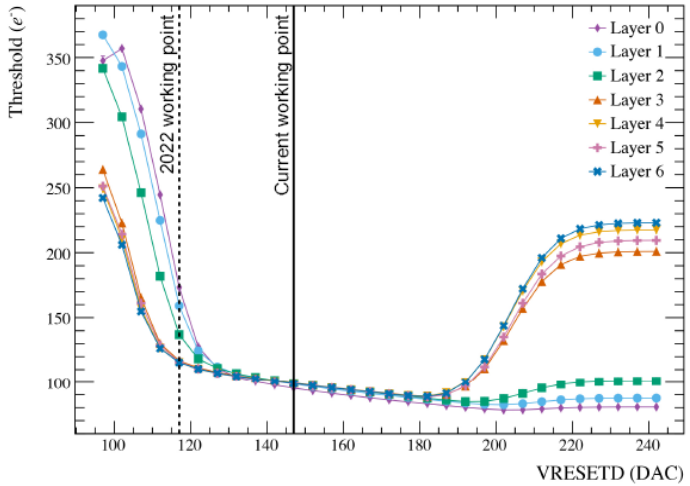


Figure 11: Results from the `VRESETD two-dimensional scan` recorded in July 2023. The x axis represents the value at which the `VRESETD` DAC is set, the y axis represents the average threshold per layer in e^-. The error bars on the average thresholds are estimated as the standard deviation of the average chip threshold distribution per layer at a given `VRESETD` divided by the square root of the number of entries in the distribution. The bars are smaller than the marker size. The solid black line indicates the DAC setting in use since 2023 during standard operations. The dashed black line indicates the DAC setting used until 2022.

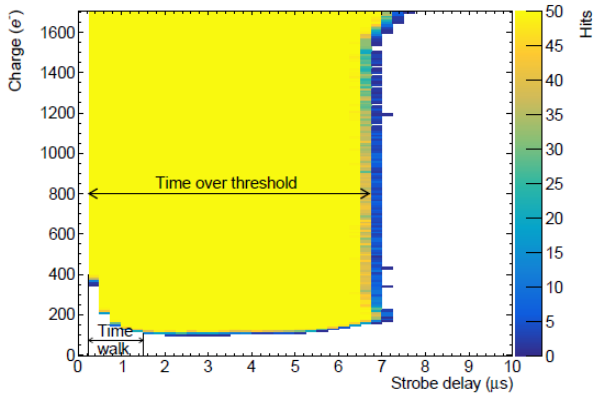


Figure 12: Two-dimensional histogram showing the injected charge as a function of the strobe delay for a pixel of ITS2. The colored scale refers to the number of hits recorded for each charge-delay pair. The ALPIDE is operated with the maximum signal clipping (`VCLIP` = 0 DAC units) and the pixel under study has a tuned threshold of about 100 e^-. The double arrows indicate the time-over-threshold for a charge of 800 e^- and the *time walk*. See text for more details.

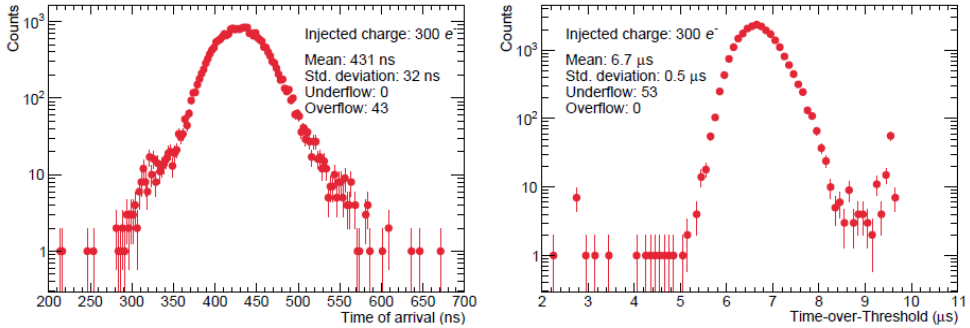


Figure 13: Left panel: distribution of the ToA of the ALPIDE analog pulse calculated chip by chip as an average of 1024 values coming from the pulse shape scan of 1 pixel row. Right panel: distribution of the ToT of the ALPIDE analog pulse calculated chip by chip as an average of 1024 values coming from the pulse shape scan of 1 pixel row. Thresholds were tuned to 100 e^- for this measurement, and a charge of 300 e^- was injected in the pixels. Error bars correspond to Poissonian errors of the counts.

Threshold Scan Trends Over Time

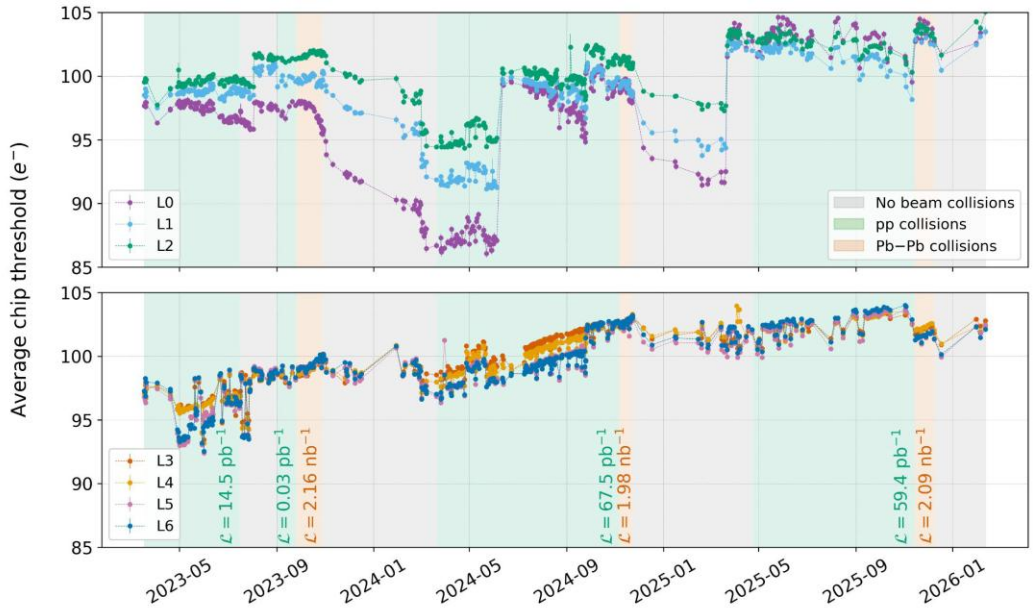


Figure 14: Evolution of the average in-pixel thresholds per layer, from March 2023 to January 2026. All the threshold runs are recorded either during long (weeks) periods without beam or at the end of every LHC fill once the beam is dumped. The error bar for the threshold is calculated as the ratio between the standard deviation and the square root of the number of entries in the per-chip threshold distributions. For the majority of the data points, the error bar is smaller than the marker size. The gray bands indicate the long periods without beam collisions, the green and orange bands indicate periods with pp and Pb–Pb collisions, respectively. For each band, the value of the integrated luminosity delivered to ALICE during the corresponding period is reported.

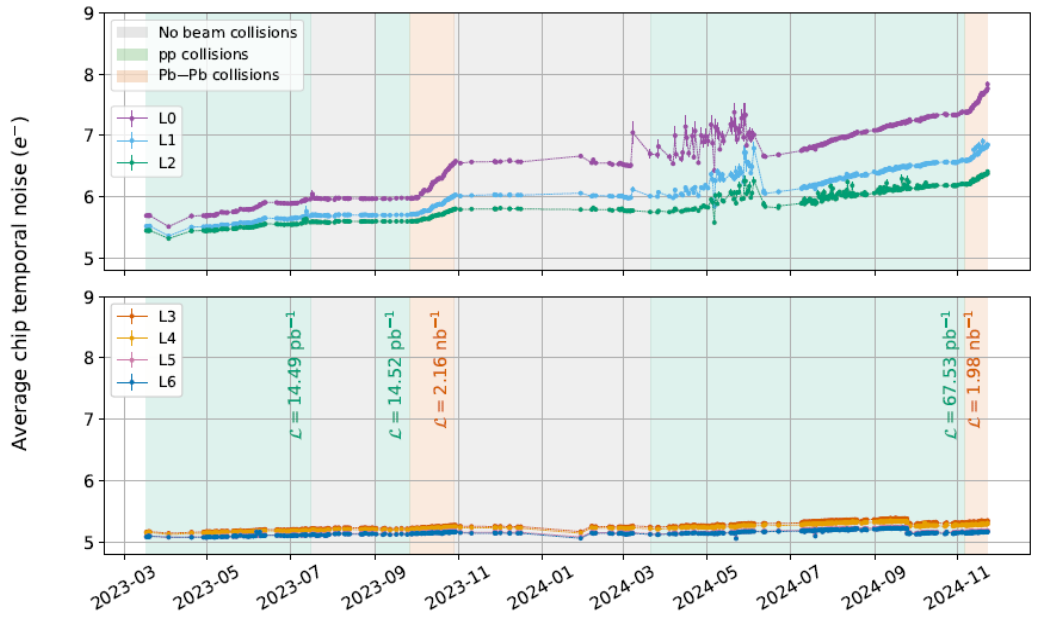
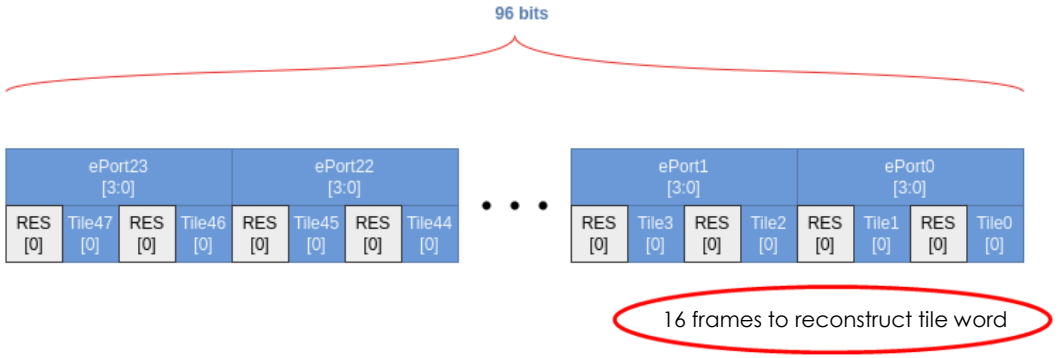


Figure 15: Evolution of the average in-pixel temporal noise per layer, from March 2023 to November 2024. The temporal noise is extracted from threshold scan runs recorded either during long (weeks) periods without beam or at the end of every LHC fill once the beam is dumped. The error bar for the noise is calculated as the ratio between the standard deviation and the square root of the number of entries in the per-pixel noise distributions. The gray bands indicate long periods without beam collisions, the green and orange bands indicate periods with pp and Pb–Pb collisions, respectively. For each band, the value of the integrated luminosity delivered to ALICE during the corresponding period is reported.

LEC Data Routing

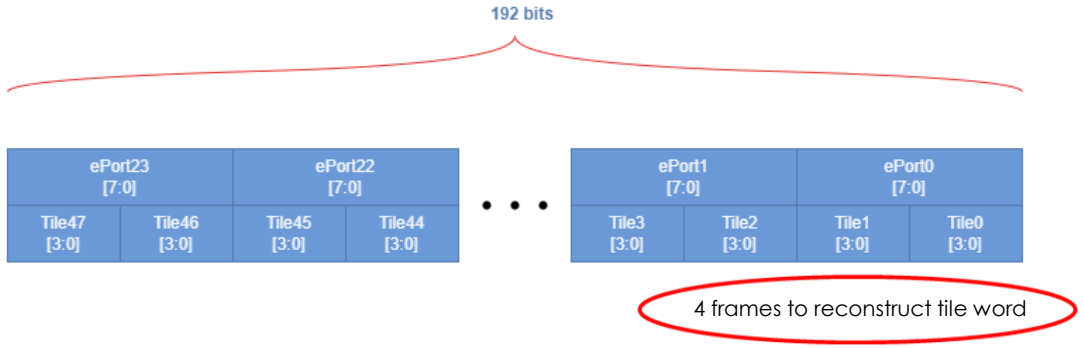
5 Gbps / 40 MHz

LAS



10 Gbps / 160 MHz

MOSAIX



1 row threshold scan:

- 1 tile header
- 1 region header
- 38 data
- 1 region header
- 38 data
- 1 region header
- 40 data
- 1 region header
- 40 data
- 1 tile trailer

= 162 (16bit) words

162 words x 16 frames x 25ns = 64.8 µs
For 1 row to transfer to aggregator board

Calibration Procedure Estimates from Jupyter Notebook (2)

Calibration: An example calibration run where the impact of a DAC setting is tested on all pixels of the sensor. A number of pulses are injected for each setting and data is read out. The calculation shows the time required to run both assuming full parallel control of all tiles of a segment, and in the case of sequential control of all tiles of a segment.

```
Parallel calibration procedure estimates (s):
```

	SC Bandwidth (MHz)	Power (s)	Config (s)	Integration (s)	Readout chip (s)	Readout RU (s)	Total (s)
0	5	0.5011	64	568	92	424	1149

20 minutes

```
Sequential calibration procedure estimates (s):
```

	SC Bandwidth (MHz)	Power (s)	Config (s)	Integration (s)	Readout chip (s)	Readout RU (s)	Total (s)
0	5	72.16	9212	81838	13258	424	104804

29 hours

In [66]:

```python
import time

DAC_RESOLUTION = 256
NUM_PULSES_PER_SETTING = 50
INTEGRATION_WINDOW = 100e-6
GAP_BETWEEN_ROWS = 1000e-6
TRANSACTIONS_PER_SETTING = 6
TRANSACTIONS_PER_ROW = PIXEL_CONFIG_CONTROL_TRANS * 2 # Disable all pixels and then enable new row

table = []
table_seq = []
for bandwidth in SC_BANDWIDTH:
    time_power = 0
    time_config = 0
    time_integration = 0
    time_readout_chip = 0
    time_readout_ru = 0
    tran_time = 1/bandwidth*SC_BITS_PER_TRAN + SC_DEADTIME

    # Powering
    time_power = POWER_ON_TRANS * tran_time
    time_power += POWER_ON_WAIT_STABLE_ADC
    time_power += POWER_MON_TRANS * tran_time
    time_power += POWER_OFF_TRANS * tran_time

    for row in range(0, NUM_ROWS):

        # Set row mask and enable pulsing
        time_config += TRANSACTIONS_PER_ROW*(tran_time)

        for dac in range(DAC_RESOLUTION):

            time_config += CALIBRATION_DATA_WORD_DURATION

            # Set config for all TRO/half-RSUs
            time_config += TRANSACTIONS_PER_SETTING*tran_time + READOUT_DONE_POLL_TIME

            # Send Pulses
            time_integration += INTEGRATION_WINDOW * NUM_PULSES_PER_SETTING

            # Readout
            time_readout_chip += TILE_ROW_READOUT_TIME * NUM_PULSES_PER_SETTING

            time_readout_ru +=  RU_ROW_READOUT_TIME * NUM_PULSES_PER_SETTING * NUM_TILE_IN_SEGMENT

        # Some gap between each row
        time_config += GAP_BETWEEN_ROWS


    total_time = time_power + time_config + time_integration + time_readout_chip + time_readout_ru

    time_power_seq = time_power*NUM_TILE_IN_SEGMENT
    time_config_seq = time_config*NUM_TILE_IN_SEGMENT
    time_integration_seq = time_integration*NUM_TILE_IN_SEGMENT
    time_readout_chip_seq = time_readout_chip*NUM_TILE_IN_SEGMENT
    time_readout_ru_seq = time_readout_ru
    total_time_seq = time_power_seq + time_config_seq + time_integration_seq + time_readout_chip_seq + time_readout_ru_seq

    table.append([int(bandwidth/1e6), round(time_power, 4), round(time_config), round(time_integration), round(time_readout_chip), round(time_readout_ru), round(total_time)])
    table_seq.append([int(bandwidth/1e6), round(time_power_seq, 2), round(time_config_seq), round(time_integration_seq), round(time_readout_chip_seq), round(time_readout_ru_se
```

Backup

Bit Counts

1 tile noise pixel:
- 1 tile header
- 1 region header
- 1 data
- 1 tile trailer

4 words
-> 64 bits per pixel

1 tile hit (4 pixels):
- 1 tile header
- 1 region header
- 4 data
- 1 tile trailer

7 words
->112 bits
-> 28 bits per pixel

1 hit (4 pixels) per region:
- 1 tile header
- 1 region header
- 4 data
- 1 region header
- 4 data
- 1 region header
- 4 data
- 1 region header
- 4 data
- 1 tile trailer

22 words
-> 352 bits
-> 22 bits per pixel

2 hits (4 pixels) in 1 region:
- 1 tile header
- 1 region header
- 8 data
- 1 tile trailer

11 words
-> 176 bits
-> 22 bits per pixel

4 hits (4 pixels each) in 1 region:
- 1 tile header
- 1 region header
- 16 data
- 1 tile trailer

19 words
-> 304 bits
-> 19 bits per pixel

1 tile hit (2 pixels):
- 1 tile header
- 1 region header
- 2 data
- 1 tile trailer

5 words
-> 80 bits
-> 40 bits per pixel

1 row threshold scan:
- 1 tile header
- 1 region header
- 38 data
- 1 region header
- 38 data
- 1 region header
- 40 data
- 1 region header
- 40 data
- 1 tile trailer

162 words
-> 2,592 bits

Calibration Procedure Estimates from Jupyter Notebook (1)

Configuration: The configuration transactions will be different for each RSU/tile/etc, no broadcast can be used to speed up the process, i.e. each tile is configured sequentially:

Posted:

	Bandwidth (MHz)	Transaction (s)	Pixel (s)	Tile (s)	Segment (s)
0	5	0.000008	0.000041	0.028052	4.039476

Non-Posted:

	Bandwidth (MHz)	Resync stages	Mean Transaction (s)	Mean Pixel (s)	Mean Tile (s)	Segment (s)
0	5	24	0.000021	0.000104	0.072381	9.554276
1	5	12	0.000018	0.000092	0.064069	8.457134
2	5	6	0.000017	0.000086	0.059536	7.858693
3	5	4	0.000017	0.000084	0.058024	7.659213
4	5	2	0.000016	0.000082	0.056513	7.459733
5	5	0	0.000016	0.000080	0.055758	7.359993

Configuration Scrubbing: To ensure that all pixels keep their configuration during a run, the configuration must be updated with regular intervals.

Posted:

	Bandwidth (MHz)	Transaction (s)	Pixel (s)	Tile (s)	Scrub cycle segment (s)	Scrub cycle segment weighted (s)
0	5	0.000008	0.000041	2.805192	403.947648	807.895296

Non-Posted:

	Bandwidth (MHz)	Resync stages	Transaction (s)	Pixel (s)	Tile (s)	Scrub cycle segment (s)	Scrub cycle segment weighted (s)
0	5	24	0.000021	0.000104	7.238088	955.427616	1910.855232
1	5	12	0.000018	0.000092	6.406920	845.713440	1691.426880
2	5	6	0.000017	0.000086	5.953556	785.869344	1571.738688
3	5	4	0.000017	0.000084	5.802434	765.921312	1531.842624
4	5	2	0.000016	0.000082	5.651313	745.973280	1491.946560
5	5	0	0.000016	0.000080	5.575752	735.999264	1471.998528

Calibration Procedure Estimates from Jupyter Notebook (2)

Monitoring:

	Bandwidth (MHz)	Resync stages	Monitoring cycle segment (s)	Monitoring cycle segment weighted (s)
0	5	24	0.26928	0.53856
1	5	12	0.25056	0.50112
2	5	6	0.24048	0.48096
3	5	4	0.23712	0.47424
4	5	2	0.23376	0.46752
5	5	0	0.23184	0.46368

Calibration: An example calibration run where the impact of a DAC setting is tested on all pixels of the sensor. A number of pulses are injected for each setting and data is read out. The calculation shows the time required to run both assuming full parallel control of all tiles of a segment, and in the case of sequential control of all tiles of a segment.

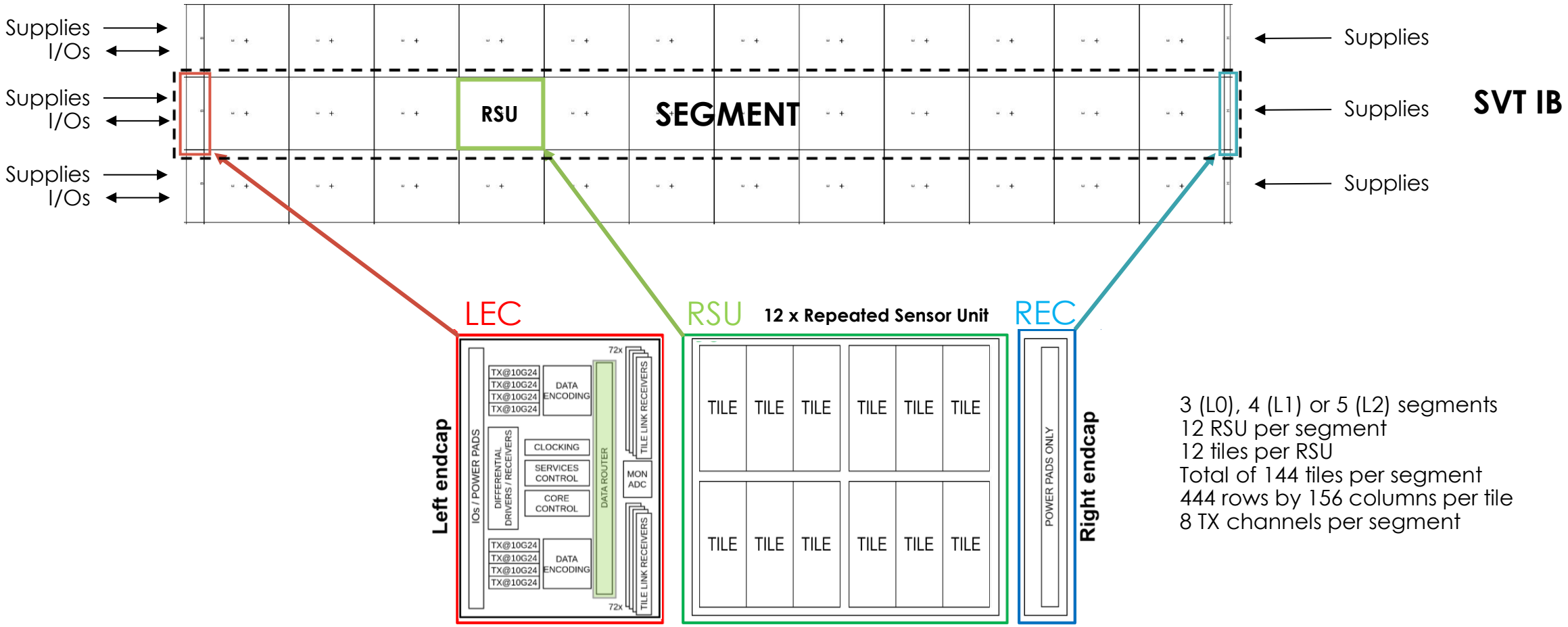
Parallel calibration procedure estimates (s):

	SC Bandwidth (MHz)	Power (s)	Config (s)	Integration (s)	Readout chip (s)	Readout RU (s)	Total (s)
0	5	0.5011	64	568	92	424	1149

Sequential calibration procedure estimates (s):

	SC Bandwidth (MHz)	Power (s)	Config (s)	Integration (s)	Readout chip (s)	Readout RU (s)	Total (s)
0	5	72.16	9212	81838	13258	424	104804

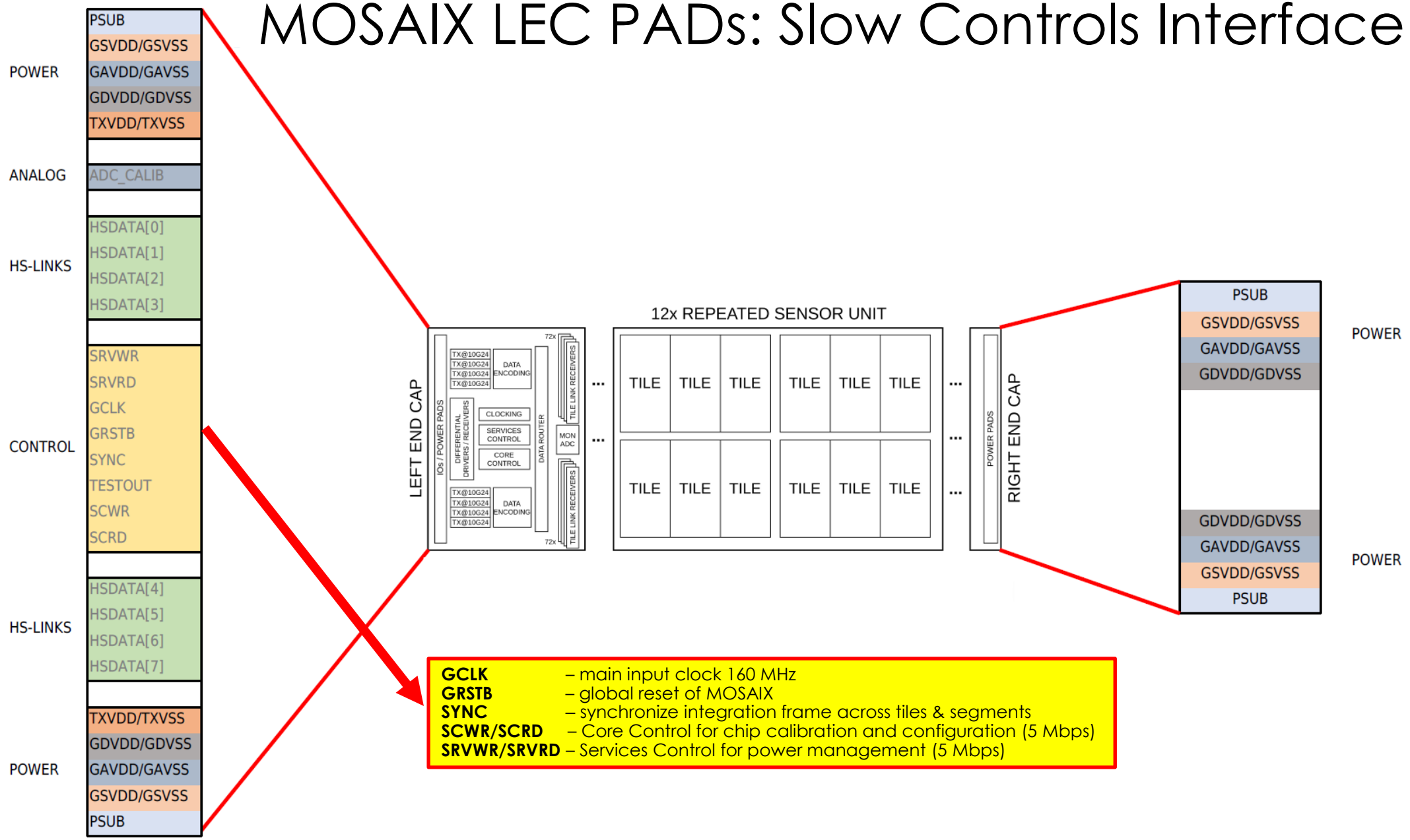
MOSAIX Stitched Sensor Detail



3 (L0), 4 (L1) or 5 (L2) segments
12 RSU per segment
12 tiles per RSU
Total of 144 tiles per segment
444 rows by 156 columns per tile
8 TX channels per segment

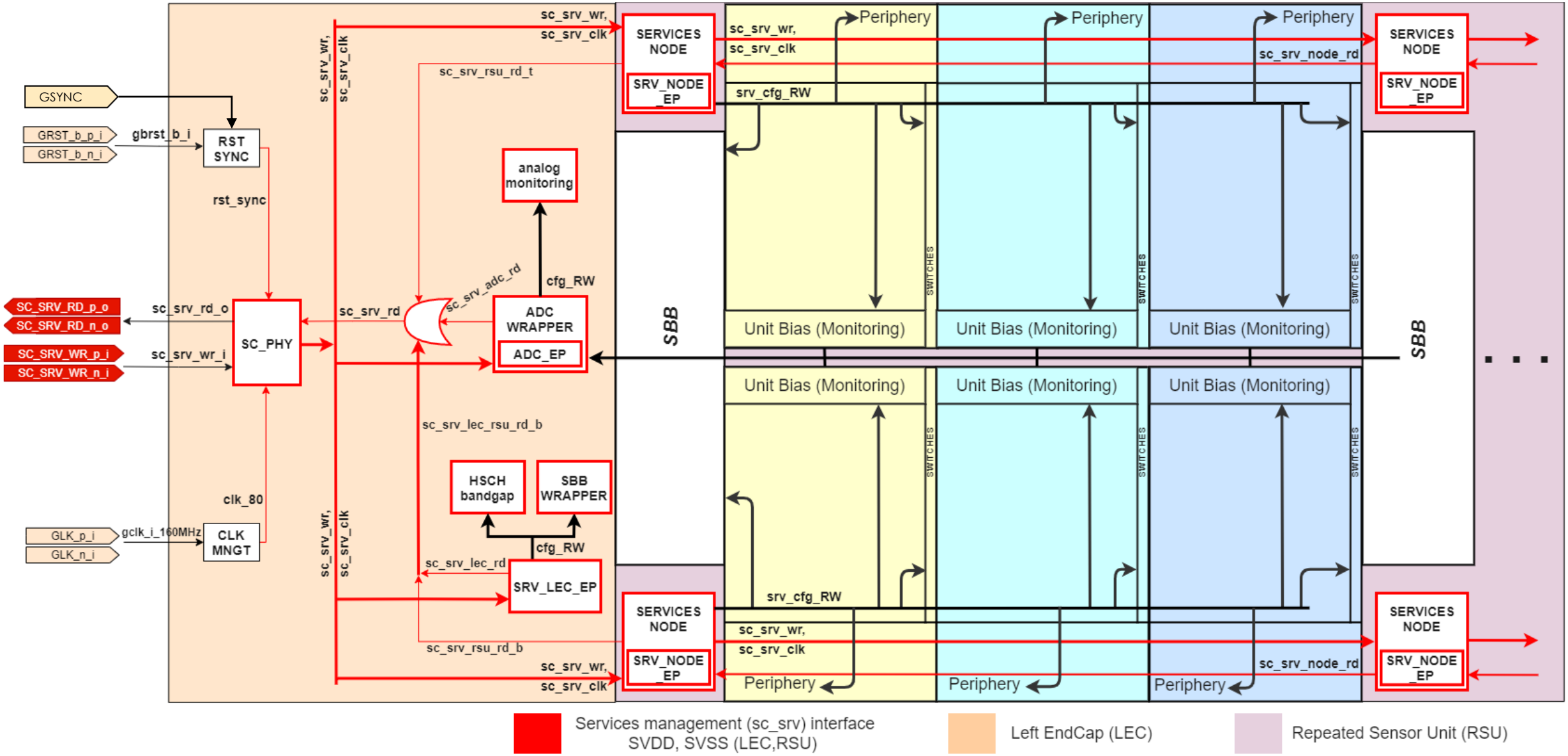
SVT **LAS** (OB, disks) is a MOSAIX with a **single** segment and only **6 RSU**, using only **1 TX** channel

MOSAIX LEC PADs: Slow Controls Interface



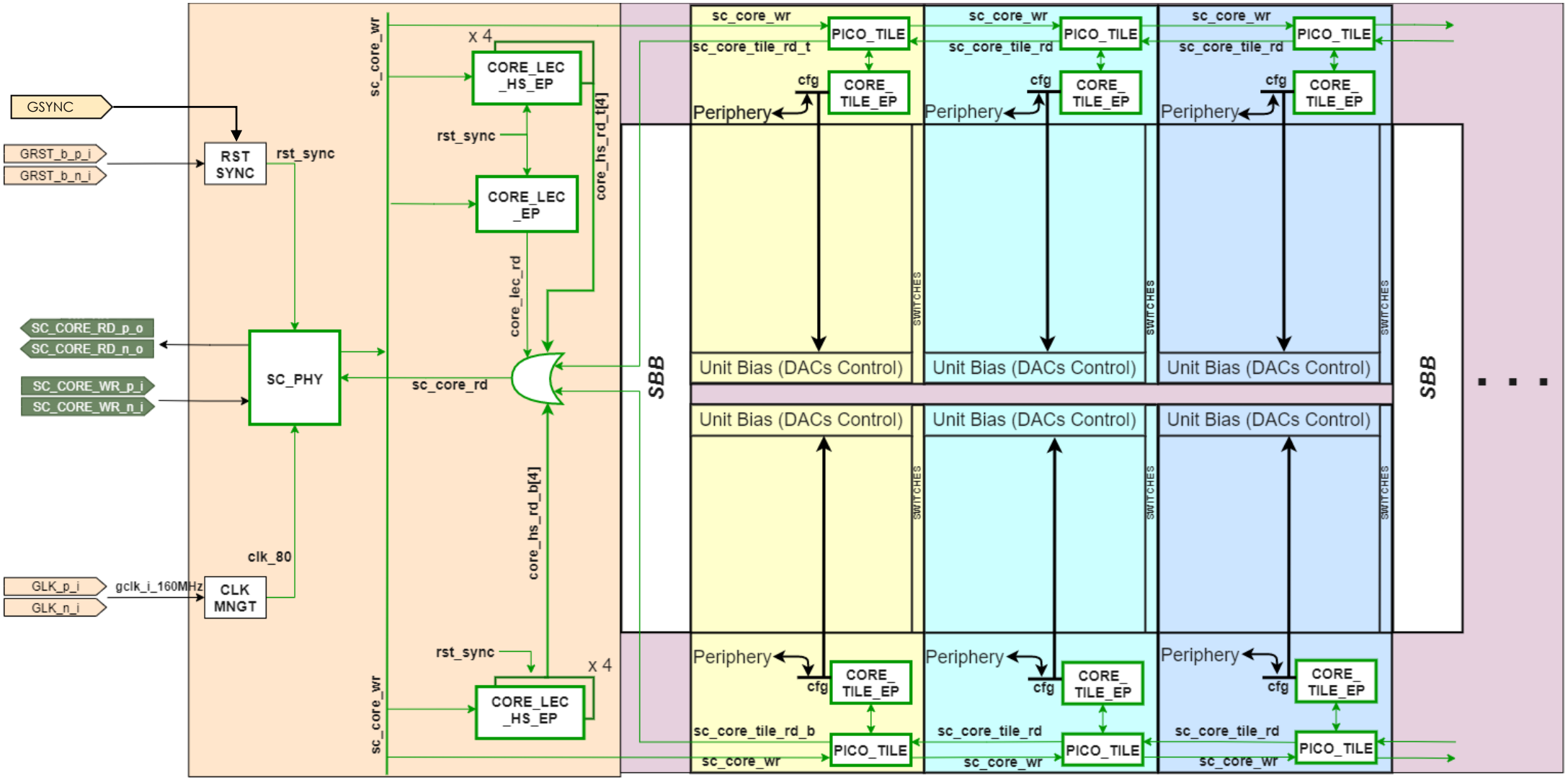
GCLK – main input clock 160 MHz
GRSTB – global reset of MOSAIX
SYNC – synchronize integration frame across tiles & segments
SCWR/SCRD – Core Control for chip calibration and configuration (5 Mbps)
SRVWR/SRVRD – Services Control for power management (5 Mbps)

Service Slow Contol Block Diagram



J. Schambach

Core Slow Control Block Diagram



Core management interface

Left EndCap (LEC)

Repeated Sensor Unit (RSU)

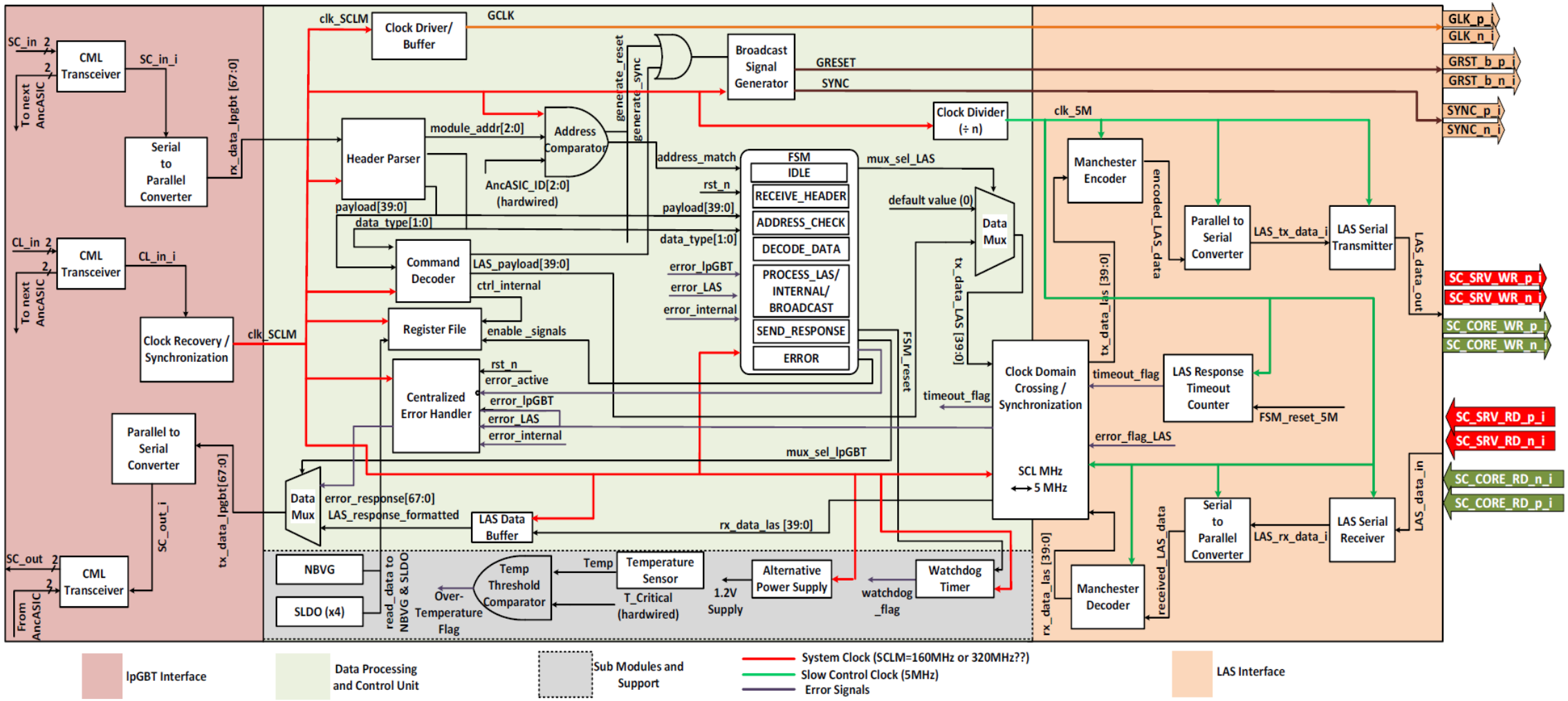
J. Schambach

Control Interface

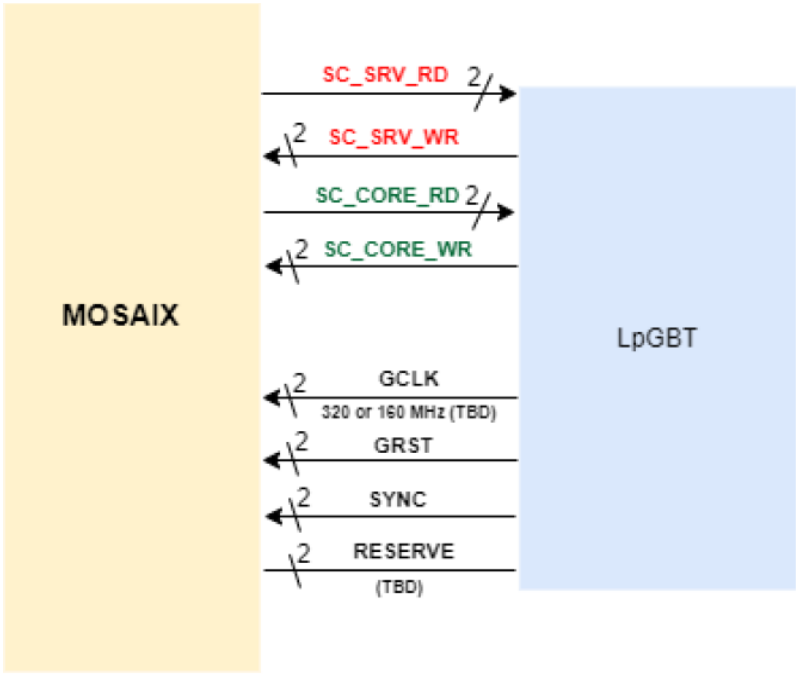
- **Services Management Interface:**
  - Reset management
  - Analog monitoring
  - Tile power switches and reset control
  - I/O configuration
  - Stitched backbone configuration
- **Core Management Interface:**
  - Readout control and configuration
  - Pixel matrix and bias configuration
  - High-speed channel operation
  - Tile and LEC logic control

Interface	Endpoint Flavor	# Instances	Broadcast	Description
Services	Services LEC	1	No	SBB and HSCH bandgap config
	Service Node	48	Yes	Tile reset, monitoring, power switch config
	Monitoring ADC	1	No	Analog monitoring ADC control
Core	Core LEC	1	No	LEC routing config
	Tile	144	Yes	Tile readout and matrix config
	High-Speed Channel	8	Yes	High-speed channel configuration

SVT AncBrain Block Diagram (**BNL** Instrumentation group)



MOSAIX Slow Controls Protocol



40-bit SC Packet

Field ID	Field Name	Width	Description
A	HDR	4	Fixed 4bit (4'hA) value that indicates a start of transaction
B	OPCODE	2	2bit field to indicate transaction type: b00 – WRITE_posted transaction, b01 – WRITE_non-posted transaction, b10/b11 – READ transaction
C	EP_ADDR	8	Endpoint Address
D	REG_ADDR	8	Register Address
E	REG_DATA	16	Register write or read data
F	PARITY	1	Transaction parity bit (bit-wise xor of RW, EP_ADDR, REG_ADDR and REG_DATA fields
G	STOP	1	Stop bit: 1'b0: Fixed 1bit (1'b0) value that indicates end of transaction

Slow Controls Physical Layer:

- CERN Low Power Signaling (CLPS)
- MSB first
- Manchester encoded
- 5 Mbps bit speed, 10 Mbaud physical

Transaction type	Direction	HDR	RW	ADDR	DATA	PARITY	STOP
WRITE_posted	Input to the ASIC	4'b1010	2'b00	Any valid register address	Data to write to the register	xor(HDR, RW,ADDR,DATA)	1'b0
WRITE_non-posted	Input to the ASIC	4'b1010	2'b01	Any valid register address	Data to write to the register	xor(HDR, RW,ADDR,DATA)	1'b0
WRITE-response	Output of the ASIC	4'b1010	2'b01	Register address as specified in the corresponding WRITE_non-posted transaction	Value of the register	xor(HDR, RW,ADDR,DATA)	1'b0
READ	Input to the ASIC	4'b1010	2'b10	Any valid register address	16'd0 (or 8'd0)	xor(HDR, RW,ADDR,DATA)	1'b0
READ-response	Output of the ASIC	4'b1010	2'b10	Register address as specified in the corresponding READ transaction	Value of the register	xor(HDR, RW,ADDR,DATA)	1'n0

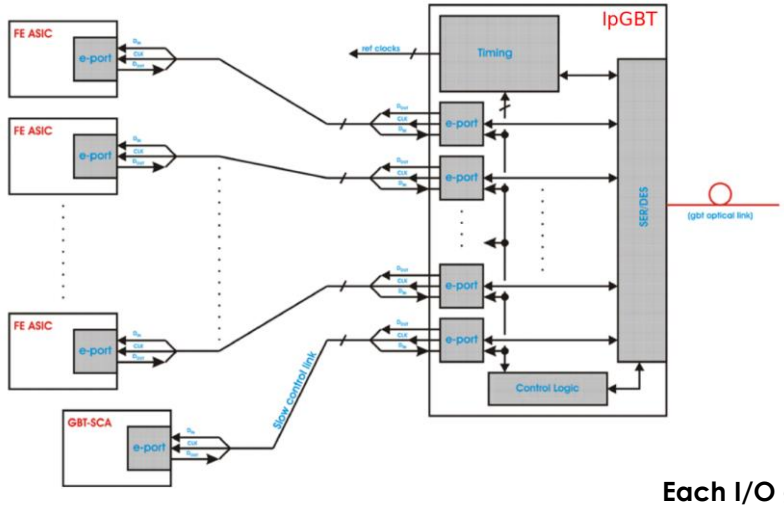
AncBrain Protocol

Field Name	Width (bits)	Description
Start of Frame (SOF)	6	Predefined bit pattern to signal the beginning of a transaction (b011101)
Frame Type ID (FTI)	2	Frame Type Identifier
Address (ADDR)	4	0x0=Reset BC, 0xF=Sync BC, 0x1-0xE=Module ID
Type	2	00=Internal, 01=LAS_Srv, 10=LAS_Core
Payload	40	Command or LAS Data Payload
CRC	8	CRC-8 (poly 0x07)
End of Frame (EOF)	6	End delimiter (b101110)

LAS Service or Core Data Format as in previous slide.
Internal:

Field Name	Width (bits)	Description
SUBMOD_ID	4	Submodule ID within ANCBrain
R/W	2	Transaction type: Write or Read
Register Address	8	Register address within the target sub-module
Control/Data	8	Sub-module control data or internal signaling
Parity	1	Transaction parity bit
Stop	1	Internal signaling (or reserved)
Unused	16	Reserved for future use

lpGBT Architecture

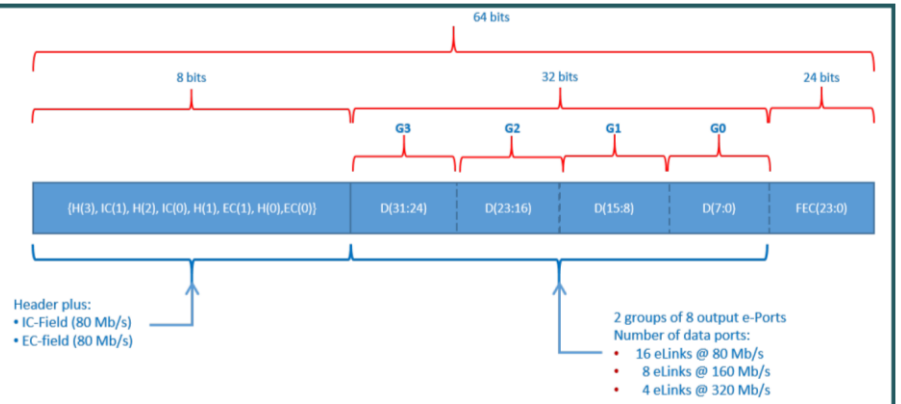


Each I/O group contains either 4, 2, or 1 elink(s)

Frame	Function	I/O Group
FRMUP[23:0]	FEC[23:0]	
FRMUP[39:24]	Data[15:0]	0
FRMUP[55:40]	Data[31:16]	1
FRMUP[71:56]	Data[47:32]	2
FRMUP[87:72]	Data[63:48]	3
FRMUP[103:88]	Data[79:64]	4
FRMUP[119:104]	Data[95:80]	5
FRMUP[121:120]	DownIC[1:0]	See text
FRMUP[123:122]	EC[1:0]	EC
FRMUP[125:124]	IC[1:0]	
FRMUP[127:126]	H[1:0]	HFH[1:0] = 2'b10

Uplink

96 out of 128 bits are data:
Payload = 3840 Mbps



Output eLinks (down-link)			
Bandwidth [Mb/s]	80	160	320
Maximum number	16	8	4

Downlink

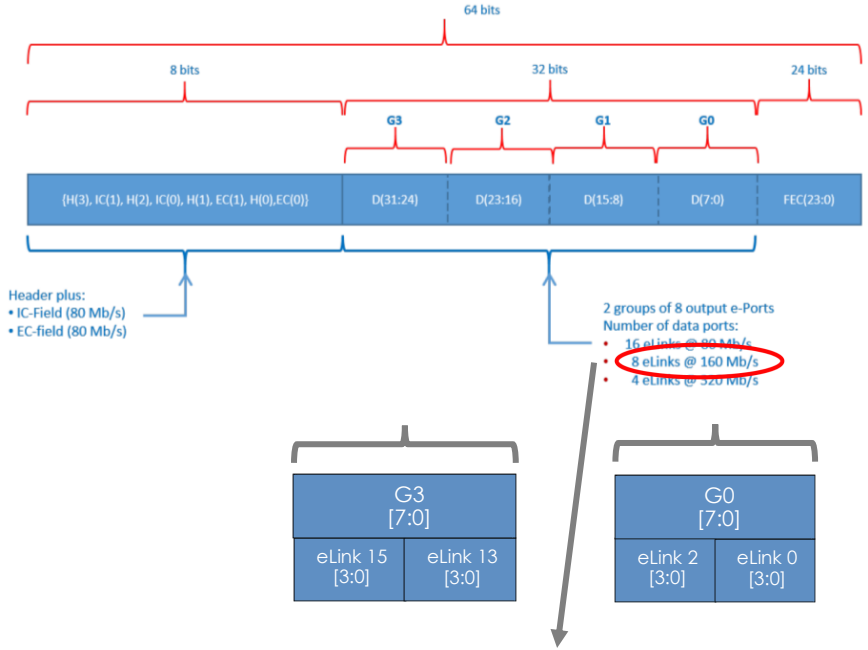
Line Rate: **2.56 Gbps**
32 out of 64 bits are data:
Payload = **1280 Mbps**

Input eLinks (up-link)												
Up-link bandwidth [Gb/s]	5.12						10.24					
FEC coding	FEC5			FEC12			FEC5			FEC12		
Bandwidth [Mb/s]	160	320	640	160	320	640	320	640	1280	320	640	1280
Maximum number	28	14	7	24	12	6	28	14	7	24	12	6

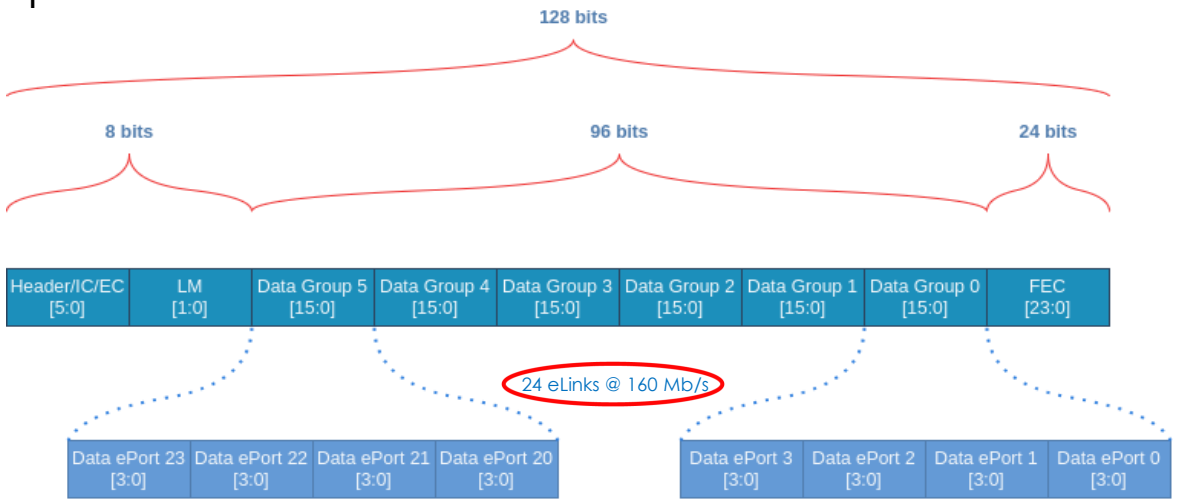
Field	5.12 Gbps FEC5	5.12 Gbps FEC12	10.24 Gbps FEC5	10.24 Gbps FEC12
Frame [bits]	128		256	
Header [bits]	2		2	
IC [bits]	2		2	
EC [bits]	2		2	
D [bits]	112	96	224	192
FEC [bits]	10	24	20	48
LM [bits]	0	2	6	10
Correction [bits]	5	12	10	24
# of eLink groups	7	6	7	6

lpGBT eLink Routing

Downlink 2.56Gbps



Uplink 5.12Gbps



Inner Barrel:
40 bits * 2 transitions * 16 (bit-stuffing) / 4 bits per frame = **320** frames to reconstruct a 40bit SC word

Outer Barrel & Disks:
17 frames to reconstruct a 68bit AncBrain word

Shadowing eLinks enabled for OB & Disks

DataRate[1:0]	DataIn[7:0]	DataOut			
		e-Link[3]	E-Link[2]	E-Link[1]	E-Link[0]
2'b10 (x4) 160Mbps	{chn2[7:4],chn0[3:0]}	-	DataIn[7:4]	-	DataIn[3:0]
2'b10 (x4) 160Mbps	{chn2[7:4],chn0[3:0]}	**DataIn[7:4]**	DataIn[7:4]	**DataIn[3:0]**	DataIn[3:0]

Fiber Aggregator Board Firmware for Slow Control

- Up to 20 lpGBT channels should be possible in one of the smaller US+ FPGAs based on resource usage in ATLAS FELIX firmware, a total of **~250** lpGBT links
- Use lpGBT firmware for SVT side, add firmware for Aurora protocol on FELIX side
- Aggregation Firmware (**uplink**):
  - Assemble lpGBT frames into 40bit SC words for each tile depending on data rate (LAS vs MOSAIX)
  - Only keep "data" tile words
  - Drop other SC words or handle locally (errors, dropped frames, empty frames, status, …)
  - Add geometric word to identify Segment and mux into TX channel towards FELIX
  - Add summary status for locally handled conditions
- Slow Controls Firmware (**downlink**):
  - Separate downlink data into individual lpGBT channels and interpret (pass-through or handle locally)
  - Add sync signal to delimit readout frames for IB
  - Add commands for readout frame generation on AncASIC for OB and disks
  - (others?) …..