ATLAS ITk Plans for Optical Links

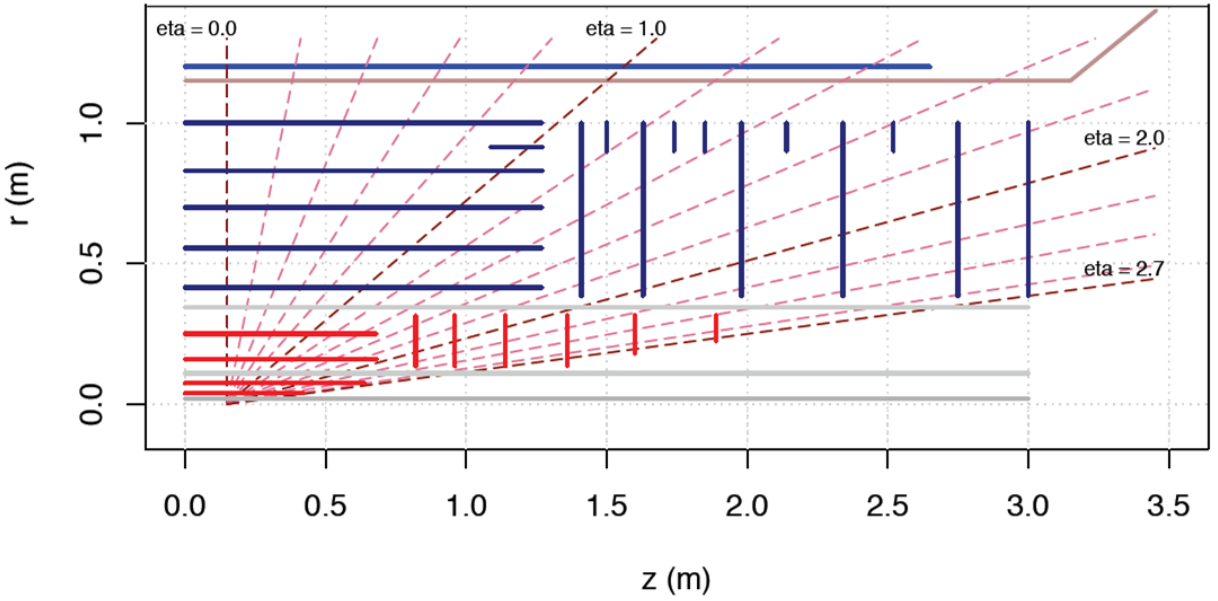
Special Diamond Jubilee edition

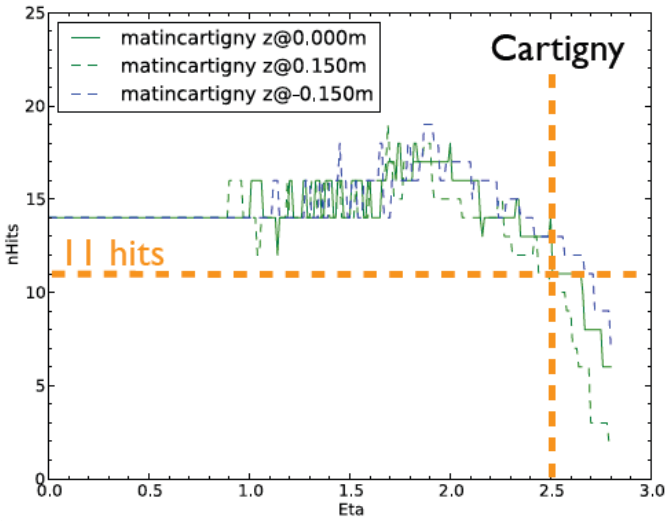
60 slides to commemorate 60 years of Her Majesty's Reign

Jubilee Outline

- Inner tracker (ITk) layout (June 2012)
  - Features differing from current detector
- Barrel Strip detector (Stave concept)
  - Forward Strip tracker (Petal concept)
- Pixel detector (stave concept)
- Summary and Issues

Phase II tracker - ‘Cartigny’ layout





- Strip system
  - Inner (short strip) barrel (3 layers)
  - Outer (long strip) barrel (2 layers)
  - ECs
  - Stubs at large r to increase BdL (this hopefully will be revised)
- Pixel
  - 2 inner pixel barrel layers (separately removable)
  - 2 outer pixel barrel layers
  - Disks

- Uniformly 14 hits up to η=2.5, pixel tracking up to up to η=2.7

Tracker elements

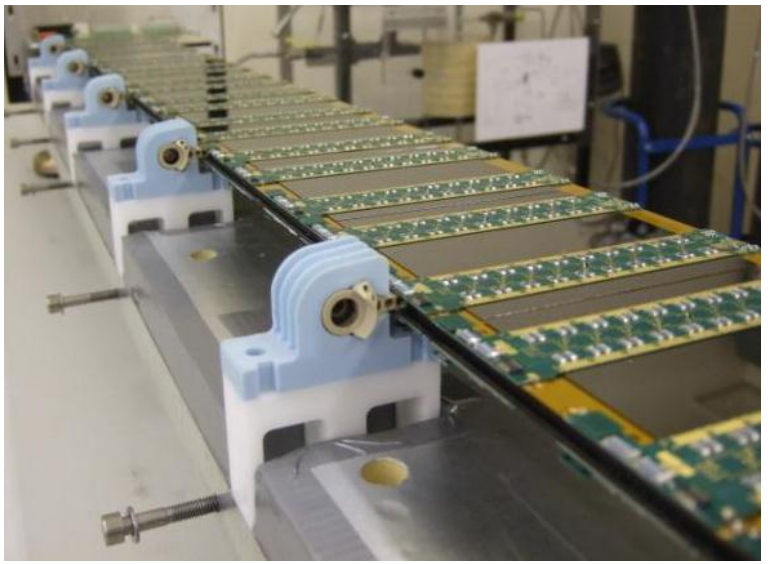
Concept:

To create integrated, fully functional objects, which can be

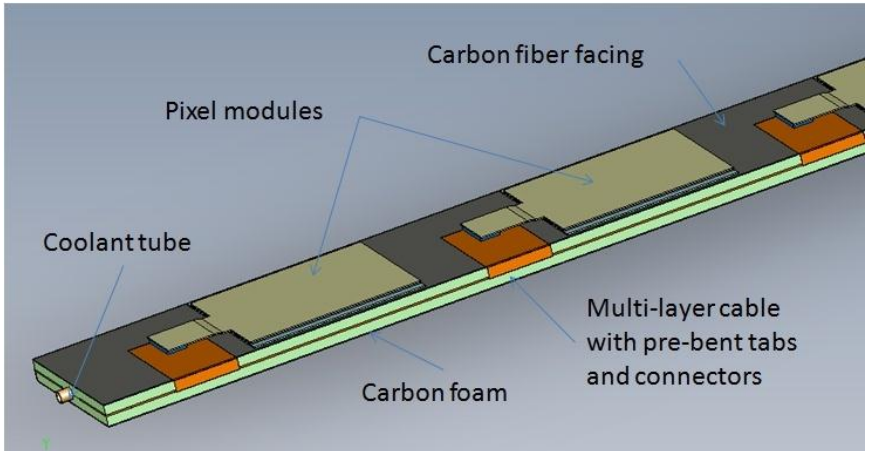
- Produced in parallel
- Tested fully early in the assembly
- Single staves are of limited value and loss of small number has small impact on project

→ Project robustness

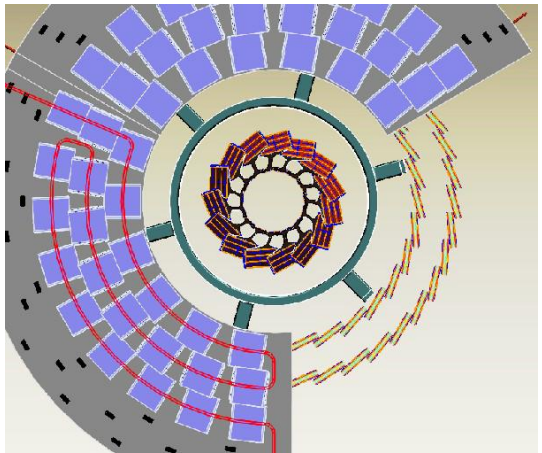
Barrel strip stave



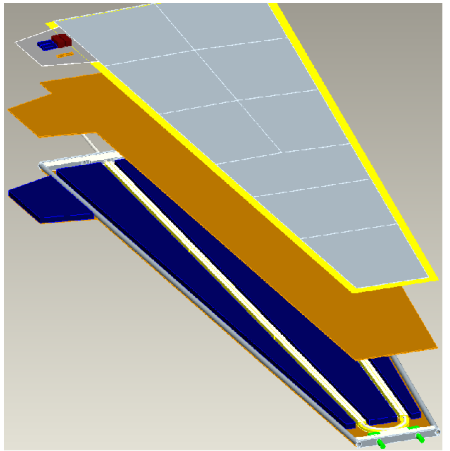
Outer pixel stave



Pixel disk

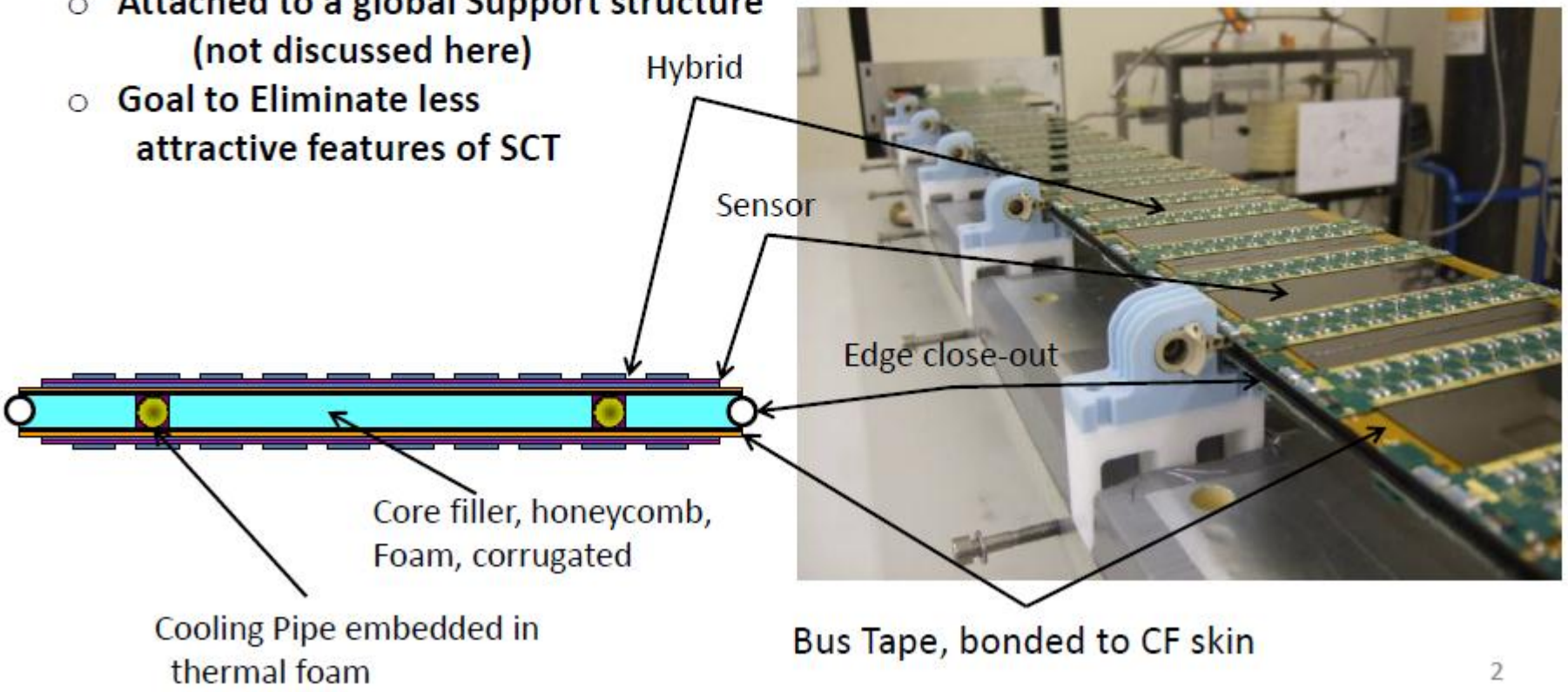


EC strip petal



Strip staves

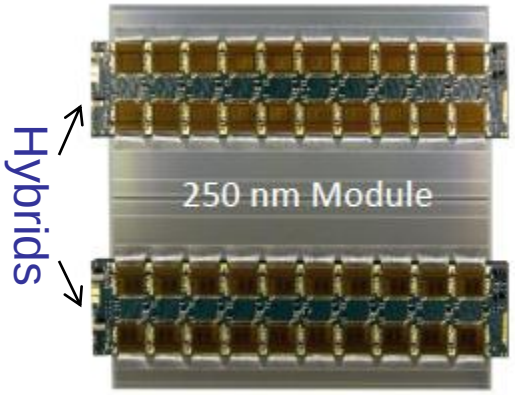
- Silicon Modules directly bonded to a cooled carbon fibre plate
  - Plate a sandwich construction for high structural rigidity with low mass
- Services integrated into plate, including power, control and data transmission.
  - Called a Stave in barrel region
  - and a Petal in the forward direction
- Attached to a global Support structure (not discussed here)
- Goal to Eliminate less attractive features of SCT



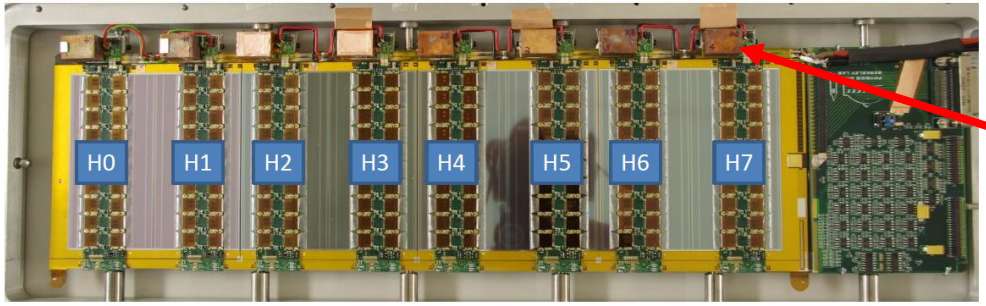
Strips electronics & readout

(prototypes – close packed text: G. Viehhauser)

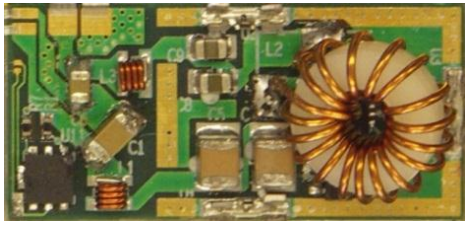
- **Sensors**: n-in-p single sided design, 98 x 98mm^2, 500V Max
- **Hybrids:** glued onto sensor
- **ASICs:** a 130 nm CMOS chipset
  - ABCn130: binary readout architecture (like SCT) but new protocol, 256 inputs for smaller hybrids, ROI and fast L1 trigger block
  - HCC: interface and module controller (1 per hybrid)
- **LV Powering:** either serial (SP) or DC-DC at each hybrid/module
  - Additional powering and protection chipset, prototyped and new versions in development
- Readout is being tested using stavelets (goal: good noise performance)



Example:
DC-DC powered stavelet
4 modules
8 hybrids
160 ABCn
20k channels



DC-DC converter board



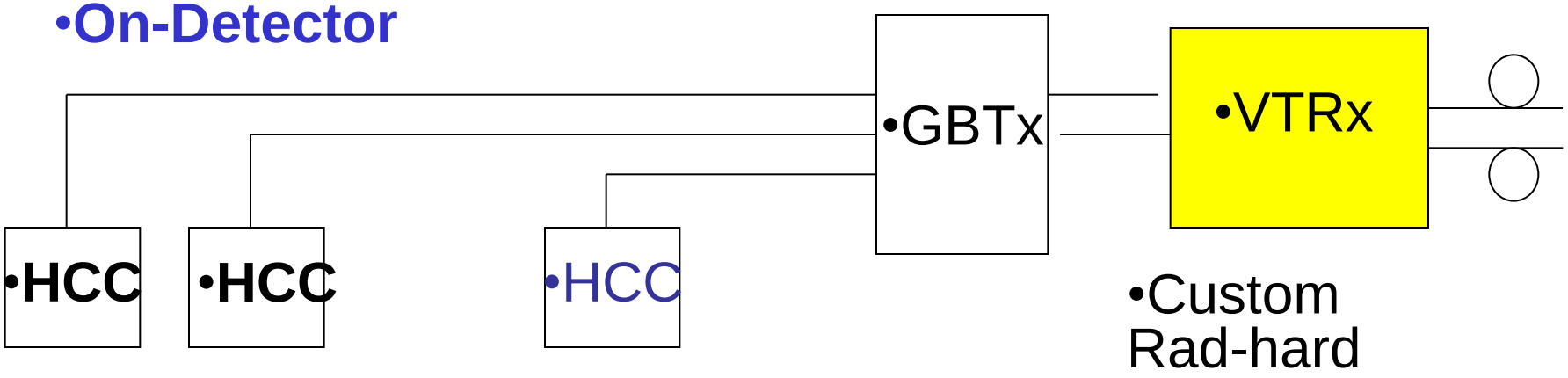
Double Trigger Noise Occupancies

	H0		H1		H2		H3		H4		H5		H6		H7	
Column	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1.0fC	0	0	0	0	DTN clean at 0.75fC, and counts at 0.5fC all <40 ☺								0	0	0	
0.75fC	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.5fC	0	1	6	36	18	5	12	38	12	2	4	9	0	0	0	4

ATLAS Strip Read-Out

(Barrel and Forward)

•**On-Detector**



•Off-Detector: COTS

•**Optical engines:**

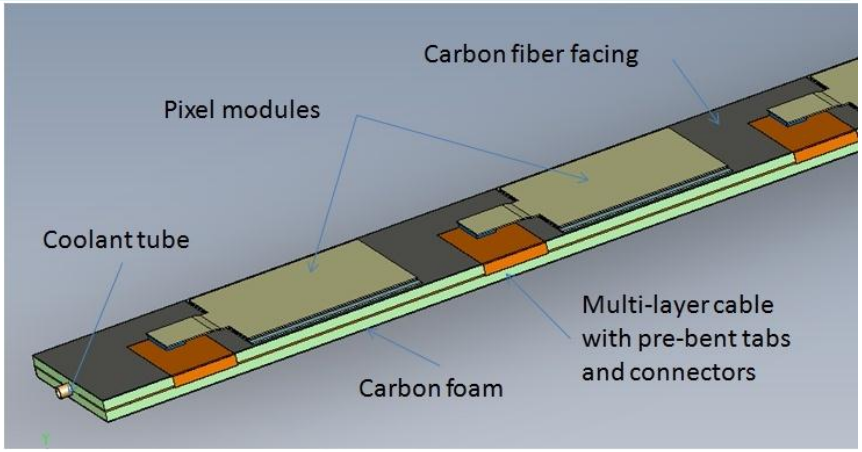
•**TX: Laser driver + laser arrays**

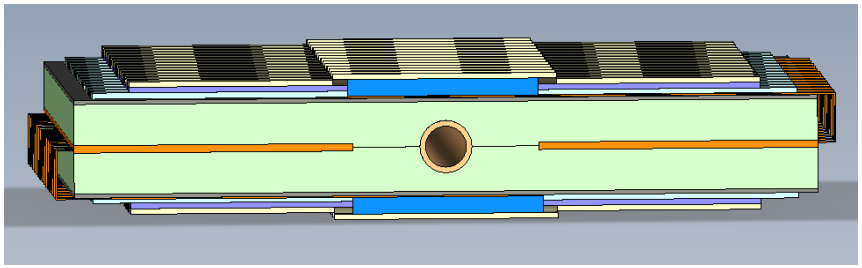
•**RX: p-i-n array + TIA/discriminator**

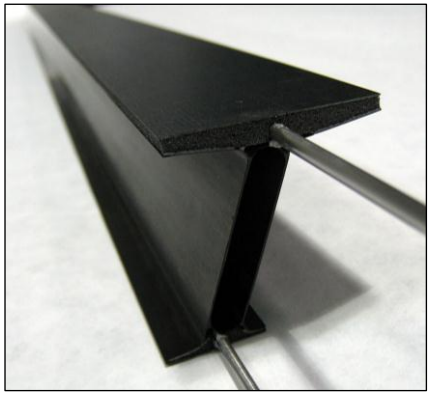
•**GBTx functionality in FPGA**

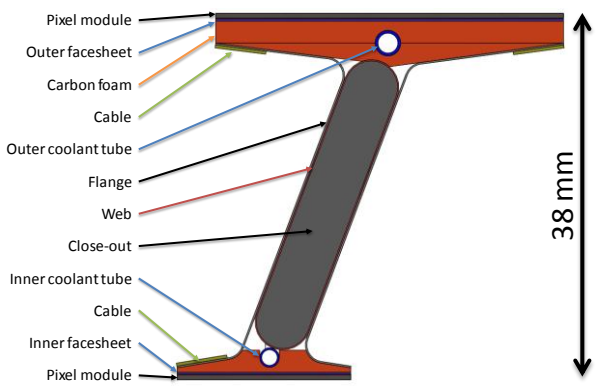
Pixel staves

- Outer pixel layers
  - About 1.4m long and 5mm thick
  - Modules on both sides, overlap for full coverage, makes module mounting easier
  - Embedded Services following a similar philosophy as strip concept.
- Inner pixel layers
  - I-beam design linking neighbouring layers
  - optimizes stiffness



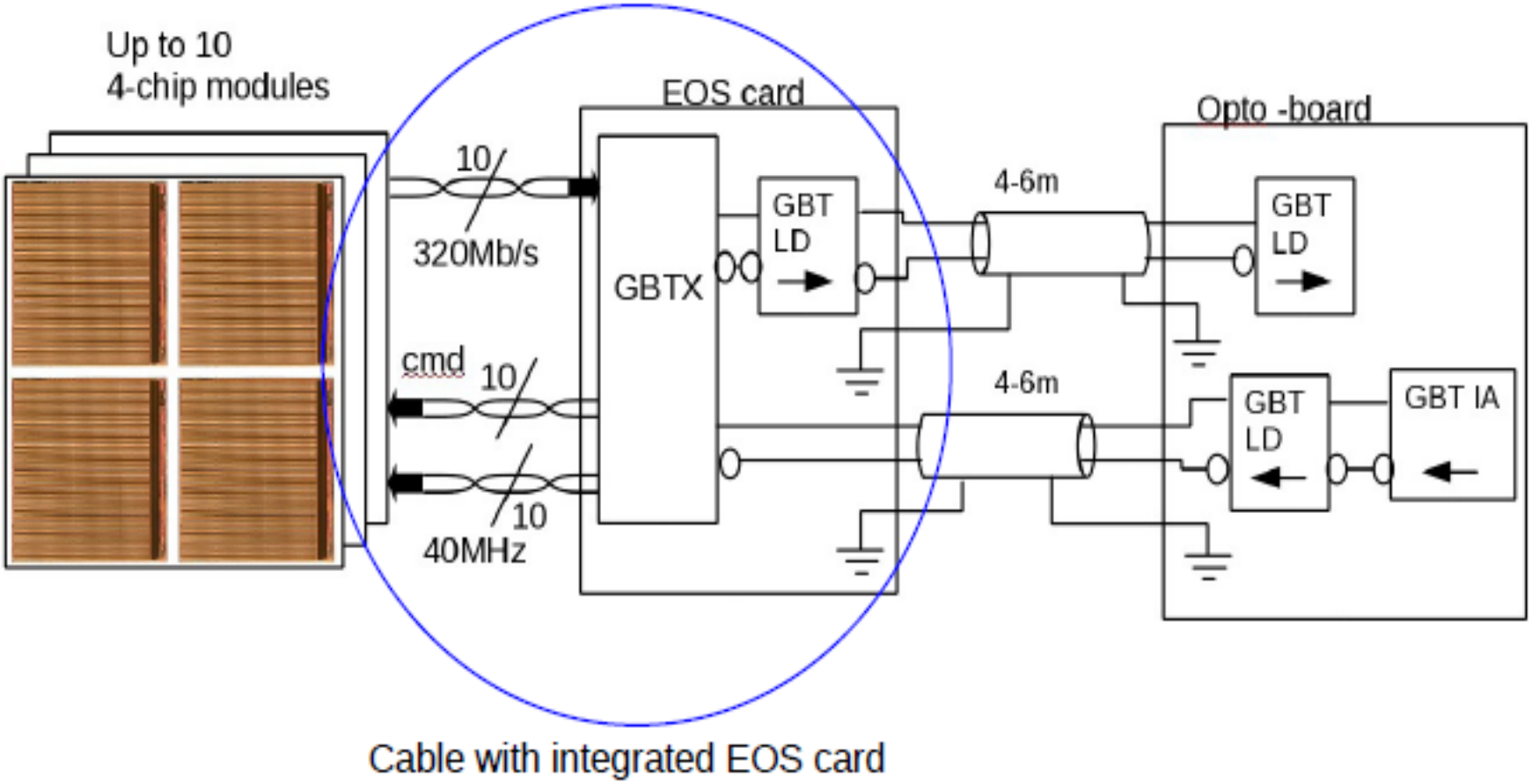






Readout Differences for Pixel

- **Need to move O/E to larger radius because of radiation damage ➔ high**



GBT, Versatile Link

(Back to Staves)

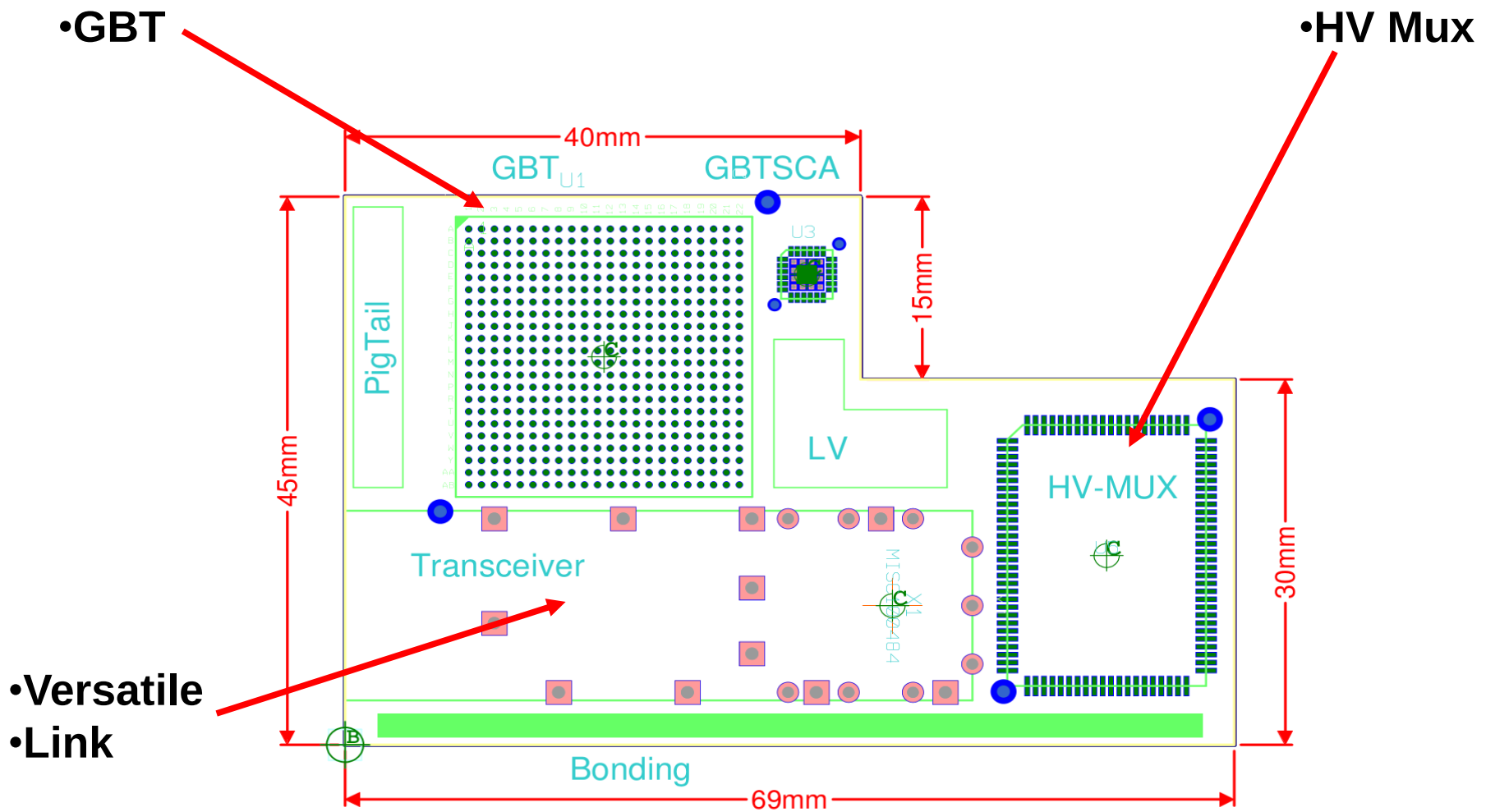
- EoS= End-of-substructure Card
  - Connects a **Stave** (Barrel) or **Petal** (Endcap) with the outside world
  - Hosts GBT, GBTSCA and Versatile Link
- EoS
  - Multi-layer PCB
  - Designed for Robustness (single point of failure)
  - Lots of components
  - X_0 ~ 4-5% (Current Estimate)

Connectivity

- Recommend to route everything through EoS
- Required
  - LV for Stave
  - LV for EoS
  - HV
  - Interlocks (2 lines)
  - Fiber link
- Alternatives for Connectivity
  - Pigtail (Preferred)
  - Connector (“Samtec”) on PCB

•Common Connector

Default Floor plan



This all Looks nice; but is our thinking joined up?

- Fixed size packets for data transfer
- Packets contain only ~50% payload
  - Optimised for 1 packet per event
- L1 data is split over as many packets as needed
- R3 data only uses 1 packet per event
- 3 packet types for data:

  L1_1BC – normal data taking – fits 3 clusters (max width 4)

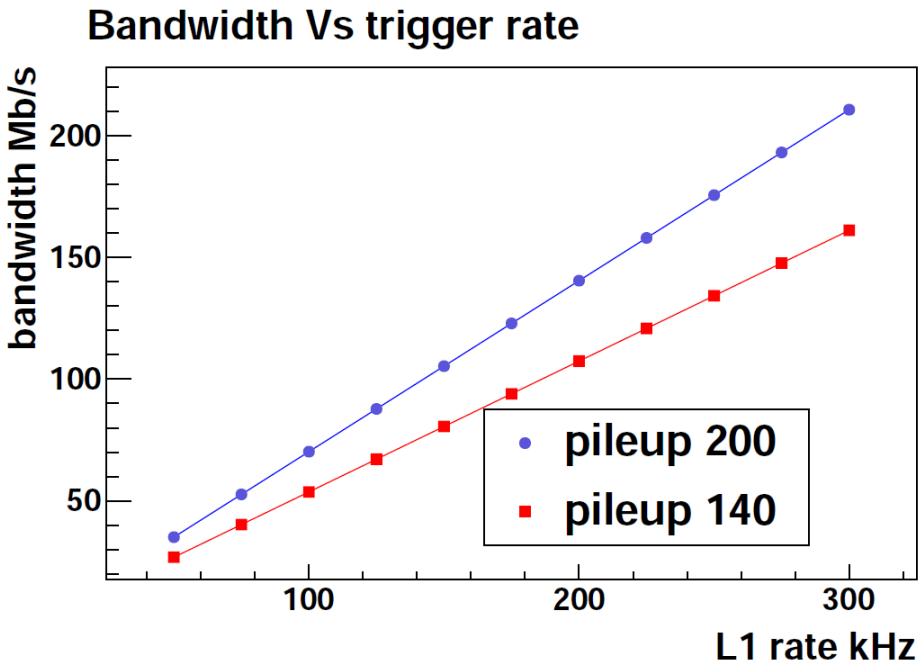
  L1_3BC – timing-in data, incl. hits from prior and next BC

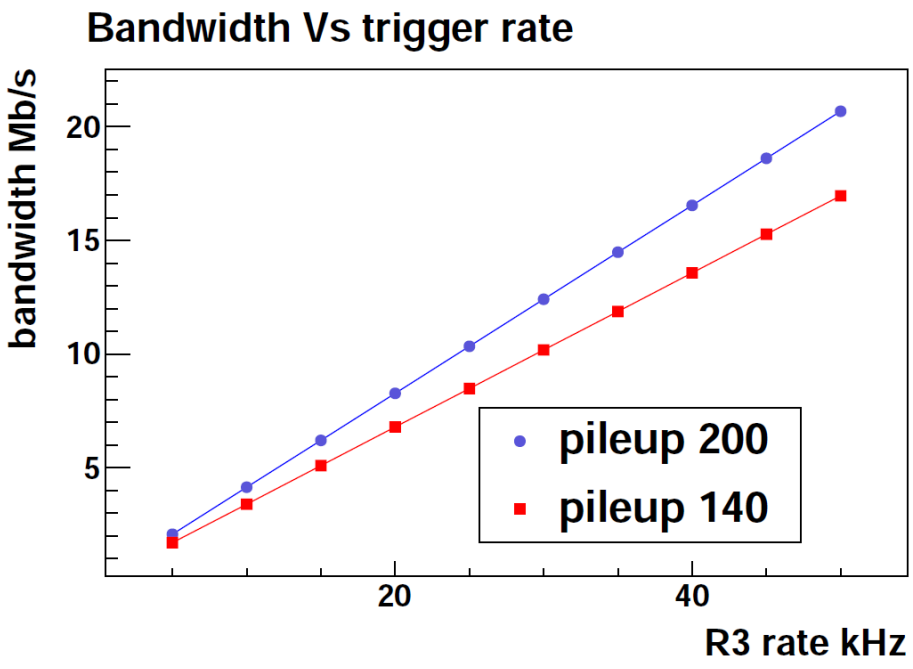
  R3 – reduced for latency – only 4 clusters reported, no widths

Overall Data Volume

- With packets 59 bit long, BW is simply:
- Trigger rate * packets/chip/event * 59 bits/pkt * 10 chips

- R3-Data BW





- @ 200kHz L1A, 25kHz R3 (5% of 500kHz L0A) :
- 200 pileup: BW=140+10= 150Mb/s
- 140 pileup: BW=107+8= 115Mb/s

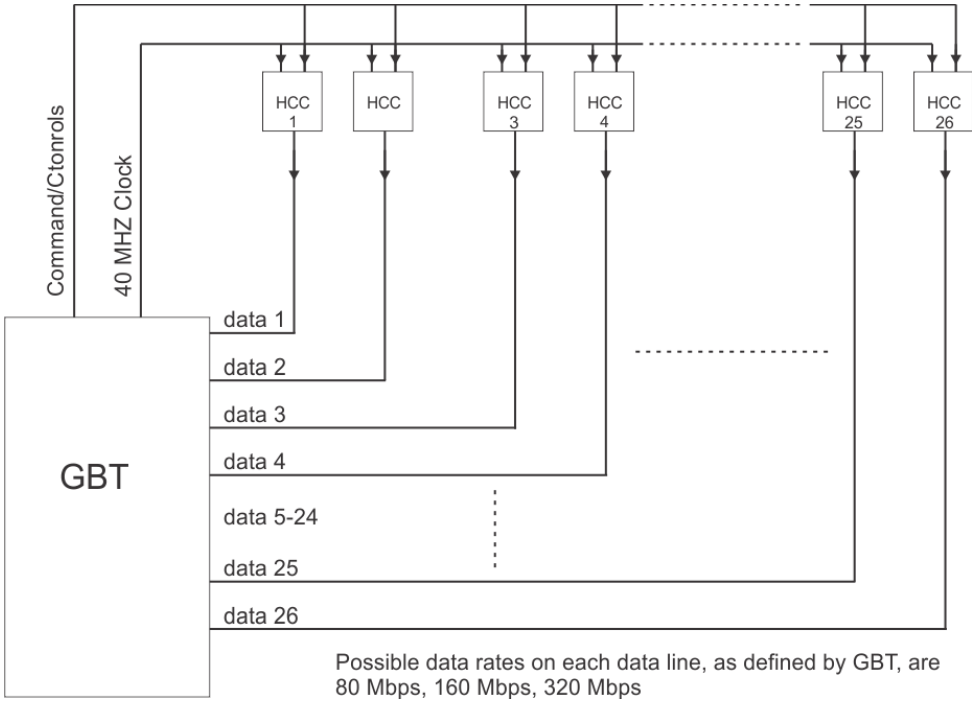
GBT Issues

- With 160 Mbps e-links, there are 20 inputs (FEC), BUT short strips need max. 26 inputs
- Inner short strip needs (160 Mbit e-links):

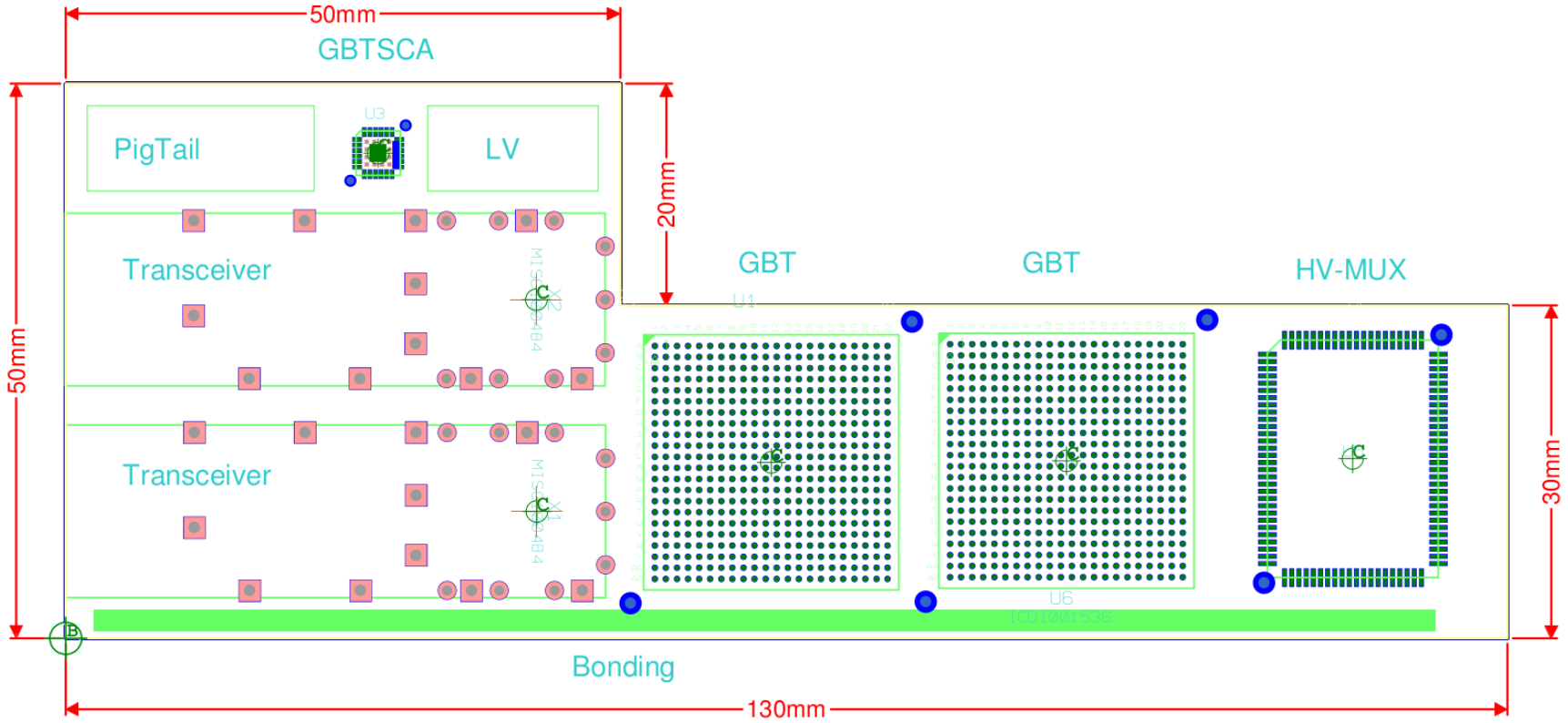
  A: Two GBTs

  B. One GBT running in wideframe mode at 160 (28 inputs)
    - No Error correction

  C. A new-GBT?



Double-Link


50mm
GBTSCA
PigTail
LV
20mm
Transceiver
Transceiver
GBT
GBT
HV-MUX
50mm
30mm
Bonding
130mm

More GBT Issues?

- Reliability
  - For this concept, it is single point of failure
  - How to make it as robust as possible
  - Self-testing and diagnosis needed
- Yield
  - BGA may be challenging
  - Can we check all balls made a good connection ?
  - How can we re-work a EoS card if needed
- Control
  - GBT needs to be in sane state upon powering, because all communication is through GBT

Strips Summary

- Reading out a strips barrel with:
  - 200 Pileup
  - 500kHz L0A, 6.4μs latency
  - 200KHz L1A, 20μs latency
  - 24 (26) 160Mb GBT e-links

IS POSSIBLE, BUT with almost 0% margin!

(because “upgrade” always means higher trigger rate)

- Deadtime/queuing losses need further study
  - BUT unlikely to be an issue
- GBT operates with no ECC, and at the limit
  - more BW would be better

Versatile Link issues (ITk perspective)

- Connecting it to the EoS…
  - Make it as robust as possible
- Monitoring
  - Can we check Versatile Link performance like light levels ….
  - May help to spot developing problems
- General
  - Having prototypes would help a lot

The End