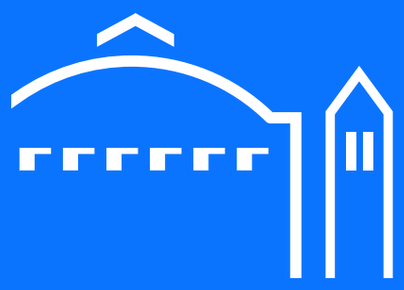
28th Feb
2024

# MLHE studies

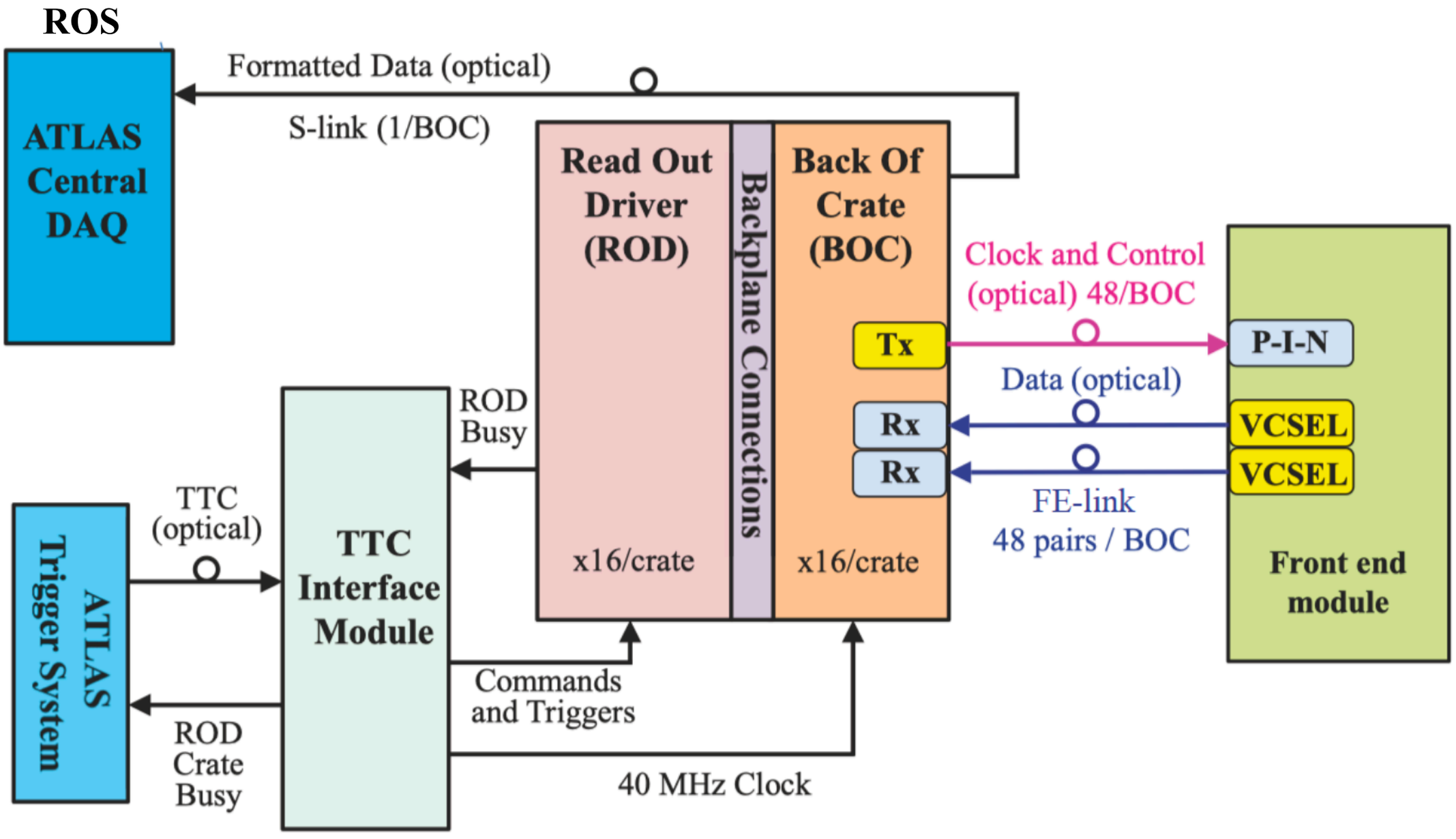
## SCT ROD Missing Link Header Error update

**Rebecca Carney**, John Joseph, *Bruce Gallop, Dave Robinson*, Koichi Nagai,
Daiya Akiyama, Sahal Yacoob, Elisabeth Schopf, Alessandro Guida, and more



BERKELEY LAB

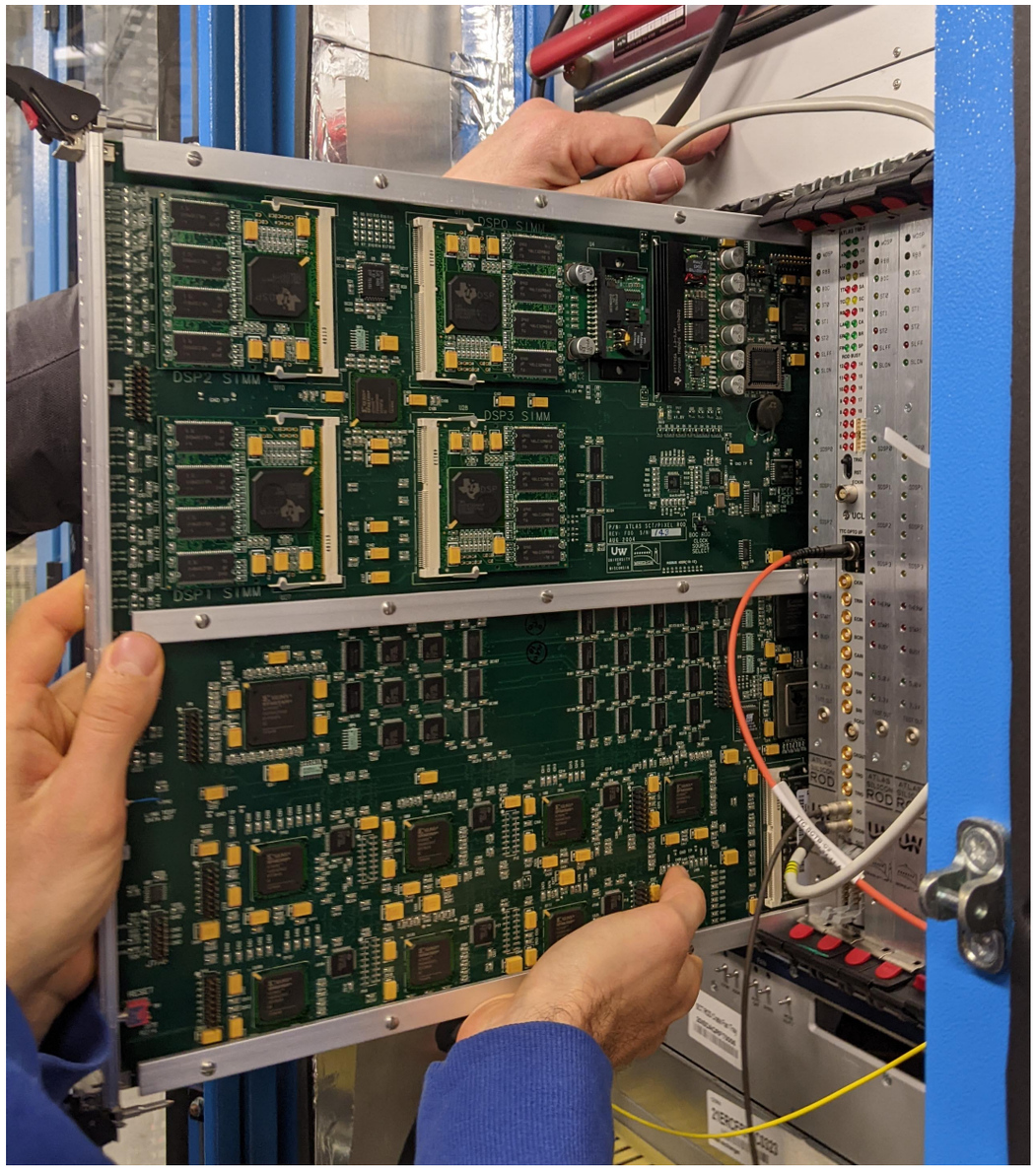
rcarney@lbl.gov



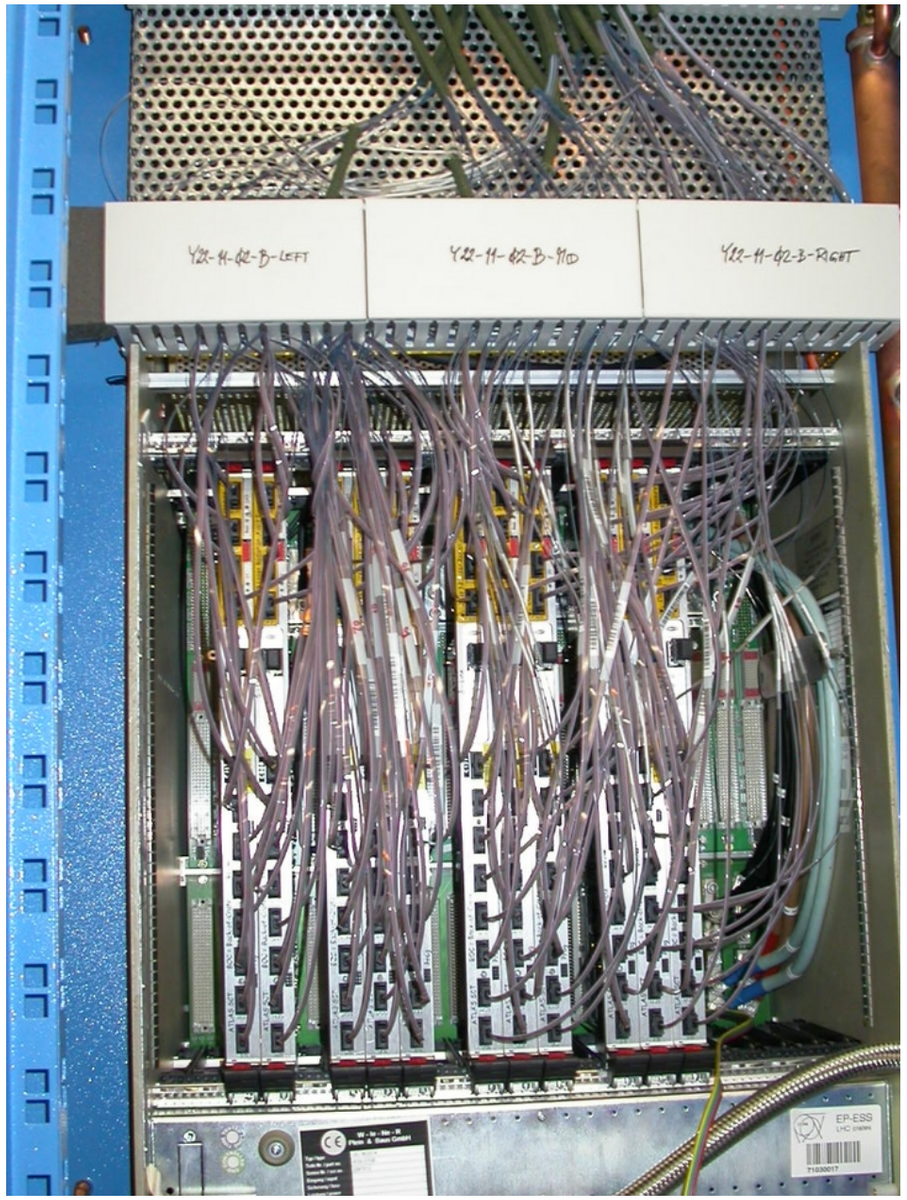

- Read-Out Driver (ROD), Timing Interface Module (TIM), Front-end (FE), Back Of Crate (BOC)
- ROD receives commands and triggers from TIM, sent to FE-links via BOC
- Upon L1, ROD reads out (via BOC) all FE-links with corresponding BCID.
- ROD then performs a variety of operations on data: adding trigger/link info, error detection, repacking into an “event fragment” and sending downstream to ROS (via BOC).



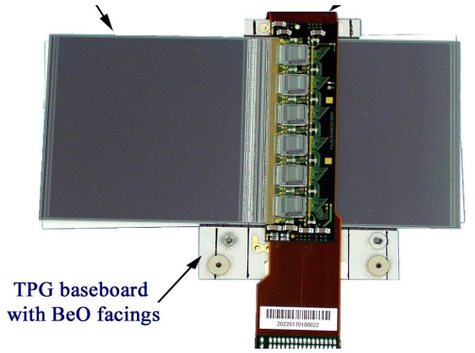
rcarney@lbl.gov

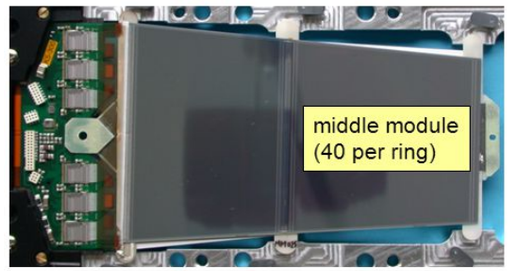


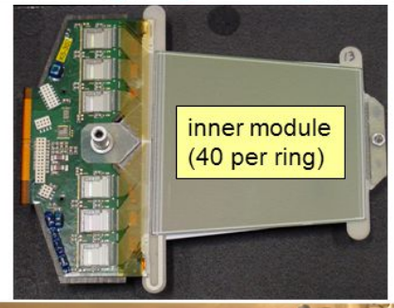

*ROD (in hand), with TIM visible (orange fibre connection)*

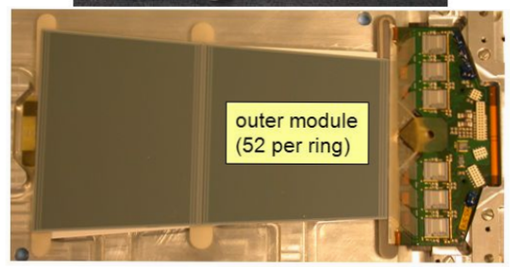



*BOCs with fibre connections*




*SCT Barrel and EC modules, ABCD3T FE visible as 6 squares on hybrid*

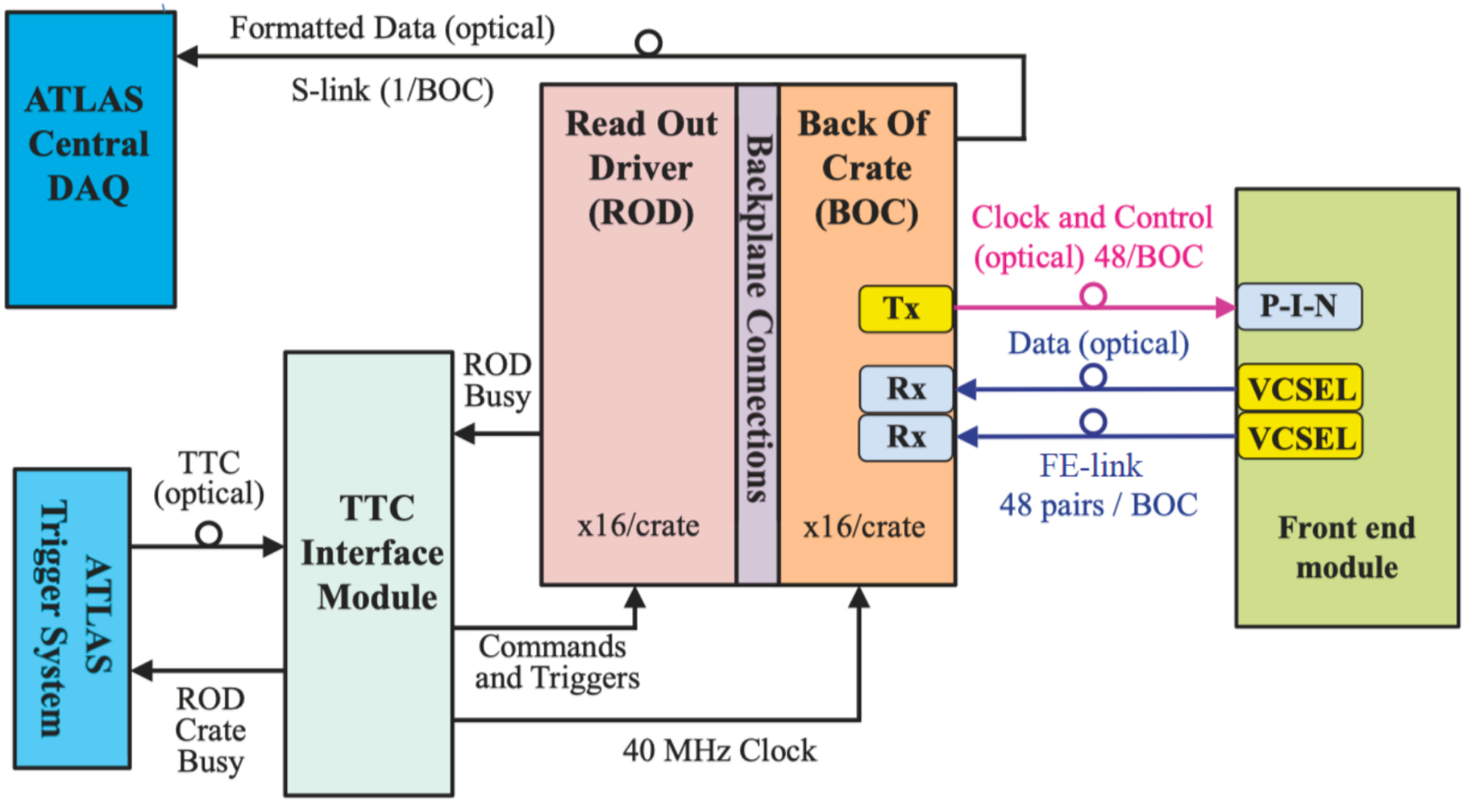
- Here's what some of these components look like in actuality (featuring SCT Run Coordinator deputy Alessandro Guida's hands on the left).



rcarney@lbl.gov

# **MLHE** what is it?

- **What comes out of the ROD?**
  - ROS receives a formatted **event fragment** from ROD (contains L1ID, detector type, trigger type) - high-level info about the event.
  - *Within* fragment header and trailer, there are 32-bit words with the following format: **Header —> [Data x *n*]—> Trailer**






rcarney@lbl.gov

# **MLHE** what is it?

- **What comes out of the ROD?**
  - Header words: contain L1ID, BCID info
  - ‘Data’ words: different formats/compression (condensed, expanded, super-condensed (used in Run 3)).
  - Data words contain FE ID info, cluster, and error info.
  - Trailer words contain error info.

| Name | Bits [15:0] or [31:16] |
|---|---|
| Header | 001pLLLLBBBBBBBB |
| Trailer | 010zhvxxxxxxxxxx |
| 1 hit condensed | 1FFFFCCCCCCCxfx0 |
| 2 hits condensed | 1FFFFCCCCCCCsfx1 |
| 1st hit cluster expanded | 1FFFFCCCCCCC0DDD |
| 1 hit cluster expanded | 1xxxxxxx0xxx1DDD |
| 2 hit cluster expanded | 1xxxxxxx1DDD1DDD |
| Flagged error | 000xxxxxxFFFFEEE |
| Raw data | 011nnnxxWWWWWWWW |

Key:

B = BCID
C = cluster base address
D = 3 bit hit data
E = ABC error code
F = FE number
f = error in condensed mode data, 1st hit
h = header trailer limit error
L = L1ID
n = count of raw data bits + 1
p = preamble error
s = error in condensed mode data, 2nd hit
v = data overflow error
W = raw data
x = Don’t care (ROD fills these with 0’s)
z = trailer bit error



rcarney@lbl.gov

# **MLHE** what is it?

- **Missing Link-Header Error:**
  - ROS receives a formatted event fragment from ROD
  - Data within fragment has the following format (much simplified, see later): Header —> Data —> Trailer
  - Data word has: link number + cluster info + error info
  - MLHE occurs when h->d->t order not obeyed and/or the link numbers are out of order. More details later.
- **Maybe misplaced link data/header error is a more accurate name..**

**Table 16: SCT Formatter Output, Bits [31:0]**

| Name | Bits [15:0] or [31:16] |
|---|---|
| Header | 001pLLLLBBBBBBBB |
| Trailer | 010zhvxxxxxxxxxx |
| 1 hit condensed | 1FFFFCCCCCCCxfx0 |
| 2 hits condensed | 1FFFFCCCCCCCsfx1 |
| 1st hit cluster expanded | 1FFFFCCCCCCC0DDD |
| 1 hit cluster expanded | 1xxxxxxx0xxx1DDD |
| 2 hit cluster expanded | 1xxxxxxx1DDD1DDD |
| Flagged error | 000xxxxxxFFFFEEE |
| Raw data | 011nnnxxWWWWWWWW |

- **Why does this matter?**
- Hit data being disconnected from the header means that you can no longer be sure which bc it came from, so that cluster data is essentially lost/incomplete.
- Could have knock-on effects in track reco

```
Key:

  B = BCID                                          n = count of raw data bits + 1
  C = cluster base address                          p = preamble error
  D = 3 bit hit data                                s = error in condensed mode data, 2nd hit
  E = ABC error code                                v = data overflow error
  F = FE number                                     W = raw data
  f = error in condensed mode data, 1st hit         x = Don't care (ROD fills these with 0's)
  h = header trailer limit error                    z = trailer bit error
  L = L1ID
```

https://twiki.cern.ch/twiki/bin/view/Main/AtlasSiliconRodGroup , see ROD Manual (it's a bit out of date but gives a solid overview)



rcarney@lbl.gov

28th Feb
2024

# Observations

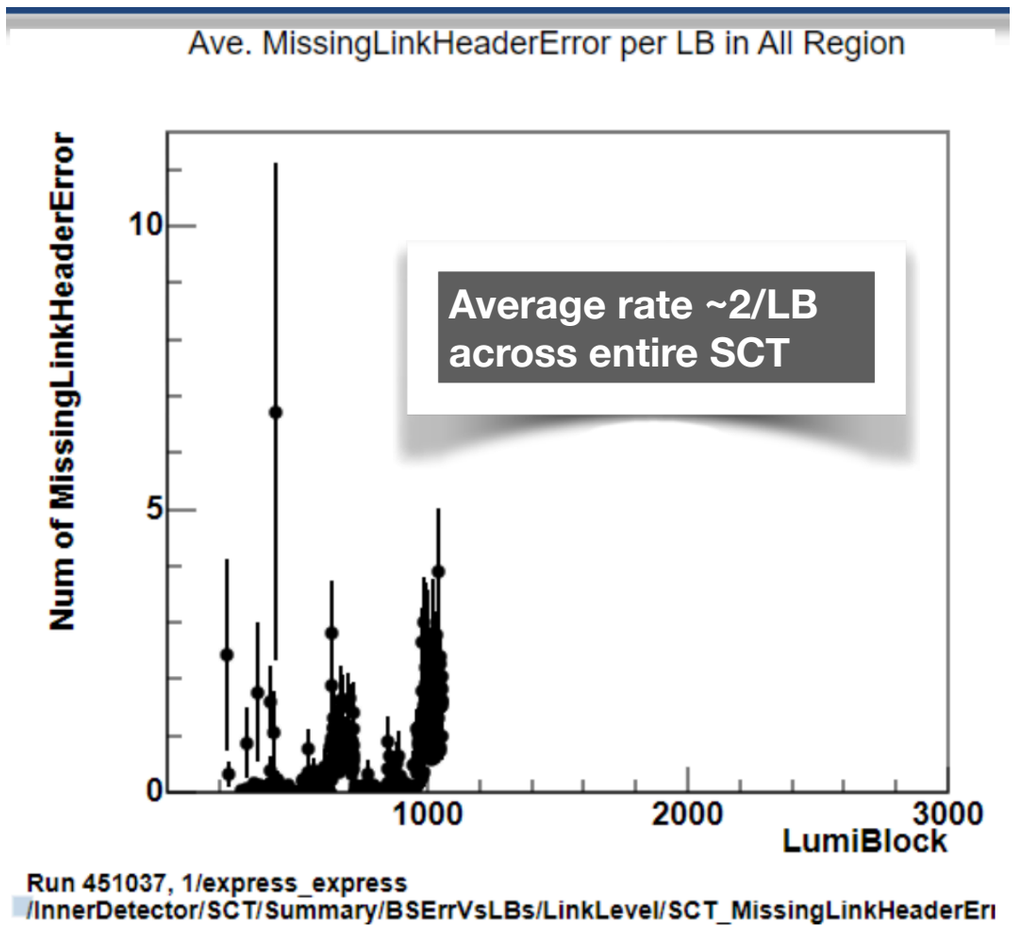
BERKELEY LAB

rcarney@lbl.gov

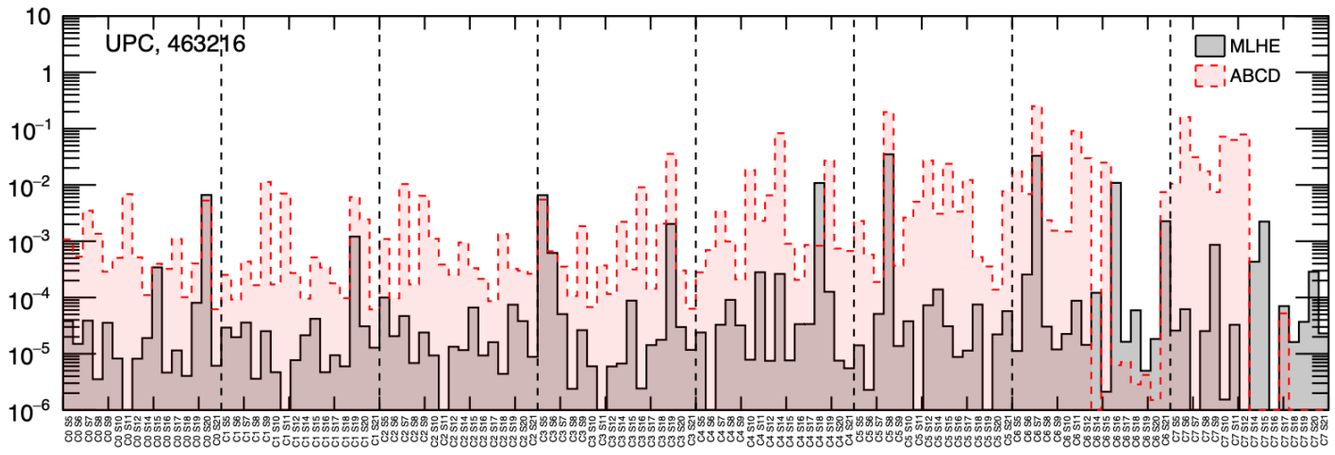
# **Some history** likely incomplete

- Please check out these great operations/DQ studies from **Shigeki Hirose** & **Daiya Akiyama** from last year, some of the results I summarize here.

https://indico.cern.ch/event/1284673/contributions/5398059/attachments/2644603/4577450/SCTWeekly_20230510.pdf
https://indico.cern.ch/event/1350440/contributions/5685219/attachments/2768261/4822473/MLHE_rod_swap_20231207.pdf




## N links w/ MLHE / Events




- Summed up the histogram entries in each slot
- Almost all slots have MLHE, and some slots have a significantly high frequency

**An issue across all links, no obvious correlation with ABCD3T errors.**
**NB on a run-by-run basis some links have more MLHE than others but tends to even out over time.**



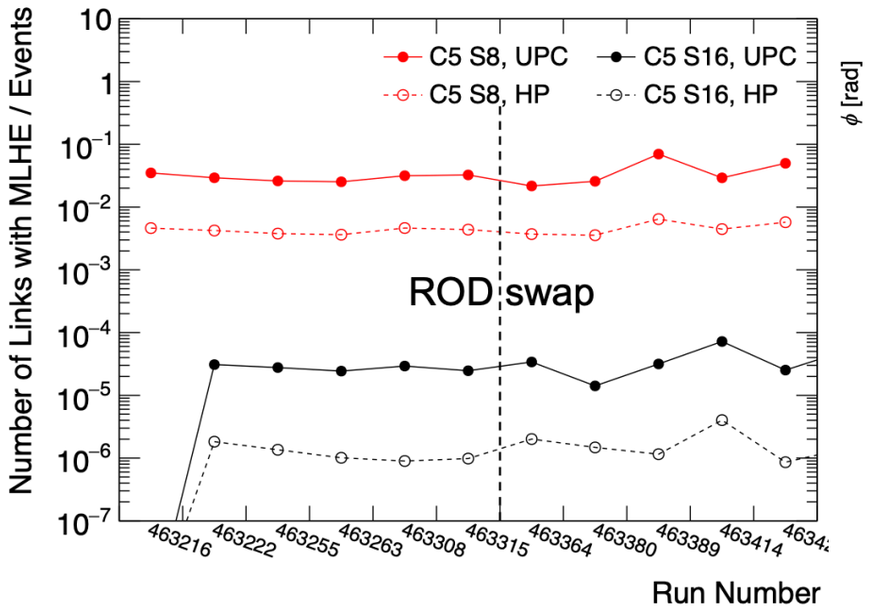
rcarney@lbl.gov

# **Some history** likely incomplete

- Please check out these great operations/DQ studies from **Shigeki Hirose** & **Daiya Akiyama** from last year, some of the results I summarize here.

https://indico.cern.ch/event/1284673/contributions/5398059/attachments/2644603/4577450/SCTWeekly_20230510.pdf
https://indico.cern.ch/event/1350440/contributions/5685219/attachments/2768261/4822473/MLHE_rod_swap_20231207.pdf

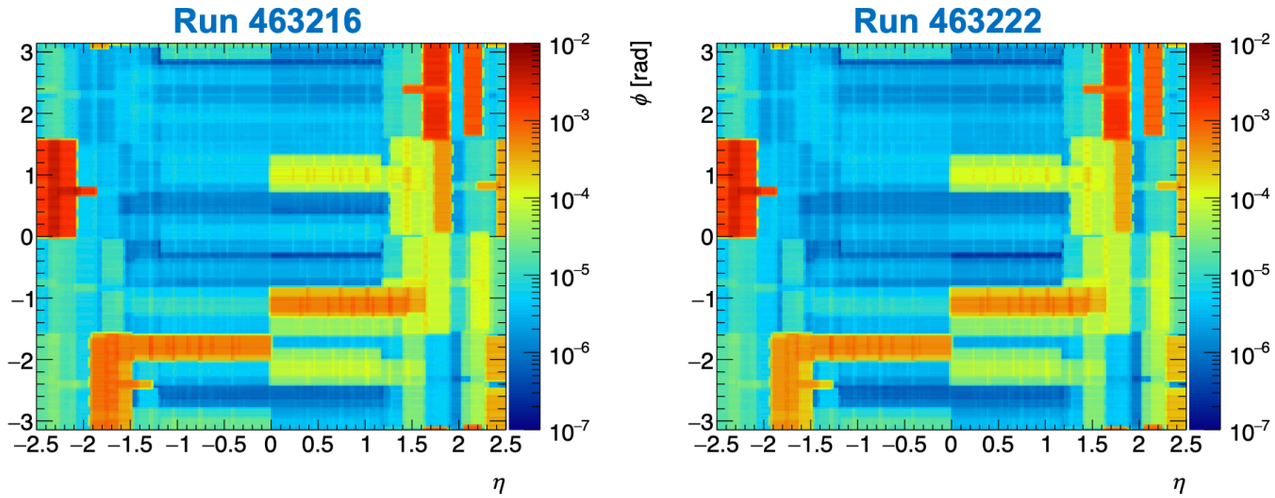
## ROD swap




- https://atlasop.cern.ch/elisa/display/528704
- The number of errors did not change in each slot, the problem is not in ROD

**This is not a ROD hardware issue: rate is matched to links, physically swapping out cards doesn't change rate.**

## Eta-phi distribution




- Many errors in the A-side

**There is some A-side dependency, seen in the last heavy-ion runs. Unclear why.**



rcarney@lbl.gov

# **Some** conclusions

- This is an infrequent but consistent issue.
- It is not caused by problematic ROD hardware, but perhaps by how ROD firmware is handling problematic module data or how the DAQ stream is working at high data-rate.
- There is some A-side dependency but no problematic modules.

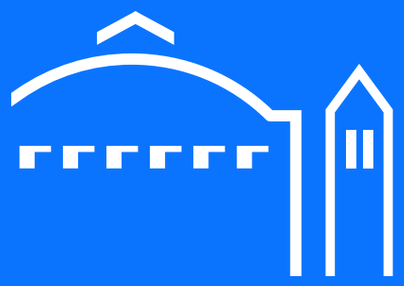
To proceed,
a little more info about the ROD…



rcarney@lbl.gov

28th Feb
2024

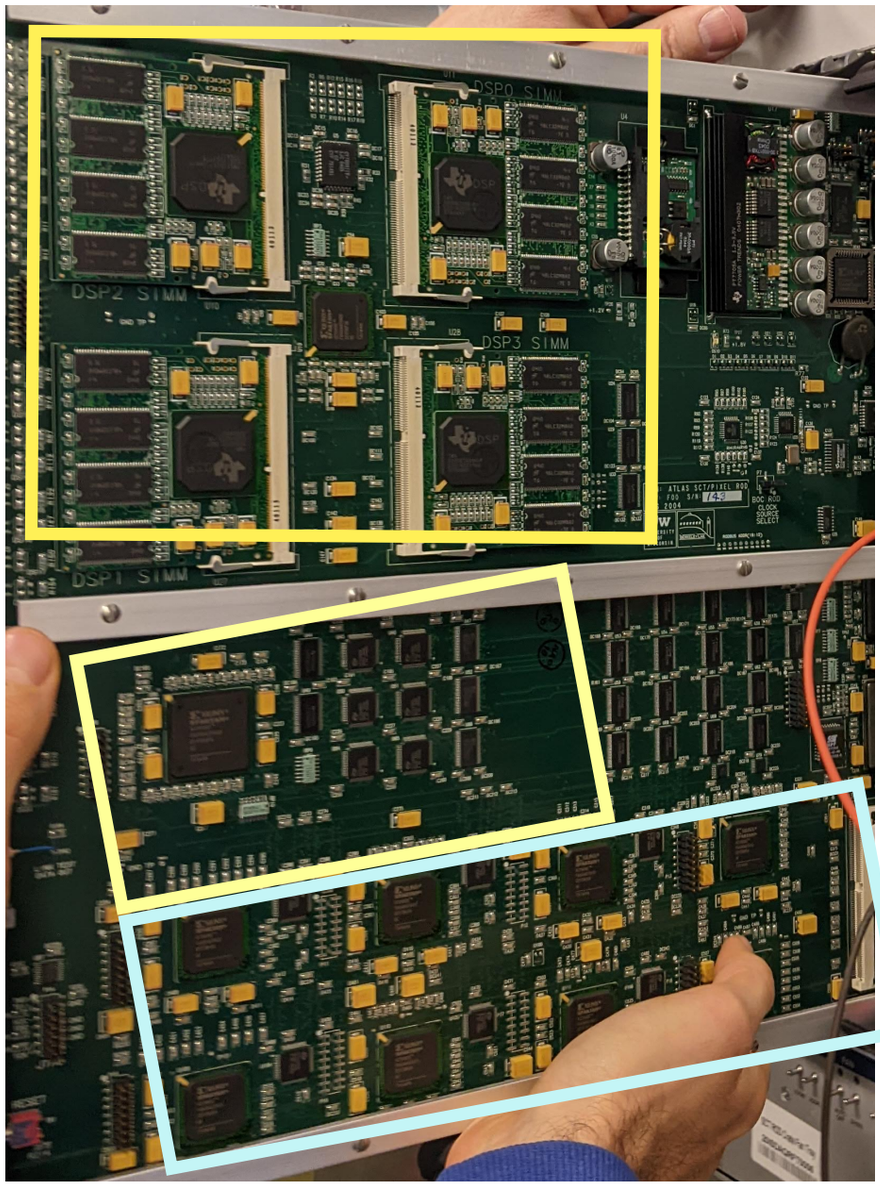
# ROD data path



BERKELEY LAB

rcarney@lbl.gov

# **ROD** data path



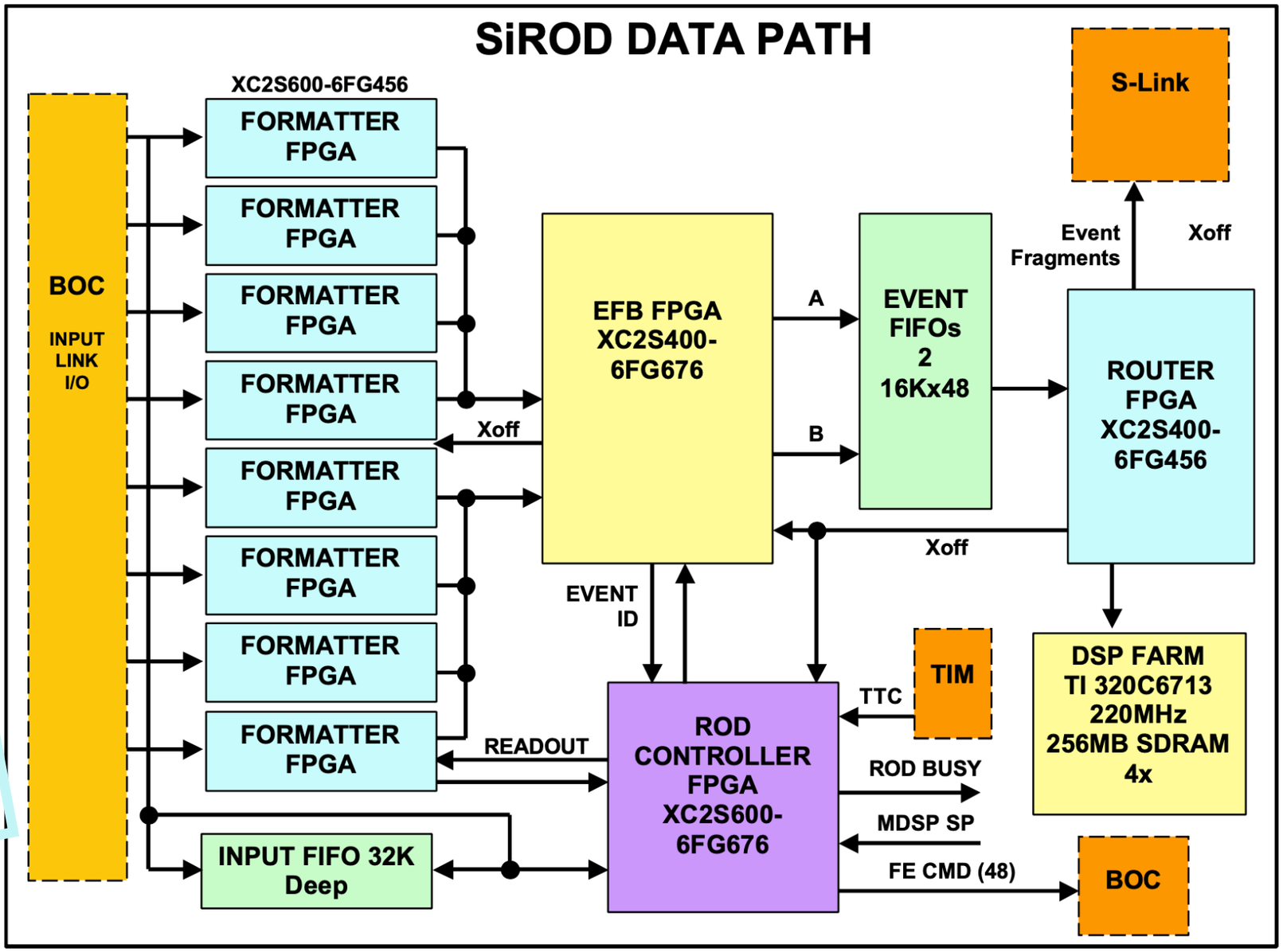




**Figure 6: SiROD Data Path**

- One of the functions of the ROD is to reformat link data into an event fragment.
- Within the ROD this is achieved using logic distributed across multiple FPGAs.
- The logic can be traced across data or control paths, focus here on the data path.



rcarney@lbl.gov

# ABCD-3T format

- ABCD3T FE output contains trigger info, chip address, hit info, hit address.

**Table 12: SCT Data Packet Format**

| Header | DT | L1ID | BC ID | S | | S | | Trailer |
|---|---|---|---|---|---|---|---|---|
| **<11101>** | <0> | <tttt> | <bbbb bbbb> | **<1>** | <data block> | **<1>** | <data block n> | **<1000 0000 0000 0000>** |

**Legend:** L1ID >> **t** = L1ID value from FE Module
BCID >> **b** = BCID value from FE Module

**Table 13: SCT Data Block (Hit in Data)**

| Leader | Chip Address | Hit Channel Address | | First Hit | | Next Chan Hit | | | Nth Chan Hit | | N+1th Chan Hit |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **<01>** | <aaaa> | <ccc cccc> | **<1>** | <ddd> | **<1>** | <ddd> | **<1>** | --- | <ddd> | **<1>** | <ddd> |

**Hit Pattern <ddd> :**

1. Detector Alignment = <1xx> or <x1x> or <xx1>
2. Level = <x1x>
3. Edge = <01x>
4. Test = <xxx>



**Formatter input** **Formatter output**

rcarney@lbl.gov

# ROD formatter

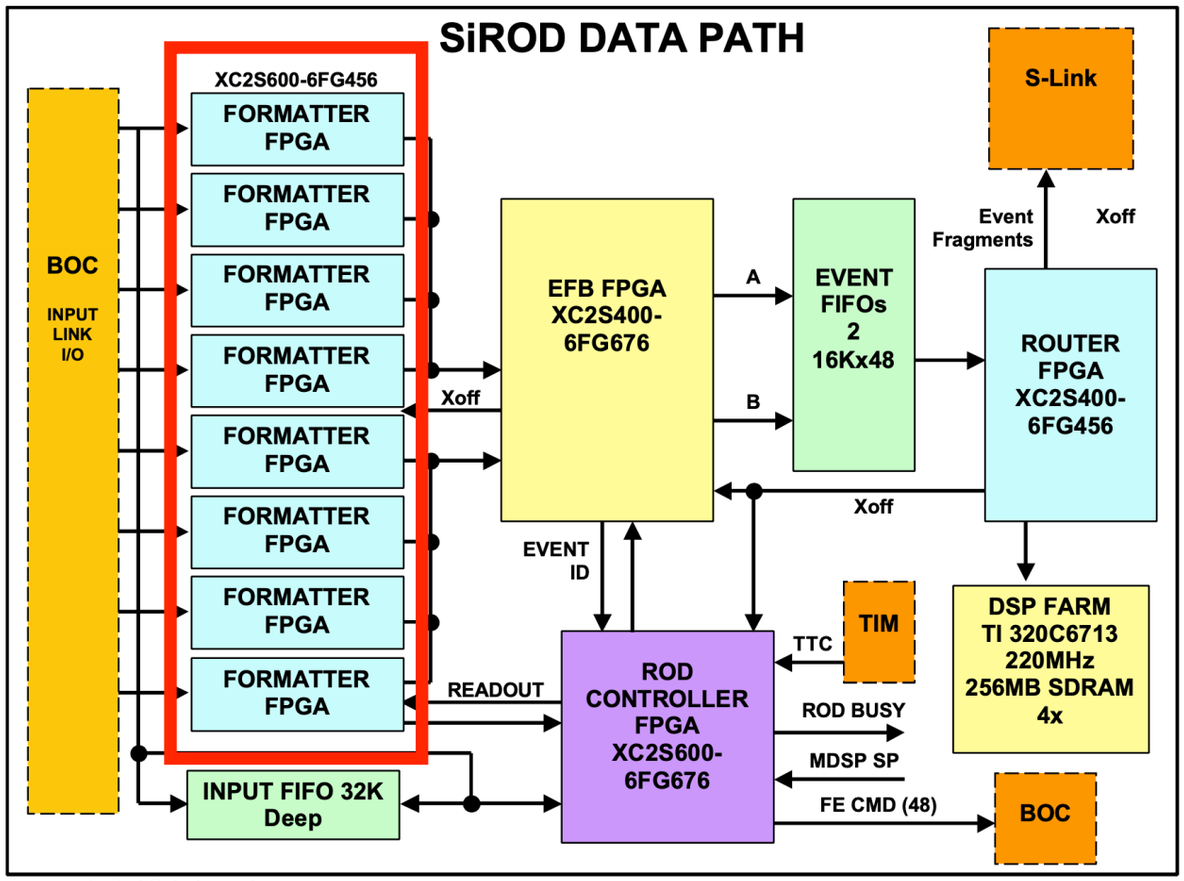



Figure 6: SiROD Data Path

- ABCD3T FE output contains trigger info, chip address, hit info, hit address.
- **Formatter** reorganizes info into clusters with a trigger ID in the header and error flags [32 bits] + [6 bits] of metadata used by EFB.
- Additionally derandomizes link stream, handles module errors.
- This also contains the expanded/ condensed/super-condensed logic packing.

| Name | Bits [15:0] or [31:16] |
|---|---|
| Header | 001pLLLLBBBBBBBB |
| Trailer | 010zhvxxxxxxxxxx |
| 1 hit condensed | 1FFFFCCCCCCCxfx0 |
| 2 hits condensed | 1FFFFCCCCCCCsfx1 |
| 1st hit cluster expanded | 1FFFFCCCCCCC0DDD |
| 1 hit cluster expanded | 1xxxxxxx0xxx1DDD |
| 2 hit cluster expanded | 1xxxxxxx1DDD1DDD |
| Flagged error | 000xxxxxxFFFFEEE |
| Raw data | 011nnnxxWWWWWWWW |

**Table 15: SCT Formatter Output Field Definitions**

| Bits | Definition | Notes |
|---|---|---|
| [31:0] | Event Data | |
| [35:32] | Link Number | (Present for all 32 bit words) |
| [36] | Time Out Error Bit | (Present for all 32 bit words) |
| [37] | Condensed Mode Bit | (Present for all 32 bit words) |

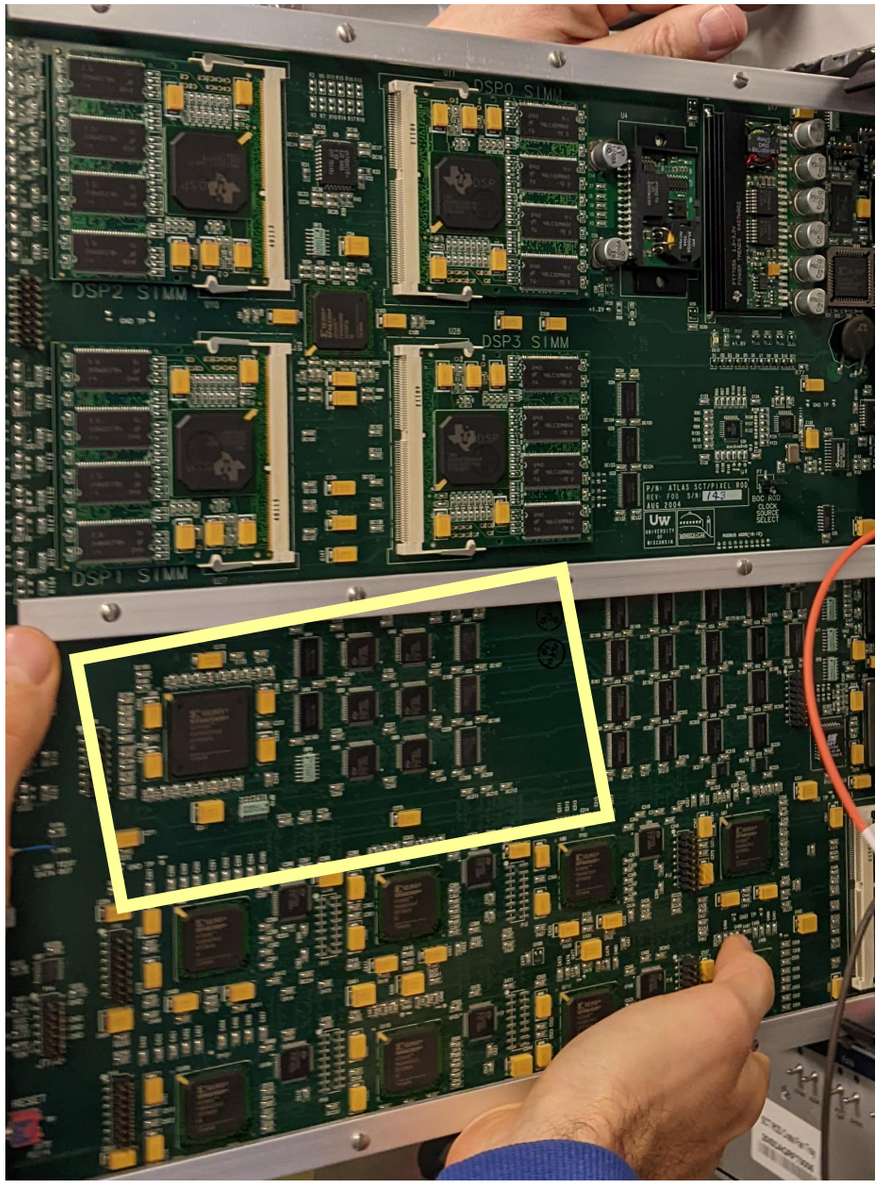


**Formatter input**

**Formatter output**

rcarney@lbl.gov

# **ROD** data path



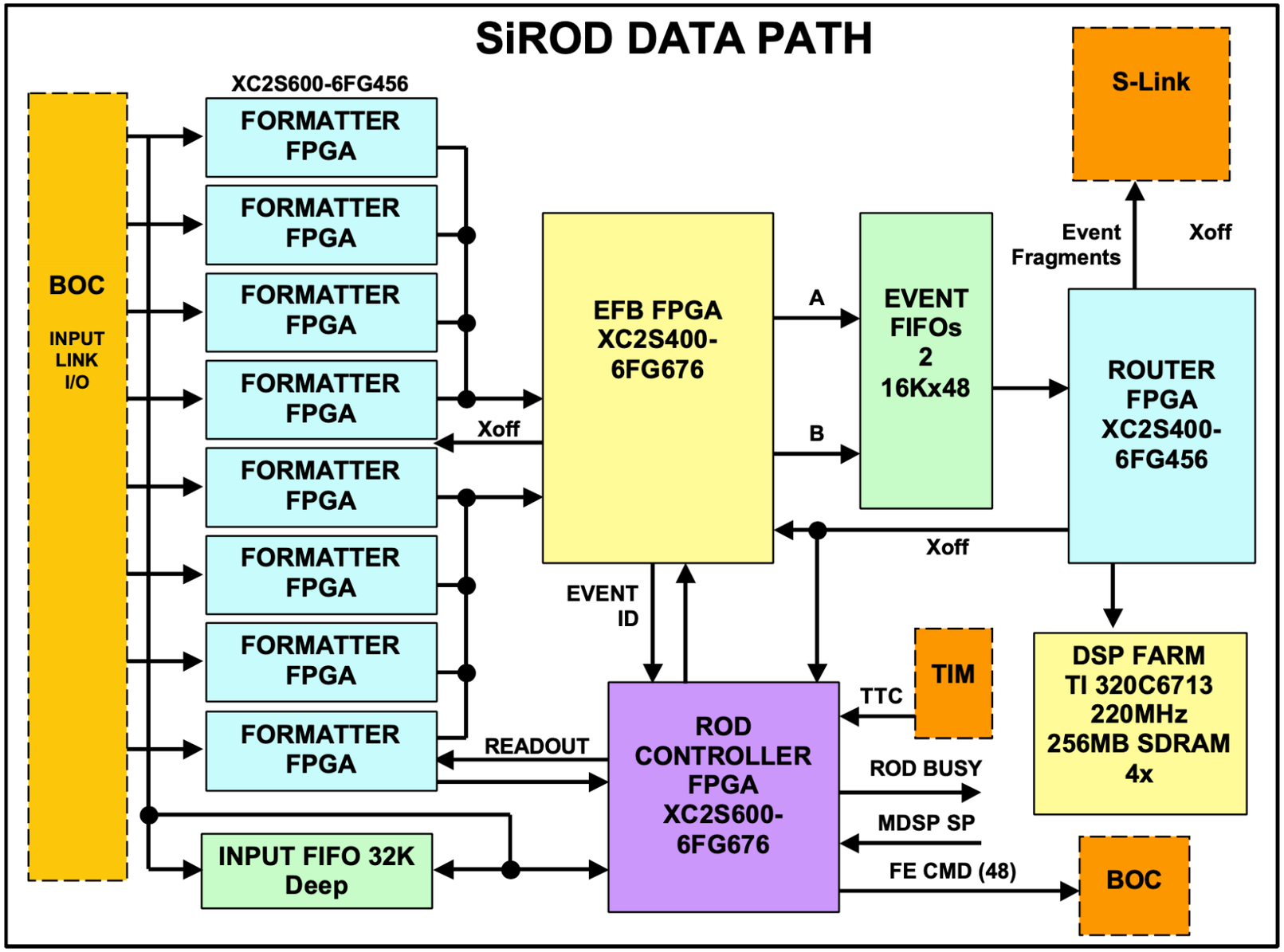




Figure 6: SiROD Data Path

- Now to EFB



rcarney@lbl.gov

# ROD EFB

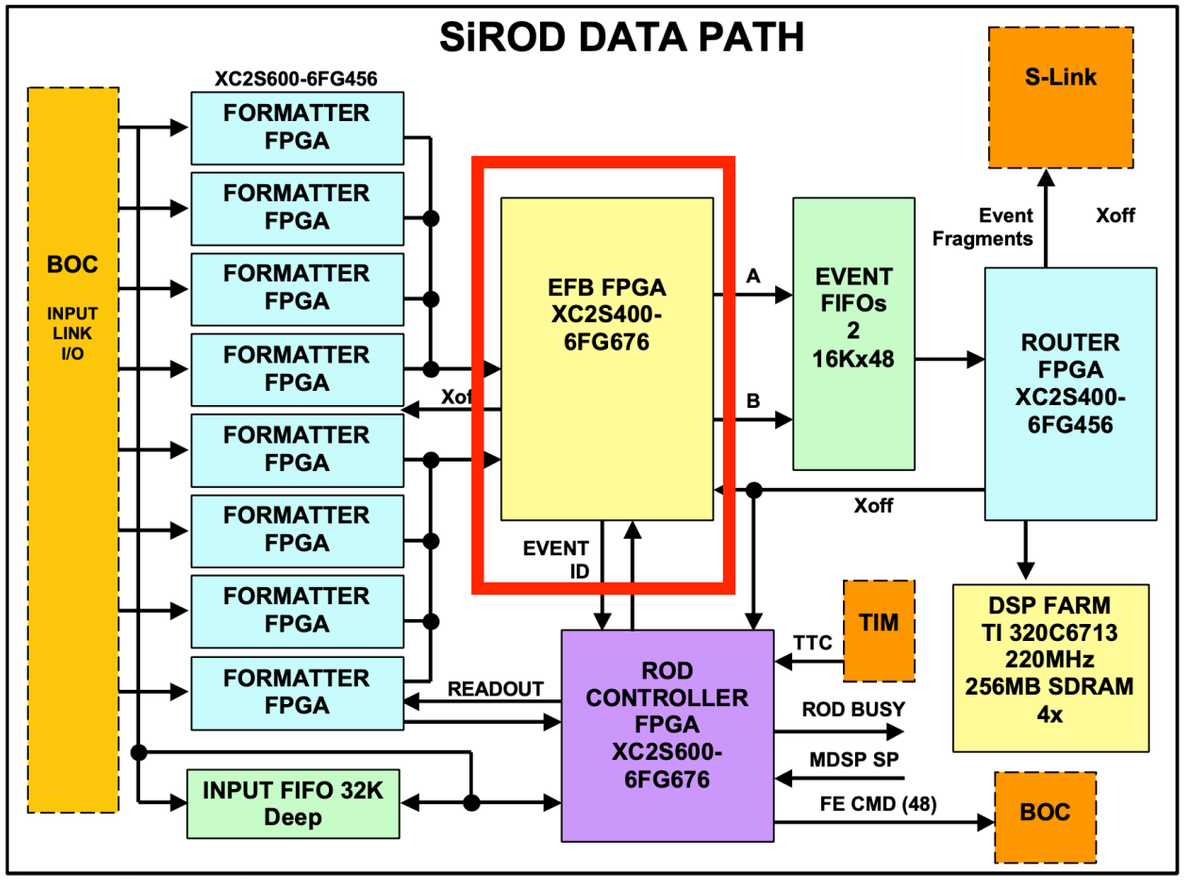



Figure 6: SiROD Data Path

- Event Fragment Builder (EFB) receives data from each formatter.
- **EFB** performs bc/L1D checks & checks for non-sequential, non-monotonic FE #, interprets errors (and raises flags), repacks data into Event Fragment format (see below) + does some zero supression. Send to event FIFO.

**Table 23: Event Header Format at the Output of the EFB**

| Word | Contents | Comment |
|---|---|---|
| 0 | 0xB0FTTTTT | Beginning of fragment marker/<br>T = Router Trap Type Data (Removed by Router) |
| 1 | 0xEE1234EE | Start of header |
| 2 | 0x9 | Header size |
| 3 | 0x30100000 | Format Version Number (Ver 3.1) |
| 4 | 0x001XNNNN Pixel<br>0x002XNNNN SCT | Source Identifier<br>N = Module ID, X = LS Nibble of Sub-detector ID |
| 5 | 0xTTSSSSSS | Run Number: T = Run Type ➔ 0x00 > Physics<br>0x01 > Calibration<br>0x02 > Cosmics<br>0x0F > Test<br>S = Sequence within Run Type |
| 6 | 0xEELLLLLL | Extended Level 1 ID: E = ECR ID, L = L1ID |
| 7 | 0x00000BBB | Bunch Counter ID |
| 8 | 0x000000AA | ATLAS Level 1 Trigger Type |
| 9 | 0x00RR000T | Detector Event Type R = ROD or T = TIM |

**Table 25: EFB Output Link Data Format - SCT**

| Bits | Definition | Notes |
|---|---|---|
| [31:0] | Event Data | |
| [38:32] | Link Number | Present for in header word only |
| [39] | Time Out Error Bit | Present for in header word only |
| [40] | Condensed Mode Bit | Present for in header word only |
| [41] | L1 ID Error Bit | Present for in header word only |
| [42] | BC ID Error Bit | Present for in header word only |

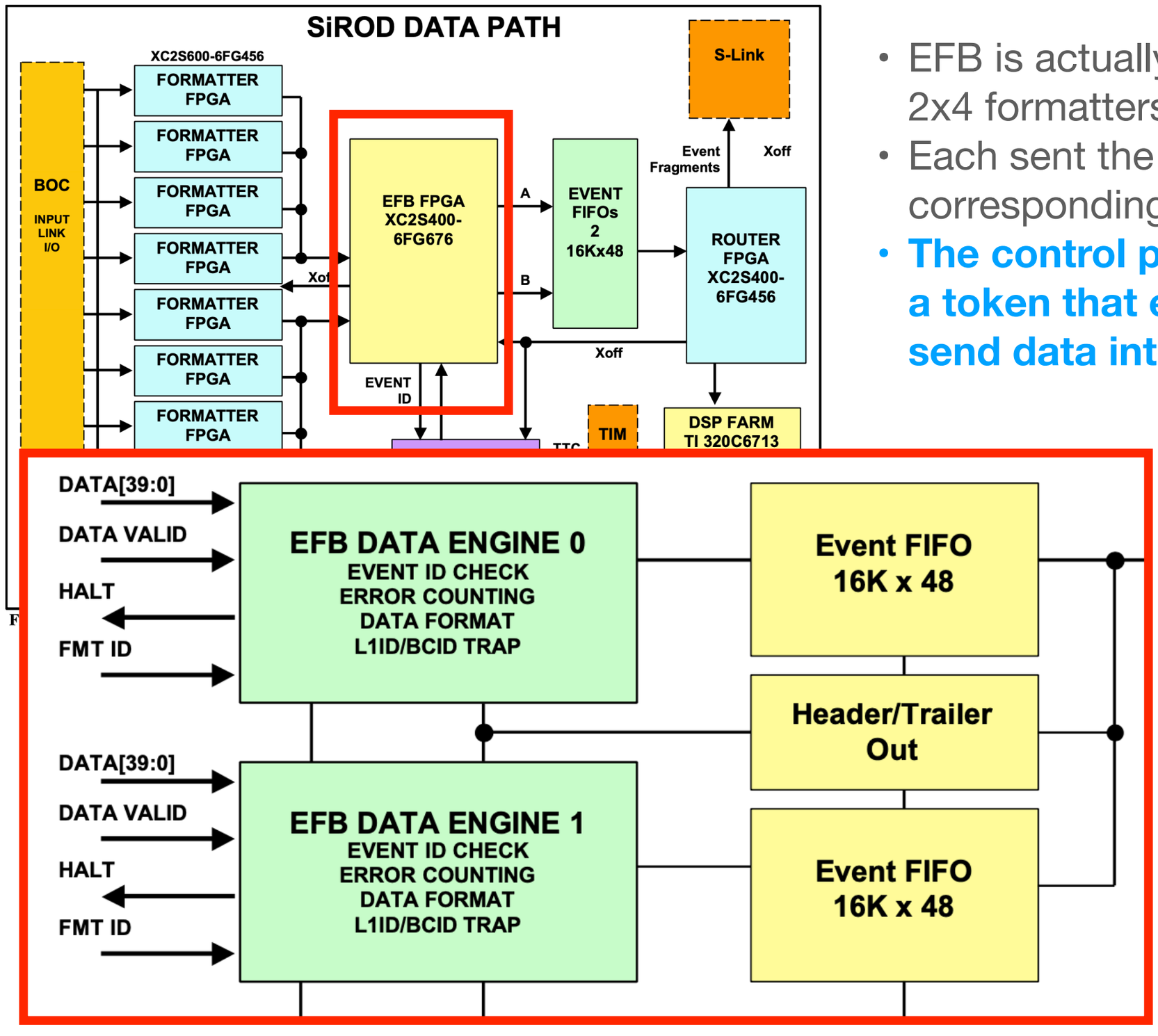


**EFB input**

**EFB output**

rcarney@lbl.gov

# ROD EFB




- EFB is actually 2 instances that deal with 2x4 formatters in parallel).
- Each sent the fragments to a corresponding external FIFO.
- **The control path (not pictured) handles a token that enables each formatter to send data into the EFB instance.**

| FSCT | Links (Rx ID from ROD perspective) |
|---|---|
| 0 | 00-11 (0x00-0x0b) |
| 1 | 16-27 (0x10-0x1b) |
| 2 | 32-43 (0x20-0x2b) |
| 3 | 48-59 (0x30-0x3b) |
| 4 | 64-75 (0x40-0x4b) |
| 5 | 80-91 (0x50-0x5b) |
| 6 | 96-105 (0x60-0x6b) |
| 7 | 126-137 (0x70-0x7b) |

**EFB input**

**EFB output**



rcarney@lbl.gov

# ROD router

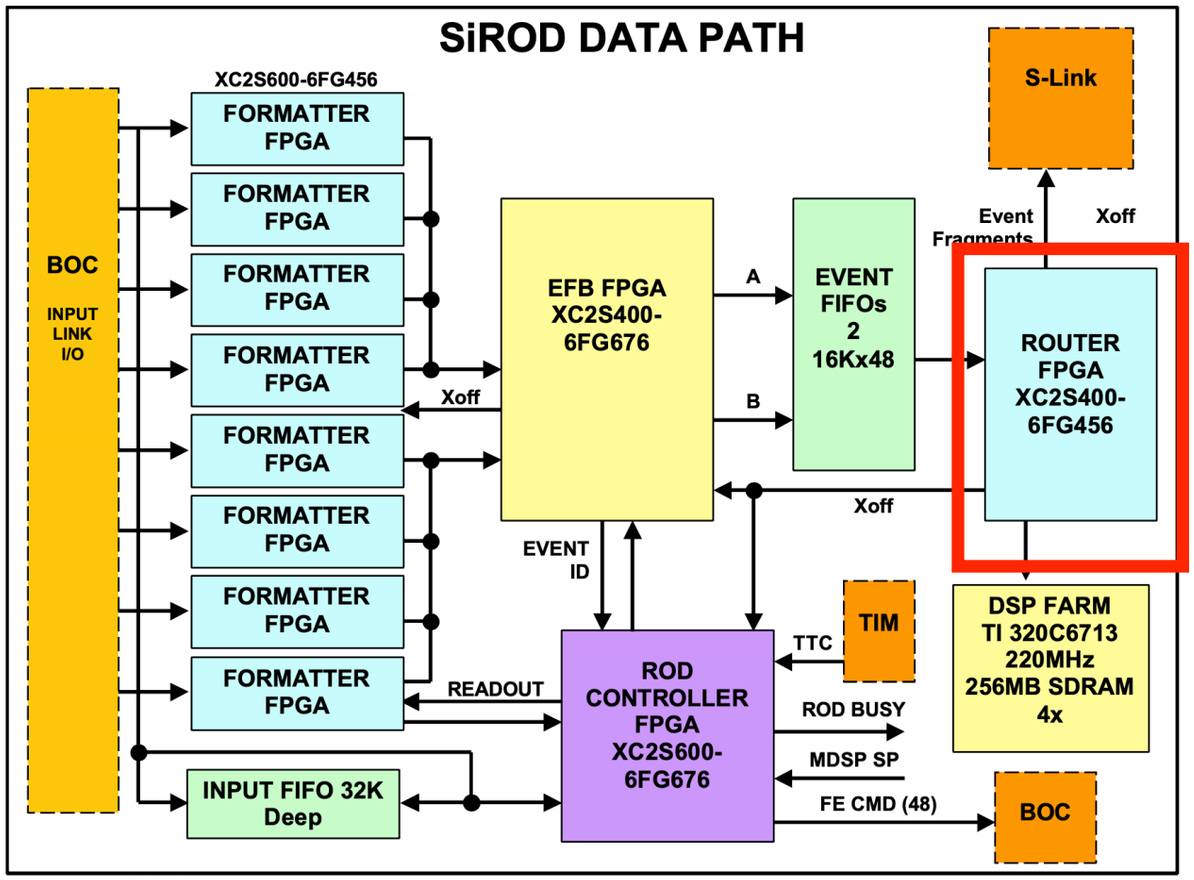



Figure 6: SiROD Data Path

- In terms of data output, the router does not modify the event fragment: it just puts the non-event data info from the EFB into the header.
- We can also “trap” data at the output of the router in the so-called DSP farms. This is useful to debug what is coming out of the router!

### 2.3.3.1 S-Link Event Header and Trailer Output Format

**Table 29: Event Header Format at the Output of the Router**

| Word | Contents | Comment |
|---|---|---|
| 0 | 0xB0F00000 + UCTRL | Beginning of fragment marker |
| 1 | 0xEE1234EE | Start of header |
| 2 | 0x9 | Header size |
| 3 | 0x30100000 | Format Version Number (Ver 3.1) |
| 4 | 0x001XMMMM Pixel<br>0x002XMMMM SCT | Source Identifier<br>M = Module ID, X = LS Nibble of Sub-detector ID |
| 5 | 0xTTSSSSSS | Run Number: T = Run Type ➔ 0x00 > Physics<br>0x01 > Calibration<br>0x02 > Cosmics<br>0x0F > Test<br>S = Sequence within Run Type |
| 6 | 0xEELLLLLL | Extended Level 1 ID: E = ECR ID, L = L1ID |
| 7 | 0x00000BBB | Bunch Counter ID |
| 8 | 0x000000AA | ATLAS Level 1 Trigger Type |
| 9 | 0x00RR000T | Detector Event Type R = ROD or T = TIM |

This is the same as the EFB

**Table 27: Router Link Header Output – SCT Format Bits[31:0]**

| Name | Bits [15:0] or [31:16] – Output to the S-Link | EFB Output |
|---|---|---|
| Header | 001ptlbKdMMMMMMM | 001pLLLLBBBBBBB |

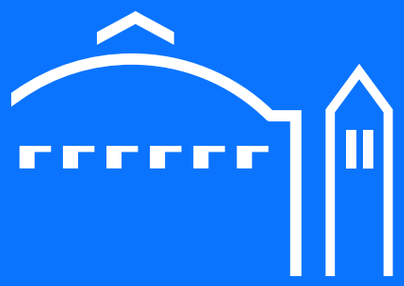
Key:
b = BCID error
l = L1 error
M = link number
t = time out error
d = link masked by DSP
B = BCID
L = L1ID
p = Preamble Error
x = don’t care



rcarney@lbl.gov

28th Feb
2024

# MLHE clues



BERKELEY LAB

rcarney@lbl.gov

# **Out of order** header

```
62210 0xaf10217b
 62211 0xc390c4d0
 62212 0xc780cc10
 62213 0xd280e860
 62214 0xece04000
 62215 Trailer found!!!
 62216 0x00000000
 62217 Status word found, skipping ahead to ROB marker.
 62218 0xdd1234dd
 62219 ROB marker
 62220 0xee1234ee
 62221 ========== ROD marker ==========
 62222 0xa9304000
 62223 ERROR: this data should have been a header!
 62224 ERROR: no header before hit data!
 62225 0x212080c0
 62226 0xade02121
 62227 0xc0d02122
 62228 0x82908610
 62229 0x8ab09730
```

This word is completely out of place

**Table 30: Event Trailer Format at the Output of the Router**

| Word | Contents | Comment |
|---|---|---|
| 0 | See Table 31 | Status 1: Flagged Data Errors |
| 1 | See Table 32 | Status 2: Count of errors in Event Fragment (C)<br>Static Error Flags and ROL Status (S) |
| 2 | 0x2 | Number of status words |
| 3 | 0x0000Ndata | Count of data words |
| 4 | 0x1 | Status block position: 0/1 = before/after data |
| 5 | 0xE0F00000 | End of fragment marker |

In this example there were no other errors for the remainder of the event. Is it possible this word is from a different event?
Counted words (trailer/event/header type, 32-bit full words) and compared to words counted by event FIFO in ROD (recorded in event trailer). Number of words is **always correct, even in events with errors like this.** So we're not losing/gaining words and it seems unlikely it's from another event. However, need to review router code to check where the word counter is implemented, the #words may be a red herring?



rcarney@lbl.gov

# Out of order links

```
183432 0x85302138
183433 ------ Link: 56
183434 0x81808b00
183435 0x2139cba0
183436 ------ Link: 57
183437 0x213a83e0
183438 ------ Link: 58
183439 0x217be4a0
183440 ------ Link: 123
183441 0x214080e0
183442 ------ Link: 64
183443 ERROR: Link headers out of order?
oldLink# = 123, new link# = 64
183444 0x21428490
183445 ------ Link: 66
183446 0x8ac097b0
183447 0x2143c050
183448 ------ Link: 67
183449 0xc3802144
183450 ------ Link: 68
183451 0x84c08650
```

| FSCT | Links |
|---|---|
| 0 | 00-11 (0x00-0x0b) |
| 1 | 16-27 (0x10-0x1b) |
| 2 | 32-43 (0x20-0x2b) |
| 3 | 48-59 (0x30-0x3b) |
| 4 | 64-75 (0x40-0x4b) |
| 5 | 80-91 (0x50-0x5b) |
| 6 | 96-105 (0x60-0x6b) |
| 7 | 126-137 (0x70-0x7b) |
| **Total** | |

| Previous link | Next link | Instances |
|---|---|---|
| 123 (0x7b) | 64 (0x40) | 12% |
| 123 (0x7b) | 96 (0x60) | 6% |
| 122 (0x7a) | 64 (0x40) | 13% |
| 59 (0x3b) | 32 (0x20) | 44% |
| 59 (0x3b) | 33 (0x21) | 12% |
| 58 (0x3a) | 33 (0x21) | 6% |
| 57 (0x39) | 32 (0x20) | 6% |
| **Total** | | **32** |

- A specific set of links seem to get moved around in the event - are they even from the same event (can we trust the word count from previous slide)?
- The links in question are highlighted in yellow above. What is interesting is that these always occur at the EFB instance boundaries!



rcarney@lbl.gov

# Out of order links

```
183432 0x85302138
183433 ------ Link: 56
183434 0x81808b00
183435 0x2139cba0
183436 ------ Link: 57
183437 0x213a83e0
183438 ------ Link: 58
183439 0x217be4a0
183440 ------ Link: 123
183441 0x214080e0
183442 ------ Link: 64
183443 ERROR: Link headers out of order?
oldLink# = 123, new link# = 64
183444 0x21428490
183445 ------ Link: 66
183446 0x8ac097b0
183447 0x2143c050
183448 ------ Link: 67
183449 0xc3802144
183450 ------ Link: 68
183451 0x84c08650
```

| FSCT | Links |
|---|---|
| 0 | 00-11 (0x00-0x0b) |
| 1 | 16-27 (0x10-0x1b) |
| 2 | 32-43 (0x20-0x2b) |
| 3 | 48-59 (0x30-0x3b) |
| 4 | 64-75 (0x40-0x4b) |
| 5 | 80-91 (0x50-0x5b) |
| 6 | 96-105 (0x60-0x6b) |
| 7 | 126-137 (0x70-0x7b) |
| **Total** | |

| Previous link | Next link | Instances |
|---|---|---|
| 123 (0x7b) | 64 (0x40) | 12% |
| 123 (0x7b) | 96 (0x60) | 6% |
| 122 (0x7a) | 64 (0x40) | 13% |
| 59 (0x3b) | 32 (0x20) | 44% |
| 59 (0x3b) | 33 (0x21) | 12% |
| 58 (0x3a) | 33 (0x21) | 6% |
| 57 (0x39) | 32 (0x20) | 6% |
| **Total** | | **32** |

- The Event Fragment Builder is actually two instantiations: each one works in parallel, processing 4 formatters at a time. The first one processing FSCT {0,3}, the second {4,7}.
- There is a token that controls which formatter is currently being read out.
- The fact that the MLHE out of order links predominantly happen at the EFB instance boundaries hints towards an issue at that point.



rcarney@lbl.gov

# **Results** summary

| Error message in file | Word type expected | Recieved | Count |
|---|---|---|---|
| ERROR: Data expected, got ROB marker instead | Data | ROB marker | 4% |
| ERROR: this should have been a header, got ROD marker instead! | Header | ROD marker | 0 |
| ERROR: this should have been hit data, got ROD marker instead! | Data | ROD marker | 0 |
| ERROR: this should have been a header, got trailer instead! | Header | Trailer | 0 |
| ERROR: no header before this trailer! | Header | Trailer | 0 |
| ERROR: no rod marker before this trailer! | ROD marker | Trailer | 0 |
| ERROR: this data should have been a header! | Header | Data | **27%** |
| ERROR: no header before hit data! | Header | Data | **27%** |
| ERROR: hit data before rod marker! | ROD marker | Data | 0 |
| ERROR: this should have been hit data, instead it is a header! | Data | Header | 4% |
| ERROR: header before rod marker! | ROD marker | Header | 0 |
| ERROR: Link headers out of order? | New link > old link | Old link > new link | **40%** |

- Most frequent error is receiving data before a header, explicitly: receiving data when a header was **expected**.
- The link headers are also sometimes appearing out of order.
- The errors above are not mutually exclusive.
- More on these points over the next few slides.



rcarney@lbl.gov

## Tools available to us

- The ROD has a small input FIFO that can be used to trap formatter data. This could allow us to capture problematic data streams that cause MLHE - but it is very shallow: would need to know exactly which stream to capture.
- However, the event fragment ordering is checked only in the ROS output monitor (off-ROD). We would need a deep buffer to store events to account for latency of any feedback and we don't have (a) a deep buffer nor (b) a mechanism to provide feedback!
- The DSP farm can trap data coming out of the router but that's only useful if we think there is an issue at the S-link (we don't).

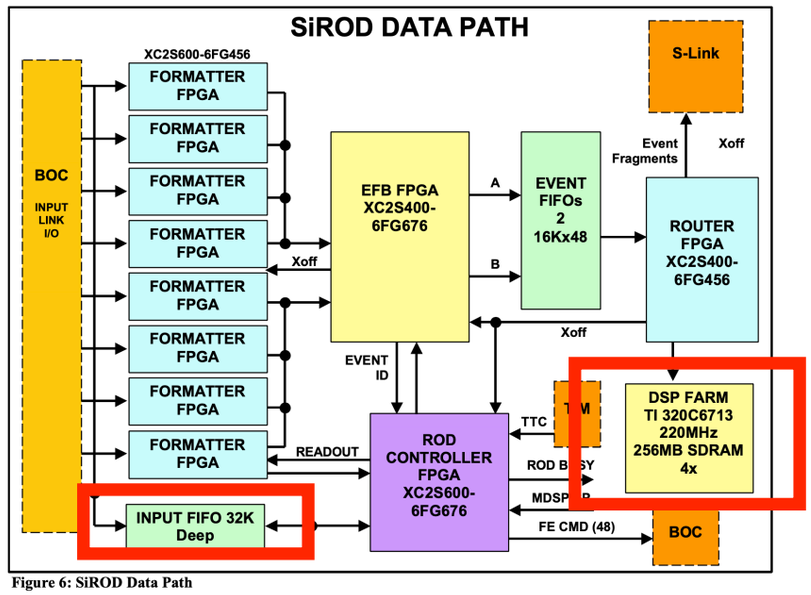


Figure 6: SiROD Data Path

## Other challenges

- The rate at which this error occurs is very low (2 MLHE/min at peak across the entire SCT detector), so it is very improbable that capturing random link streams will result in capturing one responsible for a MLHE.
- We have so far been unable to replicate this issue in (a) simulation or (b) standalone test stands at Berkeley & SR1 using generated 'bad/difficult' data NOR random link streams captured during high-PU events in Run 3.

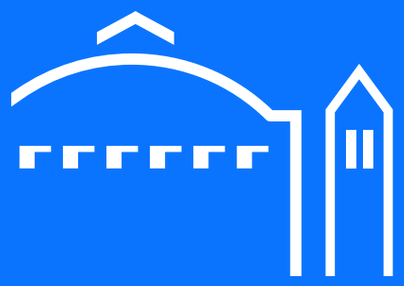
## So what now?



rcarney@lbl.gov

28th Feb
2024

# SCT as test bench



BERKELEY LAB

rcarney@lbl.gov

# Is it just **rate?**

## Is it rate or something special about collision data?

- So far, simulating high-rate events and sending them to the test-stand RODs does not cause MLHE.
- So is it something special about collision data (hit pattern/module occupancy?) or just the high occupancy in Run 3?

- During M2 week (end of Feb) we: lowered the module thresholds, set to anyHit mode, and lowered the sensor bias (i.e. increased noise, opened the modules up to accept the noise as hits) to try and emulate a high-occupant. https://atlasop.cern.ch/elisa/display/534017
- We succeeded in observing MLHE!
- Conclusion: this is purely rate-related, not something special about having beams.
- Result: We now have a test environment (!!) to try and narrow down the cause of the error!

| Time start/end | Test | Threshold | Crates/Rods affected | MLHE rate | Notes |
|---|---|---|---|---|---|
| 10:27:30/10:47:30 | All 0.7 | 0.7 | All | 39 in 20 min = 2/min | This is about the average during runtime. MonitoringScalingFactor = 10% |
| 10:59/11:19 | All 0.5 | 0.5 | All | 51 in 20 min = 2.55/min | MonitoringScalingFactor = 10% |
| 11:56/13:32 | All | 0.7 | All | 181 in 96 min = 2/min | MSF = 1 (100%) No obvious change in rate from increasing the monitoring scaling factor. |

- **Managed to get rate equivalent to that seen in Run 3**
- **Cannot get rate higher (played with HV bias, module thresholds) - this is still a rare occurrence! (Which means we can't try trapping data on a single ROD - still improbable)**



rcarney@lbl.gov

# Plans (from last week)

## Where is this happening in the ROD?

- This is either occurring in the Formatter or EFB.
- Currently we're exploring the EFB since the MLHE links always occur at the EFB instance boundary.
- The EFB contains some logic to perform zero suppression that 'repacks' 16-bit half-words. Maybe that has a small bug?
- Or maybe this is happening further downstream?

| FSCT | Links |
|---|---|
| 0 | 00-11 (0x00-0x0b) |
| 1 | 16-27 (0x10-0x1b) |
| 2 | 32-43 (0x20-0x2b) |
| 3 | 48-59 (0x30-0x3b) |
| 4 | 64-75 (0x40-0x4b) |
| 5 | 80-91 (0x50-0x5b) |
| 6 | 96-105 (0x60-0x6b) |
| 7 | 126-137 (0x70-0x7b) |
| **Total** | |

## Plan for week:

- When SCT not needed for M2 activities with ATLAS: put it into the noisy test-mode. It is great that we have a mode to test things in!
- Try flashing different debug firmware and seeing what happens: e.g. if I remove the packing logic does the MLHE rate or content change?
- Another thing we're going to explore is what the fill-level of the EFB output FIFOs is and if it correlates with MLHE.



rcarney@lbl.gov

# **One issue:** varying baselines

| Date | Time ↓ | Run | MLHE | Run duration [min] | MLHE rate [/min] | Firmware | Notes |
|---|---|---|---|---|---|---|---|
| 28/02 | 15:23 | 0 | 57 | 27 | 2.1 +/- 0.3 | Current release | STANDBY, 0.7fC |
| 04/03 | 8:58 | 0 | 140 | 31 | 4.5 +/- 0.4 | Current release | STANDBY, 0.7fC |
| | 17:12 | 7 | 87 | 33 | 2.6 +/- 0.3 | Current release | STANDBY, 0.7fC |
| 05/03 | 8:35 | 0 | 79 | 30 | 2.6 +/- 0.3 | Current release | STANDBY, 0.7fC |

| Date | Time ↓ | Run | MLHE | Run duration [min] | MLHE rate [/min] | Firmware | Notes |
|---|---|---|---|---|---|---|---|
| 28/02 | 17:34 | 1 | 41 | 26 | 1.6 +/- 0.2 | Current release (rebuilt) | STANDBY, 0.7fC |
| 01/03 | 15:21 | 0 | 61 | 28 | 2.2 +/- 0.3 | Current release (rebuilt) | STANDBY, 0.7fC |
| 04/03 | 10:00 | 1 | 121 | 30 | 4.0 +/- 0.4 | Current release (rebuilt) | STANDBY, 0.7fC |
| | 11:56 | 3 | 137 | 43 | 3.2 +/- 0.3 | Current release (rebuilt) | STANDBY, 0.7fC |
| | 14:13 | 5 | 66 | 47 | 1.4 +/- 0.2 | Current release (rebuilt) | STANDBY, 0.7fC |
| 05/03 | 10:20 | 2 | 103 | 31 | 3.3 +/- 0.3 | Current release (rebuilt) | STANDBY, 0.7fC |
| | 11:11 | 3 | 106 | 30 | 3.5 +/- 0.3 | Current release (rebuilt) | STANDBY, 0.7fC |
| | 12:01 | 4 | 403 | 71 | 5.7 +/- 0.3 | Current release (rebuilt) | STANDBY, 0.7fC |

- 1.5-2.4
- 2.5-3.4
- 3.5-4.4
- 4.5-10
- 10+

- Uncertainties are just std error on the mean.
- **Baseline rates have varied in time.**
- Have not found any obvious correlations by looking at module temp, cooling exhaust temp, cooling power. But not a true systematic study done.
- Once I noticed the baselines shifting I tried to sandwich each debug measurement with a baseline measurement. But still hard to draw conclusions..



rcarney@lbl.gov

# All tests

- Summary of MLHE rates over tests done in the past week

| Date | Time | Run | MLHE | Run duration [min] | MLHE rate [/min] | Firmware | Notes |
|---|---|---|---|---|---|---|---|
| 28/02 | 15:23 | 0 | 57 | 27 | 2.1 | Current release | STANDBY, 0.7fC |
| | 17:34 | 1 | 41 | 26 | 1.6 | Current release (rebuilt) | STANDBY, 0.7fC |
| 29/02 | 10:04 | 0 | 145 | 54 | 2.7 | Current release | *SUPSAFE*, 0.7 fC |
| 01/03 | 15:21 | 0 | 61 | 28 | 2.2 | Current release (rebuilt) | STANDBY, 0.7fC |
| | 16:23 | 1 | 433 | 25 | 17.3 | EFB repack removed | STANDBY, 0.7fC |
| 04/03 | 8:58 | 0 | 140 | 31 | 4.5 | Current release | STANDBY, 0.7fC |
| | 10:00 | 1 | 121 | 30 | 4.0 | Current release (rebuilt) | STANDBY, 0.7fC |
| | 10:54 | 2 | 119 | 38 | 3.1 | Reduced ev_afull_counter in EFB | STANDBY, 0.7fC |
| | 11:56 | 3 | 137 | 43 | 3.2 | Current release (rebuilt) | STANDBY, 0.7fC |
| | 13:01 | 4 | 135 | 50 | 2.7 | Halved ev_afull_counter | STANDBY, 0.7fC |
| | 14:13 | 5 | 66 | 47 | 1.4 | Current release (rebuilt) | STANDBY, 0.7fC |
| | 15:50 | 6 | 168 | 61 | 2.8 | Halved ev_afull_counter | STANDBY, 0.7fC |
| | 17:12 | 7 | 87 | 33 | 2.6 | Current release | STANDBY, 0.7fC |
| 05/03 | 8:35 | 0 | 79 | 30 | 2.6 | Current release | STANDBY, 0.7fC |
| | 9:27 | 1 | 122 | 32 | 3.8 | Halved ev_afull_counter | STANDBY, 0.7fC |
| | 10:20 | 2 | 103 | 31 | 3.3 | Current release (rebuilt) | STANDBY, 0.7fC |
| | 11:11 | 3 | 106 | 30 | 3.5 | Current release (rebuilt) | STANDBY, 0.7fC |
| | 12:01 | 4 | 403 | 71 | 5.7 | Current release (rebuilt) | STANDBY, 0.7fC |
| | 13:45 | 5 | 493 | 39 | 12.6 | EFB repack removed | STANDBY, 0.7fC |
| | 14:42 | 6 | 30 | 13 | 2.3 | Halved fifoRAMb afull thresh | STANDBY, 0.7fC |

- Current
- Current (rebuilt)
- Halved ev_afill counter
- EFB repack removed
- Other



rcarney@lbl.gov

# **All** tests (coloured by rate)

- Summary of MLHE rates over tests done in the past week

| Date | Time | Run | MLHE | Run duration [min] | MLHE rate [/min] | Firmware | Notes |
|---|---|---|---|---|---|---|---|
| 28/02 | 15:23 | 0 | 57 | 27 | 2.1 | Current release | STANDBY, 0.7fC |
| | 17:34 | 1 | 41 | 26 | 1.6 | Current release (rebuilt) | STANDBY, 0.7fC |
| 29/02 | 10:04 | 0 | 145 | 54 | 2.7 | Current release | *SUPSAFE*, 0.7 fC |
| 01/03 | 15:21 | 0 | 61 | 28 | 2.2 | Current release (rebuilt) | STANDBY, 0.7fC |
| | 16:23 | 1 | 433 | 25 | 17.3 | EFB repack removed | STANDBY, 0.7fC |
| 04/03 | 8:58 | 0 | 140 | 31 | 4.5 | Current release | STANDBY, 0.7fC |
| | 10:00 | 1 | 121 | 30 | 4.0 | Current release (rebuilt) | STANDBY, 0.7fC |
| | 10:54 | 2 | 119 | 38 | 3.1 | Reduced ev_afull_counter in EFB | STANDBY, 0.7fC |
| | 11:56 | 3 | 137 | 43 | 3.2 | Current release (rebuilt) | STANDBY, 0.7fC |
| | 13:01 | 4 | 135 | 50 | 2.7 | Halved ev_afull_counter | STANDBY, 0.7fC |
| | 14:13 | 5 | 66 | 47 | 1.4 | Current release (rebuilt) | STANDBY, 0.7fC |
| | 15:50 | 6 | 168 | 61 | 2.8 | Halved ev_afull_counter | STANDBY, 0.7fC |
| | 17:12 | 7 | 87 | 33 | 2.6 | Current release | STANDBY, 0.7fC |
| 05/03 | 8:35 | 0 | 79 | 30 | 2.6 | Current release | STANDBY, 0.7fC |
| | 9:27 | 1 | 122 | 32 | 3.8 | Halved ev_afull_counter | STANDBY, 0.7fC |
| | 10:20 | 2 | 103 | 31 | 3.3 | Current release (rebuilt) | STANDBY, 0.7fC |
| | 11:11 | 3 | 106 | 30 | 3.5 | Current release (rebuilt) | STANDBY, 0.7fC |
| | 12:01 | 4 | 403 | 71 | 5.7 | Current release (rebuilt) | STANDBY, 0.7fC |
| | 13:45 | 5 | 493 | 39 | 12.6 | EFB repack removed | STANDBY, 0.7fC |
| | 14:42 | 6 | 30 | 13 | 2.3 | Halved fifoRAMb afull thresh | STANDBY, 0.7fC |

- 1.5-2.4
- 2.5-3.4
- 3.5-4.4
- 4.5-10
- 10+

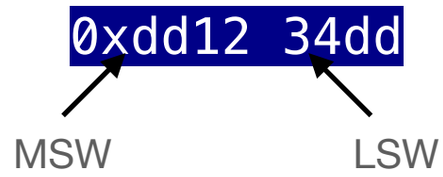


rcarney@lbl.gov

# **Some** conclusions

format_data.vhd +1 −58 View file @ 8201b7bc

```
138 138        last_word_flush    <= last_word;
139 139        if (rod_type = '0') then  -- SCT
140     - --  ********************  NEW: added in Ver19F *****************
141     - --      if (last_word_flush = '1' and extra_halfword = '1') then  -- Remove if dropping last trailer
142     - --        extra_halfword <= '0';                                  --
143     - --        data_valid_out_p2 <= '1';                               --
144     - --        data_out_p2(45 downto 0) <= "0000000"                   --
145     - --                                 & data_out_p2(38 downto 32)    --
146     - --                                 & extra_data                   --
147     - --                                 & "0100000000000000";          -- Remove if dropping last trailer
148     -         if (last_word = '1' and extra_halfword = '1') then  -- Use when dropping last trailer
149     - --      elsif (last_word = '1' and extra_halfword = '1') then     -- Remove if dropping last trailer
150     -           if (msw_valid = '1') then
151     -             if (lsw_valid = '1') then -- Usually H/T word on an odd boundary
152     - --            extra_halfword <= '1';  -- Remove if dropping last trailer
153     -               data_out_p2(45 downto 0) <= '0'
154     -                                        & data_in_d1(44 downto 32)
155     -                                        & extra_data
156     -                                        & data_in_d1(31 downto 16);
157     -             elsif (lsw_valid = '0') then -- Usually D/T word on an odd boundary
158     - --            extra_halfword <= '0';  -- Remove if dropping last trailer
159     -               data_out_p2(45 downto 0) <= "0000000"
160     -                                        & data_out_p2(38 downto 32)
161     -                                        & extra_data
162     -                                        & data_in_d1(31 downto 16);
163     -             end if;
164     -           elsif (msw_valid = '0') then -- Extra D word on an odd boundary
165     -             if (lsw_valid = '0') then
166     - --            extra_halfword <= '0';  -- Remove if dropping last trailer
167     -               data_out_p2(45 downto 0) <= "0000000"
168     -                                        & data_out_p2(38 downto 32)
169     -                                        & extra_data
170     -                                        & "0100000000000000";
171     -             end if;
172     -           end if;
173     -           data_valid_out_p2 <= '1';
174     -           extra_halfword <= '0'; -- Use when dropping last trailer
175     - --  ^^^^^^^^^^^^^^^^^^^^  NEW: added in Ver19F  ^^^^^^^^^^^^^^^^
176     -         elsif (data_valid_in_d1 = '1') then
177     -           if (msw_valid = '1' and lsw_valid = '0' and extra_halfword = '0') then
178     -             extra_halfword <= '1';
179     -             extra_data(15 downto 0) <= data_in_d1(31 downto 16);
180     -           elsif (msw_valid = '1' and lsw_valid = '0' and extra_halfword = '1') then
```




The repacking can remove empty clusters and repack with buffered data from previous word (MSW and LSW get moved to different words).

| Date | Time | Run | MLHE | Run duration [min] | MLHE rate [/ min] | Firmware | Notes |
|---|---|---|---|---|---|---|---|
| 01/03 | 16:23 | 1 | 433 | 25 | 17.3 +/- 0.8 | EFB repack removed | STANDBY, 0.7fC |
| 05/03 | 13:45 | 5 | 493 | 39 | 12.6 +/- 0.6 | EFB repack removed | STANDBY, 0.7fC |

- Two main tests run so far.
- In first, the EFB repacking (zero suppression) was removed, thereby increasing the event size: https://gitlab.cern.ch/atlas-sct-rod-daq/rod_efb/-/commit/8201b7bc1b24383c71226fbcc14bf6d084de818f
- This increased the MLHE rate significantly, even considering the changing base rate.
- **What does this tell us?** That the event repacking in the EFB is not the source of the MLHE and that event size increases the rate.



rcarney@lbl.gov

# **Some** conclusions?

ev_data_decode.vhd

```
@@ -180,9 +180,13 @@ begin
    ev_data_empty <= '1';
  elsif (clk'event AND clk = '1') then
                                          -- 64 events at 16 w/evt = 1024 words/almo
-   --if (fifocount_out > "1110000000") then  -- 56 events at 16 w/evt = 896 words/alm
-   if (fifocount_out > "1101000000") then  -- 832 words/almost full`
-
+   -- Default
+       --if (fifocount_out > "1110000000") then  -- 56 events at 16 w/evt = 896 words
+   -- Test 1:
+       --if (fifocount_out > "1101000000") then  -- 832 words/almost full`
+   -- Test 2:
+       if (fifocount_out > "0111000000") then  -- 448 words/almost full`
+
      ev_data_almost_full <= '1';
    else
```

```
-- Description:
--    ROD Dual Gatherer
--    SCT Version
--    Reads Event FIFO data from the external FIFO and writes the 8 bits
--    BCID and 4 bit L1 ID data into the Event ID FIFOs of each gatherer.
--    The full L1, BC ID, and Trig type info are sent to the logic that
--    generates the event fragment header/trailer.
--
--    A full event of data must be read out over 17 words from the
--    external FIFO since there are so many bits of data.
--    The 4K FIFO can buffer the data from 240 Events
--
--  word 0  : L1 ID[15:0]                                  ev_data_sh_reg( 15: 0)
```

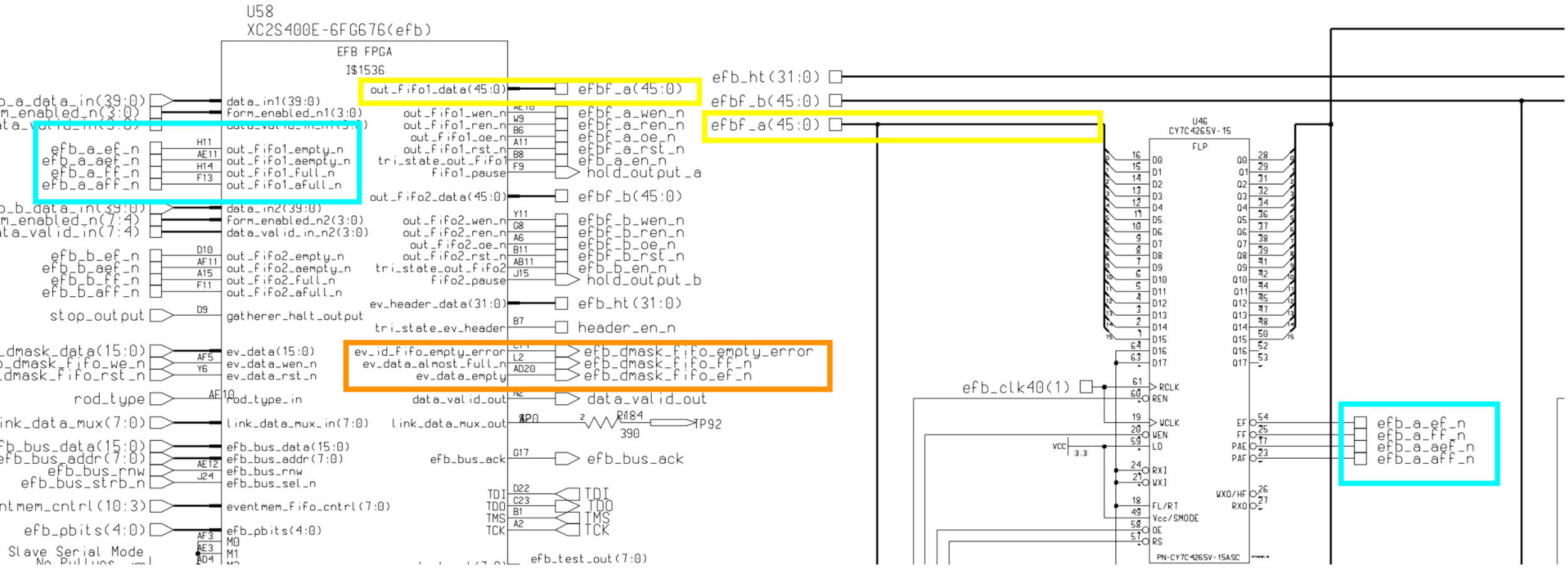
| Date | Time | Run | MLHE | Run duration [min] | MLHE rate [/ min] | Firmware | Notes |
|---|---|---|---|---|---|---|---|
| 04/03 | 13:01 | 4 | 135 | 50 | 2.7 +/- 0.2 | Halved ev_afull_counter | STANDBY, 0.7fC |
| | 15:50 | 6 | 168 | 61 | 2.8 +/- 0.2 | Halved ev_afull_counter | STANDBY, 0.7fC |
| 05/03 | 9:27 | 1 | 122 | 32 | 3.8 +/- 0.3 | Halved ev_afull_counter | STANDBY, 0.7fC |

- The second test halves the FIFO almost-full threshold in the event header mask FSM (not that the repacking was added back in for this test. Tests are mutually exclusive): https://gitlab.cern.ch/atlas-sct-rod-daq/rod_efb/-/commit/54ef31a57f5f7d4bd11e3f118d270b9163454295
- Essentially it delays the write state earlier that it normally would in the Dynamic Mask Read Out State Machine, i.e. writes to the FIFO from the EFB are buffered for longer.
- The threshold was halved from d'896 to d’448
- Hard to draw conclusions from these runs. In all cases they are sandwiched by baselines that changed. However, it is clear that any results of this change are not as significant as with repacking test.
- In each case we could maybe say that this firmware change did not change the rate since it is consistent (within uncertainty) with baseline? But really not clear.



rcarney@lbl.gov

# **Re-reading** output?




- From the data sheet here: https://www.digikey.com/en/products/detail/infineon-technologies/CY7C4265V-15ASC/2270518 the last value read from the FIFO remains on the FIFO output pins even when the FIFO is empty. It's kind of throwaway line in the datasheet: "An empty FIFO maintains the data of the last valid read on its Q 0–17 outputs even after additional reads occur."
- Another avenue to explore could be: see if that last value in the FIFO is being sampled near the start of the next event before the FIFO is ready to be read? And then we'd see double words, which I don't think we ever do (do we check for that??).
- Playing with the is_almost_empty (not actual name) threshold should change the rate in the case that this is actually the cause.
- Hard to simulate because different components!!



rcarney@lbl.gov

26th Apr 2024


Thanks for the warm support of SCT operations & experts team!



rcarney@lbl.gov

- **Firmware in git**
  - The SCT ROD has four FPGA's: Formatter, Event Fragment Builder, Router, and Controller
  - Each has its own repo of vhdl source code linked above and here: https://gitlab.cern.ch/atlas-sct-rod-daq
  - Main will always contain the current release source code. Branches are for debug or features. Releases are tagged in main.
- **How-to-build document**: https://gitlab.cern.ch/atlas-sct-rod-daq/rod_docs
  - Specifies how to install legacy Xilinx toolkit and build each project.
  - Let me know if you have any suggested edits!