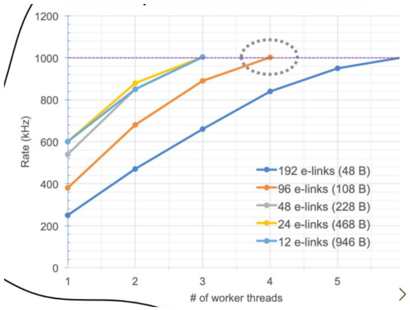
Very Brief Intro

- Thanks everyone for joining! Most things summarized on JIRA : https://its.cern.ch/jira/browse/ATLITKSW-232. These are just suggestions to get discussion started:
- Suggested steps forward (steps 1-4 needed regardless of where we eventually decode)
- 1. Find most efficient/easiste to integrate with Athena encoder (Sebastien, Ondra with help from Mathias/Neomi/Shaun?)
  - Neomi’s is already in older version of Athena (https://gitlab.cern.ch/itk-felix-sw/RD53BEmulator/-/blob/master/src/Stream.cpp),
  - Carlos’s:https://gitlab.cern.ch/itk-felix-sw/RD53BEmulator/-/blob/master/src/Stream.cpp,
  - Ondra’s: https://indico.cern.ch/event/1381390/contributions/5879525/attachments/2826566/4937870/ITkPixV2_encoding_decoding.pdf
  - Mathias is also working on a repo trying to pull out the different encoding/decoding algorithms that exist and put them into individual packages: https://gitlab.cern.ch/wittgen/endec
- 2. Include encoder in latest version of Athena (Sebastien, Ondra with help from Neomi/Shaun/Simone?), might take time?
- 3. Simulate ttbar events
- 4. Evaluate performance of three available decoders (itk-felix-sw, YARR, Orion), hopefully in HSU units as used by DAQ/EF
- 5. Integrate with Event Filter (EF) and Tracking
  - Get ttbar decoded/decompressed events in RDO File format used by Clustering at the EF

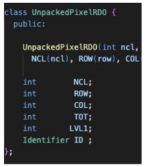
Step 5: Why integrate with EF?

- Why not decode at Data Handler (DH)? We can for subset of events for chip monitoring, but for all events it might be difficult:
  - data in and out of DH will be at 5.2TB/s @1MHz – ie we can't just decode/decompress, would also have to compress again
  - Data Handler resources tight already without sub-detector modules current PCs are full
  - Data Handler is underground – so tighter space/power requirements. Network switches after DH also limiting rate in bandwidth studies
    - maximum occupancy is 6 FELIX (2U) and 12 DH (1U) per rack – 24U occupancy per rack. Number of racks needed 54 (CBE) and 59 (MPV). Per server power usage FELIX 660W (863) for CBE (MPV), and DH 304 (334) for CBE (MPV)
  - 110 PIXEL, 76 for STRIP DH servers forseen (budget does not allow for very high performance servers, hoping on having in the future 32 core servers to implement 400 Gb/s N).
- If decoding at Event Filter, only have to do it for 10% of events after L0 decision
  - Regional tracking allows a fast initial rejection in the EF of single high-pT lepton and multi-object triggers from background processes, to reduce the rate to around 400 kHz. This system is specified to operate at **1 MHz and use up to 10% of the ITk data,** by selecting tracking modules in regions based on the results of the Level-0 trigger system. Software-based reconstruction will follow to achieve further rejection.
- Storage Handler before Event filter has a lot of space to buffer data
  - The large storage volume needed to achieve the required 7.8 TB/s throughput allows for increased event processing latency. This can be provided by 1800 SSDs providing 36 PB of storage~ > an hour of event buffering at L0 rate during EF processing
- Event Filter already has some "room" for us to decode - 2.2/3.2 HSU at 140/200 mu – technically this i just for getting decoded/decompressed events into RDO format, but TDAQ folks are open to seeing whether we can fit decoding/decompression in this budget as well, and possibly expand budget
- Should start with the CPU based Event Filter, then explore FPGA/GPU options depending on what DAQ settles on finally
- Suggested RDO format is evolving, discussions started by Haider: https://indico.cern.ch/event/1396809/contributions/5872894/attachments/2824025/4932543/Pixel%20&%20Strip%20dataformat.pdf



Scaled Data Handler perforamce – no sub-detector modules
12 CPU Core with Intel Xeon 5218 2.3 Ghz CPU - Surgei Kolos et al

"Physics data" for pixel

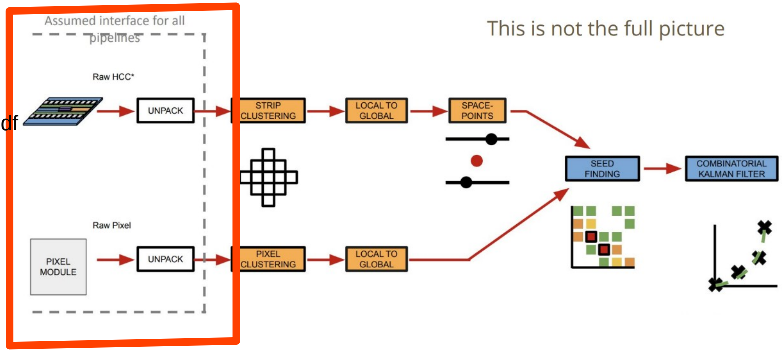


```
class UnpackedPixelRDO {
  public:

    UnpackedPixelRDO(int ncl,
      NCL(ncl), ROW(row), COL

    int         NCL;
    int         ROW;
    int         COL;
    int         TOT;
    int        LVL1;
    Identifier ID ;
};
```

NCL	Index for hit index Depends on when we reset the counter, this might have to be as large as 0-250k range
ROW	Row Index of the fired pixel. Max range is 400 So range = [0,512]
COL	Col Index of the fired pixel. Max range is 384 So range = [0,512]
TOT	Time over threshold for the pixel. At maximum 8 bits, but depends on the compression mode
LVL 1	Level 1 accept. Unclear what it is used for. But at around O(1) bit
Identifier	64 bit identifier for the pixel. This is an encoded object and probably needs to be constructed at decoding setup or in CPU

Haider's talk

This is not the full picture



$\langle\mu\rangle$	Tracking	Byte Stream Decoding	Cluster Finding	Pixel Sp.Points	Strip Sp.Points	Si Track Finding	LRT Track Finding	Total ITk
140	primary	$2.2^{(*)}$	6.1	1.0	-	13.4	-	22.7
	LRT	-	-	-	2.4	-	2.5	4.9
200	primary	$3.2^{(*)}$	8.1	1.2	-	23.2	-	35.8
	LRT	-	-	-	3.5	-	5.9	9.3

(*) Scaled from Run-2. The time spent for decoding as used in Ref. [3] is updated for the new Pixel event size.

Thanks! Thoughts?