

EPIC DAQ Protocols and Operation Scheme -- Organization

1. The first & most critical job we have is getting the data to a computing facility in some useable form → working electronics
 - a. The operation does influence the electronics
 - b. The topic of what the data will look like to the computing is important,
 - there is a lot of interest from the wider collaboration
 - this interest is important to the collaboration because their actions should reflect what the data will actually look like.
2. We need to join with software, I've been emailing with Marcus regarding the organization of this
 - a. Software is organizing around task forces
 - Something like an “agile” project management sprint with multiple software developers focusing effort on producing code for a short period and then re-organizing around next goal
 - My view is the best scheme is a subgroup of DAQ and a subgroup of S/C to meet every 2 weeks to define the specific issues and report back to the main groups.
 - On our side I know there are a lot of different ideas. We need written proposals to discuss and refine rather than fuzzy concepts.
 - b. For DAQ group, thus far interest was expressed by:
 - Jeff Landgraf, Jan Bernauer, Tonko Ljubicic
 - Let me know if you want to be added

EPIC DAQ Protocols and Operation Scheme (Jeff/Tonko strawman) (1)

The proposed organizing principle is the time frame.

- The RDO would organize headers around the concept of the time frame.

Field	Bits	Time interval	Per Fiber Data Volume	Final Data volume	Earliest Use
		500Y	-	-	
Epoch	28	1-10min	-	-	GTU/DAM
TF / Token	16	O(1) ms	O(100kb)	O(1MB)	RDO
Bin	20	10ns precision 2us max interval	-	-	ASIC (with caveats)

- $BX = \text{Epoch} * (2^{16}) + \text{TF} * (\text{BinCnt}) + \text{Bin}$
- The Full DAQ is oriented around the token, which takes the role of “event number” in a triggered system.
 - Data packets transferred by components
 - Data files indexed by token, producer
 - By token, nice properties such as time ordering, no gaps, whatever...
 - Within token packet data, format set by needs of producer
- The number of BX in a token can be configurable, but will be static for any running period.
- Assuming 3000 RDOs in system : TF (16 bits) IDX (16bits) -> min overhead ~32kb/s (~.3% overhead)

EPIC DAQ Protocols and Operation Scheme (2)

Data Format Concept

- I expect that any DATA format will be reducible logically to a directory structure
 - Star – SFS. Literally access data banks with linux file structure (e.g. /evt#/TPC/SECxx/RDOyy/RAW)
 - sPHENIX - Similar but predefined file names?
 - JLAB - ?
- For EPIC I'd propose the same:
 - /TF/DET/RDOxxx/raw
- Data Format Needed Features:
 - A data bank on its own is a valid file
 - Two files appended remain valid file
 - No required non-local parameters (e.g. No offsets, no file sizes etc at start of file)
 - If there is an index scheme, it must be reconstruct-able
 - Low-level features
 - Simple interface → write(fn, bank_ptr, bank_sz, dest_buffer)
 - Scatter-gather support → writev(bank_spec_array, dest_buffer)
- Data at any stage of processing is valid output file!

EPIC DAQ Protocols and Operation Scheme (3)

RDO

- Token known. Data packetization need not match token boundaries but token boundaries are marked.

DAM

- Depends upon processing defined for DAM, could at this stage produce formatted data, but likely not...

Readout
Computers

- Constructs TF from token. Formats “/TF/raw” as a minimum. Perhaps there are sub-banks created by DAM. If so, formatted now.
- Support local running mode, and can write valid files locally
- Produces additional banks, eg “/TF/cld”,
- May need to have input produced from other detector (eg trigger), so will need a wait for contribution concept.
- Output may already suppress “/TF/raw”
- Sends contribution to specific node based on token number

COTS

- Hierarchy to be defined (Building nodes, processing nodes, QA nodes, etc...)
- Generally speaking will have agents that receive a bank, create a new bank, and transmit either both banks, or only the new bank to another agent

Final Data File

- Full detector data in 1 file per token
- List of banks can be variable by token. For example we may define 1 in 100 tokens to contain raw data passed from DAM, but the rest only processed data.

EPIC DAQ Protocols and Operation Scheme (4)

Flow Control

- Strictly speaking any flow control breaks streaming concept!
- This doesn't remove any of the benefits of the streaming concept...

GTU

- Maintains list of tokens and mapping of tokens to TF
- Issues tokens through DAM to RDOs
- If no token issued (ie. 0) RDO does not transmit data

RDO

- Transmit data from timeframe only if token $\neq 0$

Readout Computers

- On completion of arrival of token from all RDO returns token to GTU