Functional Protocols

Jeff Landgraf

List of the protocols needed

We need to understand the protocols that describe the way the DAQ system works in terms of information flow

- Dependent on the hardware for implementation and constraints
- Independent of the hardware details

Examples in the RDO:

- Data Transfer Protocols
  - Flow Control
  - Error Handling
- Timing Synchronization and readout frames
- Low level timing fiber protocols
  - Firmware upload protocol
  - Configuration upload protocol
- Triggering
- Special run modes (calibration, ”raw” readout for evaluation of compression algorithms or AI/ML, hardware tests)
- Support for 40Mhz / 50Mhz clocks
- Potential analysis requirements in the RDO?

DATA transfer protocol (page 1)

- Low level protocol & packetization
  - My assumption is that we will have a synchronous fiber protocol between the RDO and DAM boards. Even though the data from each time frame will be a variable size, there will be no need for the upload fiber to be involved in the data transfer by, for example confirming arrival, or confirming available space. Is this true of the common FELIX data transfer protocols?
- What are the constraints and appropriate assumptions for aggregation of FEBs to the RDO?
- What is the method of aggregation?
  - The simplest streaming concept is a single file per channel with an array of ADC values, or a stream of zero suppressed hits in (bx, adc) pairs.
  - Aggregation implies additional fields (Detector #, RDO#, feb#, ch#, bx, adc) leading to redundancy
  - Reordering and suppressing repeats removes the redundancy, but breaks the simple streaming concept implying time frames, and other potential groupings.
  - Efficiency of the groupings depends upon occupancy. Can they be system wide? How do we manage them?
  - How much buffering is needed to implement appropriate aggregation?

DATA transfer protocol (page 2)

- How do we handle data overrun?
  - In synchronous transfer the source potentially knows if it will overrun, so how do we act and how do we indicate it?
  - How are such overruns on the FEB side sensed and handled?
- Flow Control
  - If data must be dropped it is may be better to apply global deadtimes. This requires communication down to GTU so it can be sent to all RDOs.
    - How can such a mechanism work?
    - Must it be deterministic, or is it ok to mark packets with dropped packets and act to avoid them in the future?
    - Can we identify when it is appropriate to drop individual packets and when it is appropriate to apply deadtimes?
- Error Handling
  - Do we need out-of-band mechanisms to avoid lockups?

Timing synchronization and readout frames

- Timing fiber is assumed to be 10Gb/s leading to 100bits / BX. One simple (potentially wasteful) scheme is 64 bx bits + 36 control bits.
  - Feedback at reviews and from others that this won't work? Why?
  - Many other schemes (eg: Rev tick + periodic bx identification). What is best?
- Fixed time frames vs configurable time frames fixed for all detectors vs detector configurable time frames?
- Define headers?

Low level timing fiber protocols

- Configuration and firmware protocols presumably fixed (i2c / jtag). Is this accurate?
- How to interlace them?
  - Modes
  - Allocated bits in the BX groupings
- Gpt on FPGA?

Triggering

- How do we support hardware trigger, and what do we even mean by hardware trigger in the context of ePIC?
- Is triggering global, or only for affected detectors? (e.g. dRICH, FB)
- Triggered System (Say STAR)
  - Wait for decision to initiate readout (e.g. STAR TPC)
    - Not appropriate for most EPIC Detectors
  - Store and select logic requires hardware support (e.g. STAR Calorimeters)
    - Not supported in ASICs
    - Would need significant buffering to implement in RDO
- Set busy, release busy, set busy logic would be sufficient for debug/testing/calibration needs
- Hardware signal distributed to DAM or Readout computers could be used to implement hardware initiated software trigger.
- Pure software triggers in DAM or Readout, will require protocols within DAM/Computing levels between nodes
- Other concepts?

Special Run Modes

- These are likely to be defined by user commands in the timing protocol
- May work different than we are used to in triggered experiments
  - Pedestal run's have been analyzed for TOF and in MPGDs
    - Sufficient bandwidth exists
    - Scheme features repeated short periods of full adc readout
  - Raw readout for evaluation of algorithms could work in similar manner
- Need to understand full needs of detectors.

Support for 40Mhz / 50 Mhz clocks or other special needs

- Assuming 40MHz clock referenced to collider clock does RDO have to support the translation?
- Some FEB/ASIC may have special needs (e.g. HG2CROC64 may need a feedback loop from a self-generated trigger.)
- Is there any analysis that needs to be run on the RDO?
  - Pedestal subtraction?
  - Pedestal/Calibration calculations?
  - T0 applications?
  - Common Mode Noise?