(proposed) ePIC DAQ Protocol Specification

Version 0.1

ePIC Electronics and DAQ Working Group

June 19, 2024

1 Introduction

This document describes the requirements for the DAQ fiber protocol connecting the Global Timing Unit (GTU), the Data Aggregation and Manipulation boards (DAM), and the Readout Boards (RDO) boards. The DAM boards are expected to be implemented using Front End exchange (FELIX) boards.

The underlying physical protocol must support the requirements of the DAQ system as a whole. The DAQ protocols must enable the data from all detectors to be synchronised. Data will be organized within time frames and must be tagged to the bunch crossing from which it originated because specific bunch crossing will have distinct polarization states. The functions of the fiber protocol also includes the transfer of the experimental data which will include both collision data and slow controls information related to the status of the readout electronics. In normal operation mode the DAQ system should eliminate global deadtime, however situations are envisioned for which flow control will be required. These situations include testing, component malfunction, or abnormal beam conditions. This requires provisions to monitor the data flow and impose system-wide deadtimes. Additionally the same links used for the above functions must also be used for the configuration and control of both the RDOs and the FEBs connected to them. These features require control commands with deterministic latencies to be applied to identified 10.15ns bunch crossings.

Some ePIC detectors require high resolution timing. The DAQ fiber protocol must provide these detectors with <5ps jitter as well a calibrated phase stability relative to the bunch crossing signal of <5ps. However, not all detectors require the same level of timing performance. The timing resolution can be relaxed to < 100ps for most detectors.

The physical protocol must also support high rate data transfers from the RDO to the DAM boards. The most demanding detectors require line throughput up to 10Gbit/sec. The data will be 8b/10b encoded resulting in 80gbps data throughput.

2 System Overview

Figure 1 show the components of the ePIC DAQ and the distribution of the signals throughout the system.

The overall timing signal is a affected by the timing characteristics of each of the components, which must individually have better time resolution than the system as a whole.

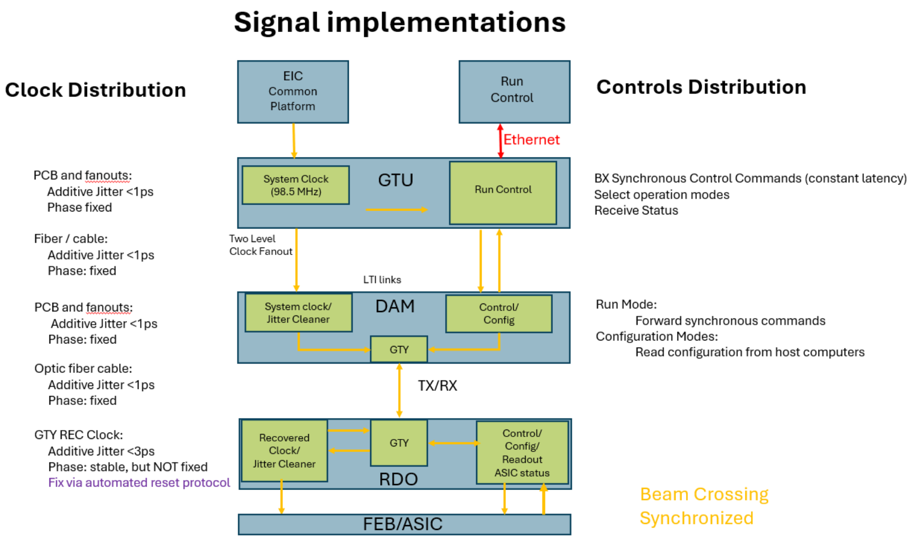


Figure 1: Signal Implementation Overview

2.1 GTU

The RF clock is distributed to the experiments using the EIC Common Platform electronics. The GTU distributes the clock it throughout the ePIC detector. The signals arriving from the EIC include the RF clock and the revtic, a signal representing the first bunch of each beam revolution. The GTU also receives status information from DAM boards. The internal state of the GTU, and external commands from the run control, the DAM status information, and Trigger information combined define the generation of control commands. These commands are distributed, along with the system clock using the FELIX Local Trigger Interface (LTI). The FELIX-155 LTI interface corresponds to 4 bi-directional fiber ports running at rates up to 14 or 25Gb/s. One of these fibers is wired with direct access to the FELIX clock chip. The clock and commands must be fanned out to O(140) DAM boards.

2.2 DAM

The DAM board has dual function of transmitting timing/control information to the RDOs and of receiving data from the RDOs. To first order these are separate, non-interacting tasks.

For the transmission function, the DAM board has the job of fanning the clock and control commands to the RDOs. The input clock from GTU is used to define the frequency of the data transmission signal to avoid the need for a separate clock fiber. This clock is to be reconstructed by the RDO from the data signal.

For the receive function, the data from the RDOs is formatted and sent via PCI to the readout computers (FBDCs). The DAM board will also send status information back to the GTU. This information may contain flags corresponding to internal DAM board conditions (such as buffer status) or to information transmitted from the RDOs and relating to RDO or FEB information. Additionally, for some detectors the firmware in the DAM boards have complicated analysis tasks such as evaluating trigger conditions. In this case the trigger information will need to be returned to the GTU via the status for redistribution to other DAM boards.

The receive and transmission functions do interact in certain modes.

- The DAM boards can be operated in stand-alone mode. In this case they emulate the behavior of the GTU.
- There must be configuration modes in which the transmit fibers are taken over by the DAM boards to transmit firmware and/or register settings to the RDO/FEBs. In these modes a synchronous GTU command will be initiate the configuration mode, but the DAM board will manage to communication to the RDO directly using data from their host computers. Control will be relinquished via status signals.

2.3 RDO

The RDO will reconstruct the high resolution clock to distribute to the FEBs. It also must retrieve, aggregate, and package the ASIC. To this end it must define the time frame structure and produce headers sufficient to convert the ASIC provided times. The maximum time frame length is set to 16 bits. to full EIC bunch crossings identified by 64 bit ids. It is also responsible for downloading firmware and configuration registers to both itself and to FEBs that require firmware. Its expected that the firmware configuration modes will take full use of the fiber, however there will be some limited control parameters that must be handled during running.

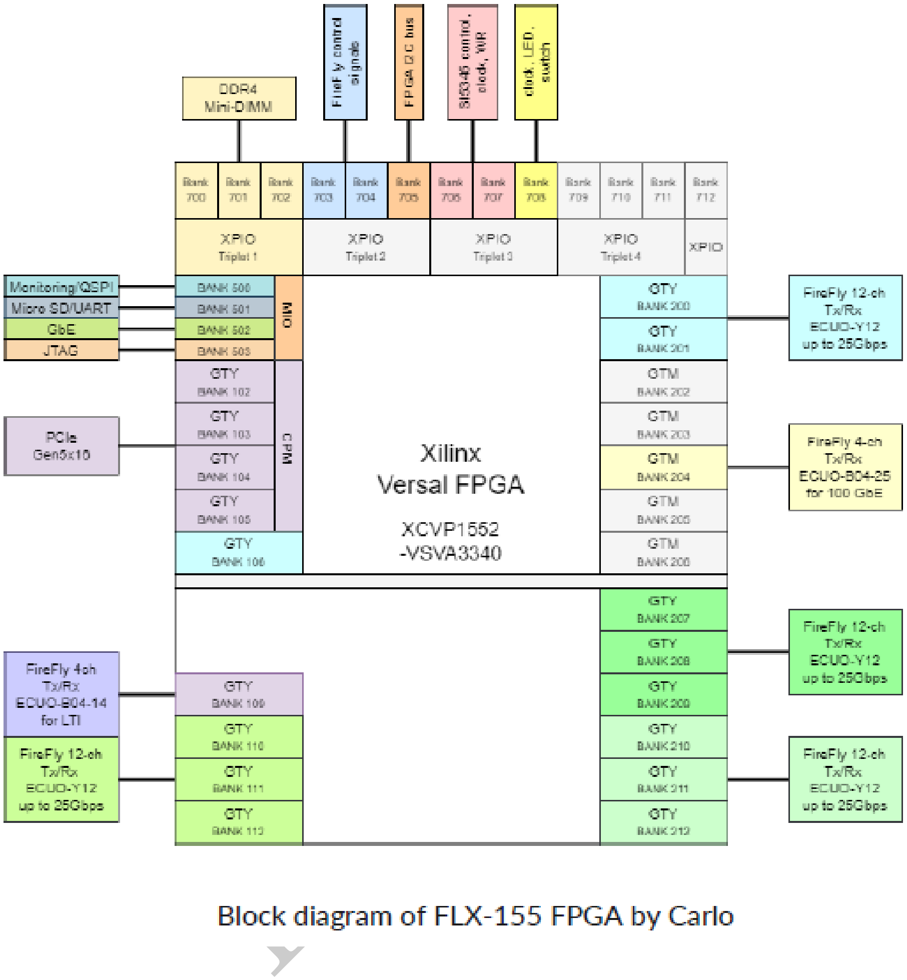


Figure 2: FELIX board to Versal output mapping. The Versal XCVP1552-VSVA3340 contains 68 GTYP transcievers. 16 are used for the PCI interface, 48 for the RDO interface, and are available for the GTU interface. And additional GTM transcievers are used for the 100Gb ethernet connection.

3 Physical Layer

3.1 GTU Fanout scheme

The fanout has not been defined in detail. However it must be able to:

1. Distribute copies of GTU signals to O(140) DAM boards.

2. Distribute copies of a dedicated clock O(140) DAM boards.

3. Independently address (some) commands to up to 32 detectors or groups of detectors

4. Receive in the GTU at least N bits of data status information from a selection of O(140) DAM boards each bunch crossing. This information is assumed to be synchronous in the sense that it each bunch crossing the status information that arrives is for a specific bunch crossing a fixed number of bunch crossings in the past. This delay is assumed to be O(10us) in order to give time for both RDO-DAM data transfer and DAM board processing. N is assumed O(64) bits.

5. Recieve in the GTU at least N bits of status information from each of O(140) DAM boards each bunch crossing. This information is for managing flow control and includes the status of the RDOs under DAM control as well as information about DAM buffering. N is assumed to O(4 bits) per DAM board.

6. Issue fast commands intended for the RDOs. These commands must reach the RDO after a fixed latency after issue by the GTU. This latency is assumed to be O(20 bunch crossings).

3.2 GTU - DAM Signal Implementation

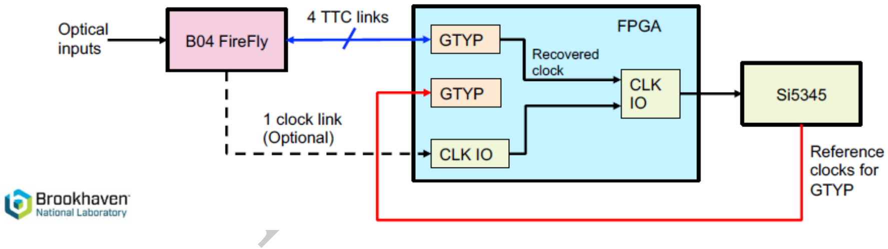
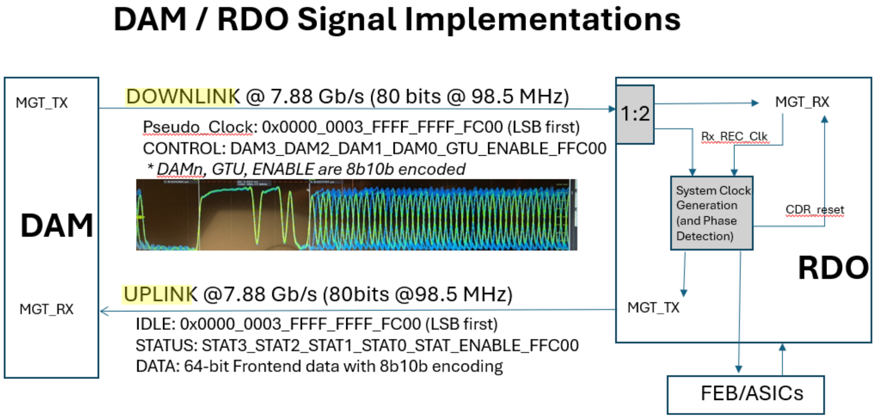


Figure 3: FELIX timing distribution. One fiber can be configured to connect directly through a clock/io chip to the Si5345 timing distribution chip to provide a dedicate timing signal.

The mapping between the Versal XCVP1552 and the Felix board output is shown in figure 2. The 68 GTYP transceivers are each capable of up to 32.75 Gb/s. Up to 4 bi-directional fiber interfaces to the GTU are connected via Samtec ECUO-B04 optical links which can be selected at 14Gb/s or 25Gb/s. Up to 48 fiber bi-directional interaces to the RDO are connected via 4 FireFly ECUO-Y12 transceivers, which can also be selected for 14Gb/s or 25Gb/s.

The timing distribution is shown in figure 3. One of these fibers connected to the GTU can be configured to write directly to the clock/io of the DAM board.

3.3 Signal Implementation


DAM / RDO Signal Implementations
MGT_TX
DAM
MGT_RX
1:2
MGT_RX
Rx_REC_Clk
System Clock Generation (and Phase Detection)
CDR_reset
RDO
MGT_TX
FEB/ASICs

DOWNLINK @ 7.88 Gb/s (80 bits @ 98.5 MHz)

Pseudo_Clock: 0x0000_0003_FFFF_FFFF_FC00 (LSB first)
CONTROL: DAM3_DAM2_DAM1_DAM0_GTU_ENABLE_FFC00
*DAMn, GTU, ENABLE are 8b10b encoded

UPLINK @7.88 Gb/s (80bits @98.5 MHz)

IDLE: 0x0000_0003_FFFF_FFFF_FC00 (LSB first)
STATUS: STAT3_STAT2_STAT1_STAT0_STAT_ENABLE_FFC00
DATA: 64-bit Frontend data with 8b10b encoding

Figure 4: Signals Between DAM/RDO

The DAM/RDO link (assuming 14Gb/s firefly connections) is shown in Figure 4. The links will run at 7.88Gbps (80 bits @98.5 MHz). These links will use a 8b10b encoding in order to carry a multiple of the EIC Clock frequency to be reconstructed at the RDO, allowing for 64 data bits per bunch crossing.

The GTU/DAM link is more flexible. One fiber will be carry a dedicated clock. The 3 remaining fibers can either use a similar 8b/10b encoding as the links to the RDO or run a serial link corresponding to the dedicated clock. Assuming 3 links using the 14Gb/s firefly connection, there could be up to 192-240 bi-directional bits available per bunch crossing. These can be allocated among the DAM_CTRL, DAM_STATUS, TRG_CTRL, TRG_STATUS, and signal paths as needed. If needed, these links could be increased by using 25 Gb/s firefly links between selected RDO and DAM.

4 Signal Paths

Figure 5 defines names for the paths for the information flow within the ePIC DAQ. These signal path definitions, are in some sense, independent of the physical connections as they describe signals according to their information content and semantics rather than their implementation. In particular, the DAM_CTRL, TRG_CTRL, DAM_STATUS and TRG_STATUS may share physical links. Additionally, the communication of the firmware trigger decision is not yet finalized. The RC connection is via ethernet and is assumed to be unsynchronized with beam. The EIC_CLOCK and DAM_CLOCK connections are assumed to be dedicated clock signals. The DAM_CTRL, DAM_STATUS, TRG_CTRL, TRG_STATUS, RDO_DATA, and RDO_CTRL signals are assumed

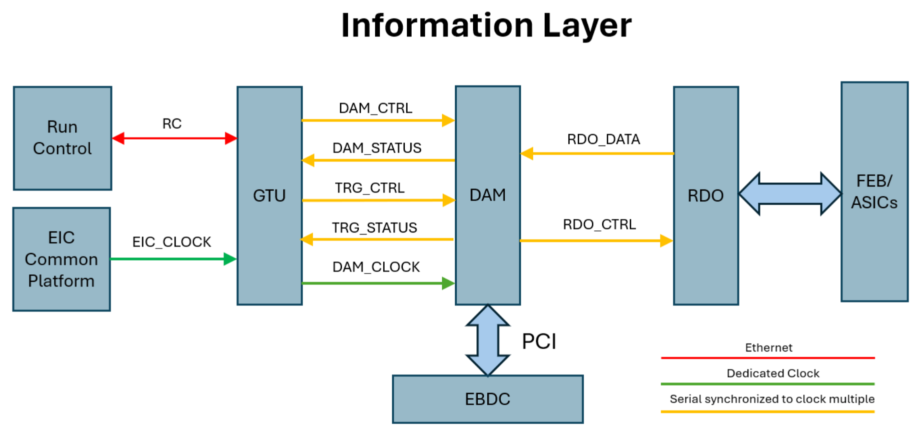


Figure 5: Definition of information signal flow. The yellow paths are expected to be synchronous with bunch crossings. The DAM/RDO links are assumed to carry 64 bits per bunch crossing. The total bits for the GTU/DAM could handle up to 240 bits per bunch.

to be serial signals synchronized to a multiple of the EIC_CLOCK. The RDO_CTRL signal path is expected to use 64 bit packets defined by a common synchronous command structure. The other signal paths, with the exception of the RC path and dedicated clock signals will also use a command structure synchronous with the bunch structure, but they may use a different number of bits. The time between issuing synchronous commands to their arrival and interpretation inside the RDO will be a fixed number of bunch crossings.

5 Feature Requirements

5.1 Bunch Definition

The GTU must give the information necessary to synchronize the bunch counters on every RDO board. This would require at the very least a synchronization command. This might well be combined with a START_RUN command. There are options for how this command might work, a single bit would be sufficient to indicate that the RDO's should start counting if the starting value were assumed (say 1).

The timeframes will have a maximum bunch crossing count of 16 bits. The RDO must also be aware of the identity of each timeframe. 24 bits of time frame identification would mean 40 bits of bunch crossing identification, or about 3 hours of unique bunch crossing identifiers.

It will also prove useful to be able to ensure that all RDOs remain synchronized. This does not have to be done every bunch crossing, but it should be ensured that the synchronization is ensured every time frame. This could be handled either by sending a full 40 bits periodically, as infrequently as .5ms or so. It could also be handled by sending a redundant BCO checksum that cycles through bits. 7 bits could of RBCO (6 bits to identify the bit, and 1 bit to identify the value) could all a full evaluation of the BCO consistency every 128 bunch crossings.

There should also be indicators to the RDO for the situation of the beam. I'd suggest 3 useful bits of information: REV_TIC, HADRON_BUNCH_FILLED, and ELECTRON_BUNCH_FILLED.

Synchronization commands, to force a re-synchronization are likely useful. These commands could come from the RDO in the case of error conditions, but also must be capable of being issued by the GTU.

5.2 Time Frame Handling

The RDO must be directed to start and stop time frames. The time-frame stop can be the same indicator as time-frame begin as there need not be any data without a time frame associated. The time frame identification could be automatically handled by using the high order bits of the BCO. Howeverm, it might prove useful to have another identifier (a token) that could be used to direct data to specific buffers and to define state models of data within the DAM/time frame builders.

5.3 RDO and DAM Data Processing Flags

There should a possibility to indicate to the RDO/DAM what type of data formatting/processing is required. This will be needed to implement time periods when unprocessed data (or different levels of processed data) should be provided. These features are more likely to be handled in the DAM than the RDO, but it would be very useful to have the ability to implement such features within the RDO, to this information should be possible to transmit to the RDO.

5.4 Firmware, Run Control Configuration / Reset

The capability of loading RDO firmware and configuring FEBs is required. Most likely the hardware configuration will be indicated by a command switching to a mode in which the DAM board takes over the RDO link and performs activity defined via the run control or slow control interfaces. A command issued by the DAM/RDO is also needed to inform the GTU it should re-take control.

An important feature will be the need to select specific detectors to run at specific times. We also need to keep in mind the requirement that some detectors must provide continuous operation to provide scalers, while others will turn on and off depending upon the beam conditions.

5.4.1 Fast Configuration

Some detectors have indicated the need to do configuration during a run. This could include calibration control, or error mitigation or recovery. It could be handled by the standard configuration procedure, switching modes for a short time, or by specialized commands.

5.5 Triggering

There are two distinct types of triggering that need to be handled by the ePIC DAQ. The first are events that need to be communicated to the FEB to perform specific actions. The second is firmware based physics event selection. We also want to have hardware inputs to the GTU as fallbacks to implement potential triggers. These should be configurable to provide signals for either of the two trigger schemes.

5.5.1 Firing Hardware Actions/Activities

These could include calibration steps such as firing a laser or pulser system. This might also include requests to read slow controls information. They might also include FEB instructions, for example to write out ADC pedestal data for a short period of time.

5.5.2 Firmware Based Trigger

The firmware based trigger concept is oriented around the dRICH and the low Q2 taggers, although the concept should be general enough to apply to other detectors if needed. The scheme for triggering is defined in section 6.8

5.6 Flow Control

The RDO must finish sending all data corresponding to a time frame within a fixed period, potentially delayed, but corresponding to the timeframe. If it is not able to send all data for the timeperiod in question, either because there is too much data to transfer on the data link, or because internal ASIC buffers have been overrun it must indicate in its headers that the data has been truncated. It's expected that some of the ASIC interfaces to the RDO might have similar possibilities for data truncation errors and these must be flagged as well. If possible, the data truncation flags would be more useful if there is an indication of how much data was dropped.

The DAM board will also have to monitor its own buffers to determine whether it can accept the data volume sent to it.

Any of these data congestion conditions must be indicated by a command to the GTU, which will have the option of applying global deadtimes. We need commands for the transfer of the truncation information between the RDO and DAM board as well as to the GTU, as well as commands defining to imposition of arbitrary global deadtimes.

5.7 DATA Link

The data link is not defined here. It is assumed to consist of headers provided by the RDO, multiplexed raw ASIC data, and Slow controls information provided by the FEBs.

6 Signal Definitions

6.1 DAM / RDO Decoded Synchronous Command Structure

The synchonous signal path between the DAM and RDO is the most constrained in the system. These are implemented using a serial protocol based upon a multiple of the EIC clock. 80 bits are transferred per bunch, but they are 8b/10b encoded to result in 64 data bits transferred per bunch. The initial definition of this command structure (5/9/2024) is shown in table 1. The fixed count of 16 commands bits is likely insufficient for all of the features that the RDO must implement so table 2 shows a schematic solution increase the data content of these commands while maintaining critical beam/time frame information in each bunch crossing.

Decoded Synchronous Command Structure							
[0:7]	[8:15]	[16:23]	[24:31]	[32:39]	[40:47]	[48:55]	[56:64]
Lower 40 bits of master BCO					CMD		Comma

Table 1: DAM/RDO Decoded Synchronous Command Structure. This is the 64 bit command structure discussed on 5/9/2024 DAQ meeting.

Decoded Synchronous Command Structure							
[0:7]	[8:15]	[16:23]	[24:31]	[32:39]	[40:47]	[48:55]	[56:64]
Flexible Command Data Encoding					FAST CMD		Comma
type	type specific				FAST CMD		Comma

Table 2: DAM/RDO Decoded Synchronous Command Structure. This is a potential schematic concept to extend the data content of the original Command Structure to allow both continuous availability of the critical beam related bits and more rare commands. The data in the 40 bits worth of flexible command data encoding remains flexible but must contain enough control bits to select what structure it has. The "type", "type specific" division is an potential holding this flexibility

The preliminary list of potential commands is shown in table 3.

Destination	Link or Command Type	Command Name
GTU → RDO	FAST Command Bit	HAD_REVTIC
GTU → RDO	FAST Command Bit	HAD_FILLED_BUNCH
GTU → RDO	FAST Command Bit	EL_REVTIC
GTU → RDO	FAST Command Bit	EL_FILLED_BUNCH
GTU → RDO	Flexible CMD	SET_FORMAT (N bits)
GTU → RDO	Flexible CMD	RDO trigger (N bits)
GTU → DAM	TRG_CTRL	FIRMWARE_TRG (N bits)
DAM → GTU	TRG_STATUS	TRG_DATA_DESC
GTU → DAM	DAM_CTRL	SEND_CONFIG
GTU → DAM	DAM_CTRL	CONFIG_DONE
GTU → DAM/RDO	DAM_CTRL/Flexible CMD	RUN_START
GTU → DAM	DAM_CTRL	RUN_STOP
GTU → DAM	DAM_CTRL	RUN_TYPE
DAM → RDO	Flexible CMD	RESET
GTU → DAM/RDO	DAM_CTRL/Flexible CMD	RESYNC
GTU → DAM/RDO	DAM_CTRL/Flexible CMD	APPLY_BUSY
GTU → DAM/RDO	DAM_CTRL/Flexible CMD	RELEASE_BUSY
GTU → RDO	FAST Command Bit	TIMEFRAME_START
RDO → DAM/GTU	DAM_DATA	TIMEFRAME_START
RDO → DAM/GTU	DAM_DATA	TIMEFRAME_SENT
DAM → GTU	DAM_STATUS	TIMEFRAME_ERR
DAM → GTU	DAM_STATUS	ASIC_OVERFLOW
DAM → GTU	DAM_STATUS	RDO_OVERFLOW
DAM → GTU	DAM_STATUS	DAM_OVERFLOW
DAM → GTU	DAM_STATUS	BUFF_STATUS
GTU → RDO	Flexible CMD	FAST_CONFIG

Table 3: Synchronous Command List

name	description
HAD_CLOCK	Hadron beam RF Signal
HAD_REVTIC	Hadron beam first bunch of revolution
HAD_FILLED	Hadron beam filled bunch
EL_CLOCK	Electron beam RF Signal
EL_REVTIC	Electron beam first bunch of revolution
EL_FILLED	Electron beam filled bunch

Table 4: EIC_CLOCK signals from EIC Controls Common Platform

name	description
RUN_START	Start a run
RUN_STOP	Stop a run
RUN_TYPE	
SEND_CONFIG	
RESET	
RESYNC	initiate a resynchonization
PAUSE	Apply global busy
RESUME	Release global busy
CALIBRATE	issue a direction to fire detector specific calibration

Table 5: RC signal examples

6.2 EIC_CLOCK Signal Path

The EIC_CLOCK signal path will consist of a small number of dedicated single bit signals defined in table 4. There will be separate signals for the hadron beam and the electron beam. There will certainly be signals defining the first bunch of each revolution and whether a bunch is filled. Other signals may also be defined including beam information such as polarization states for each bunch.

6.3 RC Signal Path

The RC Signals are ethernet commands originating from users. Example commands are given in table 5. RC commands may have complicate parameters, such as filenames to load, or specific definitions of detector specific calibration triggers to execute. It is not necessary for the GTU electronics to handle all of these parameters as run control commands can also be issued to the readout computers controlling each DAM board. The detector specific meaning of a calibrate or configure command can thus be configured at the point of receiver. The role of the GTU is simply to coordinate the synchronous execution of these commands.

6.4 DAM_CLOCK Signal Path

The DAM_CLOCK signal path carries the clock to the DAM board. It's expected that the best timing resolution will use a dedicated clock fiber for this purpose. Note, however, that this is likely a single clock, rather than a full copy of the signals included in the EIC_CLOCK signal path. The other information from the EIC will be merged into the digital DAM_CTRL signal path.

6.5 DAM_CTRL Signal Path

The DAM_CTRL signal path sends commands defined by the decoded synchronous command structure.

The generation of these commands are performed in the GTU firmware by combining the input from the EIC_CLOCK data path, the RC data path, TRG_STATUS, and the DAM_STATUS data paths. The information from the EIC_CLOCK data path is prompt and takes precedence over all other commands. The RC, TRG_STATUS, and DAM_STATUS. Are presented in separate FIFOs which are emptied according the precedence. DAM_STATUS commands relate to Time Frame and flow control processing and are handled first. TRG_STATUS commands are requests to forward trigger information and handled second. RC commands are handled last.

6.6 DAM_STATUS Signal Path

The DAM_STATUS Signal Path allows synchronous feedback from the DAM boards to the GTU in order to implement flow control features. These commands inform the GTU of overflow situations, to allow the imposition of deadtime. They also allow for the communication of buffer usage to prevent overflow situations via imposition of deadtime. The general strategy for flow control is to allow truncation of timeframe data, but to mark it and provide heuristic flow control to reduce or eliminate it.

6.7 TRG_STATUS Signal Path

The TRG_STATUS Signal Path is used for trigger information generated in DAM boards to communicate trigger information to other DAM boards. The simplest scheme for this is to return the trigger information to the GTU via the TRG_STATUS data path. The GTU then issues notifies other DAM boards and their readout computers of the triggered beam crossings via the DAM_CTRL data path. There are several possible choices for the TRG_STATUS signal path as will be discussed in Section 6.8. For now, note that the buffering of streaming data for trigger can occur in either the DAM board or in the readout computer itself. There is sufficient buffering to allow the trigger decisions with latency as high as 100ms-500ms. Although this signal path may share the data path with the synchronous DAM_STATUS path, it could also use a data path that is not synchronous to the EIC clock.

6.8 Firmware based triggering options

Triggering in the ePIC DAQ is not expected to be a feature used by generic detectors, but rather to be used for special cases. No selection criteria are ever applied before the DAM board processing, but there must be a provision to evaluate conditions in several specific DAM boards and communicate the results to another small set of DAM boards.

For some DAM boards for detectors providing the trigger some kind of algorithms will run determining a summary of the information in the detector. The time for the data associated with a bunch crossing to arrive at the DAM board and be processed is assumed to by on the order of or less than 10us. After a fixed latency time O(10us) the summary information will be transferred on the TRG_STATUS data path to the GTU from each of the triggering detectors. This information will be compiled and a trigger decision will be determined. The GTU will distribute this trigger information on the TRG_CTRL data path. The trigger sensitive detector will forward or throw away data for the appropriate crossing according to the trigger result.

6.9 RDO_CTRL Signal Path

The RDO_CTRL Signal path is the sole command path for the RDO, thus FEB and ASIC. It defines the clock, labels bunch crossings, and defines time frames. It also deliver commands to the RDO for distribution to the FEBs. These might include triggers for calibration events. Finally, this signal path also delivers data formatting instructions, and flow control instructions. All of these are generated from the GTU and copied from the RDO_CTRL signal path.

The RDO_CTRL signal path must also handle configurations tasks. For these tasks the SEND_CONFIG command is intercepted by the DAM boards which takes over control to load the configuration data. It responds with CONFIG_DONE command to the GTU when it relinquishes control of the signal path.

6.10 RDO_DATA Signal Path

The RDO_DATA signal path reads out all data. It should be a mix of Synchronous Commands with timeframe information and detector specific data contents.