# Jefferson Lab VXS L1 Trigger Protocol

## Introduction

### Motivation

The purpose of this document is to provide a solution to figure out the answers to the following questions (and more) that are constantly run into on the 12GeV VXS trigger hardware:

1) What is the status of the fiber/VXS SerDes Aurora lanes?
2) Are there any bit errors on the fiber/VXS SerDes links?
3) What is the status of the L1 trigger signal?
4) Was a SYNC signal received, or is trigger data missing from a particular module?
5) Was the L1 trigger information from multiple boards synchronized for the run that was just completed?
6) What is the L1 trigger latency at module X?
7) What does the processed trigger data look like at module X?

Furthermore, functional trigger logic is critical to the success for any experiment. A failure here represents a loss of data that directly impacts the research at Jefferson Lab. Physicists are constantly worried about trigger functionality for this reason and this alone justifies the need for diagnostic capabilities to detect problems.

System downtime is another big component that needs to be minimized. The trigger/DAQ system is a distributed system involving many different modules, crates, cables, and programming. It is very easy for mistakes to be made here especially when a large number of people may have their hands on this system. Diagnostics are needed to quickly identify problems, which can be due to a number of issues, and minimize downtime.

### L1 Trigger System

The Jefferson Lab Fast Electronics group was given the task to implement the new trigger system and core front-end DAQ modules for the 12GeV program. The main motivation for the solution was cost, scalability, and familiarity.

### VXS Based DAQ System

A VXS based solution was settled on primarily because it is an extension to the VME64x backplane (a very widely known and used DAQ card format at Jefferson Lab) and the extensions it provided to VME64x met the requirements needed to implement the new trigger features. From a DAQ point-of view this allows a degree of compatibility between new VXS based cards/crates and VME64x cards/crates which made for a much simpler transition as opposed to switching to an entirely different bus style.

## VXS L1 Trigger Requirements

**Scalable**
**High bandwidth**
**High reliability**
**Low Latency**
**Fixed Latency**

## VXS L1 Trigger Implementation Details

### VXS Interface Signals

The following payload module (P0) VXS signal mapping indicates the required and options signals needed for proper communication with the L1 trigger system:

| Signal Name | VXS-P0 Pair | Description | Required |
|---|---|---|---|
| RX0 | DP1/SWA-RX0 | Aurora Lane 2-RX | Optional |
| TX0 | DP2/SWA-TX0 | Aurora Lane 2-TX | Optional |
| RX1 | DP3/SWA-RX1 | Aurora Lane 3-RX | Optional |
| TX1 | DP4/SWA-TX1 | Aurora Lane 3-TX | Optional |
| RX2 | DP5/SWA-RX2 | Aurora Lane 0-RX | Yes |
| TX2 | DP6/SWA-TX2 | Aurora Lane 0-TX | Yes |
| RX3 | DP7/SWA-RX3 | Aurora Lane 1-RX | Optional |
| TX3 | DP8/SWA-TX3 | Aurora Lane 1-TX | Optional |
| STATUS0 | SE1/SWA | | Optional |
| STATUS1 | SE2/SWA | | Optional |
| SYNC | DP24/SWB-TX0 | Fixed latency SYNC input | Yes |
| CLK | DP25/SWB-RX1 | CLK input | Yes |

### RX [3:0] and TX [3:0]

These signals are multi-gigabit transceiver signals that convey the bulk L1 trigger information. A deviation from the VXS standard is made here by requiring the Aurora Lane number convention to be used rather than the VXS lane number convention. This is important to follow because the current VXS switch cards at Jefferson Lab (CTP and GTP) only have connections to support the first 2 Aurora lanes from each payload port.

Switch cards are capable of sending/receiving at 2.5Gbps and/or 5Gbps rates per Aurora lane.

VXS switch cards provide AC coupling capacitors on all TX and RX gigabit signals, so payload cards can drive the backplane gigabit signals with DC offsets without worry of excessive current draw and allow for optimized receiver common mode biasing.

### CLK

This signal provides a low-jitter reference that is synchronous to the global trigger system. Jitter is sufficiently low for use as a reference clock on the gigabit transceivers, which help with synchronized SerDes links eliminating overhead associated with clock correction sequences. However it is strongly recommended to keep a low-jitter local oscillator onboard in case unforeseen uses arise.

## SYNC

The SYNC signal is used by the L1 trigger and event builders.

The event builder will use this signal to reset its timestamp counter to zero while SYNC = ‘1’. When SYNC = ‘0’ the timestamp counter increments by 1 every CLK period and this value is stored when a readout trigger is received, which provides a consistency check with other modules (i.e. all timestamps should agree between all modules utilizing the VXS signal distribution).

The SYNC signal is used by the L1 trigger and its use will be described below in the protocol section.

## Line Protocol

The Xilinx Aurora line protocol is used at all VXS and fiber optic L1 trigger interfaces. This provides a well-defined, low-latency link between modules. Aurora is used in the full-duplex streaming mode, which allows for uninterrupted streams of data.

## Data Format: Front-end modules (FADC250, DCRB, VSCM, etc.):

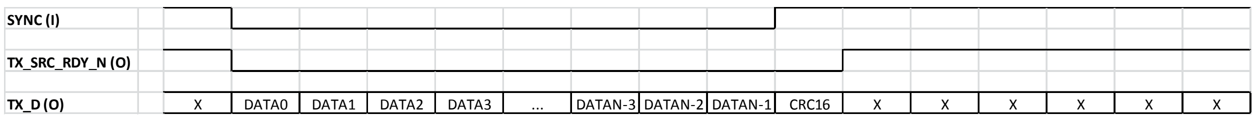
Front-end modules shall use the SYNC signal to determine when to send trigger data, send IDLE words, and when to generate a CRC.

### Trigger data:

SYNC low defines the trigger data sampling window. If SYNC=’0’ then sequential trigger data words are sent to the Aurora interface; therefore, for every clock cycle SYNC = ‘0’ forces Aurora TX_SRC_RDY_N = ‘0’. By this definition if no SYNC has been issued the front-end modules should still send trigger data, which is very helpful in diagnosing whether an issue is related to synchronization issues or related to unreliable links.

### CRC data:

Each aurora lane provides a 16bit data path. A 16bit CRC is computed for each Aurora lane begins with the first data word when SYNC = ‘0’ and includes all data words up to when SYNC = ‘1’. When SYNC transitions from ‘0’ to ‘1’, the 16bit CRC computed value is appended to the data transmission and will be used by the receiving end to verify data integrity of the Aurora lane for this SYNC period. This scheme is shown in the following figure:



### IDLE data:

During the SYNC='1' period, with exception to the one CRC value sent, data transmission is inhibited to the Aurora interface by TX_SRC_RDY_N = '1', which forces the Aurora IDLE sequence to be sent on the SerDes interface.

**Data Format: Trigger modules (CTP, SSP, GTP, etc.):**

Trigger modules act a bit different from the front-end modules because they must receive and CRC the incoming data, and also they must compute a CRC for the outgoing data just like the front-end modules.

For outgoing data things are almost identical to the front-end crates except that instead of beginning transmission and CRC computations in the falling edge of SYNC, they use the RX_SRC_RDY_N signal from their Aurora receiver. In most cases the trigger module will be receiving data from multiple sources; therefore, the received Aurora will need to be buffered until all data streams are ready to be processed coherently (i.e. all DATA0 words from all receivers processed together, all DATA1 words processed together, etc…) which means the transmitted data and CRC computation will be determined when all input channels have ready data.

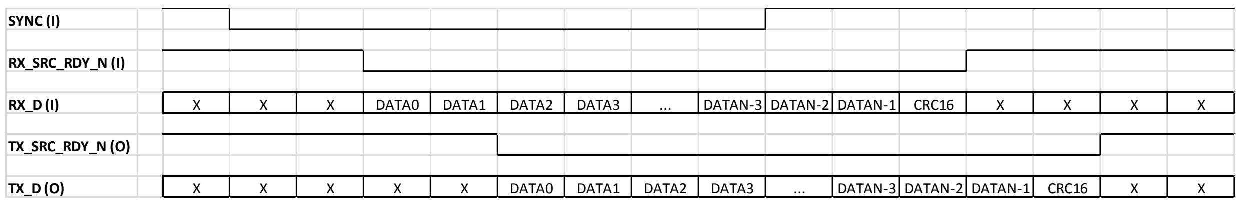
For incoming data, the trigger module must extract on the trigger DATA words and pass them on for processing, but remove the final word from data processing (however this word must go to the CRC computer because it is the CRC value that will confirm if the past data frame is valid or not).

### Trigger data:

RX_SRC_RDY_N = '0' defines the trigger data sampling window (note that this is the simplest case involving just one receiver, additional buffer and coincidence of data ready is required when multi receivers are involved in creating the outgoing trigger data). If RX_SRC_RDY_N ='0' then sequential trigger data words are sent to the Aurora interface; therefore, for every clock cycle RX_SRC_RDY_N = '0' forces Aurora TX_SRC_RDY_N = '0'.

### CRC data:

Each aurora lane provides a 16bit data path. A 16bit CRC is computed for each Aurora lane begins with the first data word when RX_SRC_RDY_N = '0' and includes all data words, except the last – which is the received CRC that must be dropped for outgoing trigger data processing, when RX_SRC_RDY_N = '1'. A16bit CRC computed value is appended to the data transmission and will be used by the receiving end to verify data integrity of the Aurora lane for this SYNC period. This scheme is shown in the following figure:


SYNC (I)
RX_SRC_RDY_N (I)
RX_D (I)
X X X DATA0 DATA1 DATA2 DATA3 … DATAN-3 DATAN-2 DATAN-1 CRC16 X X X X
TX_SRC_RDY_N (O)
TX_D (O)
X X X X X DATA0 DATA1 DATA2 DATA3 … DATAN-3 DATAN-2 DATAN-1 CRC16 X X

**Data Integrity Checking – CRC**

Type: CRC-16-CCITT ($x^{16}+x^{12}+x^{5}+1$)

Reset condition: Initialized to 0xFFFF on Falling edge of SYNC

Each receiver CRC check will compare the current CRC value to the residual and report this status to a register that can be checked during the SYNC='1' period. The CRC will provide a reasonably high degree of accuracy at determining (1 part in $2^{16}$) if a data transmission error has occurred on the L1 trigger Aurora links due to: missing received words, SerDes soft error, SerDes hard error, large SYNC skew, missing SYNC,

**Diagnostic Monitors**

The following list of diagnostic signals is described below and is intended to provide two major functions:

1) Yes/No answer on whether the trigger link has performed error free for a physics run
2) When an error has occurred isolate the source.

Most of these are trivial to include in terms of development effort and resource utilization, but there are more advanced diagnostics that will be mentioned but not deemed as necessary.

**Required Diagnostic/Control Signals**

1) **Aurora lane CRC error flags:**

**Description**

A 1 bit status flag on each Aurora channel that indicates if the CRC computer is valid, or not, for any of the channel's lanes for the past SYNC period.

**Purpose**

- This CRC flag indicates with a very high confidence level that no failure for the past SYNC period on the SerDes lane due to: soft bit errors, unsynchronized links, resetting links

**Resources (per lane)**

~30 LUT

~20 Registers

2) **Aurora lane soft error counter:**

**Description**

A 16bit counter on each Aurora lane that can be read/reset by register access and counts each time a soft error occurs on the Aurora lane.

**Purpose**

- Identify the specific lane having errors due to faulty transmitter, receiver, fiber, copper, or connector failures.
- Quantify degree of problem (certain physics runs may associate a level of confidence of the trigger accuracy as a function of this counter. For

example, a very poor performance may result in a useless trigger and therefore require that run to be trashed, but a low error count may suggest the trigger is suitable enough for data analysis)

**Resources (per lane)**

~20 LUT
~20 Registers

3) **Aurora lane up flags:**

**Description**

A 1bit status flag on each Aurora line that indicates if the Aurora full-duplex lane logic has established a link

**Purpose**

- Identify the specific lane having problems acquiring a link when a channel has failed to start.

**Resources**

Negligible

4) **Aurora channel up flag:**

**Description**

A 1bit status flag on each Aurora channel that indicates if the Aurora global logic has established a link

**Purpose**

- Identify the specific interface having problems acquiring a channel.

**Resources**

Negligible

5) **SerDes RX and TX PLL Locked Status:**

**Description**

1bit RX PLL locked status per lane, 1bit TX PLL locked per TX PLL.

**Purpose**

- Identify any PLL having problems locking due to a poor reference clock.

**Resources**

Negligible

6) **SerDes Reset:**

**Description**

1bit per channel hardware SerDes reset control

**Purpose**

- Provide a hard reset mechanism to SerDes that typically require this when a reference clock is changed.

**Resources**

Negligible

7) **Aurora Reset:**

**Description**

1bit per channel Aurora link reset control

**Purpose**

- Provide a soft reset mechanism to Aurora logic that requires this when the system clock is changed.

**Resources**

Negligible

8) **Input Aurora Link Status:**

**Description**

1bit per channel Aurora link: RX_SRC_RDY (indicates non-idle data patterns are being received)

**Purpose**

- Identify source channel not sending trigger data.

**Resources**

Negligible

9) **Output Aurora Link Status:**

**Description**

1bit per channel Aurora link: TX_SRC_RDY (indicates non-idle data patterns are being transmitted)

**Purpose**

- Identify source channel not sending trigger data.

**Resources**

Negligible

10) **Input Latency Monitor:**

**Description**

16bit per L1 trigger input channel latency measurement from falling edge of SYNC to falling edge of RX_SRC_RDY. Hold counter in reset while SYNC = ‘1’, prevent counter from overflowing at 0xFFFF.

**Purpose**

- Identify latency of trigger input source

**Resources (per channel)**

~20 LUT
~20 Registers

11) **Output Latency Monitor:**

**Description**

16bit per L1 trigger output channel latency measurement from falling edge of SYNC to falling edge of TX_SRC_RDY. Hold counter in reset while SYNC = ‘1’, prevent counter from overflowing by saturating at 0xFFFF.

**Purpose**

- Identify latency of trigger output

**Resources (per channel)**

~20 LUT
~20 Registers

**Recommended Additional Diagnostic Signals**

1) **SerDes pre-emphasis, receiver equalization control**
2) **SerDes PRBS transmitter/receiver checker:** PRBS7 sufficient
3) **SerDes receiver sampling point control**
4) **SerDes raw parallel data output capture**
5) **SerDes loopback controls**
6) **Trigger data waveform capture**

## Streaming Behavior

Trigger data that is buffered in FIFO or similar type logic that will not automatically clear on their own while SYNC='1' must be capable of being cleared by using the falling edge of SYNC. This important behavior is to ensure malfunctioning links don't allow left-over data in the trigger pipeline to be used in the next SYNC frame. The SYNC pulse length will be long enough to ensure the data will flow through the trigger system and result in empty buffered throughout the trigger pipeline under normal conditions.

SYNC is used to establish a common start of all L1 trigger data to ensure data processing of all L1 trigger links happen in a time-coherent manner (i.e. sample 0 from all modules is processed together, sample 1 from all modules is processed together, etc…). The trigger processing should not be dependent on seeing a SYNC pulse in order to initial any data transfer or processor. For example when the system is initially configured a SYNC pulse will not yet have been issued, but the L1 trigger data should still be flowing if SYNC='0' (the data synchronization across modules is lost in this case, but that will be detected by the CRC failure). The reasoning for this behavior is so links are running so we can monitor general link performance.

## Example Implementations

You can contact Benjamin Raydo (braydo@jlab.org) for details on implementation or examples. An implementation can be provided quickly that is specific to your design. This would be the preferred method as it would eliminate duplicate efforts to implement the same thing!