# Specifications for the Micromegas Trigger processor

[huth@physics.harvard.edu](mailto:huth@physics.harvard.edu)

Version 4, 16 July, 2014

**List of functions the component must perform**

1. Receive fiber optic ART signals from ADDC cards - 32 per wedge, 4 wedges total per card
2. Deserialize ART data, convert address into slope coordinates
3. Find patterns associated with viable track candidate
4. Fit slope in theta, and determine eta and phi coordinates of track candidate
5. Cut on projective slope to interaction point
6. Determine region of interest bin from eta and phi coordinate
7. Algorithm must perform task within 70 nsec
8. Range of angles covered 7-30 degrees
9. Integrate over two bunch crossings
10. Find slope difference between segment and projection to interaction point to precision of RMS 2.1 mrad.
11. Cut on slope difference d(theta) at +/- 16 mrad
12. Send d(theta) and ROI to sector logic
13. Track BCID internally using TTC signals, and check for alignment of BCID from ADDC with internal clock.
14. Keep buffer of hit and calculation information to be read out on a L1A
15. Monitor mode that keeps histograms of data such as number of hits per strip
16. Configuration buffer – contains list of dead channels, BCID offsets, generated tracks, road size, cut parameters

Note: Trigger algorithm design and performance can be found in the note An “Algorithm for Micromegas Segment Reconstruction in the Level-1Trigger of the New Small Wheel” by Brian Clark et al., resident in CDS:

<https://cds.cern.ch/record/1706160>

Format of ADDC output

Table . Rising edge data

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **GBTOUT\_RISING[55:0]** | | | |
| calibration | 7 x CALIB\_DATA0[7:0] | | | |
| Hit list | "1010" | BCID[11:0] | ERR\_FLAGS[7:0] | HIT\_LIST[31:0] |
| Hit addresses | HIT\_CNT[3:0] | BCID[11:0] | 8 x VMMIDx[4:0] | |

Table . Falling edge data

|  |  |  |
| --- | --- | --- |
|  | **GBTOUT\_FALLING[55:0]** | |
| calibration | 7 x CALIB\_DATA1[7:0] | |
| Normal run | ARTDATA\_PARITY[7:0] | 8 x ARTDATAx[5:0] |

HIT\_LIST[31:0] = 32-bit list of flags corresponding to each of the 32 VMMs. 0 - no hit, 1 - hit. A register controls if this is a filtered (i.e. 8 hits max) or an un-filtered copy of the VMM flags registered in a particular BC.

HIT\_CNT[3:0] = number of hits (range 0 - 8; 9 - 15 invalid)

ARTDATA\_PARITY[7:0] = parity bit of the ART data computed by each of the 32 ART de-serializer units. Each bit corresponds to one of the ART data field selected by the priority unit.

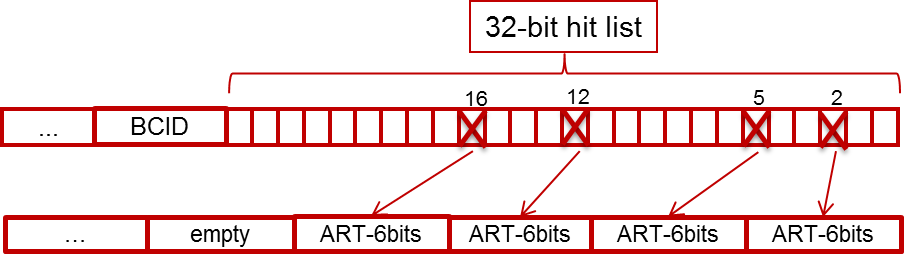


Figure 1 ADDC output format

**List of Interfaces**

Please refer to the diagram below for a description of the interfaces.



Figure 2 Configurations relevant to MM Trigger Processor.

**List of interfaces**

1. Mezzanine cards: fiber optics from ADDC
2. Mezzanine cards: fiber optics to sector logic
3. ATCA backplane: TTC via GBT links from TTC partitions fanned out by FELIX
4. ATCA backplane: SCA monitoring via Shelf Manager GbE
5. ATCA backplane: L1A readout via GBT links on backpland through FELIS
6. ATCA backplane: Configuration management via GbE links from PC farm
7. ATCA backplane: Monitoring information via 10/40 GbE links to PC Farm

**Function details**

1**, Receive fiber optic ART signals from ADDC cards - 32 per wedge, 4 wedges total per card**Receive VTRx signals, bit rate of 4.2 Gbps

ART latency 50 nsec, Xmit latency 150 nsec, of VTRx, transit time 450 nsec

Data received asynchronously

Up to six ART hits per fiber in one transmission

Data field is geographic address header of the ADDC card, number of ART signals, followed by a stream of data that is a 5 bit geographic address of VMM 11 bit address of strip, and 12 bit BCID

* 1. Steps required for receiving ART data
     1. VTTx (or VTRx) lock
  2. Failure modes and recovery
     1. Failure to get transceiver lock

1. **Deserialize ART data, convert address into slope coordinates**

Check geographic address of ADDC card in debug mode for cabling

Form 28 bit words for each ART hit

Convert word into slope coordinate

* 1. Steps required for deserialization and conversion
     1. Form 28 bit words
     2. Use look-up table to convert VMM ASIC address and strip address into a slope coordinate
     3. Compare BCID from ADDC to expected BCID from internal counter (plus delay)
  2. Failure modes and recovery
     1. Geographic address of ADDC card doesn’t match expectation (debug mode)
     2. BCID doesn’t match expected BCID from internal counter

1. **Find patterns associated with viable track candidate**
   1. Steps required
      1. Bin hits into established “slope roads” to determine coincidence for tracks
      2. Handle overlaps between hits in a given road (particularly for u and v planes)
      3. Determine when a hit is removed from the roads (times out)
      4. Determine minimum thresholds of hits per road (potentially dependent on BCID) to go to step 4.
      5. Store decision and other debug output in memory for debug mode (in SDRAM, potential)
   2. Failure modes and recovery
      1. Too many hits and roads created, reach limit in resources: flush the slope roads, flag the event as an error for the offline
      2. BCID received too late (not usable within latency): throw away the hit, flag the event as an error with a hit-specific identifier (if nhits>some number just flag event to not clog offline reporting resources). This may need to be sent to monitoring processes online
2. **Fit slope in theta, and determine eta and phi coordinates (i.e., ROI) of track candidate**
   1. Steps required
      1. Calculate local theta
      2. Average global slopes of hits (up to 4 or 2 for each plane type)
      3. Calculate Cartesian slope values of track
      4. Lookup table for Cartesian slopes to ROI
      5. Store decisions and other debug output in memory for debug mode
   2. Failure modes and recovery
      1. Corruption somewhere in the calculation: it will be discarded on step 5
3. **Cut on projective slope to interaction point**
   1. Steps required
      1. Calculate θ from global slope and local slope values recovered from horizontal strips
      2. Cut events with θ greater than a to-be-determined value
      3. Store events until latency expires
   2. Failure modes and recovery
      1. Timeout longer than 70 nsec – send BUSY to TTC partition
      2. Buffer overflow – send busy to TTC partition
4. **Send trigger segments to Sector Logic** 
   1. Steps required
      1. Construct θ, and ROI locations of each candidate in appropriate format
      2. Synchronize to BC domain and output on required BCID
      3. Serialize for transmission to Sector Logic using Xilinx GTX core
         1. Encode with 8b/10b (VHDL from FELIX)



* 1. Failure modes and recovery
     1. Too many segments for transmission – truncate list to maximum allowed; send exception message.

1. **Track BCID internally using TTC signals, and check for alignment of BCID from ADDC with internal clock.**
   1. Steps required
      1. Initial run in to establish different between specific ADDC’s and fibers as received at mezzanine cards
      2. Establish offsets for configuration database
      3. Receive TTC signals from FELIX partition
      4. Receive ADDC BCID signal
      5. Compare BCID from TTC via FELIX + offset with ADDC BCID
   2. Failure modes and recovery
      1. If BCID comparison is valid, continue
      2. If BCID comparison out of bounds, send exception message.

**TTC**

* The standard encoded TTC signal will arrive to FELIX via a standard TTC fiber and will be decoded by a TTCrq or equivalent FPGA firmware. TTC data will be stuffed, on each BC clock, with fixed latency, directly into *all* output streams with the "TTC" attribute, as follows:
  + 2-bit field: the raw TTC A and B channels. This requires the destination to decode the two serial streams.
  + 4-bit field: L1A, BCR, ECR, system[3] from the 8-bit TTC broadcast packet
  + 8-bit field: L1A, BCR, ECR, system[3..0], user[7] from the 8-bit TTC broadcast packet

The wider fields may be defined as multiple 2-bit or 4-bit E-links at lower bit rates.

* Note that the E-link clock can be 40MHz, but, for example, the 4-bit field can be transferred at 160Mb/s if the receiver generates a ****4 multiple of the 40MHz E-link clock.
* Typically, the reverse direction of the event data E-link can be used for TTC.
* Transparent upgrade to the Phase-II TTC using a mezzanine board in FELIX
* TTC data could also be sent into the LAN.
* The case of a FELIX with only TTC input and only TTC output, i.e. a TTC distributor, is an extreme, but potentially useful, case.

1. **Download simulated ART hits directly from ATCA and perform trigger algorithm steps listed above (2-10).**
   1. Convert simulated tracks generated by Athena to from-ADDC format
   2. Fake data to be received using IPbus FW and loaded into a FIFO…
      1. Fake data format: the word from ADDC on every BC
   3. Steps required
      1. Perform algorithm as in steps 1-5 and repeat
      2. Transmit to sector logic if part of debugging mode, **or** run in standalone mode
   4. Failure modes and recovery
      1. Identical recovery mode to actual running mode
2. **Keep buffer of hit and calculation information to be read out on a L1A**
   1. Steps required
      1. If a valid segment is formed, put d(theta) and ROI, and associated ART in data FIFO (i.e. “Level-1 pipeline”)
      2. On L1A, if L1A has BCID is coincident with data in the FIFO, transfer data to derandomizer FIFO;  
         if BCID is not consistent with buffer, discard data for this BCID
         1. If derandomizer FIFO almost full – send busy to TTC partition
   2. Read data from derandomizer FIFO and format for the ROD; output via an E-link to FELIX
3. **Have a monitoring mode where the entire buffer contents are read out.**
   1. Steps required
      1. Send ART hits to histogram buffer
      2. Send all information on found segments to buffer with d(theta) and ROIs associated
      3. On trigger by PC farm, readout buffer contents, form histograms for online monitoring
   2. Failure modes and recovers
      1. Identical to 5.2 above
      2. Buffer overflow - flush end of histograms as FIFO and indicate overflow flag.
4. **Configuration buffer**
   1. Steps required
      1. Configuration buffer consists of a list of dead channels, BCID offsets, algorithm parameters, must be formatted and sent to PC for download
      2. Download via GbE interface
      3. Get positive download handshake back to PC
   2. Failure modes and recovery
      1. Lack of positive handshake, interrogate existing buffer via PC and reload
5. **Monitoring of temperatures and voltages**
   * 1. To be done by ATCA Shelf Manager using IPMI
6. **Exception handler**

**The following figures should help illuminate the above text**

TTC synch.pdf

Monitor and readout paths.pdf

Debug.pdf