June 27, 2022

Robert Johnson

AESOP-Lite Tracker Logic Interface

This version is for the AESOP-Lite-2 configuration, with 8 tracker modules.

# Command Format

Each command to the Tracker is a sequence of Bytes sent over the UART link, each a minimum of three Bytes in length:

* 1 Byte FPGA address (0 to 7). This is ignored for some commands that go to all layers at once and for commands destined only for use by the master board.
* 1 Byte Command Code (1 to 255).
* 1 Byte for the number of data Bytes (0 to 15).
* Sequence of data Bytes.

The commands are first interpreted by the master Tracker board, the one at the bottom of the daisy chain, which if necessary then forwards the command up the chain until it arrives at the correct board. Some commands should only be directed to the master, which has address 0.

After any command that turns power on, a clock on, or accesses any of the ASIC registers or i2c monitoring chips, the system should wait a millisecond or so before enabling the data acquisition trigger.

# List of commands

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Hex Code | Board  Address? | Number of Data Bytes | Default | Description |
| 1 | No, goes to all boards | 1 |  | Read full tracker event. Read the event data from the most recent trigger. A read command is sent to all ASICs, to transfer the buffered data to the board FPGAs. The master sends the event header to the UART, then sends its hit data to the UART, and then one by one instructs the slave boards to send their hit data to the UART.  The data byte controls the trigger tag used in the read command sent to the ASICs:   1. 0 -> the internally generated tag will be used consistently for the trigger acknowledge and the read event, for normal operation.   (b) ‘000001xx’ -> trigger tag xx will be used, for calibration events only, to make the tag consistent with that in the calibration strobe command. |
| 2 | No, goes to all boards | 3 |  | ASIC calibration strobe. Data Byte description:  1. ASIC address (generally needs to be all ASICs and all FPGAs, so that all chips get the trigger)  2. [7:2] Trigger delay (number clock cycles) [1:0] Trigger tag  3. Specify which FPGA should send the TOT trigger information |
| 3 | Yes | 0 |  | Configuration reset; FPGA reset of all parameters to their defaults. *This must be executed after power-up for all layers in order for the readout to operate correctly.* |
| 4 | Yes | 0 |  | FPGA reset of all state machines and counters, except for the command interpreter state machine. Does NOT reset the configuration parameters. |
| 5 | No, goes to all boards | 1 |  | ASIC hard reset. Data Byte is the ASIC address (0 to 11). The “hard” reset is implemented by sending an external pulse to each ASIC, which guarantees a reset even if the command interpreter is stuck. The effect is the same as a reset command in case the ASIC command interpreter is not stuck (it is not known to get stuck and was designed in principle such that it cannot get stuck). |
| 6 | No, goes to all boards | 2 | 1  1 | Set trigger delay (1st Byte) and stretch (2nd Byte), in clock cycles; note: 2nd Byte is now ignored (always set to 1). This parameter affects just the external trigger signal. |
| 7 | Yes | 0 |  | Read trigger delay (1st Byte) and stretch (2nd Byte) settings from command 6. |
| 8 | No | 0 |  | Turn the ASIC analog power on (is off at startup) |
| 9 | No | 0 |  | Turn the ASIC analog power off for power savings |
| A | Yes | 0 |  | Read FPGA code version |
| B | Yes | 0 |  | Read FPGA code configuration (1 Byte)  Bit 0: ASIC power on or off (see commands 8 and 9)  Bit 1: Trigger logic type (see command 63)  Bit 2: Trigger mask (see command 62)  Bit 3: Trigger enable (see commands 65 and 66)  Bit 4: Trigger end status (see command 5A)  Bit 5: Dual trigger (see command 5B)  Bit 6: Trigger choice for counting (see commands 6E and 6C) |
| C | No, goes to all boards | 1 |  | Send a RESET command (soft reset) to the ASICs   1. ASIC address   In general this soft reset should be redundant and unnecessary if the hard reset has already been used. |
| D | Yes | 0 |  | Dump the RAM buffer to the UART TX, *not normally used; for debugging only* |
| E | Yes | 2 | xFFF | Load the ASIC mask (e.g. xxxx111111111111). This now affects only the TReq (as of V92). |
| F | No | 1 | 8 | Set the number of layers to read out (send to all FPGAs) |
| 10 | Yes | 2 | NA | Load ASIC calibration DAC. Data Bytes:  1. ASIC address  2. DAC contents (range bit followed by 7 voltage setting bits) |
| 11 | Yes | 2 | See below | Load ASIC threshold DAC. Data Bytes:  1. ASIC address  2. DAC contents (range bit followed by 7 voltage setting bits) |
| 12 | No, all are equal | 4 | See  below | Load ASIC configuration register. Data Bytes:  1. ASIC address  2-4. The 19-bit register followed by 5 unused bits (LSBs)  Note: the 5 unused bits must have even parity! |
| 13 | Yes | 9 | See  below | Load ASIC data mask. Data Bytes:  1. ASIC address  2-9. Mask contents |
| 14 | Yes | 9 | See  below | Load ASIC trigger mask. Data Bytes:  1. ASIC address  2-9. Mask contents |
| 15 | Yes | 9 | NA | Load ASIC calibration mask. Data Bytes:  1. ASIC address  2-9. Mask contents |
| 1E | Yes | 0 |  | Return the ASIC mask (2 Bytes) |
| 1F | Yes | 0 |  | Return number of tracker layers in the readout |
| 20 | Yes | 1 |  | Read ASIC calibration DAC. Date Byte is ASIC address. |
| 21 | Yes | 1 |  | Read ASIC threshold DAC. Data Byte is ASIC address. |
| 22 | Yes | 1 |  | Read ASIC configuration register. Data Byte is ASIC address. |
| 23 | Yes | 1 |  | Read ASIC data mask. Data Byte is ASIC address |
| 24 | Yes | 1 |  | Read ASIC trigger mask. Data Byte is ASIC address |
| 25 | Yes | 1 |  | Read ASIC calibration mask. Data Byte is ASIC address |
| 26 | No | 0 |  | Initialize ASICs to the default configuration. NO LONGER USED! DISABLED IN THE VERILOG. |
| 45 | Yes | 4 |  | Load an i2c 16-bit register. This also sets the pointer register for a subsequent register read command. Data Bytes:  1. i2c address  2. register identifier  3. register Byte 1 (Bits 15:8)  4. register Byte 2 (Bits 7:0) |
| 46 | Yes | 1 |  | Read an i2c 16-bit register. The register that will be read is determined by the chip’s pointer register, so that must be set prior to issuing this command. The data byte is the i2c address. |
| 48 | Yes | 0 |  | Read the ina226 interface error code |
| 49 | No | 0 |  | Reset the i2c state machine |
| 52 | Yes, NOT master | 0 |  | Send event data to the master internal RAM (internal use only, for the master board to command the slave boards) |
| 53 | No | 0 |  | Throw away the current event *(internal use only, for the master board to command the slave boards)* |
| 54 | Yes | 0 |  | Return the number of triggers that fill ASIC buffers, 1 Byte |
| 55 | Yes | 0 |  | Return number of ASIC buffer overflows (error condition), 1 Byte |
| 56 | Master only | 1 | 45 | Set maximum clock cycles to wait for the GO to show up, for trigger source 4. |
| 57 | Master only | 0 |  | Inquire whether an event is ready to read out  yes = 59 hex (ASCII Y)  no= 4E hex (ASCII N) |
| 58 | Master only | 0 |  | Return the number of missed GO signals (Trig Source=4), 2 Bytes |
| 59 | Master only | 0 |  | Return the maximum clock cycles to wait for the GO, 1 Byte |
| 5A | Yes | 1 | 0 except for boards 3 and 7 | Set the board to be the end (starting) layer of the internal trigger daisy chain, so that it does not expect a trigger signal from a layer further down the chain.  1 = set as the end layer  0 = not the end layer (expect trigger from another layer)  By default layers 6 and 7 are set to be the ends (i.e. the bottom layers on the bending and non-bending sides) |
| 5B | Master only | 1 | 0 | Set dual trigger mode: 1=on (default), 0=off. If on, then an OR is made of the triggers from the two views (bending and non-bending). Otherwise only the bending view will contribute triggers. |
| 5C | Yes | 0 |  | Return the count of events sent streamed to the master (not relevant for the master board) |
| 60 | Yes | 0 |  | Return the command counter |
| 61 | Yes | 2 | 7  1 | Set the trigger output length (1st Byte) and delay (2nd Byte). The trigger length needs to be big enough to ensure that coincidences of trigger primitives between layers get recognized. The delay value affects only the internal tracker trigger timing, not the trigger primitives. |
| 62 | Yes | 1 | 1 except layers 4 & 6 | Set the trigger mask (1 to include this layer, 0 do not include) |
| 63 | No, goes to all boards | 1 | 0 | Set the internal trigger logic type (0 = AND, 1 = OR). This defines the logic used in each view. An OR is always taken between the two views in the case the dual trigger (command x5B). |
| 64 | No, goes to all boards | 1 | 0 | Global trigger source setting (AESOP-Lite V3)  0 = external trigger  1 = external and internal  2 = external or internal  3 = internal  4 = internal and wait for GO (AESOP-Lite V1) |
| 65 | No, goes to all boards | 0 | 0 | Enable the trigger (not enabled on initialization by default). Note that while the trigger is enabled, commands should not be sent to the slave boards, especially in trigger source mode 4 (see command 64). |
| 66 | No | 0 |  | Disable the trigger |
| 67 | Master only | 0 |  | Trigger the readout by command (for noise measurements only)  Note: this command has NO echo! |
| 68 | Yes | 0 |  | Return the 2-byte trigger count |
| 69 | Master only | 0 |  | Return the number of GO signals received, 2 Bytes |
| 6A | Yes | 0 |  | Return the number of dumped events (no GO received), 2 Bytes |
| 6B | Yes | 0 |  | Return the number of accepted events (read commands) each layer, 2 Bytes |
| 6C | No, all boards | 0 |  | Start counting layer trigger pulses for 1 second.  Note: this command has NO echo! |
| 6D | Yes | 0 |  | Return the 2-byte trigger count from command 6C |
| 6E | No, goes to all boards | 1 | 0 | 0 = count the raw internal trigger OR of all enabled channels  1 = count the trigger going to the previous layer in the chain (for the master this will be the trigger coincidence count) |
| 71 | Yes | 0 |  | Return the trigger output length and delay set by command 61 |
| 73 | Yes | 0 |  | Return the board identifier |
| 74 | Yes | 0 |  | Return the trigger source |
| 75 | Yes | 0 |  | Return the count of missed triggers, 1 Byte |
| 76 | Yes | 0 |  | Return state machine status. Two bytes are returned:  1: {StateEv, StateRg}  2: {StateTg, StateTot} |
| 77 | Yes | 1 |  | Return error count byte. Data byte controls which error count to return.  1: time-outs in state RdEv  2: time-outs in state Reg0 or ACfg or Buss or Dump  3: read event command received with no trigger  4: time-outs in MstrEv or DMP2Ev or SendEv or DMP3Ev or DoneEv  5: overflow error in DataRAMbufferAESOP  6: time-outs in DeadTg  7: time-outs in BusyTg or WtRdTg  8: time-outs in Evcl  9: Bad command address or command code received  10: Error in the data merging |
| 78 | Yes | 0 |  | Return byte of error codes and command in progress, 2 bytes |
| 81 | Yes | 0 |  | Start calibration of the FPGA input time delay |
| 82 | Yes | 0 |  | Reset the FPGA input timing delay to the center value |
| 83 | Yes | 1 |  | Increment (1) or decrement (0) the FPGA input timing delay |

# ASIC Register Defaults

The tracker ASICs have hardwired defaults that are not useful. However, the FPGA firmware has default values for the relevant ASIC registers that can be loaded by the command x26 or by issuing an ASIC reset command (either “hard” or “soft”). The default values are contained in the Verilog code TKRinitial.v, but for them to be implemented it is necessary to compile and load a different mcs file for each board. To be certain of what the defaults are, look at the code, or even better, read them back from the ASICs after telling the FPGA to load the defaults, but here are the current default values:

## Configuration Register

The default setting is 19'b1010101000110111000. This is parsed as follows

* Polarity: non-inverted.
* 1-shot: the trigger output primitive does ***not*** stay high for the duration of the time over threshold.
* Gain: high
* Shaping time: long
* FIFO buffer speed: 3, i.e. the period of the FIFO is 4 clock cycles, or 400 ns.
* FIFO read delay: 3, the read pointer follows the write pointer by 3 cycles of the FIFO clock, or 1200 ns.
* Trigger window: 3 cycles of the FIFO clock, or 1200 ns.
* Output drive: 2, medium-high power used in the LVDS output drivers.
* Maximum number of clusters per event: 10.

## Threshold DAC

* Board A: [20.,20.,20.,20.,20.,20.,20.,20.,20.,20.,20.,20.]
* Board B: [19.,19.,18.,18.,18.,18.,19.,19.,18.,19.,20.,18.]
* Board C: [19.,19.,20.,19.,18.,22.,20.,19.,18.,19.,19.,18.]
* Board D: [19.,19.,19.,19.,19.,17.,19.,18.,18.,19.,19.,19.]
* Board E: [23.,18.,22.,19.,19.,17.,17.,21.,17.,18.,19.,19.]
* Board G: [19.,19.,19.,18.,19.,19.,17.,20.,17.,19.,20.,18.]
* Board H: [19.,19.,18.,21.,20.,19.,20.,20.,19.,18.,20.,19.]
* Board I: [18.,21.,19.,18.,20.,20.,16.,20.,22.,17.,19.,18.]

## Data Masks

By default, all 64 channels in every chip are enabled. If a particular channels if found to be so noisy that it is nearly always on, then it might be worth disabling it. So far no such channels exist.

## Trigger Masks

By default all 64 channels are enabled in the trigger, except for on Board C, currently the board inside the magnet, for which all 64 channels of the first chip and the last chip are disabled from the trigger. That is done because the strips read by those chips are entirely outside of the magnet bore.

# Timeouts

Several timeout counters exist within the FPGA firmware. In general, they were put there for debugging. In practice if a timeout occurs the system probably will not recover without a reboot. For example, failure to get a read event command will result in the ASIC logic getting confused by a mismatch between triggers and event reads (which will set an error bit in the ASIC register that can be seen by reading back the configuration register). Timeouts that occur in the FPGA logic get counted by error counters that can be read back using command x77. In general none of them should ever occur, but here is a list of possibilities:

* Trigger machine waiting for all boards to be ready for another trigger after dumping an event for lack of a GO signal; 224 clock cycles (~1.6 seconds). This timeout has now been removed.
* Trigger machine waiting for a readout to complete before enabling the trigger again; 224 clock cycles (~1.6 seconds). This timeout has now been removed.
* Command interpreter machine, waiting for all ASIC buffers to be cleared for a given board when dumping an event for lack of a GO signal; 216 clock cycles (~6.5 milliseconds).
* Command interpreter machine, waiting for the ASIC initialization to complete; 224 clock cycles (~1.6 seconds).
* Command interpreter machine, individual board waiting for its data readout sequence to complete; 216 clock cycles (~6.5 milliseconds).
* Command interpreter machine, master waiting for the full event readout to complete: 216 clock cycles (~6.5 milliseconds).
* Command interpreter machine, waiting for register readout to complete; 224 clock cycles (~1.6 seconds).
* Command interpreter machine, waiting for the local output buffer to dump to the UART: 224 clock cycles (~1.6 seconds).
* Master event output machine: waiting for the master’s data to flow out to the UART: 224 clock cycles (~1.6 seconds).
* Master event output machine: waiting for another layer’s data to flow out to the UART: 224 clock cycles (~1.6 seconds).

# Output Data Formats

## Trigger Notice for Cosmic Ray Events in the Lab

When the global trigger is set to “internal” trigger mode 3, each trigger will send a single byte 10101011 to the UART. The Python DAQ program then can poll for this byte to detect the trigger occurrence.

## Data Transmission from the Tracker to the Computer

The ASIC data output format is naturally in 6-bit chunks (see below), because of the 64-channels of the chip. So within an ASIC hit list the contents will not align in any way with Byte boundaries.

The data arrive in several packets, one for the event header and then one for each of the boards, starting with the master board.

* 1-Byte length of the header in Bytes (=5).
* 1-Byte start-of-event identifier: “11010011”
* 2-Byte event trigger number (repeats after 65,535 triggers)
* 1-Byte command count (lowest 8 bits of the counter)
* 1-Byte:
  1. 4 LSBs: Number of board hit-list data packets to follow (1 to 8)
  2. 2 MSBs: trigger pattern: bit-7 non-bending side; bit-6 bending side
* Length of the master board hit list, in Bytes.
* Master board ASIC hit list, including 6-bit CRC
* Two bits: ‘11’ followed by zeros if necessary to reach the byte boundary
* Length of the hit list of the next board, in Bytes.
* Next board ASIC hit list, including 6-bit CRC
* Two bits: ‘11’ followed by zeros if necessary to reach the byte boundary
* etc.

Definition of a board ASIC hit list:

* Start bit, is not transmitted.
* Identifier byte: 11100111
* 1-byte FPGA board address
* 12-bit FPGA Header
  1. 7-bit event tag. The two lowest-order bits are the ASIC trigger tag. The other 5 bits are from the clock counter running in the tracker-board FPGA. The counters are synched at 0 when a reset/synch signal is received by the FPGA.
  2. 1-bit error flag (set in case of a trigger tag mismatch)
  3. 4-bit number of chips reporting cluster data
* 12-bit ASIC Header (repeated, following the cluster information, if multiple chips are reporting)
  1. 1-bit cluster overflow
  2. Unused bit, set to 0 to differentiate this from a TOT packet
  3. 4-bit number of clusters
  4. Chip error bit
  5. Parity error bit
  6. 4-bit chip address
* 12-bit Cluster: 12 bits as reported by the ASIC (repeated once for each cluster)
* Next cluster from the same ASIC, if there is one, etc.
* Next ASIC header, if there is another one reporting data, etc.
* 6-bit Trailing CRC

## Monitoring Information Transmission from the Tracker FPGA to the Computer

* 1-Byte length of the transmission in Bytes.
* 1-Byte identifier: “11000111”
* Number of data Bytes being returned + 1 (for the end Byte)
* 2-Byte command count
* 1-Byte FPGA address (+ 8 for the master FPGA)
* 1-Byte code for the type of data being returned, equals the corresponding command code
* Sequence of data Bytes.
* Ending Byte: 0F (to ensure that the last Byte is not zero)

## Register read from an i2c chip on the tracker board

* 1-Byte length, should be 3
* First byte
* Second byte
* ‘\x80’

## Command Echo for Commands that Do Not Return Data

* 1-Byte length of the transmission in Bytes (=4)
* 1-Byte command echo identifier: “11110001”
* 2-Byte command count
* 1-Byte command code received

## Monitoring Information Transmission from Tracker ASIC to the Computer

* 1-Byte length of the transmission in Bytes.
* Raw bit stream from the ASIC, but without the start bit. See the ASIC documentation for the format.

## Data Transfer from the Tracker pCTFE64 ASIC to the FPGA

This internal format is hard-wired into the ASIC design, so it cannot be altered.

1. 6-bit Header:
   1. Start bit (1)
   2. Packet type: 0=data
   3. 2-bit trigger tag (also identifies the event processor that was used).
   4. Error bit. If set, read the configuration register to get the error code. This includes the two buffer-control error bits, but not the parity-error bit.
   5. Parity bit (this parity calculation includes the start bit)
2. 6-bit Number of Clusters (0 to 10) (>32 clusters is not possible for the 64-channel chip, but the number is truncated to a maximum of 10). The highest order bit is set if the data list was truncated.
3. 6-bit Cluster 1, number of strips minus 1 (0 to 63)
4. 6-bit Cluster 1, address of the first strip (0 to 63)
5. 6-bit Cluster 2, number of strips minus 1 (0 to 63)
6. 6-bit Cluster 2, address of the first strip (0 to 63)
7. Etc.

## Trigger Primitive Readout from Calibration Strobe Events

1. 5–bit Start Pattern: the receiving event builder looks for the pattern 1xxxx, where xxxx is the address of the tracker FPGA board connected to the given channel, in order to identify the start of the data stream.
   1. “1” bit
   2. 4-bit FPGA board address
2. 12-bit fixed pattern: 111111100001. This looks like an event header with the event number set to the maximum, no error, and just a single chip reporting data.
3. 12-bit fixed pattern: 010011001111. The next-to-highest-order bit differentiates this from a normal event pattern going from Tracker board to the event builder. Otherwise this looks like an ASIC header with the number of clusters equal to 3, corresponding to the following 3 12-bit words, and the chip number equal to 15.
4. 12-bit start time, giving the time elapse in clock cycles from receipt of the calibration strobe to the arrival of the earliest TReq signal.
5. 12-bit time-over-threshold, in clock cycles, of the OR of all 12 TReq signals.
6. 12-bit hit list, one bit for each chip. 0=no TReq signal was received. 1=a TReq signal was received. The TReq signal has to be at least one clock period in length to be registered, but the delay from the calibration pulse to the start of the TReq is not critical. Therefore, these data should be used for doing threshold scans, instead of the event strip data, meaning that only one channel per chip can be scanned at a time.

### Note on Time-Over-Threshold Information

Time-over-threshold information (TOT) is obtained only from the ASIC trigger output primitive signal and is, therefore, from the OR of all 64 channels. The FPGA has to do the counting, as the ASIC does nothing but output the OR signal. Meaningful TOT information can be obtained only if the 2nd bit of the ASIC configuration register is set. Otherwise the output is taken from after the edge detectors, in which case it is always only 2 or 3 clock cycles.