



Nuclear Physics Division  
Data Acquisition Group

# **Description and Technical Information for Halla SBS Trigger Supervisor (TS) Module**

J. William Gu

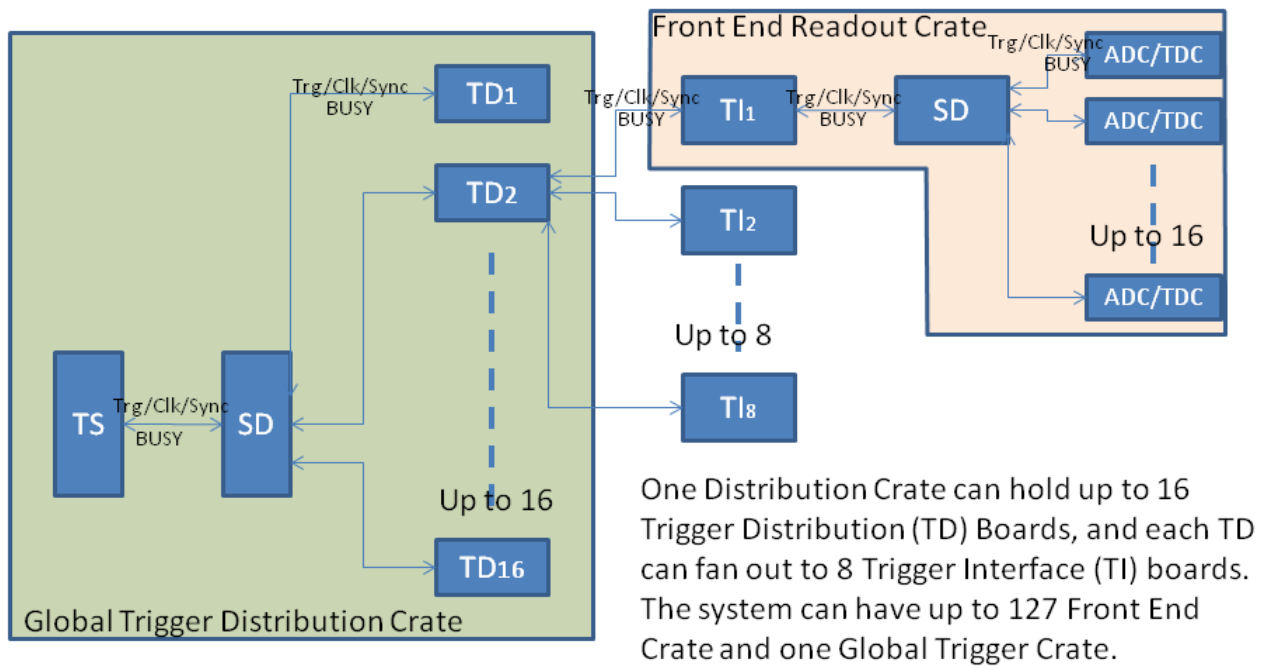
Updated on: May 7, 2015

## Table of Contents

<b>Section</b>	<b>Title</b>	<b>Page</b>
1	Introduction	3
2	Purpose of TS Module	4
3	Function Description	5
4	Specifications	19
5	TS operation	21
6	Programming Requirements	23
7	Front Panel and Backplane Pin outs	54
8	Citation	61
Appendix A	TS Schematics	62
Appendix B	Fabrication Drawing	79
Appendix C	Bill of Materials	81
Appendix D	Data format	82
Appendix D-II	TS partition data format	86
Appendix E	Revision History	89

# 1 Introduction

The Trigger Supervisor (TS) version 4 module is being designed for the Jefferson Lab 12GeV upgrade, mainly for Hall-D [ (Collaboration G. , 2009)] and Hall-B [ (Collaboration C. , 2009)], with other experimental Halls [ (experiments, 1990)] compatibility. This module is the interface between the trigger system and data acquisition system of the experiment. The TS acts as the central control point for data acquisition. It provides a synchronous clock source to the experiment, trigger signal for the data readout, and a sync signal to align the frontend electronics. Together with the Trigger Interface (TI) boards, Trigger Distribution (TD) boards and Signal Distribution (SD) boards, the clock, trigger and sync signals are distributed to the frontend electronics, and the BUSY signals from the frontend electronics are monitored. Figure 1 shows the placement of the TS, TI, TD and SD in the global trigger distribution scheme in experiment setup.



**Figure 1 Trigger distribution scheme**

There is one TS module in the system. The TS receives the trigger from GTP (in global trigger crate) through four copper cables via the VME P2 backplane and a P2 data transition card. The TS sends event readout signal through the distribution system. Depending on the frontend DAQ status, not all the triggers will generate event readout signals. Normally, the TS board is located in the last payload slot (Payload slot#18) in the trigger distribution crate. The trigger/clock/sync signals from TS pass to the Signal Distribution (SD) module in switch slot #B. The SD will fan out the signals to (up to) sixteen Trigger Distribution (TD) modules located in the payload slot #1-16. Each TD can fan out to (up to) eight Trigger Interface (TI) modules in frontend crates. The trigger distribution system can have up to 127 frontend crates, the global trigger crate, and the global distribution crate. The busy signals from frontend electronics can propagate back to the TS in the other direction. Figure 2 shows the crate level diagram.



signals to initiate data acquisition process, taking into account of the front-end electronics status (BUSY for example).

In addition to the trigger generation from GTP and front panel, the TS can also accept calibration triggers by VME command, random triggers, and predetermined number of triggers by VME. The TS supports more than 256 different event types (8 bits). The 8-bit trigger type, together with another three-bit trigger source, will be sent to the front end Trigger Interface (TI) via SD/TD modules. If the front-end electronics buffer is (almost) full (defined as BUSY), or the trigger generation is inhibited, the TS will not generate event readout signals to initiate further data readout, instead, it will start a dead time counter as data acquisition efficiency. The trigger word (16-bit word) will be serialized by the derived 62.5 MHz (250 MHz divided by four) clock.

The TS will supply the experiment with a synchronous 250 MHz clock as system clock, which is sourced either by an on-board oscillator or by an external input through the front panel. The external input could be the synchronous 250 MHz clock from the accelerator. If the accelerator synchronized clock is used, the detector timing measurement (Time of Flight for example) could be easier.

The TS will generate a SYNC signal to align the trigger signals received by the TI boards through their deserializers. The details of the SYNC can be found in the trigger distribution documentation. Using the SYNC signal, and the fiber length measurement, the event readout trigger signals can be aligned to within 4 ns across all the Trigger Interface (TI) boards (or front end crates).

Another purpose of the TS is to control the data acquisition. If the data acquisition cannot keep up with the readout trigger, the data buffer on frontend electronics may overflow. The front end can assert a BUSY signal, and the BUSY will propagate to the TS. The TS can make decision to limit the trigger rate (for data readout), or even stop the trigger completely. If the trigger is inhibited (by asserting the inhibit signal from front panel), the TS will block all the triggers for data acquisition.

There are two optional optic links, which use the same front panel space as clock input and clock output. With these two links, the TS can directly connect to the Trigger Interface board (without SD board and TD board). This is especially useful when integrating with Global Trigger Processor and testing the trigger system. The TS will behave like a “mini” trigger distribution crate. The TS can also drive the P0 connector in pulse mode, so that the TS board behaves like a TI board driving one data acquisition frontend crate.

## **3 Functional Descriptions**

### **3.1 General description**

Figure 3 shows the block diagram of the TS module, indicating the major components used in the design. A Xilinx XC5VFX70T-FG1136 FPGA is used to generate the data readout trigger (from three sources). It also interfaces with VME, serializes the trigger words at 16ns intervals, and distributes to the frontend electronics. It generates the SYNC signals to align the whole experiment. The FPGA may also limit or disable the trigger depending on the front-end status. The clock is distributed by the MC100LVEP111, a multiplexer and fan out clock driver component. The simple fan-out structure will not introduce unnecessary jitter on the clock.

## Halla SBS TS board block diagram

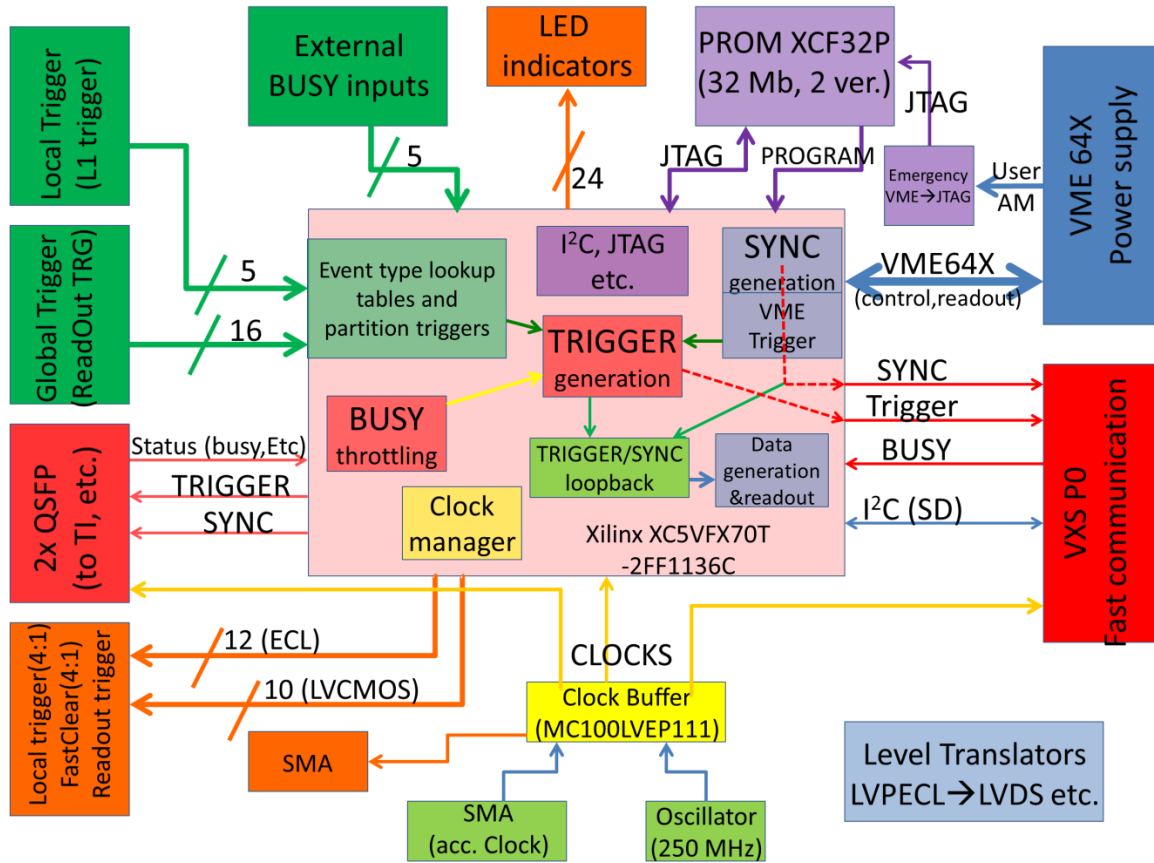


Figure 3 Trigger Supervisor (TS) functional diagram

### 3.2 Fiber links

The AFBR-79Q4Z is a multi-channel (4 Rx, 4 Tx) fiber optic link. It is technologically more advanced than HFBR-7924, and it is cheaper too, as it is widely used in Infiniband applications. For the TS, two modules are put on the front panel. The two QSFP modules give the TS extra flexibility, that is, it can bypass the SD-TD fan outs, and distribute triggers directly to two TI modules in the frontend crates. The TS will serve as a “mini” trigger distribution crate with full Global Trigger Processor (GTP) board inputs. If this feature is not necessary, the two QSFP modules are connected to FPGA MGT transceivers, each can be a four channel full duplex transceivers for data transfer testing. With the potential of MGT, the total data rate can be more than 40 Gbps. (The XC5VFX70T MGT limit is 6 Gbps per channel)

The AFBR-79Q4Z will have Trigger, Clock, Sync as Tx channels, trigger acknowledge and BUSY as Rx channel. Another generic Rx/Tx pair for future expansion (as the extra Xilinx MGT modules in the FPGA allows), or four channel full duplex transceivers for future development.

### 3.3: Clock generation:

One of the TS's major functions is the pipeline clock generation. The clock is generated by its on-board low jitter oscillator (Crystek CCPD-034M-50-250, 250 MHz). The TS can also accept an external clock input through front panel. The external clock input can also give the flexibility of using the accelerator synced clock if we can get one and we need to.

The TS uses a MC100LVEP111 to distribute clocks to the FPGA, two optical links, and SD through P0 connector. The FPGA will generate other frequencies as needed from the 250 MHz clock.

### 3.4 Local Trigger generation (Level one trigger generation):

The Local trigger is generated by the five local trigger inputs. It can be any combination of the five inputs. This is loaded with a 32-bit word (VME register offset 0x6C). For example, whenever input#1 is true, there will be a Local Trigger, the 32-bit will be: 0xAAAAAAAA. A simple rule can be applied to the Local Trigger, so that the same event will not be readout twice. The rule is: there is no more than ONE local trigger in time T. The time T is set register 0x70, bit (6:0) in 4 ns steps. For example, if bit(6:0) is set to 0x12, that means "There will be no more than one local trigger in 72 ns". Because the inputs are NOT synchronized with the TS clock (to preserve the rising edge of the trigger), the trigger rule should cover the width of the input signals.

### Halla SBS Trigger block diagram

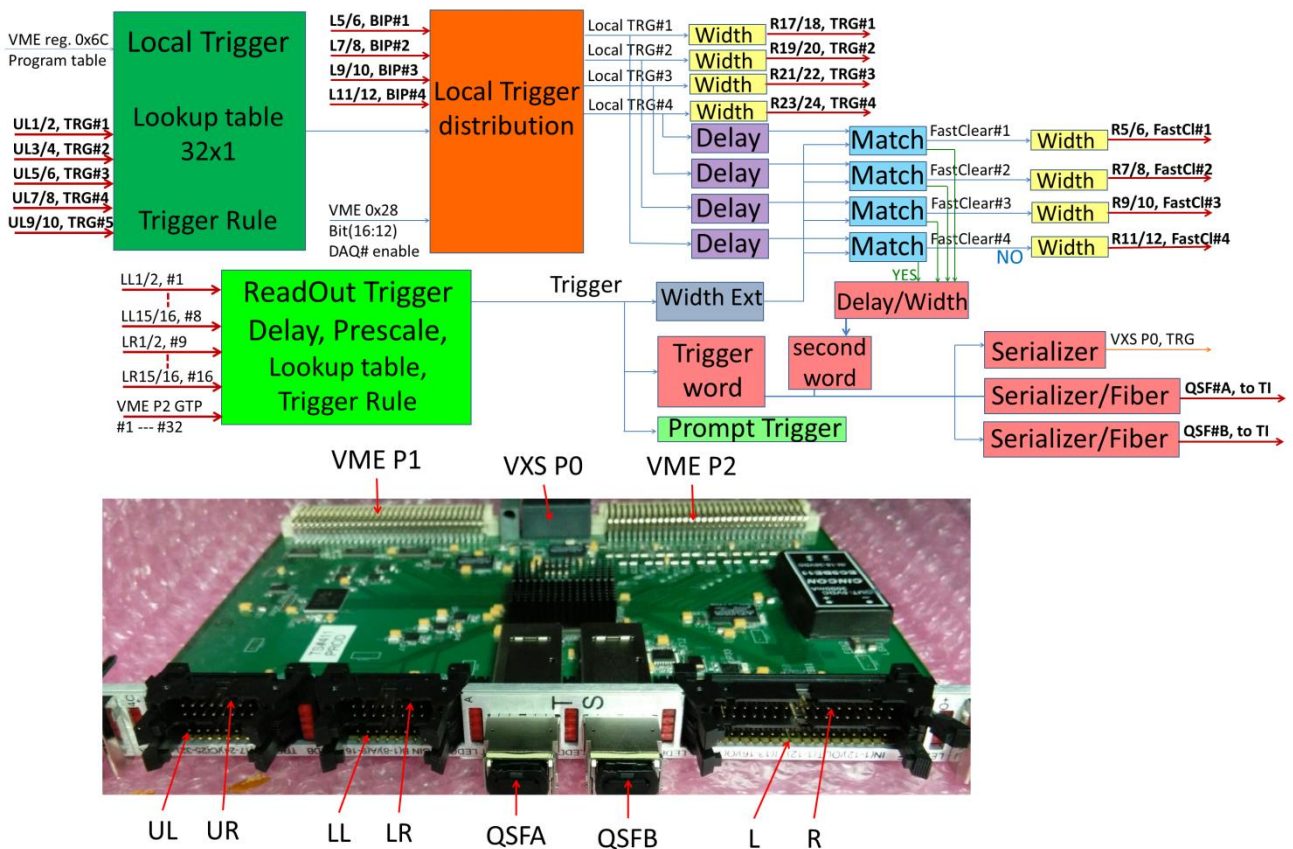


Figure 4 Local trigger distribution for fastbus DAQ duplication

The Local Trigger can be distributed to (up to) four branches in round-robin manner. If one branch is BUSY (CIP, or BIP), it will automatically skip that branch and look for the next non-BUSY branch. This can use the DAQ duplication more efficiently. The register offset 0x28 bit(15-12) is used to set the available DAQ duplication branches. Each branch has its own Local Trigger and Fast Clear, though the Readout trigger is common (see 3.6 for details). For Fast trigger response without DAQ duplication, only one (any one of the four) branch should be enabled.

The Local Trigger signal width can be set using register (vme register offset) 0x70, bit(15:8) in steps of 4 ns. For example, if the bit(15:8)=0x30, the local trigger signal width will be 200 ns.

### 3.5 Fast Clear (abort ADC conversion, or Data Buffering):

The Fast Clear is provided to abort the ADC conversion (Lecroy 1881M) or TDC buffering (Lecroy 1887S) to minimize the dead time. There are four Fast Clear signals corresponding to the four Local Trigger signals for the four duplication branches respectively. The Fast Clear signal is delayed Local trigger vetoed by the Readout trigger (see section 3.5).

The Fast Clear latency is set by VME register offset 0x118 bits(18:10) in 4 ns steps. The Fast Clear signal width can be set by register offset 0x70, bits(23:16) in 4 ns steps. The Fast Clear signal width should be at least 40 ns for LeCroy 1877S TDC, so the bit(23:16) has to be set to 8 or larger.

Upon ReadOut trigger, a window will be open to veto the Fast Clear so that the DAQ boards can save the digitized data and be read out. The veto window size is set by the vme register offset 0x70, bits(31:24) in 4 ns steps.

### 3.6 Readout triggers generation (Event type generation):

The readout trigger generation is the most important function for the TS. The Data Acquisition system treats each trigger as one event, and the pipeline frontend data are grouped by events, or triggers. The trigger system (electronics) generates the level one trigger primitives. The TS matches the level one trigger with the DAQ events. The TS should maximize the DAQ efficiency without messing up the events.

The trigger input signals can be stretched in time to cover the full event. Because of the trigger primitive natural latency uncertainties (detector signal flight time etc.), this signal extension will match the event to the real beam collision. An extra inhibit window can also be set to veto the next event, so no two events will be generated from the same collision. These two parameters can be adjusted independently in ranges from 8ns to 500ns.

The earliest level one trigger primitive timing will be used to tag as the event timing. The tag precision is 4ns. This timing information will be sent as part of the trigger word to the TI, so the TI can generate trigger (decode trigger) in 4ns precision, although the trigger word is transferred every 16ns (on 62.5 MHz clock).

The level one trigger primitives (GTP inputs or external inputs) can be stretched in time and pre-scaled before being used to form a readout trigger type. The trigger type is a look up table with 64 address inputs and two 11 bits event type outputs. This is implemented using the block RAM in the FPGA. The lookup table can be reloaded every time the power is up, or the trigger pattern needs be changed. In standard VME A24D32 access, the block RAM loading takes less than one second. It is possible to reload the lookup tables by an external non-volatile flash memory. The TS can be reconfigured without re-compiling the FPGA.



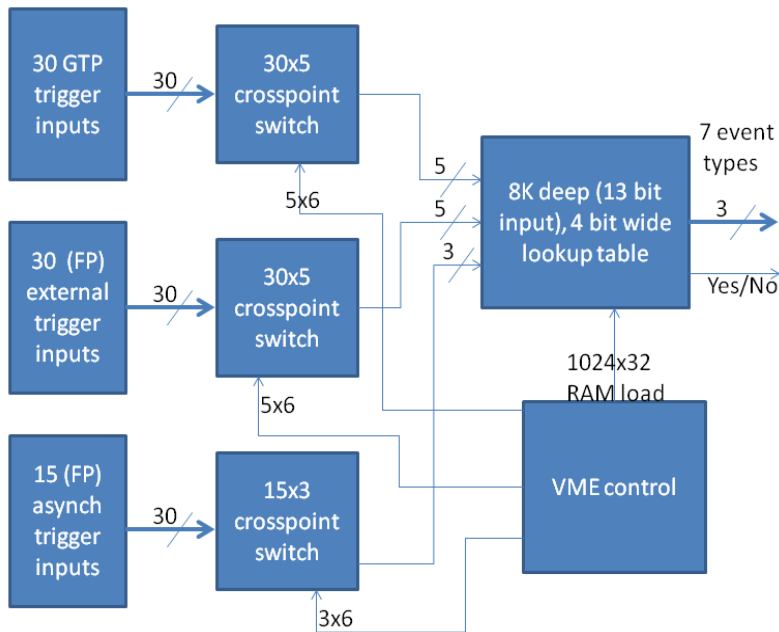
The 16 channel front panel and 32 channel VME P2 backplane (GTP trigger primitive inputs) can be used for Readout trigger generation. In HallA SBS case, the 16 front panel inputs should be sufficient.

The readout trigger collisions happen when two or more event readout triggers are needed to send in the same time window (16 ns), or the front end electronics is busy, or the trigger is inhibited. If the trigger collision happens, the trigger will be lost. The trigger collision condition is sent out. The TS will record it as busy time. Together with the live time, the TS is recording the trigger efficiency, or data acquisition efficiency. When a higher priority trigger is generated and the lower priority trigger is lost, there is no dead time introduced.

### 3.7: TS partitions (Sub-TS)

To facilitate the detector commissioning and sub-detector tests, the TS can also generate triggers as four mini trigger supervisor boards (Sub-TS). The sub-TS works in parallel with the main TS functions. Each sub-TS has its own event type look up table and data stream.

Each sub-TS can have up to 13 level-one trigger inputs. These 13 trigger inputs can generate up to 7 different event types by a lookup table implemented by the FPGA block RAM. The user can choose any five inputs from the 30 GTP level one triggers, any five from the 30 front panel synchronous inputs (external trigger), and any three from the 15 front panel asynchronous trigger inputs. The seven event type is encoded in three bits (code 0 is excluded). Figure 5 shows the diagram of one sub-TS event type generation.



**Figure 5 sub-TS event type generation**

The four sub-TS work in parallel. Trigger word with bits (15:12) = “1011” indicates that the trigger word is a sub-TS trigger strobe. The detailed trigger word is in section 3.8. Each sub-TS event type uses three bits of the lower 12-bit of the sub-TS trigger strobe word. There is no sub-TS trigger timing information, nor trigger content word. The sub-TS can also work together with the normal TS, though the normal TS trigger strobe has higher priority than the sub-TS trigger strobe word.

In this way, the TI can decode both the standard TS trigger strobe word and sub-TS trigger strobe word. The TI needs to know which sub-TS to enable (extra trigger source). Actually, one TI can respond to multiple sub-TS (partition) triggers.

The TS has trigger data for every event, which can be read out by the A32D32 access (and up to 2esst). Each sub-TS has its own event data, which can be read out by VME A24 D32 single word access.

### 3.8: Serial Flash Memory

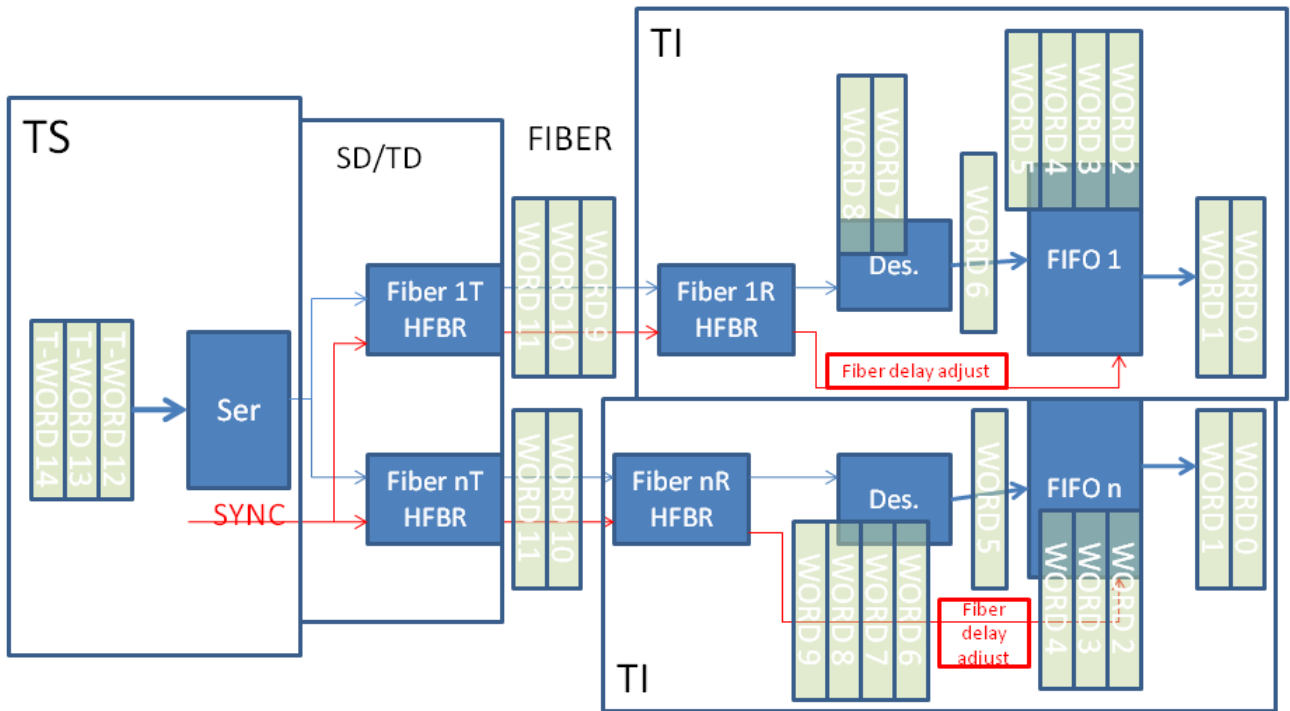
A 4 Mb (at least) serial flash memory can be used to store the trigger lookup tables. When the TS is powered down, the memory retains the trigger table. When the TS is powered up, the board will automatically load the lookup table back to the FPGA to initialize the block RAM (as the block-ram is much faster than the external flash memory). If the trigger table is going to be changed (for example, different trigger modes), the user need load the new trigger table into the flash memory, then send a VME command to load the block-ram by the flash memory. In this case, the TS is still operating in the previous trigger mode while the new trigger mode (lookup table) is being loaded. The lookup table needs less than one second to load through VME (A24D32), there is really little point to use this serial flash memory. (The Serial flash memory read needs 1 second if the clock is 4 MHz). Although the serial flash memory is populated on the TS, but the firmware has not been implemented to use it yet.

### 3.9: Fixed Latency CLKS SYNC

The CLKS SYNC signal is a 250 Mbps serial line that operates in synchronous mode. This serial link allows a 4-bit command to be sent at chosen 4 ns points in time. CLKS SYNC is synchronized to the master clock CLK250 and is sampled every 4ns. The line is considered to be idle when more than 4 samples in a row are read '1'. A command is sent between idle times by sending first a '0' followed by the 4-bit command, LSB first. After the command has been sent, a final '1' is sent so that the line will return to the IDLE state. The encoding portion of this serial protocol is performed on the TS in the FPGA. The CLKS SYNC output from the FPGA can be delayed (time adjustment) in the precision of 80 ps by the FPGA IODELAY. As the clocks are synchronized and sourced by the TS, this delay setting is not expected to change between different firmware versions. The TS will send the CLKS SYNC to the TD boards via P0 and SD. The TD will fan out to the TI boards via optical transceivers. The CLKS SYNC is Manchester encoded to balance the differential lines.

The CLKS SYNC is synchronized across the TI boards by applying different delays on the TI boards. The delays are determined by the fiber latency measurement. The CLKS SYNC link is used in conjunction with a synchronous FIFO to enforce a fixed latency on the serial trigger link. On the TI board, the trigger word is clocked into a FIFO using the FPGA built in MGT transceivers and clocked out of the FIFO using a 62.5MHz clock derived from (and in phase with) the 250 MHz system clock. At startup the FIFO is reset (0 words) and reading the FIFO is disabled. No words are written into the FIFO since the TS is not yet transmitting data words on the trigger link (i.e. received data valid signal is not asserted). Acceptance of triggers by the TS is also disabled. The serial trigger link is idle words only. On trigger start, the TS starts transmission (trigger words and/or timing words) on the trigger link. The TI will write the deserialized data (valid data, that is non-idle data word) to the FIFO. After some delay (VME register controlled) from the trigger start, the TS issues a TRG\_START command on the CLKS SYNC line. When TI receives the TRG\_START from CLKS SYNC line, the TI resets the trigger FIFO readout address, and enables continuous readout of the FIFO. As the CLKS SYNC lines are fiber length adjusted and the 62.5MHz clocks are phase aligned, the trigger words from the TI board FIFO are

synchronized across the system. In TS and TI MGT transceivers, the phase alignment is used to remove the elastic buffer delays as the Tx/Rx modules are using the same clock (or the clocks generated from the same clock). The trigger word alignment is illustrated in figure 6:



**Figure 6 Illustration of the Global Trigger Synchronization process**

The CLKSINK link is also used to synchronize the lower frequency clocks derived in the TI from the distributed system clock (CLK250). At startup, the TS issues a CLKSINK command CLKRESET, which is phase aligned with the 62.5MHz clock on TS. The CLKRESET resets the clock distribution chip (AD9510) on the same CLK250 edge, assuring that the lower frequency clocks (125 MHz, 62.5 MHz, 41.67 MHz and 31.25 MHz) are in phase across all the TI boards, and in phase with the 62.5 MHz clock on the TS board. This command is sent before the TI sync command. The delay between them is determined by the maximum MGT module reset recovery time, as the MGT clock is changed (AD9510 clock reset). The time could be several milliseconds.

### 3.10: Encoded Trigger Word

A 1.25Gbps serial link operating over the fiber is used to distribute the 16bit trigger word every 16ns. There is also a link going in the opposite direction, allowing status words to be sent back to the global trigger distribution crate. The 16bit trigger words are decoded as follows (TS->TD->TI flow):

Trigger strobe word – generated by the TS in response to the acceptance of a level 1 trigger as a DAQ event. Upon receipt, the TS encodes this word with highest priority. The TS transmits this word with fixed latency relative to the accepted trigger. Two-bit timing information is added in the word to distinguish which

quadrant of the 16 ns period the trigger is generated, so it is fully compatible with the 4ns pipeline design architecture. The TI distributes trigger signal in 4ns precision.

Trigger content word – additional information about this trigger event for use by the ROCs. It is queued in a FIFO and sent in any frame not used by a trigger strobe word. The TI matches the trigger strobe and trigger content words by the order of their reception. The Trigger content words could include an event counter. The TI can use this content word to check its synchronization if any trigger is missing by the TI/TD. The TI will send trigger acknowledge back to the TD for missed trigger check, so this check seems redundant. Right now, the trigger content word has not been used yet.

Control Word – some commands. They can be queued in a FIFO and sent in any frame not used by a trigger strobe word. Right now, this is connected to a VME register directly. The decoding is depending on the TI. In current firmware, the control word can be used to generate a single VME trigger.

Master Time Word – since the link must always be transmitting valid data (non-idle data) to keep its latency fixed, bits [13:2] of the TS time is transmitted whenever no other word types are available. Bit(1:0) is transmitted in trigger strobe words. By continuously receiving bits [13:2] of the TS time, each TI can promptly detect if its subsystem has lost global synchronization (i.e. compare the global time with its own time). Otherwise, the loss of synchronization could only be detected at the event building stage. The continuous transmission of ‘known’ (i.e. predictable) data also allows one to monitor the integrity of the link. The necessity to implement the checking depends on the reliability of the link.

This table shows the format of the trigger word

Bit15	Bit14	Bit13	Bit12	Meaning of Bit[11:0]				Word Type
0	1	0	0	Lower 12 bits of TS time (bit(13:2))				Time
0	1	0	1	Control or Command				Control
0	1	1	0	TI legacy trigger strobe				(TS) VME Trigger Strobe
				Tquad(1:0)	trigger type (Trg2, Trg1) Bit(9:8) Event Type Bit(7:2) source (random, regular ...) Bit(1:0)			
0	1	1	1	Additional Trigger data				Trigger Content
				DataAV(4:1)	Number of Local Trigger (7:0)			For duplication
1	0	0	1	Tquad(1:0)	Trigger type A, (GTP major)			Trigger Strobe A
1	0	1	0	Tquad(1:0)	Trigger Type B, (EXT major)			Trigger Strobe B
1	0	1	1	subTS#4	subTS#3	subTS#2	subTS#1	

Trigger Type A (or Trigger Type B): Bit(9:8):

00: not defined yet

01: Trigger1 will be generated by TI;

10: trigger2 will be generated by TI;

11: Sync\_event, and trigger1 will be generated by TI too. either periodic sync event or forced sync event

The trigger format can be easily changed as long as the TS encoding and TI decoding are consistent.

### 3.11 VME interface

The TS is a VXS payload board. It is compatible with VME64x backplane. Normally, it is a VME slave board, with interrupt capability. The TS is implemented the same way as the TI/TD board.

For simplicity, three kinds of VME address modifier codes are implemented. (1), User defined address modifier. This is similar to the A24 address modifier. It is used to load the PROM by the on-board discrete components. (2), Standard A24 address modifier. This is used for registers read/write on the FPGA, slow controls of the TS peripherals, and Sub-TS data readout. (3), A32 data transfer. This is used to transfer data to the ROC (Read Out Controller). For the details of A32 mode support, refer to the VME interface design (Ed, 2010).

### 3.12 The Xilinx PROM programming.

The Xilinx XCF32P PROM is used to program the FPGA. It can save one version of the (XC5VFX70T) FPGA firmware when it is used in non-compression mode. It can save two versions of the firmware when bit stream compression is used. It can even save four versions of the firmware when bit stream compression is used and the FPGA resource usage is not too high. The PROM is programmed using VME with emergency logic decoding (that is: discrete logic VME to JTAG engine). It can be addressed in the VME64x crates by its geographical address. If it is in the crate without geographical address, only one TS or one TD in the crate should be addressed as geographical slot#0. To avoid conflict with other VME addressing, the user-defined address modifiers are used for the PROM loading. The emergency logic supports A24D32 user defined address modifier codes: 0x19, 0x1A, 0x1D and 0x1E (Similar to 0x39, 0x3A, 0x3D and 0x3E). Out of the data, bit[1] is used for TDI, bit[0] is used for TMS, and all the other 30 bits are unused. This approach is fully tested on TI/TD boards. This emergency loading provides VME remote firmware loading (and re-loading) even if the FPGA is not working. One 33 MHz on-board oscillator is used to program the FPGA, and used by the FPGA for slow control. The PROM needs be in master mode if bit compression is used.

The PROM can also be programmed by the on-board JTAG connector and VME-JTAG engine (after the FPGA is programmed and working) implemented in the FPGA. The JTAG engine in the FPGA provides VME remote firmware loading with more efficient VME data transfer, but the FPGA has to be working.

### 3.13: Serial data communication with SD and other switch slot modules

As there is no VME access to the switch slots (SD for example), the TS has implemented an I<sup>2</sup>C bus to each switch slot. The TS will act as I<sup>2</sup>C master, and the switch slot module acts as slave. The I<sup>2</sup>C works in FAST mode with higher clock frequency. For details, refer to the VME to I2C implementation [ (GU, 2010)] documented earlier.

The TS can communicate with the switch slot module via a 250Mbps link, (it is possible to increase it to 500 Mbps using DDR techniques). This is implemented using the Xilinx SelectIO standard differential IO pin pair. The switch slot module can also send data to TS using this link. The direction of the data link depends on the firmware (TS and SD) implementation. This can serve as a complimentary to the slow I2C interface as one possible usage. Right now, this is implemented as SD to TS one way data transfer. Using this link, the SD can update a block memory on TS directly. [ (GU, 2013)]

### 3.14: Trigger data receivers

The Texas Instruments SN651vdt100 (Micrel SY55855 on TS prototype boards) LVDS receiver is used to receive the GTP data from P2 backplane. The receiver output (LVDS) goes directly to the FPGA for processing.

The Maxim max9602 differential discriminator chip is used to receive the trigger input from the front panel (external trigger in or generic trigger input). The max9602 uses -5 as VEE, +5 as VCC, and +2.5V as VOUT. It can receive any differential input with common voltage of -2V to +3V (which includes ECL, LVDS, LVPECL ...), and the LVPECL25 output is compatible with the extended LVDS input of the FPGA.

### 3.15: Readout Synchronization

In addition to the event number and trigger time stamp, the readout can be forced to get synchronized by synchronization events (SyncEvent). There are three ways to generate the SyncEvent.

First, the SyncEvent can be generated periodically by TS (or TImaster). The last event of the every N blocks is generated as the SyncEvent. (for now, the block level, or number of event per block, is up to 255) The N is any value between 1 and 65535, which is set and enabled by A24 register offset 0xD4. When the periodic SyncEvent is marked, the original event type is kept.

Second, the SyncEvent can be forced by VME command to the TS (or TImaster). This event may be any event in the trigger block. It is an added event with event type “00000000”. The forced SyncEvent can be generated by VME A24 register offset 0x100, bit#20.

Third, the SyncEvent can also be generated by a front panel signal input.

The TS (or TImaster) will assert a short BUSY (3us) after generating the SyncEvent, and stop further triggers. After receiving the SyncEvent, the TI will generate BUSY, and set the SyncEvent marker in the ROC polling register A24 offset 0x34. The BUSY of TI will propagate to the TS through fibers and P0 backplane. The latency of this should be less than 3us, so that it will overlap with the short BUSY on TS. After the front end data is cleared, and the Readout acknowledge will clear the BUSY on the TI. In pipeline readout mode, If there are more than one trigger blocks to be read out, the very last acknowledgement clears the SyncEvent BUSY on the TI. After all the TI boards clear their BUSY, the TS will distribute trigger again. Depending on the data acquisition mode and the amount of data backed up on the frontend, this process may take tens of microseconds to many milliseconds.

Upon SyncEvent, the trigger distribution will be paused. Users can change the DAQ settings during this time. If no reset in this time period, the event number and trigger time will be continued.

Figure 7 is the function diagram of the TS FPGA, which includes the detailed diagrams in this section (like, figure 4, 5, etc.)

TS FPGA Block diagram

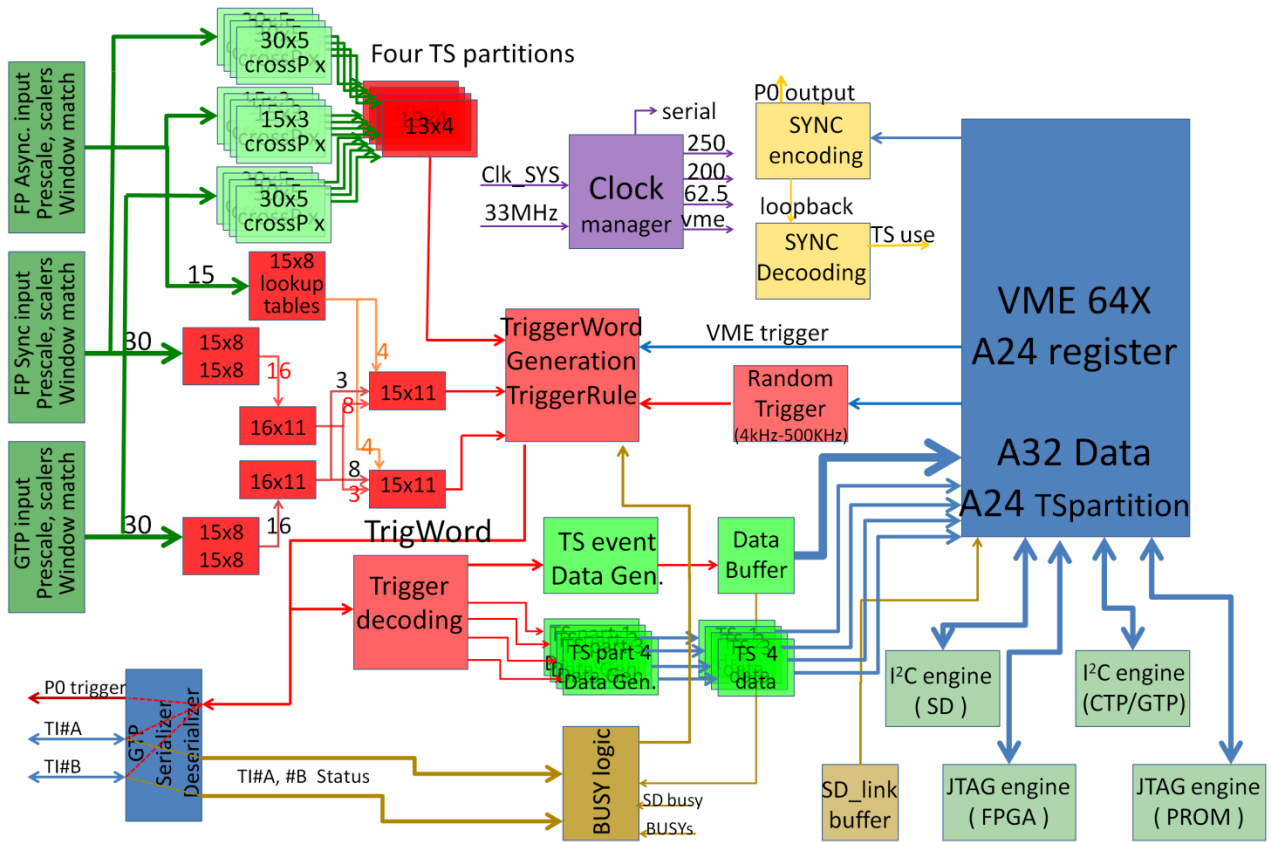


Figure 7 FPGA function diagram

### 3.16: Simplified event type look up tables

The original design is using ~90% of the FPGA block RAM resources. Taking into account the real TS application, the event type lookup table was simplified to save FPGA resources without sacrificing the flexibility. The following figures shows the new lookup table structure.

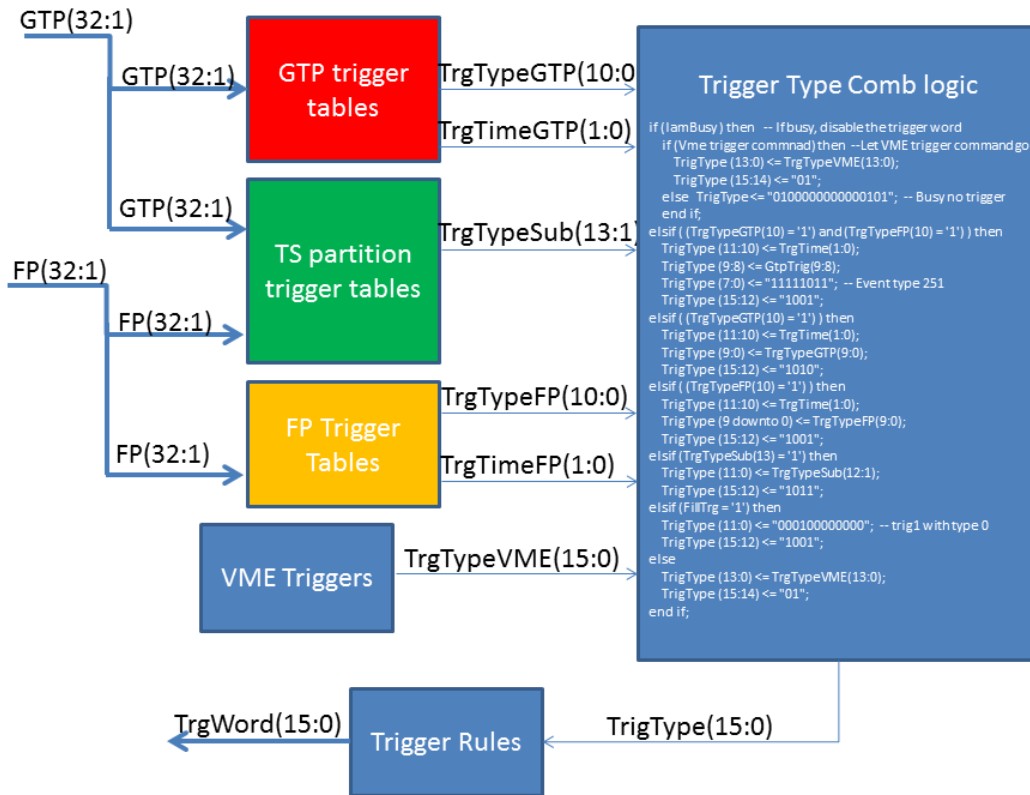


Figure 8 Overall design of the simplified event type lookup table

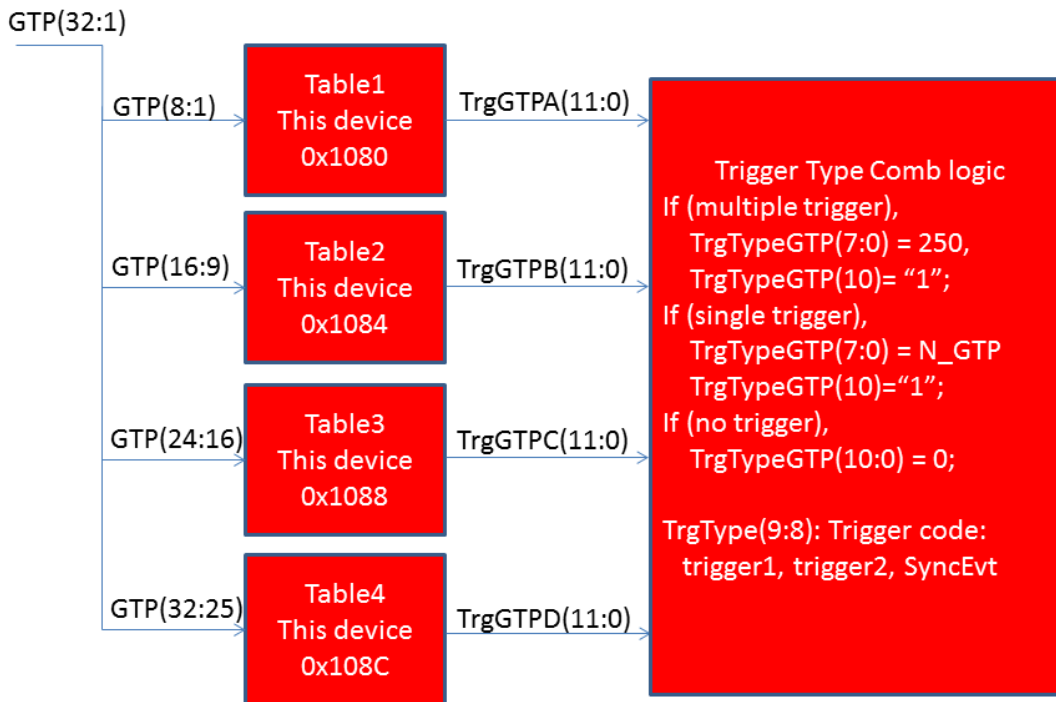
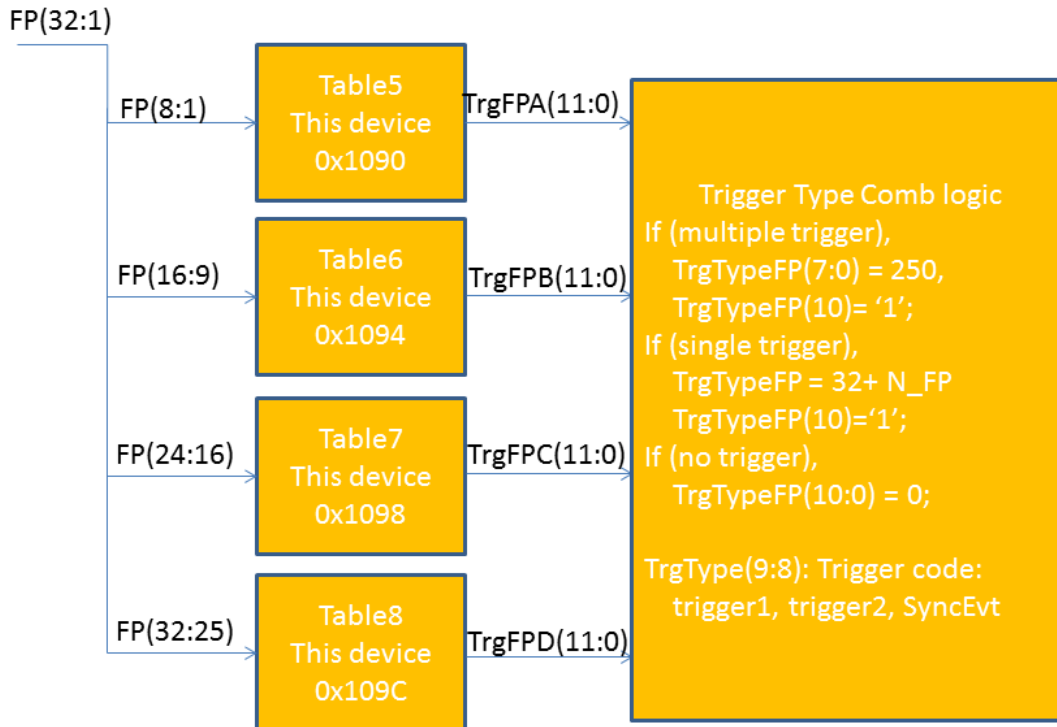
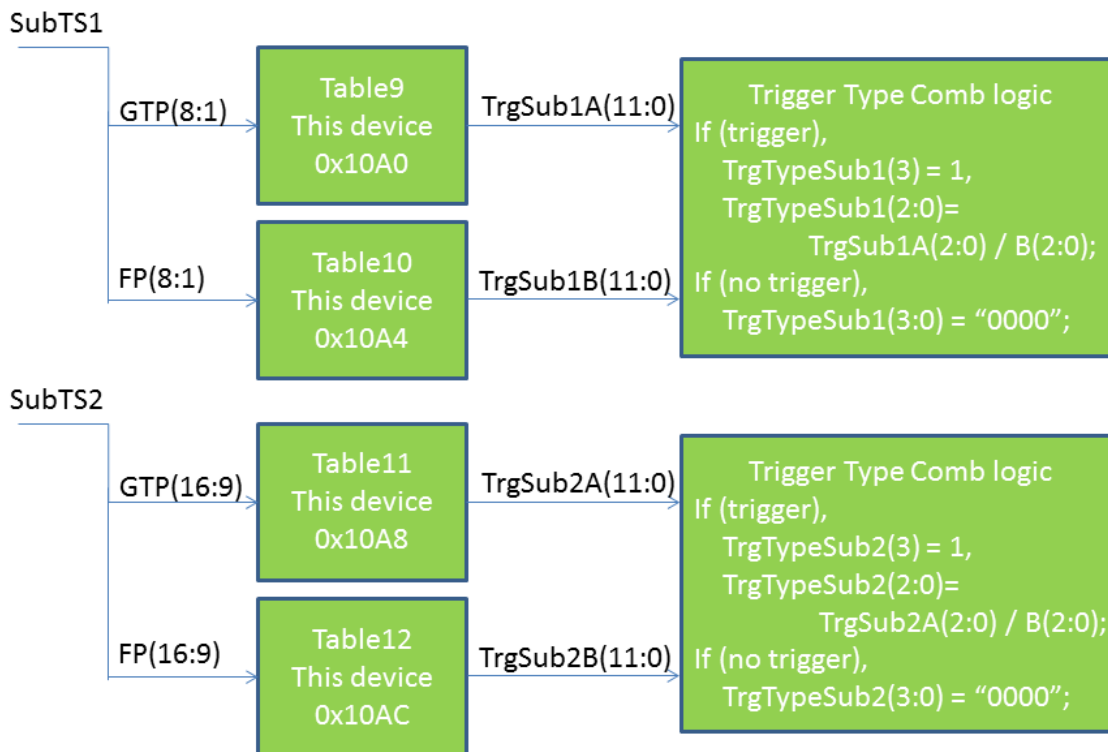


Figure 9 The GTP input (from VME P2) event type look up table

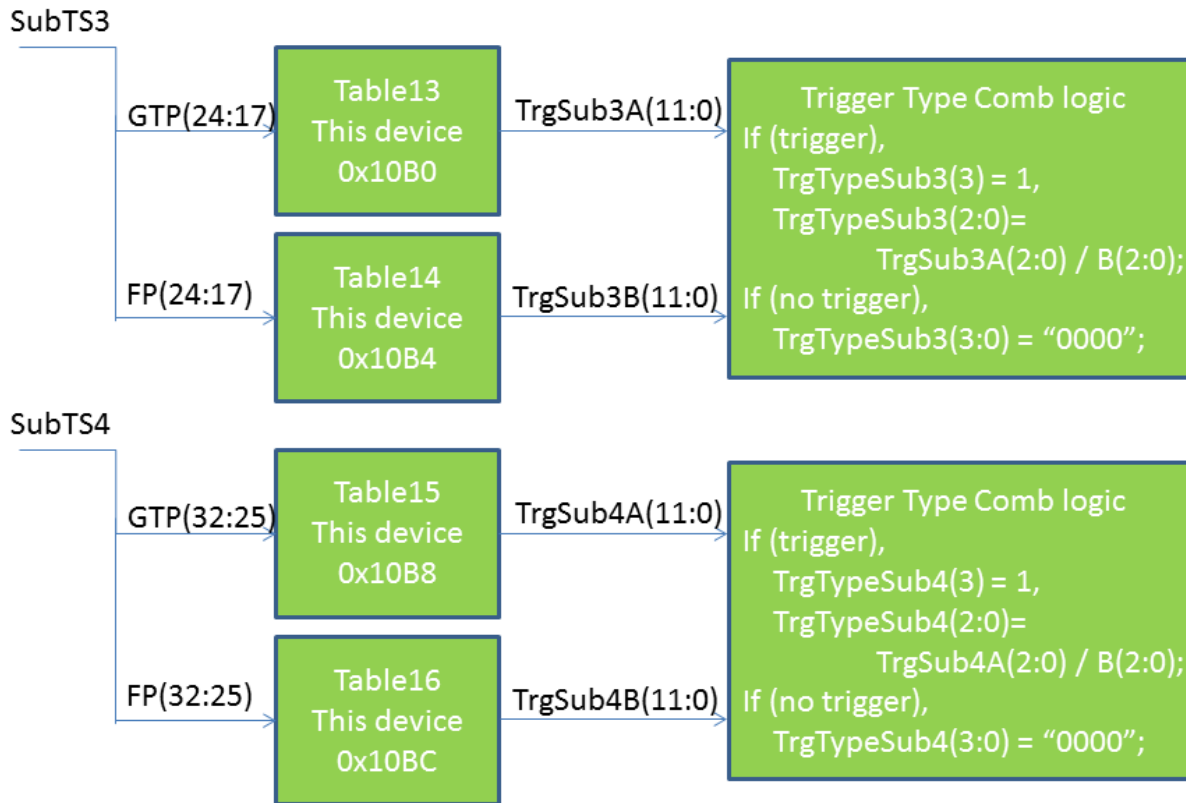




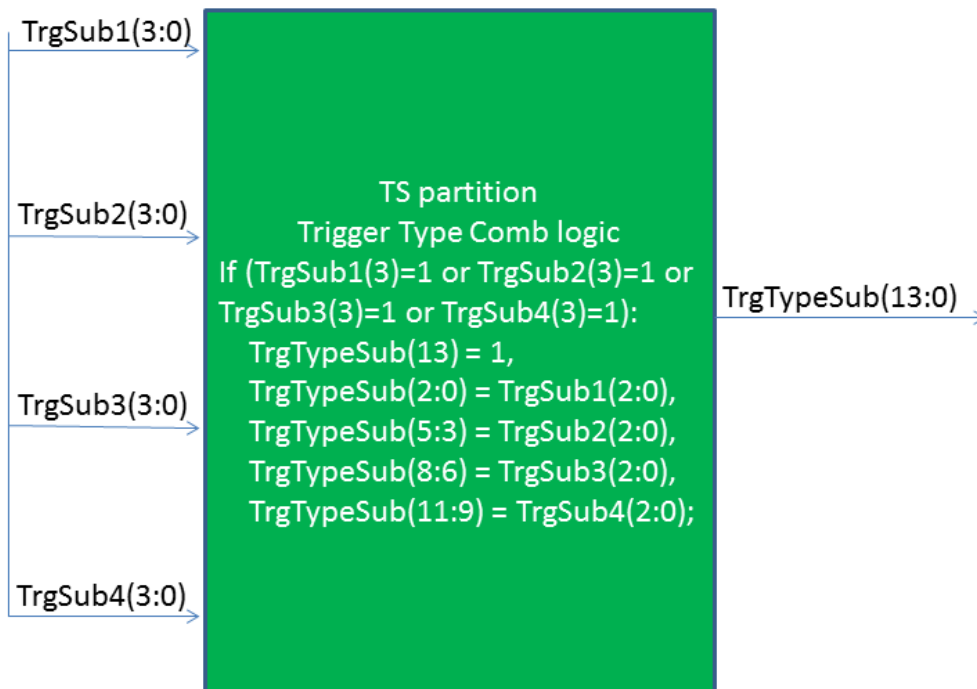
**Figure 10 Front panel input trigger lookup table**



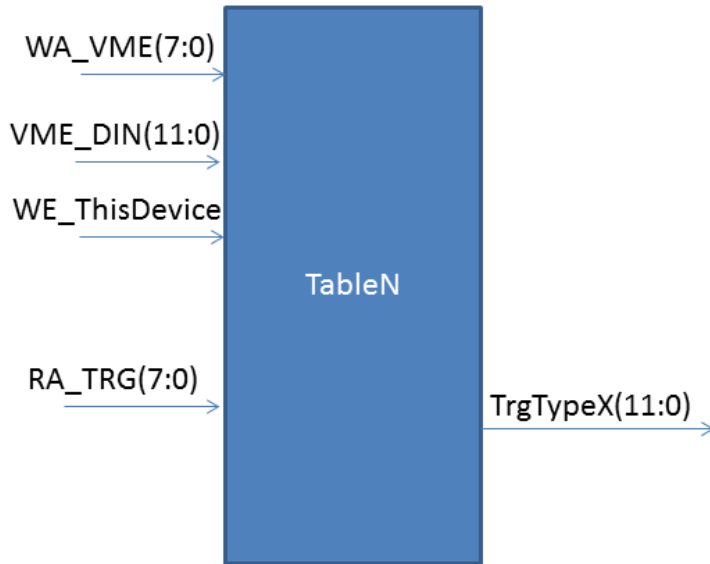
**Figure 11 TS partition #1 and partition#2 event type lookup table**



**Figure 12 TS partition#3 and partition#4 event type lookup table**



**Figure 13 TS partition event type combination**



TrgTypeX(11): Multiple triggers ( $\geq 2$ )  
 TrgTypeX(10): At least one trigger ( $\geq 1$ )  
 TrgTypeX(9:8): Trigger type: trigger1, trigger2, SyncEvt  
 TrgTypeX(7:0): trigger type

**Figure 14 Read/write data port for each lookup table cell (FPGA Block RAM)**

In summary, 8 lookup table cells for the TS event type, and 8 cells for TS partition event type. The event types are combined using FPGA logic, which is not user programmable. The user can trick the lookup table by using the property that the 8-inputs cells are implemented in the fully addressed way. That is by grouping the inputs, users can have a much more complicated scheme (trigger inputs and event types correspondence).

### 3.17 Buffer mode readout for Fastbus electronics

An additional trigger content word was generated by the TS and sent immediately after the trigger strobe word. This content word has Bit(15:12) = '0111' as special marker; Bit(11:8) indicates which duplication branch has the corresponding Local trigger, and matched with the readout trigger, that is have data; Bit(7:0) is Local trigger number counter since the previous readout trigger. (ratio of Local Trigger to Readout Trigger).

The assumption is made that "there is no more than one readout trigger in 32 ns". This trigger content word is decoded by the TI, and put in event word#4 bit(31:20) in event\_by\_event basis. By re-assigning the word#4, there are 36-bit for readout event number, which correspond to about 4 days data taking at 200 KHz. The ROC can decode this word, and knows which Fastbus duplication branch has data for this event. For block readout mode, the ROC needs put an event marker according to this data available information.

## 4. Specification Sheet

### 4.1 Mechanical

Single-width VITA 41 Payload Module with the size of 6Ux160mm. It will be positioned in the PP18 of global clock/trigger distribution crate. If there is no SD in the crate, the TS can be plugged in any of the payload slots to act as a “mini” trigger distribution crate.

### 4.2 High speed serial P0 inputs:

BUSY LVDS signals on P0;

I<sup>2</sup>C to VXS Switch slot# B, switch slot #A;

High speed LVDS data link (can be output too).

### 4.3 High speed serial P0 outputs (LVPECL):

Two 250 MHz clock to Switch slot #B; one to the switch slot #A;

1.25 Gbps Encoded (serialized) TRIGGER Signal, or pulsed Trigger1 and Trigger2;

250 Mbps Manchester encoded SYNC Signal;

Pulsed trigger, which used the Trigger2 line on SD to distribute trigger to TD boards.

### 4.4 High speed P2 GTP level one trigger primitive inputs:

250 Mbps GTP trigger primitive inputs via a backplane transition board;

P2 Row A and Row C are used, 32-pairs for 32 GTP inputs (the 1<sup>st</sup> and 17<sup>th</sup> pairs are used as data source synchronized clock);

LVPECL Signal;

Some of the 32-pairs may not be used.

### 4.5 External trigger inputs:

30-bit differential external trigger in four groups with 7/8 bits per group;

Two source synced clocks

### 4.6 Front panel general inputs:

16-bit general purpose differential inputs, which could be additional 15-bit asynchronous trigger inputs and one trigger inhibit (or busy) bit on the 34-pin connector;

250MHz Clock Input (SMA), LVPECL input;

### 4.7 Front panel general outputs:

10 LVCMOS single-ended outputs (TS status);

12 ECL differential outputs (TS status);

250 MHz clock output (SMA) LVPECL.

## 4.8 Fiber channel signals: two AFBR-79Q4Z modules (They share the same front panel space as the clock input/output SMA connectors)

### CLKSYNC Fixed Latency Link

250Mbps Serial Communication

Manchester Encoded

CLKSYNC to CLK skew variation < 1ns

This link can optionally be connected to the FPGA's MGT Rx/Tx directly;

### TRIGGER\_TX/STATUS\_RX

1.25Gbps Trigger Word Line

Provides 16bit parallel data every 16ns

A BUSY status word in the opposite direction, which also includes trigger acknowledge, and frontend data readout acknowledge.

This link is connected to the FPGA's MGT Rx/Tx directly;

### CLK

250MHz low jitter clock

This link can optionally be connected to the FPGA's MGT Rx/Tx directly;

### SPARE

Generic Rx/Tx pair connected to the FPGA MGT transceiver with loopback capability.

## 4.9 Indicators: Front Panel:

Six sets of LEDs on front panel:

Set 1 (LEDA): TS board status

FPGA is ready;

Board OK;

Lengthened trigger active;

Lengthened VME DTACK indicator.

Set 2 (LEDB): External 30-bit trigger input indicator;

Set 3 (LEDC): GTP 30-bit trigger input indicator;

Set 4 (LEDD): AFBR-79Q4 status

Set 5 (LEDE): 10 LVCMOS, 12 ECL outputs status

Set 6 (LEDF): 15-bit asynchronous trigger inputs and inhibit

On board:

Power OK near each regulator or DC-DC converter; (OFF, if power is OK)

FPGA program DONE; (OFF, if the FPGA is programmed)

#### 4.10 Programming:

VME64x, A24/D32, A32/D64 slave

#### 4.11 Firmware loading:

VME to JTAG A24D32 with user defined AM (Address Modifier) for remote firmware loading with redundant On board JTAG connector;

Custom VME to JTAG engine implemented in the FPGA using A24 standard AM.

It is possible that two revisions of the firmware be stored in the PROM simultaneously, with one as backup; or as two different configurations of TS. But as the TS firmware got more complicated, the FPGA resource usage increased to a point that the PROM is no longer large enough to hold two compressed firmware designs.

#### 4.12 Power requirements:

+3.3v @ 1.5 Amps; +5V @ 4 Amp; +12V, -12V @ 0.3 Amp (From Backplane), total < 35 W;

DC-DC converters for +1.8V, +2.5V, +3.3V and -5V, and Local regulators for other required voltages: +1.0V, +1.2V, +3.3V clock.

#### 4.13 Environment:

Forced air cooling: Weiner standard

Commercial grade components ( 0-75 Celsius or better)

### **5 TS operation procedure:**

The TS needs be properly set and plugged into the proper crate and slot. Damage may happen to the TS, the crate, or other PCBs in the crate if the right procedure is not followed.

#### 5.1 TS Power supply:

The TS use +3.3V directly from VME64x crate. It can also generate its own +3.3V supply by a DC-DC converter from +5V from VME. This is selected by loading the on-board FUSE and/or DC-DC converter. To avoid conflict, do not populate both (VXS crate is a VME64x compatible backplane).

#### 5.2 Hardware setting (Switch etc.):

An 8-bit switch is used to set the TS working mode:

Bit8: LVPECL, open=high, Clock source selection. When high, the clock source is on board oscillator; when low, the clock source is front panel clock input. The switch setting can be overwritten by a software setting (FPGA pin drive).

Bit7: open=high, A24 address space selection. When LOW, use the VME64x GA(4:0); HIGH, use the switch bit(5:1) as A24 address space setting;

Bit6: FPGA firmware revision selection, low: revision 0, high: revision 1; This can be overwritten by the FPGA MULTIBOOT RS0 if the revision 1 should fail (and if rev1 is selected). As the PROM can only save one firmware version, this switch needs be set to LOW.

Bit(5:1): A24 address space A(19:23). As the TS and TD are the only VME custom modules in the global trigger distribution crate, the switch Bit(5:1) and Bit7 will most likely be left unused as both the TS and TD support the Geographic address GA(4:0), and the VXS crate is VME64x compatible.

### 5.3 Software setting:

After the board is properly set, and plugged in the right slot, some software setting needs be applied for the board to work. The TS software setup sequence: (The TS may not work if the sequence is not followed)

- TS clock selection: This is important for the trigger distribution system to work properly
- Trigger table loading: This is the most important step, and it is required every time the TS is powered up, or the FPGA is reprogrammed;
- Some other reset/initialize in specified order. TS initialization is a whole trigger distribution issue (TS/TD/TI), not just TS itself. See the software example for proper order.

## 6. VME Programming Requirements

There are three categories of Address Modifier codes are supported on the TS: the user-defined Address Modifier codes (A24) for emergency firmware loading; Standard A24 for FPGA register read/write and slow control; A32 block transfer for VME data readout. The A24 address space is set by the 5-bit Geographical address of the VME64x backplane or the onboard binary switch (as mentioned in section 5.2).

### 6.1 VME to JTAG emergency loading:

The AM[5:0] user defined codes are used for this logic. This works even before the FPGA is programmed and working. It is almost the same as A24D32 mode. The valid AM codes are: 0x19, 0x1A, 0x1D and 0x1E. These AM codes are user defined, and similar to the AM codes 0x39, 0x3A, 0x3D and 0x3E.

The valid address bits are A[31:24] do not care; A[23:19]=GA[4:0] for VME64x crates, or A[23:19]=0 for non-VME64x crates; A[18:2]=b'0001111111111111.

Data bit[1] is used for TDI; data bit[0] is used for TMS.

For example, if the board is in slot#5 (that is  $\sim$ GA(4:0)= 11010), you need write to A(23:0)=0x28fffc.

If data(1:0)=00, both TMS and TDI will be low;

if data(1:0)=01, TMS is high, TDI is low;

if data(1:0)=10, TMS is low, TDI is high;

if data(1:0)=11, both TDI and TMS are high.

The normal A24 address should try to avoid this address (0x0fffc).

A more advanced example: Instruction register shift (8-bit, shift in 0x5a) starting from/end up at the 'reset idle' mode: 14 consecutive writes to the address 0x28fffc with AM=0x19, 1a, 1d or 1e, the data are 1, 1, 0, 0, 0, 2, 0, 2, 2, 0, 2, 1, 1, 0 respectively.

Data	1	1	0	0	0	2	0	2	2	0	2	1	1	0
TMS	H	H	L	L	L	L	L	L	L	L	L	H	H	L
TDI	0x	0x	0x	0x	0	1	0	1	1	0	1	0	0x	0x

“TMS H” means logic High, “TMS L” means logic Low, “TDI 0” means 0 or Low, “TDI 1” means 1 or High, and “TDI 0x” means DO NOT CARE by the JTAG, but the set value is 0.

## 6.2 Configuration Registers:

A24D32 are used for register read/write. Similar to the emergency loading logic, the base address is determined by the Geographic Address in VME64x crate, and external switch for non-VME64x crate. That is, A[23:19]=GA[4:0], or SW[5:1]. [0xXX, BIT(m,n)] means the register offset and bits in old TI firmware design.

### ➤ Address offset: 0x00000: Board ID:

Bit 7-0 (R/W): Crate ID; Reset default 0x00; [0x00, BIT(7:0)]

Bit 12-8 (R): A24 address, higher 5 bits; Reset default 000 [0x54, BIT(4:0)]

Bit 13 (R): ‘1’: TS is in running mode, it is GA parity bit for TI;

Bit 15-14 (R): ‘10’: TS is in running mode (no more register changes), others: TS not in running mode;

Bit 31-16 (R): hex 0x71D5 (0x71: TI, 0x75: TS, 0x7D: TD, 0x71D5: TIDS). [0x54, BIT(31:16)]

### ➤ Address offset: 0x00004: Optic transceiver enable:

Bit 7-0 (R/W): on TI: HFBR#8, #7, ... #1 Enable, Reset default 0xFF all enabled. [0x04, BIT(7:0)]

Bit 0: ‘1’ enable HFBR#1, ‘0’ disable HFBR#1; (‘disable’ means ‘power down’)

Bit 1: ‘1’ enable HFBR#2, ‘0’ disable HFBR#2;

Bit 2: ‘1’ enable HFBR#3, ‘0’ disable HFBR#3;

Bit 3: ‘1’ enable HFBR#4, ‘0’ disable HFBR#4;

Bit 4: ‘1’ enable HFBR#5, ‘0’ disable HFBR#5;

Bit 5: ‘1’ enable HFBR#6, ‘0’ disable HFBR#6;

Bit 6: ‘1’ enable HFBR#7, ‘0’ disable HFBR#7;

Bit 7: ‘1’ enable HFBR#8, ‘0’ disable HFBR#8;

On TS, they set the mode for TS’s two QSFP optic transceiver:

Bit 0: unused

Bit 1: FiberLP; Low Power select; but not valid for this transceiver (Class 1 power level module, 1.5W)

Bit 2: FiberMS, low true, module select; this select the module for I2C interface active.

Bit 3: FiberRst, low true, optic transceiver reset;

On TSrev2 and production TS, this register is not used

Fiber Reset is done through offset 0x100, bit 5;

Fiber module select and monitoring are done through offset 0xD8;



Bit8: Enable the P0 (SW#A, SW#B) trigger and SyncReset signal, default is 1 to enable; When disabled, the Trigger/SyncReset stay LOW.

➤ Address offset: 0x00008: Interrupt setting:

Bit 7-0 (R/W): Interrupt ID; Reset default 0xC8 [0x08, BIT(7:0)]

Bit 10-8 (R/W): Interrupt level; Reset default 5; [0x08, BIT(10:8)]

Bit 16 (R/W): IRQ enable. Reset default: 0, used to enable INTERRUPT; [0x04, BIT(13)]

➤ Address offset: 0x0000C: Trigger delay and Pulse width:

Bit 7-0 (R/W): Trigger\_1 delay, (n+1)\*4 ns; Reset default 0x07; [0x0C, BIT(7:0)]

Bit 15-8 (R/W): Trigger\_1 Pulse width (n+1)\*4 ns; Reset default 0x07; [0x0C, BIT(15:8)]

Bit 23-16 (R/W): Trigger\_2 delay, (n+1)\*4 ns; Reset default 0x07; [0x0C, BIT(23:16)]

Bit 31-24 (R/W): Trigger\_2 Pulse width (n+1)\*4 ns. Reset default 0x07; [0x0C, BIT(27:24)]

➤ Address offset: 0x00010: A32 address space:

Bit 13-5 (R/W): Address Max; Reset default 0x1FF; [0x10, BIT(13:5)]

Bit 22:14 (R/W): Address Min; Reset default 0x000; [0x10, BIT(22:14)]

Bit 31-23 (R/W): Base Address. Reset default 0x100; [0x10, BIT(31:23)]

➤ Address offset: 0x00014: Block size:

Bit 7-0 (R/W): Block size. Reset default 0x01; This is used on TD only. It's read/write register but does not affect anything in the FPGA.

Bit 23-16 (R): Block size (block level). This is the block level used on the FPGA.

Bit 31-24 (R): Block size set by TS (or TImaster). This is an intermediate value.

➤ Address offset: 0x00018: TI data format control: Reset default 011; [0x00, BIT(31:29)]

Bit 0: if '1', two block placeholder words are enabled; '0' disabled. (Discontinued)

Bit 2-1: Event format control:

00: Shortest words per trigger; (32-bit event number following the event header)

01: The TS timing word (bit 31-0) is enabled;

10: The TS timing word (bit 47-32) is enabled as lower 16-bit readout, Event number bit 47-32 as higher 16-bit readout;

11: The TS timing word (bit(47:0) is enabled as two words; The event number bit 47-32 is read out as higher 16-bit word.

Bit 3: '1' to enable the GTP trigger input readout;

Bit 4: '1' to enable the EXT (front panel synchronous) trigger inputs readout;

Bit 5: '1' to enable the FP (front panel Asynchronous) trigger inputs readout;

➤ Address offset: 0x0001C: VME setting; Reset default 0x011:

Bit 0 (R/W): '1' enable Bus\_Error\_En, so the block read can be terminated by event block trailer; [0x10, BIT(0)]

Bit 1 (R/W): '1' en\_token\_in is true, '0' en\_token\_in is false; [0x10, BIT(1)]

Bit 2 (R/W): '1' enable 'Multi-board' readout, '0' disable 'Multi-board'; [0x10, BIT(2)]; asset to enable multi-board token passing protocol;

Bit 3 (R/W): '1' enable en\_A32m, '0' disable en\_A32m; [0x10, BIT(3)]; assert to enable common A32 multi-board addressing of module;

Bit 4 (R/W): '1' enable en\_A32, '0' disable en\_A32; [0x10, BIT(4)]

Bit 7 (R/W): '1' enable VME bus interrupt for module error? (not used)

Bit 8 (R/W): '1' I2C device address 0x1101xxx, '0' I2C device address 0x0000xxx; [0x04, BIT(8)]

Bit 9 (R/W): '1' token\_in high, '0' token\_in low; [0x04, BIT(9)]

Bit 10 (R/W): '1' first\_board true, '0' first\_board false; [0x04, BIT(10)]

Bit 11 (R/W): '1' last\_board true, '0' last board false; [0x04, BIT(11)]

Bit 15 (R/W): '1' disable data readout buffer full [0x04, BIT(15)]

Bit 16: Enable the front panel (external input) signal to latch the trigger input scalars;

Bit 17: Enable the front panel (same signal as controlled by bit0) to reset the trigger input scalars; These two bits are kind of related to the OneShotVme command, offset 0x100, bit 24 and bit25.

Bit 20: Enable the VME P2 row #Z output to TSIO board for signals TDIO(16:1). '1' to enable

➤ Address offset: 0x00020: Trigger source register:

Bit 15-0 (R/W): Trigger source enables: Reset default 0x0000; [0x00, BIT(23:16)]

Bit 0: P0 trigger input;

Bit 1: HFBR#1 trigger input;

Bit 2: TS loopback trigger input; (No effect on TS)

Bit 3: Front Panel trigger input;

Bit 4: VME trigger;

Bit 5: Front Panel Trigger Codes (as Supervisor) inputs;

Bit 6: TS\_rev2 trigger input;

Bit 7: Random Trigger.

Bit 8: FP/Ext/GTP trigger enable on TS;

Bit 9: P2 Busy used as Trigger input [0x28, BIT(6)]

Bit 10, SubTS trigger enable (on TI);

Bit 11, Enable the automatic trigger1 generation by trigger2; (TSp)

Bit 12: Enable SubTS#1 generation on TS, enable SubTS#1 trigger on TS/TI;

Bit 13: Enable SubTS#2 generation on TS, enable SubTS#2 trigger on TS/TI;

Bit 14: Enable SubTS#3 generation on TS, enable SubTS#3 trigger on TS/TI;

Bit 15: Enable SubTS#4 generation on TS, enable SubTS#4 trigger on TS/TI;

Bit 31-16 (R): Trigger source monitor. Trigger word (15:0)

➤ Address offset: 0x00024: Sync Source register:

Bit 15-0 (R/W): Sync Source enables: Reset default 0x02; [0x00, BIT(28:24)]

Bit 0: P0 sync input;

Bit 1: HFBR#1 sync input;

Bit 2: HFBR#5 sync input;

Bit 3: Front panel sync input;

Bit 4: TS loopback SYNC enable;

Bit 6: Automatic SyncReset enable;

Bit 7: '1', Sync code 0x99 and 0xCC enable

Bit 31-16 (R): Sync source monitoring. Sync Code monitor

- Bit 19-16: TS generated SYNC code,
- Bit 21-20: TS generated SYNC code valid (repeat),
- Bit 22: loopback sync code valid,
- Bit 23: Sync history FIFO empty? 1: yes, 0: no,
- Bit 31-24: SyncSourceMon(7:0),

➤ Address offset: 0x00028: Busy source registers:

Bit 15-0 (R/W): Busy source enables: [0x04, BIT(31:16)]

- Bit 0: '1' enable the Switch Slot #A BUSY input, '0' disable;
- Bit 1: '1' enable the Switch Slot #B BUSY input, '0' disable;
- Bit 2: '1' enable the VME P2 BUSY input, '0' disable;
- Bit 3: '1' enable the FTDC front panel BUSY input, '0' disable;
- Bit 4: '1' enable the FADC front panel BUSY input, '0' disable;
- Bit 5: '1' enable the Front Panel BUSY, which is the same as TsRev2 busy;  
(Enable SD TrgOut as SyncResetRequest;)

- Bit 6: '1' Use P2 BUSY input as trigger1 input (useful and valid for TD only);
- Bit 7: '1' enable TS feedback BUSY, '0' disable the busy. (useful in TM mode);

Bit 15-8: HFBR #8-#1 BUSY enables: '1' enable the HFBR BUSY input, '0' disable;

- Bit 8: TI\_A busy
- Bit 9: TI\_B busy
- Bit 10: TS data block readout interrupt N reaches 0x8000;
- Bit 12: Duplication branch#1 enable; (for HallA SBS)
- Bit 13: Duplication branch#2 enable; (for HallA SBS)
- Bit 14: Duplication branch#3 enable; (for HallA SBS)
- Bit 15: Duplication branch#4 enable; (for HallA SBS)

Bit 31-16 (R): BUSY source monitoring. [0x28, BIT(15:0)]

- Bit 16: switch slot#A busy inserted;
- Bit 17: Switch slot#B busy inserted;
- Bit 18: Loopback sync reset requested;
- Bit 19: TI\_A sync reset requested;
- Bit 20: TI\_B sync reset requested;
- Bit 21: Front panel busy inserted;
- Bit 22: Loopback readout busy;
- Bit 23: Loopback readout acknowledged lagged too much (over threshold);
- Bit 24: TI\_A data readout acknowledges lagged too much (over threshold);
- Bit 25: TI\_B data readout acknowledges lagged too much (over threshold);
- Bit 26: Interrupt counter Bit(15), that is too many interrupt waiting;
- Bit 27: Internal loop back BUSY;
- Bit 28: TI\_A BUSY asserted;
- Bit 29: TI\_B BUSY asserted;
- Bit 30: TI\_A trigger missing/acknowledge required (lock mode);
- Bit 31: TI\_B trigger missing/acknowledge required (lock mode);

- Address offset: 0x0002C: Clock source selection:
  - Bit 1-0 (R/W): software bit switch to control the clock source. Reset default 00; [0x08, BIT(13:12)]
    - Bit[1:0] = 00: oscillator clock;
    - Bit[1:0] = 01: HFBR#5 clock input; (not valid for current setup)
    - Bit[1:0] = 10: HFBR#1 clock input;
    - Bit[1:0] = 11: Front panel 34-pin connector clock input.
    - 00, 11: oscillator clock on TS;
    - 01, 10: SMA external clock input on TS;
- Address offset: 0x00030: Trigger\_1 pre-scale:
  - Bit 15-0 (R/W): pre-scale factor: Rate = Rate\_0 / (Bit(15:0)+1). [0x0C, BIT(31:28)]
- Address offset: 0x00034: Trigger block inhibit:
  - Bit 7-0 (R/W): TS trigger inhibit threshold (in the unit of event blocks); Reset default 0x01; [0x08, BIT(31:24)]
  - Bit 23-8 (R): Number of blocks in the DAQ ready to be readout. [0x14, BIT(31:24)]  
(on TI, Bit 23-16 (R): Number of events before the a block is formed. [0x14, BIT(23:16)] )
  - Bit 27-24 (R): Number of missing block acknowledge;
  - Bit 28 (R): if '1', the run stopped because the number of block (readout) has reached (set by 0xFC);
  - Bit 29 (R): if '1', the event block is being filled by FillTrg;
  - Bit 30 (R): SyncReset requested, waiting for TS CPU to issue SyncReset;
  - Bit 31 (R): SyncEvent received, and the system is BUSY. Waiting for ROC to clear the frontend data.
- Address offset: 0x00038: Trigger rules:
  - Bit 7-0 (R/W): No more than 1 Trigger in (Bit(6:0)\*(16/500 ns)); Bit7 determines 16ns or 500ns step.  
Reset default 0x03; [R:0x58, BIT(23:16)] [W:0x810, BIT(7:0)]
  - Bit 15-8 (R/W): no more than 2 trigger in (Bit(14:8)\*(16/500ns)); Bit15 determines 16ns or 500ns step.  
Reset default 0x03; [W:0x810, BIT(15:8)]
  - Bit 23-16 (R/W): no more than 3 triggers in (Bit(22:16)\*(16/500 ns)); Bit23 determines 16ns or 500ns step.  
Reset default 0x03; [R:0x58, BIT(31:24)] [W:0x810, BIT(23:16)]
  - Bit 31-24 (R/W): no more than 4 triggers in (Bit(30:24)\*(16/500 ns)). Bit31 determines 16ns or 500ns step.  
Reset default 0x03; [W:0x810, BIT(31:24)]
- Address offset: 0x0003C: Trigger coincidence window: (in 4 ns steps)
  - Bit 7-0 (R/W): Trigger input coincidence window; Reset default 0x01;
  - Bit15-8 (R/W): Trigger inhibit window (extra to bit(7:0)). Reset default 0x00;
  - These two parameters are used to determine the event resolution
  - Bit 24-16 (R/W): the latency setting when trigger1 is generated by trigger2 automatically. (TSp V1.3)
- Address offset: 0x00040: GTP trigger input enable:
  - Bit 31-0 (R/W): 32-bit GTP input enables. Reset default 0x00000000;
- Address offset: 0x00044: Front panel external trigger input enable:

Bit 31-0 (R/W): Front panel 32-bit external trigger input enable. Reset default 0x00000000;

- Address offset: 0x00048: Front panel generic trigger input enable:

Bit 15-0 (R/W): Front panel generic trigger input enable (34-pin TI). Reset default 0x00000000;

Bit 20-16 (R/W): Widen the GTP, EXT, FP bit pattern for readout; Actually, it is the width of the valid signal for the trigger bit FIFO output. (in 4 ns steps)

- Address offset: 0x0004C: Blocks for VME interrupt:

Bit 15-0 (R/W): 4-bit output to the front panel generic output connector; Reset default 0x0000; [0x2C, BIT(3:0)]

Bit 23-16 (R): Number of data blocks ready for VME interrupts. [0x2C, BIT(31:24)]

Bit 31-24 (R): on TI: Number of events of a partial block (or, before the block is formed)

On TS: bit(15:8) of the number of data blocks ready for VME interrupt.

- Address offset: 0x00050: Sync delay setting (to compensate for the fiber length):

Bit 7-0 (R): on TI: SYNC phase of HFBR#1 input; [0x1C, BIT(7:0)]

Bit 15-8 (R/W): HFBR#1 SYNC input delay; Reset default 0x00; [0x1C, BIT(15:8)]

Bit 23-16 (R/W): TM (internal loopback) SYNC delay; Reset default 0x00; [0x1C, BIT(23:16)]

Bit 23-16 (R): on TI: SYNC phase of HFBR#5 input; [0x1C, BIT(23:16)]

Bit(31:24 (R/W): HFBR#5 SYNC input delay. Reset default 0x00; [0x1C, BIT(31:24)]

- Address offset: 0x00054: GTP input pre-scale: Reset default 0x00000000;

**The prescale factor is: 1 (0000), 2 (0001), 3 (0010), 5 (0011), 9 (0100), 17 (0101), 16385 (1111).**

**Or  $(1 + 2^{(x-1)})$  for  $x > 0$ ,  $x$  is the decimal value of the pre-scale setting). This is valid for all the register offsets from 0x54 to 0x74.**

Bit 3-0 (R/W): GTP input #1 pre-scale;

Bit 7-4 (R/W): GTP input #2 pre-scale;

Bit 11-8 (R/W): GTP input #3 pre-scale;

Bit 15-12 (R/W): GTP input #4 pre-scale;

Bit 19-16 (R/W): GTP input #5 pre-scale;

Bit 23-20 (R/W): GTP input #6 pre-scale;

Bit 27-24 (R/W): GTP input #7 pre-scale;

Bit 31-28 (R/W): GTP input #8 pre-scale.

- Address offset: 0x00058: GTP input pre-scale: Reset default 0x00000000;

Bit 3-0 (R/W): GTP input #9 pre-scale;

Bit 7-4 (R/W): GTP input #10 pre-scale;

Bit 11-8 (R/W): GTP input #11 pre-scale;

Bit 15-12 (R/W): GTP input #12 pre-scale;

Bit 19-16 (R/W): GTP input #13 pre-scale;

Bit 23-20 (R/W): GTP input #14 pre-scale;

Bit 27-24 (R/W): GTP input #15 pre-scale;

Bit 31-28 (R/W): GTP input #16 pre-scale.

- Address offset: 0x0005C: GTP input pre-scale: Reset default 0x00000000;
  - Bit 3-0 (R/W): GTP input #17 pre-scale;
  - Bit 7-4 (R/W): GTP input #18 pre-scale;
  - Bit 11-8 (R/W): GTP input #19 pre-scale;
  - Bit 15-12 (R/W): GTP input #20 pre-scale;
  - Bit 19-16 (R/W): GTP input #21 pre-scale;
  - Bit 23-20 (R/W): GTP input #22 pre-scale;
  - Bit 27-24 (R/W): GTP input #23 pre-scale;
  - Bit 31-28 (R/W): GTP input #24 pre-scale.
- Address offset: 0x00060: GTP input pre-scale: Reset default 0x00000000;
  - Bit 3-0 (R/W): GTP input #25 pre-scale;
  - Bit 7-4 (R/W): GTP input #26 pre-scale;
  - Bit 11-8 (R/W): GTP input #27 pre-scale;
  - Bit 15-12 (R/W): GTP input #28 pre-scale;
  - Bit 19-16 (R/W): GTP input #29 pre-scale;
  - Bit 23-20 (R/W): GTP input #30 pre-scale;
  - Bit 27-24 (R/W): GTP input #31 pre-scale;
  - Bit 31-28 (R/W): GTP input #32 pre-scale.
- Address offset: 0x00064: Front Panel external trigger input pre-scale: Reset default 0x00000000
  - Bit 3-0 (R/W): FP external trigger input #1, B#1 pre-scale;
  - Bit 7-4 (R/W): FP external trigger input #2, B#2 pre-scale;
  - Bit 11-8 (R/W): FP external trigger input #3, B#3 pre-scale;
  - Bit 15-12 (R/W): FP external trigger input #4, B#4 pre-scale;
  - Bit 19-16 (R/W): FP external trigger input #5, B#5 pre-scale;
  - Bit 23-20 (R/W): FP external trigger input #6, B#6 pre-scale;
  - Bit 27-24 (R/W): FP external trigger input #7, B#7 pre-scale;
  - Bit 31-28 (R/W): FP external trigger input #8, B#8 pre-scale.
- Address offset: 0x00068: Front Panel external trigger input pre-scale: Reset default 0x00000000
  - Bit 3-0 (R/W): FP external trigger input #9, A#1 pre-scale;
  - Bit 7-4 (R/W): FP external trigger input #10, A#2 pre-scale;
  - Bit 11-8 (R/W): FP external trigger input #11, A#3 pre-scale;
  - Bit 15-12 (R/W): FP external trigger input #12, A#4 pre-scale;
  - Bit 19-16 (R/W): FP external trigger input #13, A#5 pre-scale;
  - Bit 23-20 (R/W): FP external trigger input #14, A#6 pre-scale;
  - Bit 27-24 (R/W): FP external trigger input #15, A#7 pre-scale;
  - Bit 31-28 (R/W): FP external trigger input #16, A#8 pre-scale.
- Address offset: 0x0006C: Front Panel external trigger input pre-scale: Reset default 0x00000000
  - Bit 3-0 (R/W): FP external trigger input #17, D#1 pre-scale;
  - Bit 7-4 (R/W): FP external trigger input #18, D#2 pre-scale;

Bit 11-8 (R/W): FP external trigger input #19, D#3 pre-scale;  
 Bit 15-12 (R/W): FP external trigger input #20, D#4 pre-scale;  
 Bit 19-16 (R/W): FP external trigger input #21, D#5 pre-scale;  
 Bit 23-20 (R/W): FP external trigger input #22, D#6 pre-scale;  
 Bit 27-24 (R/W): FP external trigger input #23, D#7 pre-scale;  
 Bit 31-28 (R/W): FP external trigger input #24, D#8 pre-scale.

- Address offset: 0x0006C: (HallA SBS): Local Trigger combination logic:

*Bit 31-0 (R/W): Local trigger combination (5-bit in, one bit out lookup table), default 0xAAAAAAAA, which means that Trigger input#1 will generate Local Trigger, other 4 inputs are ignored.*

- Address offset: 0x00070: Front Panel external trigger input pre-scale: Reset default 0x00000000

Bit 3-0 (R/W): FP external trigger input #25, C#1 pre-scale;  
 Bit 7-4 (R/W): FP external trigger input #26, C#2 pre-scale;  
 Bit 11-8 (R/W): FP external trigger input #27, C#3 pre-scale;  
 Bit 15-12 (R/W): FP external trigger input #28, C#4 pre-scale;  
 Bit 19-16 (R/W): FP external trigger input #29, C#5 pre-scale;  
 Bit 23-20 (R/W): FP external trigger input #30, C#6 pre-scale;  
 Bit 27-24 (R/W): FP external trigger input #31, C#7 pre-scale;  
 Bit 31-28 (R/W): FP external trigger input #32, C#8 pre-scale.

- Address offset: 0x00070: Local trigger and Fast Clear setting, reset default: 0x00000000

*Bit 6-0 (R/W): Local Trigger rule, no more than one trigger in  $4 * (\text{bit}(6:0) + 2)$  ns;  
 Bit 15-8 (R/W): Local Trigger Pulse width,  $4 * (\text{bit}(15:8) + 2)$  ns;  
 Bit 23-16 (R/W): Fast Clear Pulse width,  $4 * (\text{bit}(23:16) + 2)$  ns;  
 Bit 31-24 (R/W): Fast Clear veto (by Readout Trigger) window width,  $4 * (\text{bit}(31:24) + 2)$  ns.*

- Address offset: 0x00074: Front Panel generic trigger input pre-scale: Reset default 0x00000000

Bit 3-0 (R/W): FP Generic trigger input #1 and #9 pre-scale; (#1 is not trigger input, it is INHIBIT)  
 Bit 7-4 (R/W): FP Generic trigger input #2 and #10 pre-scale;  
 Bit 11-8 (R/W): FP Generic trigger input #3 and #11 pre-scale;  
 Bit 15-12 (R/W): FP Generic trigger input #4 and #12 pre-scale;  
 Bit 19-16 (R/W): FP Generic trigger input #5 and #13 pre-scale;  
 Bit 23-20 (R/W): FP Generic trigger input #6 and #14 pre-scale;  
 Bit 27-24 (R/W): FP Generic trigger input #7 and #15 pre-scale;  
 Bit 31-28 (R/W): FP Generic trigger input #8 and #16 pre-scale.

From V14.2, this is redefined as:

Bit 7-0 (R/W): Event Type when GTP or Front Panel has multiple inputs (default 0xFA = 250);  
 Bit 15-8 (R/W): Event Type when GTP/FP (combined) has multiple inputs (default 0xFB = 251);  
 Bit 23-16 (R/W): Event Type for VME periodic trigger (default 0xFD = 253);  
 Bit 31-24 (R/W): Event Type for Random trigger (default 0xFE = 254);

- Address offset: 0x00078: VME Sync Load[0x900, BIT(7:0)]

Bit 7-4 == Bit 3-0 (R/W): 4-bit sync code; Decoding of the Sync command (bit[7:0]):  
 0x11: VME clock DCM reset, and full reset;  
 0x22: CLK250 resync (AD9510, DCM resync and MGT reset);  
 0x33: AD9510 re-sync (slower clock phase adjustment), part of 0x22 function;  
 0x44: Reset the MGT status\_B registers;  
 0x55: Trigger link enable (serial link started), FIFO read counter reset;  
 0x77: Trigger link disable, trigger FIFO write counter reset;  
 0x88: PCIe core reset. (be careful when using it).  
 0xAA: reset the TI\_trigger\_enabled registers on TD.  
 0xBB: Event number reset, and trigger input scalar reset;  
 0xDD: (SyncReset), FPGA logic and counter reset, this reset all goes to SD, CTP/GTP;  
 0x99: Force SyncReset high if this feature is enabled (by offset 0x24, bit 7);  
 0xCC: set the SyncReset low if it is forced high by code 0x99.  
 0xEE: set the syncReset low for ~4 us.  
 0x66: to be assigned;  
 0x00, 0xff: reserved, not to be assigned

- Address offset: 0x0007C: VME Sync Delay. The latency before being serialized.  
 Bit 6-0 (R/W): latency, in 4ns steps. Reset default 000,0111 [0x904, BIT(6:0)]
- Address offset: 0x00080: Reset pulse width: Reset default 00,0111  
 Bit 7-0 (R/W): Reset (to SW#A, SW#B and on-board) pulse width. Pulse width is (Bit(6:0)\*(4/32 ns)),  
 Bit(7) determines the steps (4ns or 32ns); [0x904, BIT(13:8), bit(13) set to pulse to 1us wide]
- Address offset: 0x00084: VME Trigger Command Register  
 Bit 11-0 (R/W): Trigger Command code transmitted in the trigger link; In the 12 bits, 0xABC, the 0xA determines the command type. For example:  
 Bit(11-0) = 0x123: (A=1) one trigger1 (readout trigger) pulse will be generated, and the event type = 0x23;  
 Bit(11-0) = 0x221: (A=2) one trigger2 pulse will be generated. The 0x21 is ignored;  
 Bit(11-0) = 0x812: (A=8) Set the block level (size), and the block level is set to 0x12.
- Address offset: 0x00088 (R/W): VME Random Trigger Command Register: [0x808, BIT(15:0)]  
 Bit 3-0: Random trigger\_1 rates:  $500\text{KHz}/(2^{\text{Bit}(3:0)})$ ;  
 Bit 6-4: same as Bit(2-0) for redundancy check. No match, no trigger\_1;  
 Bit 7: enable/disable random trigger\_1;  
 Bit (7:0) = 0x80 for highest rate (~500 kHz), and 0xFF for lowest rate (~15 Hz).  
 Bit 11-8: Random trigger\_2 rates:  $500\text{KHz}/(2^{\text{Bit}(11:8)})$ ;  
 Bit 14-12: same as Bit(10-8) for redundancy check. No match, no trigger\_2.  
 Bit 15: enable/disable random trigger\_2;
- Address offset: 0x0008C(R/W): VME Trigger Generation: [0x804, BIT(31:0)]  
 Bit 15-0: Number of trigger\_1s to be generated, when set to 0xFFFF, no limit;



Bit 31-18: (trigger rate control) Time between triggers.  $T = (120 + 120 * \text{Bit}(30:18)) * 256^{\text{Bit}(31)}$  ns. (Assuming that the ClkVme=33MHz or 30ns period). The factor of 256 was changed to 2048 on Apr. 9, 2015 (V16.1 or later).

Bit 17-16: Not used.

- Address offset: 0x00090(R/W): VME Trigger\_2 Generation: [0x80C, BIT(31:0)]

Bit 15-0: Number of trigger\_2s to be generated, when set to 0xFFFF, no limit;

Bit 31-18: (trigger rate control) Time between triggers.  $T = (120 + 120 * \text{Bit}(30:18)) * 256^{\text{Bit}(31)}$  ns. (Assuming that the ClkVme=33MHz or 30ns period)

Bit 17-16: Not used.

- Address offset: 0x00094 (R): Number of Blocks in the DAQ system: [0x14, BIT(15:0)]

Bit 31-24: Number of events of a partial data block;

Bit 23-0: Number of full data blocks the TIDS has ever generated;

- Address offset: 0x00098 (R): SYNC command history;

Bit (3-0)&(31-21): Time stamp of the Sync Code, (in steps of 4096 ns),

Bit 20: Time stamp overflow,

Bit 19: TS generated SYNC code valid,

Bit 18-15: TS generated SYNC code,

Bit 14: Loopback SYNC code valid,

Bit 13-10: Loopback SYNC code,

Bit 9-6: SYNC history FIFO status: full, program\_almost full, empty, registered empty,

Bit 5-4: assigned as "00".

- Address offset: 0x0009C (R/W): The FPGA running mode;

Bit 7-0: TS in running mode if set to 0x75; if other value, not in running mode. Reset default 0x00;

TI in running mode if set to 0x71. TI starts clock monitoring in 'running' mode.

- Address offset: 0x000A0 (R): Local trigger busy timer

*Bit 31-0: Local trigger busy timer. The busy can be caused by local trigger rule, and all the duplication DAQ branches busy (CIP or BIP). In 1024 ns steps.*

- Address offset: 0x000A4 (R): The FASTBUS DAQ duplication status

*Bit 23-20: FASTBUS DAQ duplication branch BUSY (OR of the different BUSYs);*

*Bit 17: All the FASTBUS DAQ branches are BUSY, (cannot accept new Local trigger);*

*Bit 16: Local Trigger rule caused BUSY;*

*Bit 15-12: FASTBUS DAQ duplication branch BUSY from TDC/ADC boards (CIP or BIP);*

*Bit 11-8: FASTBUS DAQ duplication branch BUSY after its Fast Clear;*

*Bit 7-4: FASTBUS DAQ duplication branch BUSY after its Local Trigger;*

*Bit 3-0: 4-bit mask for available FASTBUS DAQ duplication;*

- Address offset: 0x000A8 (R): Trigger live timer: (updated by ScalarLatch)

Bit 31-0 (r): board live time counter. The real time is  $\text{Bit}(31:0) * 256 * 30\text{ns}$  (step of 7.68 us).

- Address offset: 0x000AC (R): Trigger busy (trigger dead) timer: (updated by ScalarLatch)
  - Bit 31-0 (r): TID busy (cannot accept trigger, or trigger dead) time counter. The real time is  $\text{Bit}(31:0) * 256 * 30\text{ns}$  (step of 7.68 us). This counter and the live time counter make up the total time counter, which is the total time since any one of the trigger sources is enabled.
- Address offset: 0x000B0 (R): MGT STATUS\_A: [0x30]
  - Bit 7-0: MGT[7:0] reset done;
  - Bit 11-8: MGT PLL lock detected (two MGTs per PLL lock);
  - Bit 25:16: loopback trigger data buffer length, similar to offset 0xB8 in meaning.
  - Bit 31-26, 15-12 : not used yet;}
- Address offset: 0x000B4 (R): MGT STATUS\_B registers: [0x34]
  - Bit 7-0: Channel bonding sequence detected in MGT[7:0];
  - Bit 15-8: received data is not an 8B/10B character, or has disparity error in MGT[7:0];
  - Bit 23-16: RX disparity error has occurred in MGT[7:0];
  - Bit 31-24: Rx data not in 8B/10B table has occurred in MGT[7:0].
- Address offset: 0x000B8 (R): MGT trigger data buffer length: [0x38]
  - Bit 9-0: Global trigger data buffer length (to be minimized to 0 for the longest fiber) (in 16 ns steps);
  - Bit 11-10: Data generation fifo full (in TRGDAQ module);
  - Bit 13-12: Data Readout fifo prog\_almost\_full (in VME module);
  - Bit 15:14: Data Readout fifo full; The order should be: DataReadoutFifoProgFull → DataGenFifoFull → DataReadoutFifoFull
  - Bit 25-16: Sub-system trigger data buffer length;
  - Bit 27: TI is in running mode if '1';
  - Bit 28: HFBR#1 MGT receiver error;
  - Bit 29: CLK250 DCM locked;
  - Bit 30: Clk125 DCM locked;
  - Bit 31: VME CLK (33MHz or 25MHz) DCM locked
- Address offset: 0x000BC (R): TS input trigger counter: [0x3C]
  - Bit 31-0: Number of triggers received by TS (before BUSY inhibits).
- Address offset: 0x000C0 (R): valid for TM (or with TS function), not valid for TI: [0x40]
  - Bit 7-0: Number of blocks to be readout on HFBR#1;
  - Bit 15-8: Number of blocks is still missing on HFBR#1
  - Bit 23-16: Number of blocks to be readout on HFBR#2
  - Bit 31-24: Number of blocks is still missing on HFBR#2
- Address offset: 0x000C4 (R): valid for TM (or with TS function), not valid for TI: [0x44]
  - Bit 7-0: Number of blocks to be readout on HFBR#3;
  - Bit 15-8: Number of blocks is still missing on HFBR#3;
  - Bit 23-16: Number of blocks to be readout on HFBR#4;

Bit 31-24: Number of blocks is still missing on HFBR#4;

For TS, Bit 31-0: External Trigger counter, Number of triggers from External trigger before lookup table.

- Address offset: 0x000C8 (R): valid for TM (or with TS function), not valid for TI: [0x48]

Bit 7-0: Number of blocks to be readout on HFBR#5;

Bit 15-8: Number of blocks is still missing on HFBR#5;

Bit 23-16: Number of blocks to be readout on HFBR#6;

Bit 31-24: Number of blocks is still missing on HFBR#6;

For TS, Bit 31-0: GTP Trigger counter, Number of triggers from GTP trigger before lookup table.

- Address offset: 0x000CC (R): valid for TM (or with TS function), not valid for TI: [0x4C]

Bit 7-0: Number of blocks to be readout on HFBR#7;

Bit 15-8: Number of blocks is still missing on HFBR#7;

Bit 23-16: Number of blocks to be readout on HFBR#8;

Bit 31-24: Number of blocks is still missing on HFBR#8;

For TS, Bit 31-0: Front panel (similar to TI) Trigger counter, Number of triggers from Front Panel (the 34-pin connector, not the 16-pin connectors) trigger before lookup table.

- Address offset: 0x000D0 (R): valid for TM (or with TS function)

Bit 4-0: A24 address used for the module (to match with A23-A19); [0x54, BIT(4:0)]

Bit 9-5: A24 address set by the onboard hardware switch; [0x54, BIT(9:5)]

Bit 14-10: GA(4:0), VME64x geographic address; [0x54, BIT(14:10)]

Bit 15: parity of GA(4:0); [0x54, BIT15]

Bit 23-16: Number of blocks to be readout on TM itself; [0x50, BIT(23:16)]

Bit 31-24: Number of blocks is still missing on TM itself; [0x50, BIT(31:24)]

- Address offset: 0x000D4 (R/W): Periodic Sync Event register

Bit 23-0: Number of data blocks to assert a sync event; If bit(23:0) is set to 0, the periodic Sync Event generation is disabled. On TI(master), Bit19-0 are used (shorter cycles), so the VME trigger generation on TS is also limited to the lower 20 bits, as the VME trigger generation is the same design as TImaster trigger design. After the TImaster expands from 20 bits to 24 bits, the TS VME trigger will also be expanded to 24 bits.

- Address offset: 0x000D8: TIA/TIB optic transceiver modules status and Event number;

Bit 0 (r/w): Fiber module #A select, '1' to select;

Bit 1 (r/w): Fiber module #B select, '1' to select if #A is NOT selected;

Bit 4 (r): Fiber # A transceiver present, '0' means present;

Bit 5 (r): Fiber # A interrupt, '0' means interrupt requested;

Bit 6 (r): Fiber # B transceiver present, '0' means present;

Bit 7 (r): Fiber # B interrupt, '0' means interrupt requested;

Bit 31-16 (r): Higher 16 bits of the L1 trigger number (event number)

- Address offset: 0x000DC (R): Event number

Bit 31-0: Lower 32 bits of the L1 trigger number (event number).

- Address offset: 0x000E0 (R): TI#A ID
  - Bit 7-0: TI#A Trigger Source Enable;
  - Bit 15-8: TI#A Crate ID.
- Address offset: 0x000E4 (R): TI#B ID
  - Bit 7-0: TI#B Trigger Source Enable;
  - Bit 15-8: TI#B Crate ID.
- Address offset: 0x000E8 (R): TS ID
  - Bit 7-0: TS (itself) Trigger Source Enable, should be the same as bit7-0 of offset 0x20;
  - Bit 15-8: TS (itself) Crate ID, should be the same as bit7-0 of offset 0x00.
- Address offset: 0x000EC: (not used for TS yet)
  - Bit 7-0: ROC available setting, Bit 0 is for VME ROC readout. Set on TI, not used on TS.
- Address offset: 0x000F0 (R): valid for TM (or with TS function)
  - Bit 31-0: Number of valid code from Front Panel Async trigger inputs (repeat of reg 0xCC)
- Address offset: 0x000F4 (R): valid for TS
  - Bit 31-0: Number of valid code from GTP trigger inputs (repeat of reg 0xC8)
- Address offset: 0x000F8 (R): valid for TS
  - Bit 31-0: Number of valid code from Front Panel sync trigger (External) inputs (repeat of reg 0xC4)
- Address offset: 0x000FC (R/W): End\_of\_run block number limit
  - Bit 31-0: when the limit reached, the trigger is disabled automatically.
- Address offset: 0x00100 (W): Reset and one-shot registers. The signal will be one ClkVme cycle. If the ClkVme is 33 MHz, the one-shot will be 30ns wide. Positive logic. [0x100]
  - Bit 0: not used;
  - Bit 1: if '1', RESET signal to reset the VME\_to\_I2C engine;
  - Bit 2: if '1', RESET signal to reset the VME\_to\_JTAG engine;
  - Bit 3: if '1', RESET signal to reset the VME\_to\_SFM engine;
  - Bit 4: if '1', RESET signal to reset the VME registers (TID settings) to their default values;
  - Bit 5: if '1', RESET signal to reset the fiber optic transceivers; (only for TS rev2 and later)
  - Bit 6: if '1', clear the SYNC command history buffer;
  - Bit 7: if '1', this register will generate a BUSY reset, and Trg\_Ack pulse (TS rev2 compatible).
  - Bit 8: if '1', Reset the CLK250/Clk200 DCM.
  - Bit 9: if '1', Reset the CLK125 DCM.
  - Bit 10: if '1', Reset the MGT (MultiGigabit Transceiver,) inside the FPGA.
  - Bit 11: if '1', Auto alignment of SYNC phase from HFBR#1; auto align P0 sync input for TD.
  - Bit 12: if '1', TI: Auto alignment of SYNC phase from HFBR#5;
    - TS: reset the BRAM loading address to 0 (very beginning).

Bit 13: if '1', Auto alignment of fiber latency measurement signals;  
 Bit 14: if '1', Reset the IODELAY;  
 Bit 15: if '1', Measure the fiber latency  
 Bit 16: if '1', this register will generate a 'TAKE\_TOKEN'  
 Bit 17: if '1', the available number of data blocks will decrease by 1,  
 Bit 20: if '1', Force a sync event with event type 0,  
 Bit 23: if '1', indicates that the TS data is out of sync, and Sync\_Reset is required;  
 Bit 24: if '1', all the trigger input scalars are latched (ready for read out), The livetimer/busytimer are latched;  
 Bit 25: if '1', all the trigger input scalars are reset. (Bit 24 and Bit 25 can be set simultaneously); The event number (L1A trigger counter) is reset  
 Bit 26: if '1', this register will generate a SubTS#1 ROC acknowledgement; (comparison: Bit 7: for TS)  
 Bit 27: if '1', this register will generate a SubTS#2 ROC acknowledgement;  
 Bit 28: if '1', this register will generate a SubTS#3 ROC acknowledgement;  
 Bit 29: if '1', this register will generate a SubTS#4 ROC acknowledgement;  
 Bit 31: if '1', the end\_of\_run command. If the readout block is not full, dummy trigger will be generated to fill the block.

- Address offset: 0x008CX (0x8C0 – 0x8FC) (W): Trigger table loading: (prototype TS / TImaster) [0x8CX]

Bit 31-0: 32-bit wide table loading.  
 Address bits(5-2) are used to load 16 32-bit words;  
 6-bit read addressing with 8-bit trigger type (byte wide)

- Address offset: 0x104 (R/W): Front Panel trigger input channel delay (in 4ns steps)

Bit 8-0: Channel#1 delay (B#1);  
 Bit 18-10: Channel#2 Delay (B#2);  
 Bit 28-20: Channel#3 Delay (B#3);

- Address offset: 0x108 (R/W): Front Panel trigger input channel delay (in 4ns steps)

Bit 8-0: Channel#4 delay (B#4);  
 Bit 18-10: Channel#5 Delay (B#5);  
 Bit 28-20: Channel#6 Delay (B#6);

- Address offset: 0x10C (R/W): Front Panel trigger input channel delay (in 4ns steps)

Bit 8-0: Channel#7 delay (B#7);  
 Bit 18-10: Channel#8 Delay (B#8);  
 Bit 28-20: Channel#9 Delay (A#1);

- Address offset: 0x110 (R/W): Front Panel trigger input channel delay (in 4ns steps)

Bit 8-0: Channel#10 delay (A#2);  
 Bit 18-10: Channel#11 Delay (A#3);  
 Bit 28-20: Channel#12 Delay (A#4);

- Address offset: 0x114 (R/W): Front Panel trigger input channel delay (in 4ns steps)

Bit 8-0: Channel#13 delay (A#5);  
Bit 18-10: Channel#14 Delay (A#6);  
Bit 28-20: Channel#15 Delay (A#7);

- Address offset: 0x118 (R/W): Front Panel trigger input channel delay (in 4ns steps)

Bit 8-0: Channel#16 delay (A#8);  
Bit 18-10: Channel#17 Delay (D#1);  
Bit 28-20: Channel#18 Delay (D#2);

- Address offset: 0x118 (R/W): For HallA SBS

*Bit 18-10: Local Trigger delay to generate Fast Clear signal (in 4 ns steps); up to ~2us.*

- Address offset: 0x11C (R/W): Front Panel trigger input channel delay (in 4ns steps)

Bit 8-0: Channel#19 delay (D#3);  
Bit 18-10: Channel#20 Delay (D#4);  
Bit 28-20: Channel#21 Delay (D#5);

- Address offset: 0x11C (R/W): For HallA SBS

*Bit 9-0: Length of BUSY asserted after Local Trigger (waiting for the Fastbus boards assert BUSY, in 4 ns steps); up to ~4us*

*Bit 29-20: Length of BUSY asserted after Fast Clear (just in case that the Fastbus boards de-assert BUSY too early, in 4 ns steps); up to ~4us.*

- Address offset: 0x120 (R/W): Front Panel trigger input channel delay (in 4ns steps)

Bit 8-0: Channel#22 delay (D#6);  
Bit 18-10: Channel#23 Delay (D#7);  
Bit 28-20: Channel#24 Delay (D#8);

- Address offset: 0x124 (R/W): Front Panel trigger input channel delay (in 4ns steps)

Bit 8-0: Channel#25 delay (C#1);  
Bit 18-10: Channel#26 Delay (C#2);  
Bit 28-20: Channel#27 Delay (C#3);

- Address offset: 0x128 (R/W): Front Panel trigger input channel delay (in 4ns steps)

Bit 8-0: Channel#28 delay (C#4);  
Bit 18-10: Channel#29 Delay (C#5);  
Bit 28-20: Channel#30 Delay (C#6);

- Address offset: 0x12C (R/W): Front Panel trigger input channel delay (in 4ns steps)

Bit 8-0: Channel#31 delay (C#7);  
Bit 18-10: Channel#32 Delay (C#8);

- Address offset: 0x130 (R/W): Trigger input scope readout setting (not implemented yet)

Bit 7-0: scope window size (in 4 ns steps);

Bit(15-8): scope delay (in 4ns steps)

- Address offset: 0x134 (R): SubTS#1 Data ready polling register (reg 0x34: for TS)
  - Bit 31-24: SubTS#1 number of event before forming a data block.
  - Bit 23-8: SubTS#1 number of blocks of data to be read out (to be acknowledged).
  - Bit 7-0: number of events per block (block level).
- Address offset: 0x140 (R/W): SubTS#1 Front Panel Async trigger input selection
  - Bit 17-12: SubTS#1 trigger signal #13 source selections from FP inputs.
  - Bit 11-6: SubTS#1 trigger signal #12 source selections from FP inputs.
  - Bit 5-0: SubTS#1 trigger signal #11 source selection from FP inputs.
    - When '010001', VME periodic trigger is selected;
    - When '010010', Random trigger is selected.
  - This is true for 0x340, 0x540 and 0x740 registers too.
- Address offset: 0x144 (R/W): SubTS#1 GTP trigger input selection
  - Bit 29-24: SubTS#1 trigger signal #5 source selection from GTP inputs.
  - Bit 23-18: SubTS#1 trigger signal #4 source selection from GTP inputs.
  - Bit 17-12: SubTS#1 trigger signal #3 source selection from GTP inputs.
  - Bit 11-6: SubTS#1 trigger signal #2 source selection from GTP inputs.
  - Bit 5-0: SubTS#1 trigger signal #1 source selection from GTP inputs.
- Address offset: 0x148 (R/W): SubTS#1 Front Panel External Sync trigger input selection
  - Bit 29-24: SubTS#1 trigger signal #10 source selection from EXT inputs.
  - Bit 23-18: SubTS#1 trigger signal #9 source selection from EXT inputs.
  - Bit 17-12: SubTS#1 trigger signal #8 source selection from EXT inputs.
  - Bit 11-6: SubTS#1 trigger signal #7 source selection from EXT inputs.
  - Bit 5-0: SubTS#1 trigger signal #6 source selection from EXT inputs.
- Address offset: 0x14C (W): SubTS#1 lookup table loading
  - Bit 31-0: SubTS#1 lookup table context. 1024 operations are required for one Block RAM loading.
- Address offset: 0x150 (R): SubTS#1 data readout
  - Bit 31-0: SubTS#1 event data.
- Address offset: 0x154 (R): SubTS#1 Data Block available register
  - Bit 31-24: SubTS#1 number of event before forming an data block.
  - Bit 23-8: SubTS#1 number of blocks of data generated.
  - Bit 7-0: SubTS#1 number of data blocks to be read out.
- Address offset: 0x158 (R/W): SubTS#1 Busy setting
  - Bit 31-24: SubTS#1: buffer threshold for number of data blocks on TS.
  - Bit 23: TI\_B busy enable (in case that the TI\_B is configured as partition SubTS #1);
  - Bit 22: TI\_A busy enable (in case that the TI\_A is configured as partition subTS#1);

Bit 21: Front panel inhibit enable;  
 Bit 20: VXS switch slot#B busy enable;  
 Bit 17: SubTS#1: '1' enables the number of data block threshold.  
 Bit 16: SubTS#1: '1' enable the data fifo Programmable\_almost\_full backpressure.  
 Bit 15-0: SubTS#1: TD\_busy input 15-0 enables for SubTS#1 (or, is this TD belong to SubTS#1)

- Address offset: 0x15C (R): SubTS#1 Busy time
  - Bit 15-0: SubTS#1: Busy time. Busy percentage: Bit(15-0)/0xf000; This number is updated ~every second
- Address offset: 0x00180 (R): FP input trigger scalar for #1 (INHIBIT, not valid):
  - Bit 31-0: 32-bit scalar, number of input counter
- Address offset: 0x00184 (R): FP input trigger scalar for #2:
  - Bit 31-0: 32-bit scalar, number of FP input#2 counter
- Address offset: 0x00188 (R): FP input trigger scalar for #3:
  - Bit 31-0: 32-bit scalar, number of FP input#3 counter
- Address offset: 0x0018C (R): FP input trigger scalar for #4:
  - Bit 31-0: 32-bit scalar, number of FP input#4 counter
- Address offset: 0x00190 (R): GTP input trigger scalar for #1 (clock, not valid):
  - Bit 31-0: 32-bit scalar, number of GTP input#1 counter
- Address offset: 0x00194 (R): GTP input trigger scalar for #2:
  - Bit 31-0: 32-bit scalar, number of GTP input#2 counter
- Address offset: 0x00198 (R): GTP input trigger scalar for #3:
  - Bit 31-0: 32-bit scalar, number of GTP input#3 counter
- Address offset: 0x0019C (R): GTP input trigger scalar for #4:
  - Bit 31-0: 32-bit scalar, number of GTP input#4 counter
- Address offset: 0x001A0 (R): GTP input trigger scalar for #5:
  - Bit 31-0: 32-bit scalar, number of GTP input#5 counter
- Address offset: 0x001A4 (R): GTP input trigger scalar for #6:
  - Bit 31-0: 32-bit scalar, number of GTP input#6 counter
- Address offset: 0x001A8 (R): GTP input trigger scalar for #7:
  - Bit 31-0: 32-bit scalar, number of GTP input#7 counter
- Address offset: 0x001AC (R): GTP input trigger scalar for #8:
  - Bit 31-0: 32-bit scalar, number of GTP input#8 counter
- Address offset: 0x001B0 (R): External input trigger scalar for #1 (B1):



- Bit 31-0: 32-bit scalar, number of External trigger input#1 counter
- Address offset: 0x001B4 (R): External input trigger scalar for #2 (B2):
  - Bit 31-0: 32-bit scalar, number of External trigger input#2 counter
- Address offset: 0x001B8 (R): External input trigger scalar for #3 (B3):
  - Bit 31-0: 32-bit scalar, number of External trigger input#3 counter
- Address offset: 0x001BC (R): External input trigger scalar for #4 (B4):
  - Bit 31-0: 32-bit scalar, number of External trigger input#4 counter
- Address offset: 0x001C0 (R): External input trigger scalar for #5 (B5):
  - Bit 31-0: 32-bit scalar, number of External trigger input#5 counter
- Address offset: 0x001C4 (R): External input trigger scalar for #6 (B6):
  - Bit 31-0: 32-bit scalar, number of External trigger input#6 counter
- Address offset: 0x001C8 (R): External input trigger scalar for #7 (B7):
  - Bit 31-0: 32-bit scalar, number of External trigger input#7 counter
- Address offset: 0x001CC (R): External input trigger scalar for #8 (B8):
  - Bit 31-0: 32-bit scalar, number of External trigger input#8 counter
- Address offset: 0x334 (R): SubTS#2 Data ready polling register (reg 0x34: for TS)
  - Bit 31-24: SubTS#2 number of event before forming a data block.
  - Bit 23-8: SubTS#2 number of blocks of data to be read out (to be acknowledged).
  - Bit 7-0: number of events per block (block level).
- Address offset: 0x340 (R/W): SubTS#2 Front Panel Async trigger input selection
  - Bit 17-12: SubTS#2 trigger signal #13 source selection from FP inputs.
  - Bit 11-6: SubTS#2 trigger signal #12 source selection from FP inputs.
  - Bit 5-0: SubTS#2 trigger signal #11 source selection from FP inputs.
    - When '010001', VME periodic trigger is selected;
    - When '010010', Random trigger is selected.
- Address offset: 0x344 (R/W): SubTS#2 GTP trigger input selection
  - Bit 29-24: SubTS#2 trigger signal #5 source selection from GTP inputs.
  - Bit 23-18: SubTS#2 trigger signal #4 source selection from GTP inputs.
  - Bit 17-12: SubTS#2 trigger signal #3 source selection from GTP inputs.
  - Bit 11-6: SubTS#2 trigger signal #2 source selection from GTP inputs.
  - Bit 5-0: SubTS#2 trigger signal #1 source selection from GTP inputs.
- Address offset: 0x348 (R/W): SubTS#2 Front Panel External Sync trigger input selection
  - Bit 29-24: SubTS#2 trigger signal #10 source selection from EXT inputs.

Bit 23-18: SubTS#2 trigger signal #9 source selection from EXT inputs.

Bit 17-12: SubTS#2 trigger signal #8 source selection from EXT inputs.

Bit 11-6: SubTS#2 trigger signal #7 source selection from EXT inputs.

Bit 5-0: SubTS#2 trigger signal #6 source selection from EXT inputs.

- Address offset: 0x34C (W): SubTS#2 lookup table loading

Bit 31-0: SubTS#2 lookup table context. 1024 operations are required for one Block RAM loading.

- Address offset: 0x350 (R): SubTS#2 data readout

Bit 31-0: SubTS#2 event data.

- Address offset: 0x354 (R): SubTS#2 Data Block available register

Bit 31-24: SubTS#2 number of event before forming an data block.

Bit 23-8: SubTS#2 number of blocks of data generated.

Bit 7-0: SubTS#2 number of data blocks to be read out.

- Address offset: 0x358 (R/W): SubTS#2 Busy setting

Bit 31-24: SubTS#2: buffer threshold for number of data blocks on TS.

Bit 23: TI\_B busy enable (in case that the TI\_B is configured as partition SubTS #2);

Bit 22: TI\_A busy enable (in case that the TI\_A is configured as partition subTS#2);

Bit 21: Front panel inhibit enable;

Bit 20: VXS switch slot#B busy enable;

Bit 17: SubTS#2: '1' enables the number of data block threshold.

Bit 16: SubTS#2: '1' enable the data fifo Programmable\_almost\_full backpressure.

Bit 15-0: SubTS#2: TD\_busy input 15-0 enables for SubTS#2 (or, is this TD belong to SubTS#2)

- Address offset: 0x35C (R): SubTS#2 Busy time

Bit 15-0: SubTS#2: Busy time. Busy percentage: Bit(15-0)/0xf000; This number is updated ~every second

- Address offset: 0x00380 (R): FP input trigger scalar for #5:

Bit 31-0: 32-bit scalar, number of FP input#5 counter

- Address offset: 0x00384 (R): FP input trigger scalar for #6:

Bit 31-0: 32-bit scalar, number of FP input#6 counter

- Address offset: 0x00388 (R): FP input trigger scalar for #7:

Bit 31-0: 32-bit scalar, number of FP input#7 counter

- Address offset: 0x0038C (R): FP input trigger scalar for #8:

Bit 31-0: 32-bit scalar, number of FP input#8 counter

- Address offset: 0x00390 (R): GTP input trigger scalar for #9:

Bit 31-0: 32-bit scalar, number of GTP input#9 counter

- Address offset: 0x00394 (R): GTP input trigger scalar for #10:

- Bit 31-0: 32-bit scalar, number of GTP input#10 counter
- Address offset: 0x00398 (R): GTP input trigger scalar for #11:
  - Bit 31-0: 32-bit scalar, number of GTP input#11 counter
- Address offset: 0x0039C (R): GTP input trigger scalar for #12:
  - Bit 31-0: 32-bit scalar, number of GTP input#12 counter
- Address offset: 0x003A0 (R): GTP input trigger scalar for #13:
  - Bit 31-0: 32-bit scalar, number of GTP input#3 counter
- Address offset: 0x003A4 (R): GTP input trigger scalar for #14:
  - Bit 31-0: 32-bit scalar, number of GTP input#14 counter
- Address offset: 0x003A8 (R): GTP input trigger scalar for #15:
  - Bit 31-0: 32-bit scalar, number of GTP input#15 counter
- Address offset: 0x003AC (R): GTP input trigger scalar for #16:
  - Bit 31-0: 32-bit scalar, number of GTP input#16 counter
- Address offset: 0x003B0 (R): External input trigger scalar for #9 (A1):
  - Bit 31-0: 32-bit scalar, number of External trigger input#9 counter
- Address offset: 0x003B4 (R): External input trigger scalar for #10 (A2):
  - Bit 31-0: 32-bit scalar, number of External trigger input#10 counter
- Address offset: 0x003B8 (R): External input trigger scalar for #11 (A3):
  - Bit 31-0: 32-bit scalar, number of External trigger input#11 counter
- Address offset: 0x003BC (R): External input trigger scalar for #12 (A4):
  - Bit 31-0: 32-bit scalar, number of External trigger input#12 counter
- Address offset: 0x003C0 (R): External input trigger scalar for #13 (A5):
  - Bit 31-0: 32-bit scalar, number of External trigger input#13 counter
- Address offset: 0x003C4 (R): External input trigger scalar for #14 (A6):
  - Bit 31-0: 32-bit scalar, number of External trigger input#14 counter
- Address offset: 0x003C8 (R): External input trigger scalar for #15 (A7):
  - Bit 31-0: 32-bit scalar, number of External trigger input#15 counter
- Address offset: 0x003CC (R): External input trigger scalar for #16 (A8):
  - Bit 31-0: 32-bit scalar, number of External trigger input#16 counter
- Address offset: 0x534 (R): SubTS#3 Data ready polling register (reg 0x34: for TS)

Bit 31-24: SubTS#3 number of event before forming a data block.  
Bit 23-8: SubTS#3 number of blocks of data to be read out (to be acknowledged).  
Bit 7-0: number of events per block (block level).

➤ Address offset: 0x540 (R/W): SubTS#3 Front Panel Async trigger input selection

Bit 17-12: SubTS#3 trigger signal #13 source selection from FP inputs.  
Bit 11-6: SubTS#3 trigger signal #12 source selection from FP inputs.  
Bit 5-0: SubTS#3 trigger signal #11 source selection from FP inputs.  
When '010001', VME periodic trigger is selected;  
When '010010', Random trigger is selected.

➤ Address offset: 0x544 (R/W): SubTS#3 GTP trigger input selection

Bit 29-24: SubTS#3 trigger signal #5 source selection from GTP inputs.  
Bit 23-18: SubTS#3 trigger signal #4 source selection from GTP inputs.  
Bit 17-12: SubTS#3 trigger signal #3 source selection from GTP inputs.  
Bit 11-6: SubTS#3 trigger signal #2 source selection from GTP inputs.  
Bit 5-0: SubTS#3 trigger signal #1 source selection from GTP inputs.

➤ Address offset: 0x548 (R/W): SubTS#3 Front Panel External Sync trigger input selection

Bit 29-24: SubTS#3 trigger signal #10 source selection from EXT inputs.  
Bit 23-18: SubTS#3 trigger signal #9 source selection from EXT inputs.  
Bit 17-12: SubTS#3 trigger signal #8 source selection from EXT inputs.  
Bit 11-6: SubTS#3 trigger signal #7 source selection from EXT inputs.  
Bit 5-0: SubTS#3 trigger signal #6 source selection from EXT inputs.

➤ Address offset: 0x54C (W): SubTS#3 lookup table loading

Bit 31-0: SubTS#3 lookup table context. 1024 operations are required for one Block RAM loading.

➤ Address offset: 0x550 (R): SubTS#3 data readout

Bit 31-0: SubTS#3 event data.

➤ Address offset: 0x554 (R): SubTS#3 Data Block available register

Bit 31-24: SubTS#3 number of event before forming an data block.  
Bit 23-8: SubTS#3 number of blocks of data generated.  
Bit 7-0: SubTS#3 number of data blocks to be read out.

➤ Address offset: 0x558 (R/W): SubTS#3 Busy setting

Bit 31-24: SubTS#3: buffer threshold for number of data blocks on TS.  
Bit 23: TI\_B busy enable (in case that the TI\_B is configured as partition SubTS #3);  
Bit 22: TI\_A busy enable (in case that the TI\_A is configured as partition subTS#3);  
Bit 21: Front panel inhibit enable;  
Bit 20: VXS switch slot#B busy enable;  
Bit 17: SubTS#3: '1' enables the number of data block threshold.  
Bit 16: SubTS#3: '1' enables the data fifo Programmable\_almost\_full backpressure.

Bit 15-0: SubTS#3: TD\_busy input 15-0 enables for SubTS#3 (or, is this TD belong to SubTS#3)

- Address offset: 0x55C (R): SubTS#3 Busy time

Bit 15-0: SubTS#3: Busy time. Busy percentage: Bit(15-0)/0xf000; This number is updated ~every second

- Address offset: 0x00580 (R): FP input trigger scalar for #9:

Bit 31-0: 32-bit scalar, number of FP input#9 counter

- Address offset: 0x00584 (R): FP input trigger scalar for #10:

Bit 31-0: 32-bit scalar, number of FP input#10 counter

- Address offset: 0x00588 (R): FP input trigger scalar for #11:

Bit 31-0: 32-bit scalar, number of FP input#11 counter

- Address offset: 0x0058C (R): FP input trigger scalar for #12:

Bit 31-0: 32-bit scalar, number of FP input#12 counter

- Address offset: 0x00590 (R): GTP input trigger scalar for #17:

Bit 31-0: 32-bit scalar, number of GTP input#17 counter

- Address offset: 0x00594 (R): GTP input trigger scalar for #18:

Bit 31-0: 32-bit scalar, number of GTP input#18 counter

- Address offset: 0x00598 (R): GTP input trigger scalar for #19:

Bit 31-0: 32-bit scalar, number of GTP input#19 counter

- Address offset: 0x0059C (R): GTP input trigger scalar for #20:

Bit 31-0: 32-bit scalar, number of GTP input#20 counter

- Address offset: 0x005A0 (R): GTP input trigger scalar for #21:

Bit 31-0: 32-bit scalar, number of GTP input#21 counter

- Address offset: 0x005A4 (R): GTP input trigger scalar for #22:

Bit 31-0: 32-bit scalar, number of GTP input#22 counter

- Address offset: 0x005A8 (R): GTP input trigger scalar for #23:

Bit 31-0: 32-bit scalar, number of GTP input#23 counter

- Address offset: 0x005AC (R): GTP input trigger scalar for #24:

Bit 31-0: 32-bit scalar, number of GTP input#24 counter

- Address offset: 0x005B0 (R): External input trigger scalar for #17 (D1):

Bit 31-0: 32-bit scalar, number of External trigger input#17 counter

- Address offset: 0x005B4 (R): External input trigger scalar for #18 (D2):

- Bit 31-0: 32-bit scalar, number of External trigger input#18 counter
- Address offset: 0x005B8 (R): External input trigger scalar for #19 (D3):
  - Bit 31-0: 32-bit scalar, number of External trigger input#19 counter
- Address offset: 0x005BC (R): External input trigger scalar for #20 (D4):
  - Bit 31-0: 32-bit scalar, number of External trigger input#20 counter
- Address offset: 0x005C0 (R): External input trigger scalar for #21 (D5):
  - Bit 31-0: 32-bit scalar, number of External trigger input#21 counter
- Address offset: 0x005C4 (R): External input trigger scalar for #22 (D6):
  - Bit 31-0: 32-bit scalar, number of External trigger input#22 counter
- Address offset: 0x005C8 (R): External input trigger scalar for #23 (D7):
  - Bit 31-0: 32-bit scalar, number of External trigger input#23 counter
- Address offset: 0x005CC (R): External input trigger scalar for #24 (D8):
  - Bit 31-0: 32-bit scalar, number of External trigger input#24 counter
  
- Address offset: 0x734 (R): SubTS#1 Data ready polling register (reg 0x34: for TS)
  - Bit 31-24: SubTS#4 number of event before forming a data block.
  - Bit 23-8: SubTS#4 number of blocks of data to be read out (to be acknowledged).
  - Bit 7-0: number of events per block (block level).
- Address offset: 0x740 (R/W): SubTS#4 Front Panel Async trigger input selection
  - Bit 17-12: SubTS#4 trigger signal #13 source selection from FP inputs.
  - Bit 11-6: SubTS#4 trigger signal #12 source selection from FP inputs.
  - Bit 5-0: SubTS#4 trigger signal #11 source selection from FP inputs.
    - When '010001', VME periodic trigger is selected;
    - When '010010', Random trigger is selected.
- Address offset: 0x744 (R/W): SubTS#4 GTP trigger input selection
  - Bit 29-24: SubTS#4 trigger signal #5 source selection from GTP inputs.
  - Bit 23-18: SubTS#4 trigger signal #4 source selection from GTP inputs.
  - Bit 17-12: SubTS#4 trigger signal #3 source selection from GTP inputs.
  - Bit 11-6: SubTS#4 trigger signal #2 source selection from GTP inputs.
  - Bit 5-0: SubTS#4 trigger signal #1 source selection from GTP inputs.
- Address offset: 0x748 (R/W): SubTS#4 Front Panel External Sync trigger input selection
  - Bit 29-24: SubTS#4 trigger signal #10 source selection from EXT inputs.
  - Bit 23-18: SubTS#4 trigger signal #9 source selection from EXT inputs.
  - Bit 17-12: SubTS#4 trigger signal #8 source selection from EXT inputs.

Bit 11-6: SubTS#4 trigger signal #7 source selection from EXT inputs.

Bit 5-0: SubTS#4 trigger signal #6 source selection from EXT inputs.

- Address offset: 0x74C (W): SubTS#4 lookup table loading

Bit 31-0: SubTS#4 lookup table context. 1024 operations are required for one Block RAM loading.

- Address offset: 0x750 (R): SubTS#4 data readout

Bit 31-0: SubTS#4 event data.

- Address offset: 0x754 (R): SubTS#4 Data Block available register

Bit 31-24: SubTS#4 number of event before forming an data block.

Bit 23-8: SubTS#4 number of blocks of data generated.

Bit 7-0: SubTS#4 number of data blocks to be read out.

- Address offset: 0x758 (R/W): SubTS#4 Busy setting

Bit 31-24: SubTS#4: buffer threshold for number of data blocks on TS.

Bit 23: TI\_B busy enable (in case that the TI\_B is configured as partition SubTS #4);

Bit 22: TI\_A busy enable (in case that the TI\_A is configured as partition subTS#4);

Bit 21: Front panel inhibit enable;

Bit 20: VXS switch slot#B busy enable;

Bit 17: SubTS#4: '1' enable the number of data block threshold.

Bit 16: SubTS#4: '1' enable the data fifo Programmable\_almost\_full backpressure.

Bit 15-0: SubTS#4: TD\_busy input 15-0 enables for SubTS#4 (or, is this TD belong to SubTS#4)

- Address offset: 0x75C (R): SubTS#4 Busy time

Bit 15-0: SubTS#4: Busy time. Busy percentage: Bit(15-0)/0xf000; This number is updated ~every second

- Address offset: 0x00780 (R): FP input trigger scalar for #13:

Bit 31-0: 32-bit scalar, number of FP input#13 counter

- Address offset: 0x00784 (R): FP input trigger scalar for #14:

Bit 31-0: 32-bit scalar, number of FP input#14 counter

- Address offset: 0x00788 (R): FP input trigger scalar for #15:

Bit 31-0: 32-bit scalar, number of FP input#15 counter

- Address offset: 0x0078C (R): FP input trigger scalar for #16:

Bit 31-0: 32-bit scalar, number of FP input#16 counter

- Address offset: 0x00790 (R): GTP input trigger scalar for #25:

Bit 31-0: 32-bit scalar, number of GTP input#25 counter

- Address offset: 0x00794 (R): GTP input trigger scalar for #26:

Bit 31-0: 32-bit scalar, number of GTP input#26 counter

- Address offset: 0x00798 (R): GTP input trigger scalar for #27:  
Bit 31-0: 32-bit scalar, number of GTP input#27 counter
- Address offset: 0x0079C (R): GTP input trigger scalar for #28:  
Bit 31-0: 32-bit scalar, number of GTP input#28 counter
- Address offset: 0x007A0 (R): GTP input trigger scalar for #29:  
Bit 31-0: 32-bit scalar, number of GTP input#29 counter
- Address offset: 0x007A4 (R): GTP input trigger scalar for #30:  
Bit 31-0: 32-bit scalar, number of GTP input#30 counter
- Address offset: 0x007A8 (R): GTP input trigger scalar for #31:  
Bit 31-0: 32-bit scalar, number of GTP input#31 counter
- Address offset: 0x007AC (R): GTP input trigger scalar for #32:  
Bit 31-0: 32-bit scalar, number of GTP input#32 counter
- Address offset: 0x007B0 (R): External input trigger scalar for #25 (C1):  
Bit 31-0: 32-bit scalar, number of External trigger input#25 counter
- Address offset: 0x007B4 (R): External input trigger scalar for #26 (C2):  
Bit 31-0: 32-bit scalar, number of External trigger input#26 counter
- Address offset: 0x007B8 (R): External input trigger scalar for #27 (C3):  
Bit 31-0: 32-bit scalar, number of External trigger input#27 counter
- Address offset: 0x007BC (R): External input trigger scalar for #28 (C4):  
Bit 31-0: 32-bit scalar, number of External trigger input#28 counter
- Address offset: 0x007C0 (R): External input trigger scalar for #29 (C5):  
Bit 31-0: 32-bit scalar, number of External trigger input#29 counter
- Address offset: 0x007C4 (R): External input trigger scalar for #30 (C6):  
Bit 31-0: 32-bit scalar, number of External trigger input#30 counter
- Address offset: 0x007C8 (R): External input trigger scalar for #32 (C7):  
Bit 31-0: 32-bit scalar, number of External trigger input#31 counter
- Address offset: 0x007CC (R): External input trigger scalar for #32 (C8):  
Bit 31-0: 32-bit scalar, number of External trigger input#32 counter
- Address offset: 0x01000 (W): Global trigger (Event type) table: first 15x11 table, cell #0 loading  
Bit 31-0: 32-bit wide table loading. 1024 words in a row



- Address offset: 0x01004 (W): Global trigger (Event type) table: first 15x11 table, cell #1 loading  
Bit 31-0: 32-bit wide table loading. 1024 words in a row
- Address offset: 0x01008 (W): Global trigger (Event type) table: first 15x11 table, cell #2 loading  
Bit 31-0: 32-bit wide table loading. 1024 words in a row
- Address offset: 0x0100C (W): Global trigger (Event type) table: first 15x11 table, cell #3 loading  
Bit 31-0: 32-bit wide table loading. 1024 words in a row
- Address offset: 0x01010 (W): Global trigger (Event type) table: first 15x11 table, cell #4 loading  
Bit 31-0: 32-bit wide table loading. 1024 words in a row
- Address offset: 0x01014 (W): Global trigger (Event type) table: first 15x11 table, cell #5 loading  
Bit 31-0: 32-bit wide table loading. 1024 words in a row
- Address offset: 0x01018 (W): Global trigger (Event type) table: first 15x11 table, cell #6 loading  
Bit 31-0: 32-bit wide table loading. 1024 words in a row
- Address offset: 0x0101C (W): Global trigger (Event type) table: first 15x11 table, cell #7 loading  
Bit 31-0: 32-bit wide table loading. 1024 words in a row
- Address offset: 0x01020 (W): Global trigger (Event type) table: first 15x11 table, cell #8 loading  
Bit 31-0: 32-bit wide table loading. 1024 words in a row
- Address offset: 0x01024 (W): Global trigger (Event type) table: first 15x11 table, cell #9 loading  
Bit 31-0: 32-bit wide table loading. 1024 words in a row
- Address offset: 0x01028 (W): Global trigger (Event type) table: first 15x11 table, cell #10 loading  
Bit 31-0: 32-bit wide table loading. 1024 words in a row
- Address offset: 0x01040 (W): Global trigger (Event type) table: second 15x11 table, cell #0 loading  
Bit 31-0: 32-bit wide table loading. 1024 words in a row
- Address offset: 0x01044 (W): Global trigger (Event type) table: second 15x11 table, cell #1 loading  
Bit 31-0: 32-bit wide table loading. 1024 words in a row
- Address offset: 0x01048 (W): Global trigger (Event type) table: second 15x11 table, cell #2 loading  
Bit 31-0: 32-bit wide table loading. 1024 words in a row
- Address offset: 0x0104C (W): Global trigger (Event type) table: second 15x11 table, cell #3 loading  
Bit 31-0: 32-bit wide table loading. 1024 words in a row
- Address offset: 0x01050 (W): Global trigger (Event type) table: second 15x11 table, cell #4 loading  
Bit 31-0: 32-bit wide table loading. 1024 words in a row

- Address offset: 0x01054 (W): Global trigger (Event type) table: second 15x11 table, cell #5 loading  
Bit 31-0: 32-bit wide table loading. 1024 words in a row
- Address offset: 0x01058 (W): Global trigger (Event type) table: second 15x11 table, cell #6 loading  
Bit 31-0: 32-bit wide table loading. 1024 words in a row
- Address offset: 0x0105C (W): Global trigger (Event type) table: second 15x11 table, cell #7 loading  
Bit 31-0: 32-bit wide table loading. 1024 words in a row
- Address offset: 0x01060 (W): Global trigger (Event type) table: second 15x11 table, cell #8 loading  
Bit 31-0: 32-bit wide table loading. 1024 words in a row
- Address offset: 0x01064 (W): Global trigger (Event type) table: second 15x11 table, cell #9 loading  
Bit 31-0: 32-bit wide table loading. 1024 words in a row
- Address offset: 0x01068 (W): Global trigger (Event type) table: second 15x11 table, cell #10 loading  
Bit 31-0: 32-bit wide table loading. 1024 words in a row

*This part (offset 0x10BC – 0x1080) is used for the simplified trigger tables. In this case, the other A24 spaces (0x1068 – 0x1000, 0x131C-0x1100) will not be used*

- Address offset: 0x01080 (W): GTP input bit(8:1) table loading  
*Bit 31-0: 32-bit wide table loading. 256 words in a row, Bit(11:0) are actually used.*
- Address offset: 0x01084 (W): GTP input bit(16:9) table loading  
*Bit 31-0: 32-bit wide table loading. 256 words in a row, Bit(11:0) are actually used.*
- Address offset: 0x01088 (W): GTP input bit(24:17) table loading  
*Bit 31-0: 32-bit wide table loading. 256 words in a row, Bit(11:0) are actually used.*
- Address offset: 0x0108C (W): GTP input bit(31:25) table loading  
*Bit 31-0: 32-bit wide table loading. 256 words in a row, Bit(11:0) are actually used.*
- Address offset: 0x01090 (W): FrontPanelExt input bit(8:1) table loading  
*Bit 31-0: 32-bit wide table loading. 256 words in a row, Bit(11:0) are actually used.*
- Address offset: 0x01094 (W): FrontPanelExt input bit(16:9) table loading  
*Bit 31-0: 32-bit wide table loading. 256 words in a row, Bit(11:0) are actually used.*
- Address offset: 0x01098 (W): FrontPanelExt input bit(24:17) table loading  
*Bit 31-0: 32-bit wide table loading. 256 words in a row, Bit(11:0) are actually used.*
- Address offset: 0x0109C (W): FrontPanelExt input bit(31:25) table loading  
*Bit 31-0: 32-bit wide table loading. 256 words in a row, Bit(11:0) are actually used.*

- Address offset: 0x010A0 (W): SubTS#1 input bit(8:1) (GTP) table loading  
Bit 31-0: 32-bit wide table loading. 256 words in a row, Bit(11:0) are actually used.
- Address offset: 0x010A4 (W): SubTS#1 input bit(16:9) (EXT) table loading  
Bit 31-0: 32-bit wide table loading. 256 words in a row, Bit(11:0) are actually used.
- Address offset: 0x010A8 (W): SubTS#2 input bit(8:1) (GTP) table loading  
Bit 31-0: 32-bit wide table loading. 256 words in a row, Bit(11:0) are actually used.
- Address offset: 0x010AC (W): SubTS#2 input bit(16:9) (EXT) table loading  
Bit 31-0: 32-bit wide table loading. 256 words in a row, Bit(11:0) are actually used.
- Address offset: 0x010B0 (W): SubTS#3 input bit(8:1) (GTP) table loading  
Bit 31-0: 32-bit wide table loading. 256 words in a row, Bit(11:0) are actually used.
- Address offset: 0x010B4 (W): SubTS#3 input bit(16:9) (EXT) table loading  
Bit 31-0: 32-bit wide table loading. 256 words in a row, Bit(11:0) are actually used.
- Address offset: 0x010B8 (W): SubTS#4 input bit(8:1) (GTP) table loading  
Bit 31-0: 32-bit wide table loading. 256 words in a row, Bit(11:0) are actually used.
- Address offset: 0x010BC (W): SubTS#4 input bit(16:9) (EXT) table loading  
Bit 31-0: 32-bit wide table loading. 256 words in a row, Bit(11:0) are actually used.
  
- Address offset: 0x01100 (W): GTP Trigger table first 15x8 table, cell #0 loading  
Bit 31-0: 32-bit wide table loading. 1024 words in a row
- Address offset: 0x01104 (W): GTP Trigger table first 15x8 table, cell #1 loading  
Bit 31-0: 32-bit wide table loading. 1024 words in a row
- Address offset: 0x01108 (W): GTP Trigger table first 15x8 table, cell #2 loading  
Bit 31-0: 32-bit wide table loading. 1024 words in a row
- Address offset: 0x0110C (W): GTP Trigger table first 15x8 table, cell #3 loading  
Bit 31-0: 32-bit wide table loading. 1024 words in a row
- Address offset: 0x01110 (W): GTP Trigger table first 15x8 table, cell #4 loading  
Bit 31-0: 32-bit wide table loading. 1024 words in a row
- Address offset: 0x01114 (W): GTP Trigger table first 15x8 table, cell #5 loading  
Bit 31-0: 32-bit wide table loading. 1024 words in a row
- Address offset: 0x01118 (W): GTP Trigger table first 15x8 table, cell #6 loading  
Bit 31-0: 32-bit wide table loading. 1024 words in a row

- Address offset: 0x0111C (W): GTP Trigger table first 15x8 table, cell #7 loading  
Bit 31-0: 32-bit wide table loading. 1024 words in a row
- Address offset: 0x01140 (W): GTP Trigger table second 15x8 table, cell #0 loading  
Bit 31-0: 32-bit wide table loading. 1024 words in a row
- Address offset: 0x01144 (W): GTP Trigger table second 15x8 table, cell #1 loading  
Bit 31-0: 32-bit wide table loading. 1024 words in a row
- Address offset: 0x01148 (W): GTP Trigger table second 15x8 table, cell #2 loading  
Bit 31-0: 32-bit wide table loading. 1024 words in a row
- Address offset: 0x0114C (W): GTP Trigger table second 15x8 table, cell #3 loading  
Bit 31-0: 32-bit wide table loading. 1024 words in a row
- Address offset: 0x01150 (W): GTP Trigger table second 15x8 table, cell #4 loading  
Bit 31-0: 32-bit wide table loading. 1024 words in a row
- Address offset: 0x01154 (W): GTP Trigger table second 15x8 table, cell #5 loading  
Bit 31-0: 32-bit wide table loading. 1024 words in a row
- Address offset: 0x01158 (W): GTP Trigger table second 15x8 table, cell #6 loading  
Bit 31-0: 32-bit wide table loading. 1024 words in a row
- Address offset: 0x0115C (W): GTP Trigger table second 15x8 table, cell #7 loading  
Bit 31-0: 32-bit wide table loading. 1024 words in a row
- Address offset: 0x01180 (W): GTP Trigger table first 15x11 table, cell #0 loading  
Bit 31-0: 32-bit wide table loading. 1024 words in a row
- Address offset: 0x01184 (W): GTP Trigger table first 15x11 table, cell #1 loading  
Bit 31-0: 32-bit wide table loading. 1024 words in a row
- Address offset: 0x01188 (W): GTP Trigger table first 15x11 table, cell #2 loading  
Bit 31-0: 32-bit wide table loading. 1024 words in a row
- Address offset: 0x0118C (W): GTP Trigger table first 15x11 table, cell #3 loading  
Bit 31-0: 32-bit wide table loading. 1024 words in a row
- Address offset: 0x01190 (W): GTP Trigger table first 15x11 table, cell #4 loading  
Bit 31-0: 32-bit wide table loading. 1024 words in a row
- Address offset: 0x01194 (W): GTP Trigger table first 15x11 table, cell #5 loading  
Bit 31-0: 32-bit wide table loading. 1024 words in a row

- Address offset: 0x01198 (W): GTP Trigger table first 15x11 table, cell #6 loading  
Bit 31-0: 32-bit wide table loading. 1024 words in a row
- Address offset: 0x0119C (W): GTP Trigger table first 15x11 table, cell #7 loading  
Bit 31-0: 32-bit wide table loading. 1024 words in a row
- Address offset: 0x011A0 (W): GTP Trigger table first 15x11 table, cell #8 loading  
Bit 31-0: 32-bit wide table loading. 1024 words in a row
- Address offset: 0x011A4 (W): GTP Trigger table first 15x11 table, cell #9 loading  
Bit 31-0: 32-bit wide table loading. 1024 words in a row
- Address offset: 0x011A8 (W): GTP Trigger table first 15x11 table, cell #10 loading  
Bit 31-0: 32-bit wide table loading. 1024 words in a row
- Address offset: 0x011C0 (W): GTP Trigger table second 15x11 table, cell #0 loading  
Bit 31-0: 32-bit wide table loading. 1024 words in a row
- Address offset: 0x011C4 (W): GTP Trigger table second 15x11 table, cell #1 loading  
Bit 31-0: 32-bit wide table loading. 1024 words in a row
- Address offset: 0x011C8 (W): GTP Trigger table second 15x11 table, cell #2 loading  
Bit 31-0: 32-bit wide table loading. 1024 words in a row
- Address offset: 0x011CC (W): GTP Trigger table second 15x11 table, cell #3 loading  
Bit 31-0: 32-bit wide table loading. 1024 words in a row
- Address offset: 0x011D0 (W): GTP Trigger table second 15x11 table, cell #4 loading  
Bit 31-0: 32-bit wide table loading. 1024 words in a row
- Address offset: 0x011D4 (W): GTP Trigger table second 15x11 table, cell #5 loading  
Bit 31-0: 32-bit wide table loading. 1024 words in a row
- Address offset: 0x011D8 (W): GTP Trigger table second 15x11 table, cell #6 loading  
Bit 31-0: 32-bit wide table loading. 1024 words in a row
- Address offset: 0x011DC (W): GTP Trigger table second 15x11 table, cell #7 loading  
Bit 31-0: 32-bit wide table loading. 1024 words in a row
- Address offset: 0x011E0 (W): GTP Trigger table second 15x11 table, cell #8 loading  
Bit 31-0: 32-bit wide table loading. 1024 words in a row
- Address offset: 0x011E4 (W): GTP Trigger table second 15x11 table, cell #9 loading  
Bit 31-0: 32-bit wide table loading. 1024 words in a row

- Address offset: 0x011E8 (W): GTP Trigger table second 15x11 table, cell #10 loading  
Bit 31-0: 32-bit wide table loading. 1024 words in a row
- Address offset: 0x01200 (W): External Trigger table first 15x8 table, cell #0 loading  
Bit 31-0: 32-bit wide table loading. 1024 words in a row
- Address offset: 0x01204 (W): External Trigger table first 15x8 table, cell #1 loading  
Bit 31-0: 32-bit wide table loading. 1024 words in a row
- Address offset: 0x01208 (W): External Trigger table first 15x8 table, cell #2 loading  
Bit 31-0: 32-bit wide table loading. 1024 words in a row
- Address offset: 0x0120C (W): External Trigger table first 15x8 table, cell #3 loading  
Bit 31-0: 32-bit wide table loading. 1024 words in a row
- Address offset: 0x01210 (W): External Trigger table first 15x8 table, cell #4 loading  
Bit 31-0: 32-bit wide table loading. 1024 words in a row
- Address offset: 0x01214 (W): External Trigger table first 15x8 table, cell #5 loading  
Bit 31-0: 32-bit wide table loading. 1024 words in a row
- Address offset: 0x01218 (W): External Trigger table first 15x8 table, cell #6 loading  
Bit 31-0: 32-bit wide table loading. 1024 words in a row
- Address offset: 0x0121C (W): External Trigger table first 15x8 table, cell #7 loading  
Bit 31-0: 32-bit wide table loading. 1024 words in a row
- Address offset: 0x01240 (W): External Trigger table second 15x8 table, cell #0 loading  
Bit 31-0: 32-bit wide table loading. 1024 words in a row
- Address offset: 0x01244 (W): External Trigger table second 15x8 table, cell #1 loading  
Bit 31-0: 32-bit wide table loading. 1024 words in a row
- Address offset: 0x01248 (W): External Trigger table second 15x8 table, cell #2 loading  
Bit 31-0: 32-bit wide table loading. 1024 words in a row
- Address offset: 0x0124C (W): External Trigger table second 15x8 table, cell #3 loading  
Bit 31-0: 32-bit wide table loading. 1024 words in a row
- Address offset: 0x01250 (W): External Trigger table second 15x8 table, cell #4 loading  
Bit 31-0: 32-bit wide table loading. 1024 words in a row
- Address offset: 0x01254 (W): External Trigger table second 15x8 table, cell #5 loading  
Bit 31-0: 32-bit wide table loading. 1024 words in a row

- Address offset: 0x01258 (W): External Trigger table second 15x8 table, cell #6 loading  
Bit 31-0: 32-bit wide table loading. 1024 words in a row
- Address offset: 0x0125C (W): External Trigger table second 15x8 table, cell #7 loading  
Bit 31-0: 32-bit wide table loading. 1024 words in a row
- Address offset: 0x01280 (W): External Trigger table first 15x11 table, cell #0 loading  
Bit 31-0: 32-bit wide table loading. 1024 words in a row
- Address offset: 0x01284 (W): External Trigger table first 15x11 table, cell #1 loading  
Bit 31-0: 32-bit wide table loading. 1024 words in a row
- Address offset: 0x01288 (W): External Trigger table first 15x11 table, cell #2 loading  
Bit 31-0: 32-bit wide table loading. 1024 words in a row
- Address offset: 0x0128C (W): External Trigger table first 15x11 table, cell #3 loading  
Bit 31-0: 32-bit wide table loading. 1024 words in a row
- Address offset: 0x01290 (W): External Trigger table first 15x11 table, cell #4 loading  
Bit 31-0: 32-bit wide table loading. 1024 words in a row
- Address offset: 0x01294 (W): External Trigger table first 15x11 table, cell #5 loading  
Bit 31-0: 32-bit wide table loading. 1024 words in a row
- Address offset: 0x01298 (W): External Trigger table first 15x11 table, cell #6 loading  
Bit 31-0: 32-bit wide table loading. 1024 words in a row
- Address offset: 0x0129C (W): External Trigger table first 15x11 table, cell #7 loading  
Bit 31-0: 32-bit wide table loading. 1024 words in a row
- Address offset: 0x012A0 (W): External Trigger table first 15x11 table, cell #8 loading  
Bit 31-0: 32-bit wide table loading. 1024 words in a row
- Address offset: 0x012A4 (W): External Trigger table first 15x11 table, cell #9 loading  
Bit 31-0: 32-bit wide table loading. 1024 words in a row
- Address offset: 0x012A8 (W): External Trigger table first 15x11 table, cell #10 loading  
Bit 31-0: 32-bit wide table loading. 1024 words in a row
- Address offset: 0x012C0 (W): External Trigger table second 15x11 table, cell #0 loading  
Bit 31-0: 32-bit wide table loading. 1024 words in a row
- Address offset: 0x012C4 (W): External Trigger table second 15x11 table, cell #1 loading  
Bit 31-0: 32-bit wide table loading. 1024 words in a row

- Address offset: 0x012C8 (W): External Trigger table second 15x11 table, cell #2 loading  
Bit 31-0: 32-bit wide table loading. 1024 words in a row
- Address offset: 0x012CC (W): External Trigger table second 15x11 table, cell #3 loading  
Bit 31-0: 32-bit wide table loading. 1024 words in a row
- Address offset: 0x012D0 (W): External Trigger table second 15x11 table, cell #4 loading  
Bit 31-0: 32-bit wide table loading. 1024 words in a row
- Address offset: 0x012D4 (W): External Trigger table second 15x11 table, cell #5 loading  
Bit 31-0: 32-bit wide table loading. 1024 words in a row
- Address offset: 0x012D8 (W): External Trigger table second 15x11 table, cell #6 loading  
Bit 31-0: 32-bit wide table loading. 1024 words in a row
- Address offset: 0x012DC (W): External Trigger table second 15x11 table, cell #7 loading  
Bit 31-0: 32-bit wide table loading. 1024 words in a row
- Address offset: 0x012E0 (W): External Trigger table second 15x11 table, cell #8 loading  
Bit 31-0: 32-bit wide table loading. 1024 words in a row
- Address offset: 0x012E4 (W): External Trigger table second 15x11 table, cell #9 loading  
Bit 31-0: 32-bit wide table loading. 1024 words in a row
- Address offset: 0x012E8 (W): External Trigger table second 15x11 table, cell #10 loading  
Bit 31-0: 32-bit wide table loading. 1024 words in a row
- Address offset: 0x01300 (W): FP Trigger table cell #0 loading  
Bit 31-0: 32-bit wide table loading; 1024 words in a row
- Address offset: 0x01304 (W): FP Trigger table cell #1 loading  
Bit 31-0: 32-bit wide table loading. 1024 words in a row
- Address offset: 0x01308 (W): FP Trigger table cell #2 loading  
Bit 31-0: 32-bit wide table loading. 1024 words in a row
- Address offset: 0x0130C (W): FP Trigger table cell #3 loading  
Bit 31-0: 32-bit wide table loading. 1024 words in a row
- Address offset: 0x01310 (W): FP Trigger table cell #4 loading  
Bit 31-0: 32-bit wide table loading. 1024 words in a row
- Address offset: 0x01314 (W): FP Trigger table cell #5 loading  
Bit 31-0: 32-bit wide table loading. 1024 words in a row



- Address offset: 0x01318 (W): FP Trigger table cell #6 loading  
Bit 31-0: 32-bit wide table loading. 1024 words in a row
- Address offset: 0x0131C (W): FP Trigger table cell #7 loading  
Bit 31-0: 32-bit wide table loading. 1024 words in a row
- Address offset: 0x02000-0x021FC (R): SD mirror memory  
Bit 15-0: first SD register data;  
Bit 31-16: second SD register data;
- Address offset: 0x02800-0x029FC (R): CTP mirror memory  
Bit 15-0: first CTP register data;  
Bit 31-16: second CTP register data.

### 6.3 VME to Serial engines:

A24D32 are used for VME to serial engines. The engines include: VME to JTAG engine for the FPGA, VME to JTAG engine for the PROM, VME to I2C for the switch slot #A, VME to I2C engine for the switch slot #B. In the I2C engines design, only the lower one-byte or 2-byte of data are used, and the higher bytes are wasted.

Address offset A[18:0]: 0x1XXXX: JTAG for PROM. Refer to the programming manual for VME to JTAG design for details.

Address offset A[18:0]: 0x2XXXX: JTAG for FPGA.

Address offset A[18:0]: 0x3XXXX: I2C for VXS switch slot #A. Refer to the programming manual for VME to I2C design for details.

Address offset A[18:0]: 0x4XXXX: I2C for VXS switch slot #B.

Address offset A[18:0]: 0x5XXXX: I2C for fiber transceivers. (Valid for TS rev2 and later)

Address offset A[18:0]: 0x6XXXX: non-volatile Flash memory interface (maybe parallel interface).

### 6.4 VME data acquisition:

For data acquisition, the A32 block reads are used. The base address is set by an A24 register.

## 7 Front panel and Backplane pin out tables:

Figure 7 shows the prototype Trigger Supervisor (TS) board. The 30 level one trigger primitive inputs from GTP are going through the P2 VME connector. There are two external trigger input connectors with two 8-input block each on the front panel, and a generic 2x34-pin IO connector on the front panel with 16 differential inputs, 12 differential outputs and 10 single-ended outputs. The external clock input and output are through pairs of coax cables with SMA connectors. There are six quad LED packs for information display. The SMA connectors are overlapping with the two optional optical transceivers.

Figure 8 shows the TS P2 transition card (TSIO). This is the signal transition from GTP to TS. There is a differential fan-out chip which can be used for TS to send pulsed trigger1 to TD boards, or something else. The LVCMOS to LVPECL translator can be used to interface the TS's LVCMOS signal to TD's differential signals. The direction of the signal can be selected in groups of four.

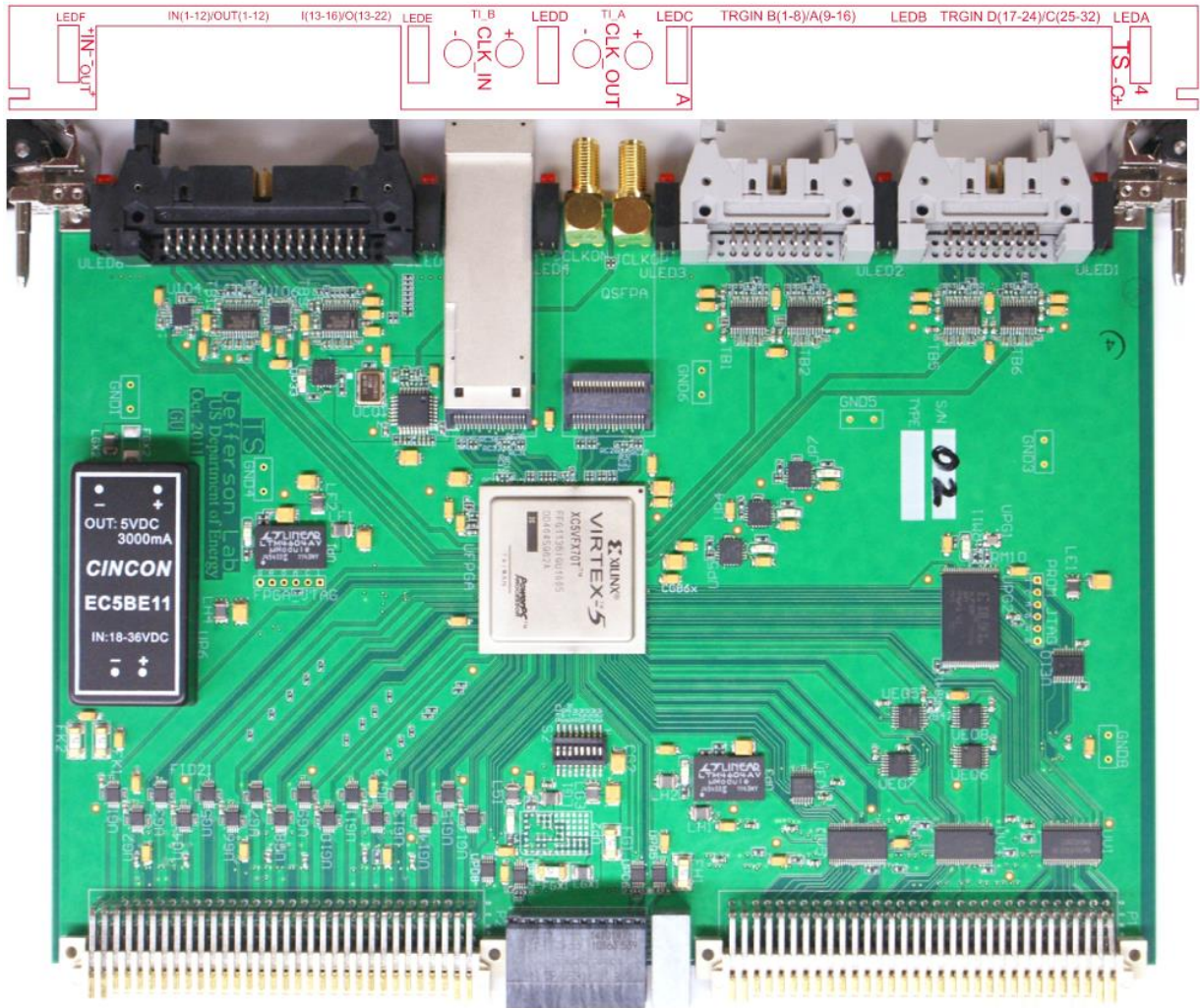


Figure 15, Trigger Supervisor (TS) PCB with its front panel drawing

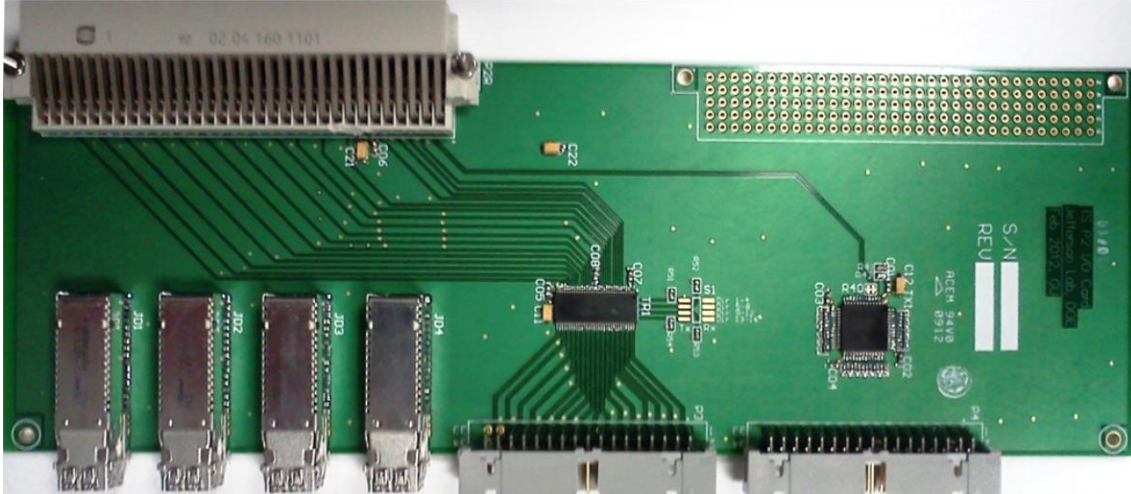


Figure 16, TSIO board, which connects to TS Via VME crate P2 connector

### 7.1 VXS P0 Pinout Table

Payload slot#18 for TS module			
Pin name	Signal Description	Signal Level	Direction
DP1 (A1+, B1-)	CLOCK_C	LVPECL	PP18 → SWA
DP3 (B2+, C2-)	TP2_LINK	LVDS	PP18 ← SWA
DP4 (E2+, F2-)	SYNC	LVPECL	PP18 → SWA
DP5 (A3+, B3-)	TRIGGER_1	LVPECL	PP18 → SWA
DP7 (B4+, C4-)	TP_LINK	LVDS	PP18 ↔ SWA
DP8 (E4+, F4-)	BUSY	LVDS	PP18 ← SWA
SE1 (G1)	I2C_SCL_SWA	LVTTTL (pull up)	PP18 → SWA
SE2 (G3)	I2C_SDA_SWA	LVTTTL (pull up)	PP18 ↔ SWA
DP23 (B12+, C12-)	CLOCK_A (250MHz)	LVPECL(DP)	PP18 → SWB
DP24 (E12+, F12-)	CLOCK_B (250MHz)	LVPECL(DP)	PP18 → SWB
DP25 (A13+,B13-)	TOKEN_OUT	LVDS(DP)	PP18 → SWB
DP26 (D13+, E13-)	SYNC	LVPECL(DP)	PP18 → SWB
DP27 (B14+, C14-)	TRIGGER_1	LVPECL(DP)	PP18 → SWB
DP28 (E14+, F14-)	TRIGGER_2	LVPECL(DP)	PP18 → SWB
DP29 (A15+,B15-)	SD_Data_Link	LVDS, 250Mbps	PP18 ↔ SWB
DP30 (D15+,E15-)	BUSY	LVDS	PP18 ← SWB
SE7 (G13)	I2C_SCL_SWB	LVTTTL (pull up)	PP18 → SWB
SE8 (G15)	I2C_SDA_SWB	LVTTTL (pull up)	PP18 ↔ SWB

### 7.2 VME P2 Pin out Table

<b>Payload slot#18, for TS module</b>			
Pin name	Signal Description	Signal Level	Direction
A1-, A2+	GTP #32	LVPECL(DP)	IN
A3-, A4+	GTP #30	LVPECL(DP)	IN
A5-, A6+	GTP #28	LVPECL(DP)	IN
A7-, A8+	GTP #26	LVPECL(DP)	IN
A9-, A10+	GTP #24	LVPECL(DP)	IN
A11-, A12+	GTP #22	LVPECL(DP)	IN
A13-, A14+	GTP #20	LVPECL(DP)	IN
A15-, A16+	GTP #18	LVPECL(DP)	IN
A17-, A18+	GTP #16	LVPECL(DP)	IN
A19-, A20+	GTP #14	LVPECL(DP)	IN
A21-, A22+	GTP #12	LVPECL(DP)	IN
A23-, A24+	GTP #10	LVPECL(DP)	IN
A25-, A26+	GTP #8	LVPECL(DP)	IN
A27-, A28+	GTP #6	LVPECL(DP)	IN
A29-, A30+	GTP #4	LVPECL(DP)	IN
A31-, A32+	GTP #2	LVPECL(DP)	IN
C1-, C2+	GTP #31	LVPECL(DP)	IN
C3-, C4+	GTP #29	LVPECL(DP)	IN
C5-, C6+	GTP #27	LVPECL(DP)	IN
C7-, C8+	GTP #25	LVPECL(DP)	IN
C9-, C10+	GTP #23	LVPECL(DP)	IN
C11-, C12+	GTP #21	LVPECL(DP)	IN
C13-, C14+	GTP #19	LVPECL(DP)	IN
C15-, C16+	GTP #17 CLOCK	LVPECL(DP)	IN (Clock)
C17-, C18+	GTP #15	LVPECL(DP)	IN
C19-, C20+	GTP #13	LVPECL(DP)	IN
C21-, C22+	GTP #11	LVPECL(DP)	IN
C23-, C24+	GTP #9	LVPECL(DP)	IN
C25-, C26+	GTP #7	LVPECL(DP)	IN
C27-, C28+	GTP #5	LVPECL(DP)	IN
C29-, C30+	GTP #3	LVPECL(DP)	IN
C31-, C32+	GTP #1 CLOCK	LVPECL(DP)	IN (Clock)
D6, D7, D8, D9, D10	Power output	+3.3V	Out
D21, D22, D23, D24, D25	Power output	-5V	Out
D1+, D2-	Trigger1	LVDS	Out
Z1 (50 Ohm to FPGA)	TD1BP / FP_Trq#1	LVTTL	IN/OUT
Z3 (50 Ohm to FPGA)	TD2BP / FP_Trq#2	LVTTL	IN/OUT
Z5 (50 Ohm to FPGA)	TD3BP / FP_Trq#3	LVTTL	IN/OUT

Z7 (50 Ohm to FPGA)	TD4BP / FP_Trg#4	LVTTL	IN/OUT
Z9 (50 Ohm to FPGA)	TD5BP / FP_Trg#9	LVTTL	IN/OUT
Z11 (50 Ohm to FPGA)	TD6BP / FP_Trg#10	LVTTL	IN/OUT
Z13 (50 Ohm to FPGA)	TD7BP / FP_Trg#11	LVTTL	IN/OUT
Z15 (50 Ohm to FPGA)	TD8BP / FP_Trg#12	LVTTL	IN/OUT
Z17 (50 Ohm to FPGA)	TD9BP / FP_Trg#17	LVTTL	IN/OUT
Z19 (50 Ohm to FPGA)	TD10BP / FP_Trg#18	LVTTL	IN/OUT
Z21 (50 Ohm to FPGA)	TD11BP / FP_Trg#19	LVTTL	IN/OUT
Z23 (50 Ohm to FPGA)	TD12BP / FP_Trg#20	LVTTL	IN/OUT
Z25 (50 Ohm to FPGA)	TD13BP / FP_Trg#25	LVTTL	IN/OUT
Z27 (50 Ohm to FPGA)	TD14BP / FP_Trg#26	LVTTL	IN/OUT
Z29 (50 Ohm to FPGA)	TD15BP / FP_Trg#27	LVTTL	IN/OUT
Z31 (50 Ohm to FPGA)	TD16BP / FP_Trg#28	LVTTL	IN/OUT

7.3 Front panel General IO Pinout Table (lower 34-pin connector). The TS PCB rev1 board has the opposite polarity as the this table.

<b>Payload slot#18 for TS module</b>			
Pin name	Signal Description	Signal Level	Direction
L1+,L2-	Inhibit (BUSY)	Differential (V: -2V ~ 3V)	IN
L3+,L4-	Scalar Control	Differential (V: -2V ~ 3V)	IN
L5+,L6-	SBS, DAQ#1 BUSY	Differential (V: -2V ~ 3V)	IN
L7+,L8-	SBS, DAQ#2 BUSY	Differential (V: -2V ~ 3V)	IN
L9+,L10-	SBS, DAQ#3 BUSY	Differential (V: -2V ~ 3V)	IN
L11+,L12-	SBS, DAQ#4 BUSY	Differential (V: -2V ~ 3V)	IN
L13+,L14-	FPIN, to be assigned	Differential (V: -2V ~ 3V)	IN
L15+,L16-	FPIN, to be assigned	Differential (V: -2V ~ 3V)	IN
L17+,L18-	FPIN, to be assigned	Differential (V: -2V ~ 3V)	IN
L19+,L20-	FPIN, to be assigned	Differential (V: -2V ~ 3V)	IN
L21+,L22-	Async FP_Trig#1	Differential (V: -2V ~ 3V)	IN
L23+,L24-	Async FP_Trig#2	Differential (V: -2V ~ 3V)	IN
L25+,L26-	Async FP_Trig#3	Differential (V: -2V ~ 3V)	IN
L27+,L28-	Async FP_Trig#4	Differential (V: -2V ~ 3V)	IN
L29+,L30-	Async FP_Trig#5	Differential (V: -2V ~ 3V)	IN
L31+,L32-	Async FP_Trig#6	Differential (V: -2V ~ 3V)	IN
L33,L34	GND		
<ul style="list-style-type: none"> <li>The signal polarity (L1-L32) for rev1 TS is different from that of the rev2 (production) TS</li> </ul>			
<b>Payload slot#18 for TS module</b>			
Pin name	Signal Description	Signal Level	Direction

R1+,R2-	BUSY	ECL	OUT
R3+,R4-	Readout Trigger	ECL	OUT
R5+,R6-	SBS DAQ#1 FastClear	ECL	OUT
R7+,R8-	SBS DAQ#2 FastClear	ECL	OUT
R9+,R10-	SBS DAQ#3 FastClear	ECL	OUT
R11+,R12-	SBS DAQ#4 FastClear	ECL	OUT
R13+,R14-	Output#7, Clk250	ECL	OUT
R15+,R16-	Prompt Trigger	ECL	OUT
R17+,R18-	SBS DAQ#1 LocalTrig	ECL	OUT
R19+,R20-	SBS DAQ#2 LocalTrig	ECL	OUT
R21+,R22-	SBS DAQ#3 LocalTrig	ECL	OUT
R23+,R24-	SBS DAQ#4 LocalTrig	ECL	OUT
R25	Output#S1	LVCOMS (50Ohm//0.1uf)	OUT
R26	Output#S2	LVCOMS (50Ohm//0.1uf)	OUT
R27	Output#S3	LVCOMS (50Ohm//0.1uf)	OUT
R28	Output#S4	LVCOMS (50Ohm//0.1uf)	OUT
R29	Output#S5	LVCOMS (50Ohm//0.1uf)	OUT
R30	Output#S6	LVCOMS (50Ohm//0.1uf)	OUT
R31	Output#S7	LVCOMS (50Ohm//0.1uf)	OUT
R32	Output#S8	LVCOMS (50Ohm//0.1uf)	OUT
R33	Output#S9	LVCOMS (50Ohm//0.1uf)	OUT
R34	Output#S10	LVCOMS (50Ohm//0.1uf)	OUT

\* L: means left side 34-pin connector; R: means right side 34-pin connector.

7.4 Front panel External Trigger Input Table (upper two 2x16 pin connector). For HallA SBS, the UL1-UL10 are used for Local trigger (or called level one trigger) inputs, LL1-LL16 and LR1-LR16 are used for readout trigger (or called level two trigger) inputs.

Pin name	Signal Description	Signal Level	Direction
UL1+, UL2-	Local Trigger #1 (D1)	Differential (V: -2V ~ 3V)	IN
UL3+, UL4-	Local Trigger #2 (D2)	Differential (V: -2V ~ 3V)	IN
UL5+, UL6-	Local Trigger #3 (D3)	Differential (V: -2V ~ 3V)	IN
UL7+, UL8-	Local Trigger #4 (D4)	Differential (V: -2V ~ 3V)	IN
UL9+, UL10-	Local Trigger #5 (D5)	Differential (V: -2V ~ 3V)	IN
UL11+, UL12-	Ext_Trig#22 (D6)	Differential (V: -2V ~ 3V)	IN
UL13+, UL14-	Ext_Trig#23 (D7)	Differential (V: -2V ~ 3V)	IN
UL15+, UL16-	Ext_Trig#24 (D8)	Differential (V: -2V ~ 3V)	IN
UR1+, UR2-	Ext_Trig#25 (C1)	Differential (V: -2V ~ 3V)	IN
UR3+, UR4-	Ext_Trig#26 (C2)	Differential (V: -2V ~ 3V)	IN
UR5+, UR6-	Ext_Trig#27 (C3)	Differential (V: -2V ~ 3V)	IN
UR7+, UR8-	Ext_Trig#28 (C4)	Differential (V: -2V ~ 3V)	IN

UR9+, UR10-	Ext_Trig#29 (C5)	Differential (V: -2V ~ 3V)	IN
UR11+, UR12-	Ext_Trig#30 (C6)	Differential (V: -2V ~ 3V)	IN
UR13+, UR14-	Ext_Trig#31 (C7)	Differential (V: -2V ~ 3V)	IN
UR15+, UR16-	Ext_Trig#32 (C8)	Differential (V: -2V ~ 3V)	IN
	CLOCK		
LL1+, LL2-	Ext_Trig#1 (B1)	Differential (V: -2V ~ 3V)	IN
LL3+, LL4-	Ext_Trig#2 (B2)	Differential (V: -2V ~ 3V)	IN
LL5+, LL6-	Ext_Trig#3 (B3)	Differential (V: -2V ~ 3V)	IN
LL7+, LL8-	Ext_Trig#4 (B4)	Differential (V: -2V ~ 3V)	IN
LL9+, LL10-	Ext_Trig#5 (B5)	Differential (V: -2V ~ 3V)	IN
LL11+, LL12-	Ext_Trig#6 (B6)	Differential (V: -2V ~ 3V)	IN
LL13+, LL14-	Ext_Trig#7 (B7)	Differential (V: -2V ~ 3V)	IN
LL15+, LL16-	Ext_Trig#8 (B8)	Differential (V: -2V ~ 3V)	IN
LR1+, LR2-	Ext_Trig#9 (A1)	Differential (V: -2V ~ 3V)	IN
LR3+, LR4-	Ext_Trig#10 (A2)	Differential (V: -2V ~ 3V)	IN
LR5+, LR6-	Ext_Trig#11 (A3)	Differential (V: -2V ~ 3V)	IN
LR7+, LR8-	Ext_Trig#12 (A4)	Differential (V: -2V ~ 3V)	IN
LR9+, LR10-	Ext_Trig#13 (A5)	Differential (V: -2V ~ 3V)	IN
LR11+, LR12-	Ext_Trig#14 (A6)	Differential (V: -2V ~ 3V)	IN
LR13+, LR14-	Ext_Trig#15 (A7)	Differential (V: -2V ~ 3V)	IN
LR15+, LR16-	Ext_Trig#16 (A8)	Differential (V: -2V ~ 3V)	IN

\* UL: Upper Left connector, UR: Upper Right connector;

\* LL: Lower Left connector, LR: Lower Right connector.

## 7.5 Front panel LED outputs:

LEDA(1)	FPGA is ready, Clock DCMs are locked
LEDA(2)	VME Data Acknowledge
LEDA(3)	Trigger_one (loopback readout trigger detected)
LEDA(4)	Sync_Reset
LEDB(1)	Synchronous front panel input(8:2) or B(8:2) input available
LEDB(2)	Synchronous front panel input(16:9) or A(8:1) input available
LEDB(3)	Synchronous front panel input(24:18) or D(8:2) input available
LEDB(4)	Synchronous front panel input(32:25) or C(8:1) input available
LEDC(1)	Synchronous GTP (VME P2) input(8:2) input available
LEDC(2)	Synchronous GTP (VME P2) input(16:9) input available
LEDC(3)	Synchronous GTP (VME P2) input(24:18) input available
LEDC(4)	Synchronous GTP (VME P2) input(32:25) input available
LEDD(1)	TI_A MGT (channel#1) Rx error OR TI_B MGT RxErr
LEDD(2)	TI_A spare MGT (Channel#4) Rx error OR TI_B spare MGT RxErr

LEDD(3)	Switch bit#7, VXS P0 Trigger2 source. LED OFF (switch is low/on) → pulsed trigger1, LED ON (switch is high/off) → trigger2
LEDD(4)	TSBusy, ON → Busy, off → not busy
LEDE(1)	TS Partition#1 triggered
LEDE(2)	TS Partition#2 triggered
LEDE(3)	TS Partition#3 triggered
LEDE(4)	TS Partition#4 triggered
LEDF(1)	Asynchronous front panel input(8:2) available
LEDF(2)	Asynchronous front panel input(16:9) available
LEDF(3)	Switch bit#6, Firmware revision setting, Off → Rev0, ON → Rev1
LEDF(4)	Switch bit#8, TS clock source, off → front pane, ON → onboard osc.

## 8. Citations:

### Works Cited

Collaboration, C. (2009). CLAS12 experiment. *Journal*, 1-25.

Collaboration, G. (2009). GlueX experiment. *journal*, 1-20.

Ed. (2010). *VME Interface*.

experiments, C. (1990). CEBAF experiements. *journal*, 1-10.

GU. (2013). *SD\_link implementation*.

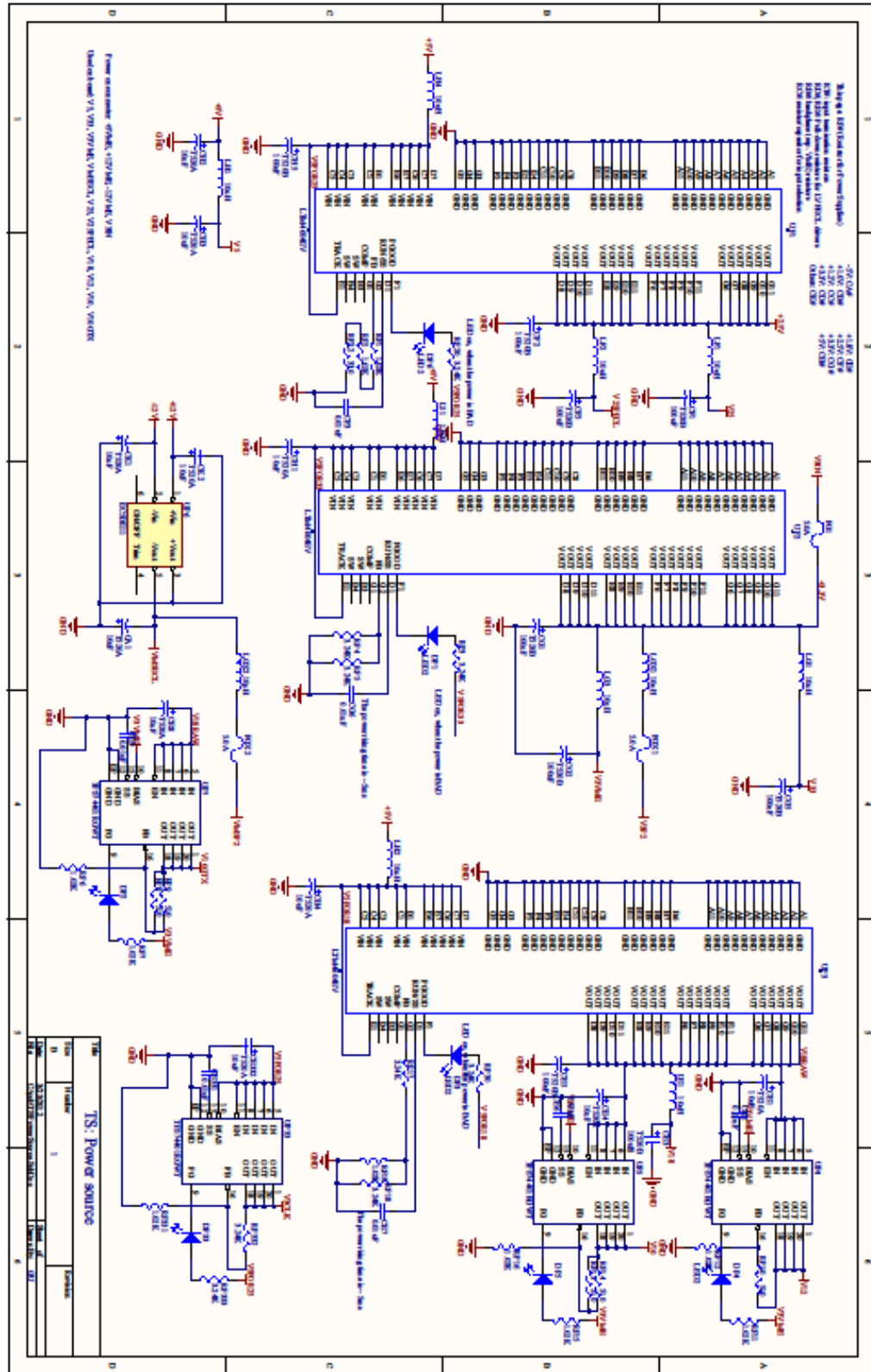
GU, J. W. (2010, August). *VME to I2C engine*. Retrieved from JLAB:  
<http://www.jlab.org/~jgu/works/VMEtoI2C.pdf>

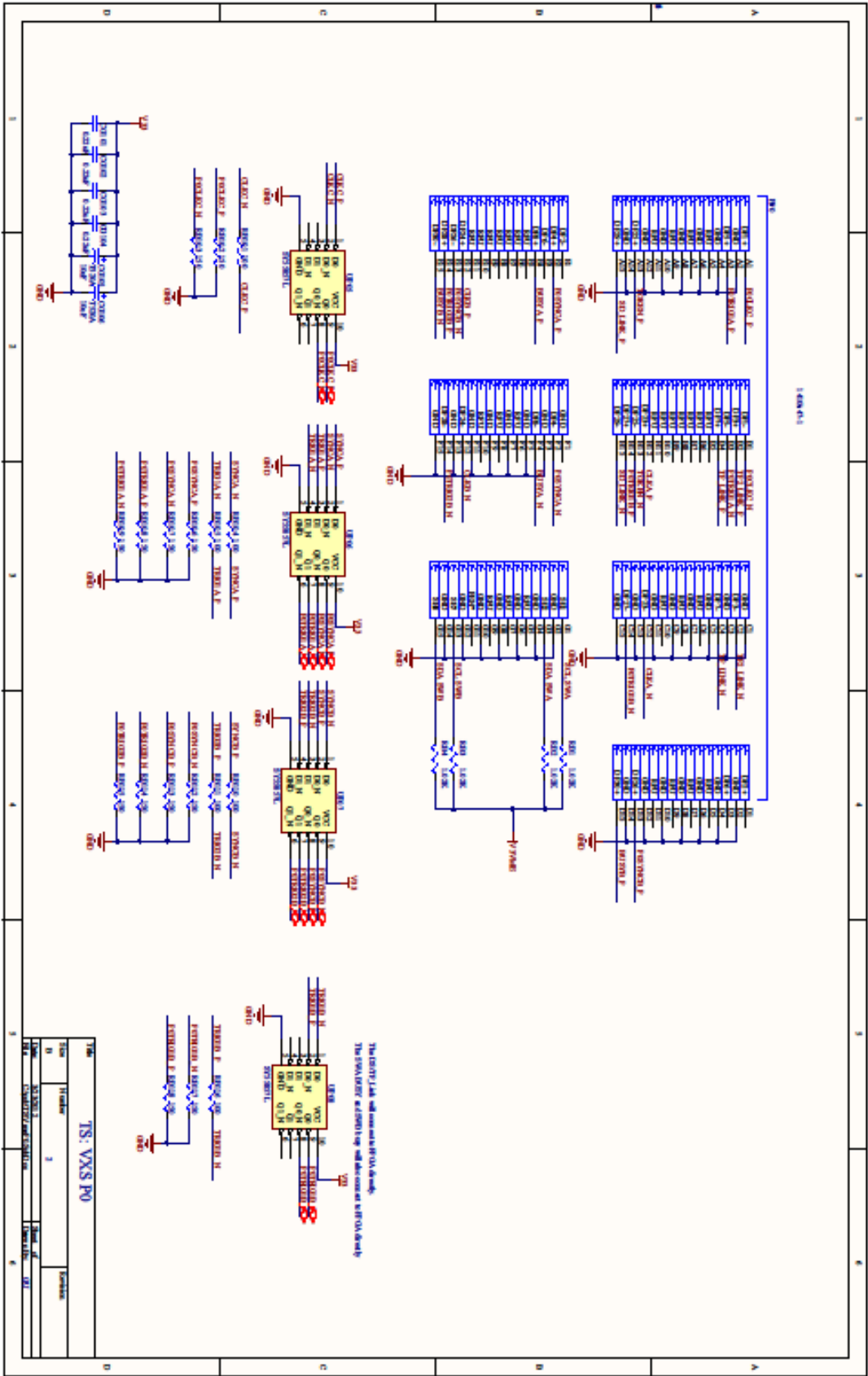
GU, (2010). VME to JTAG implementation

GU, (2010). VME to I2C implementation

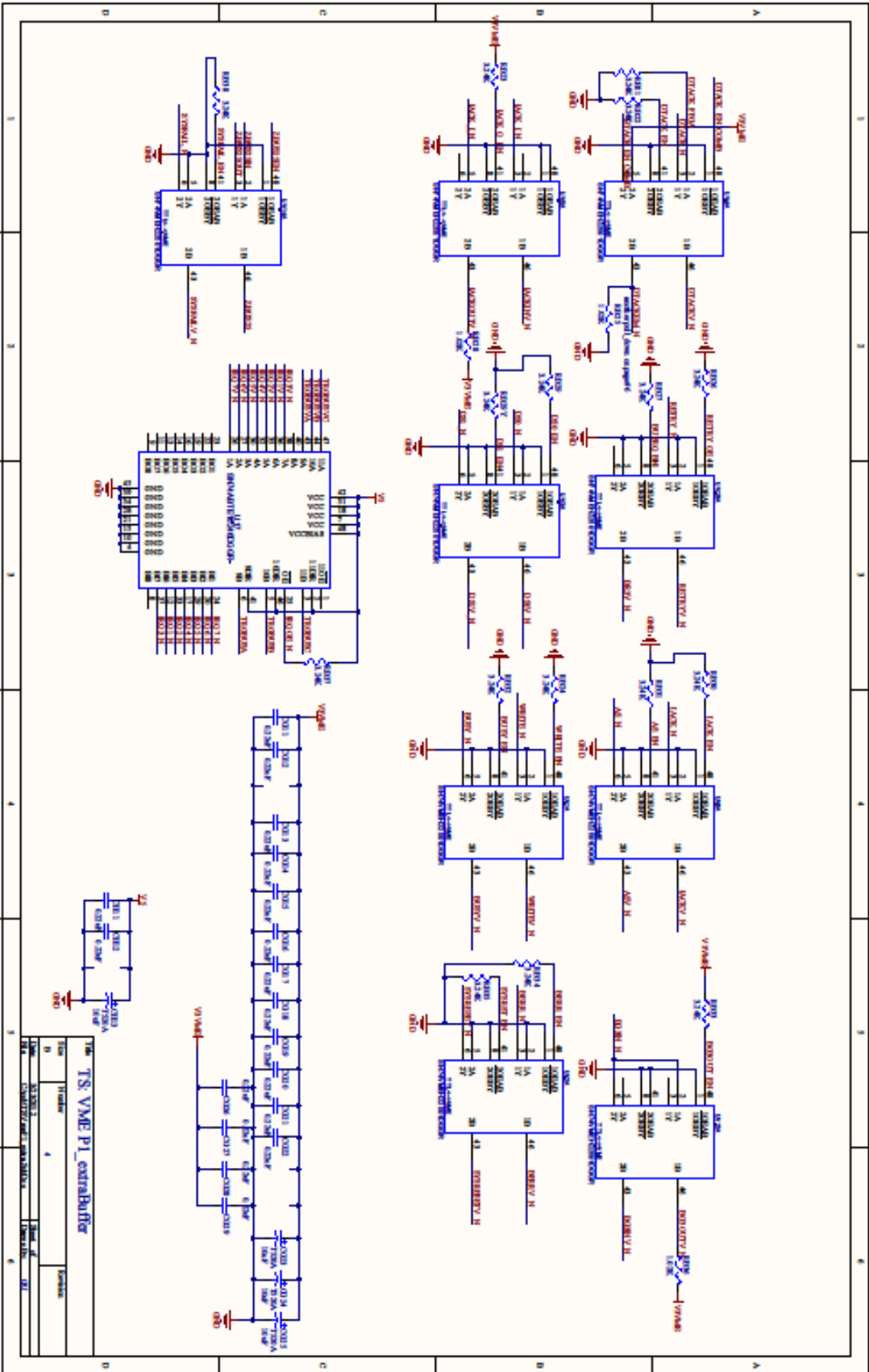


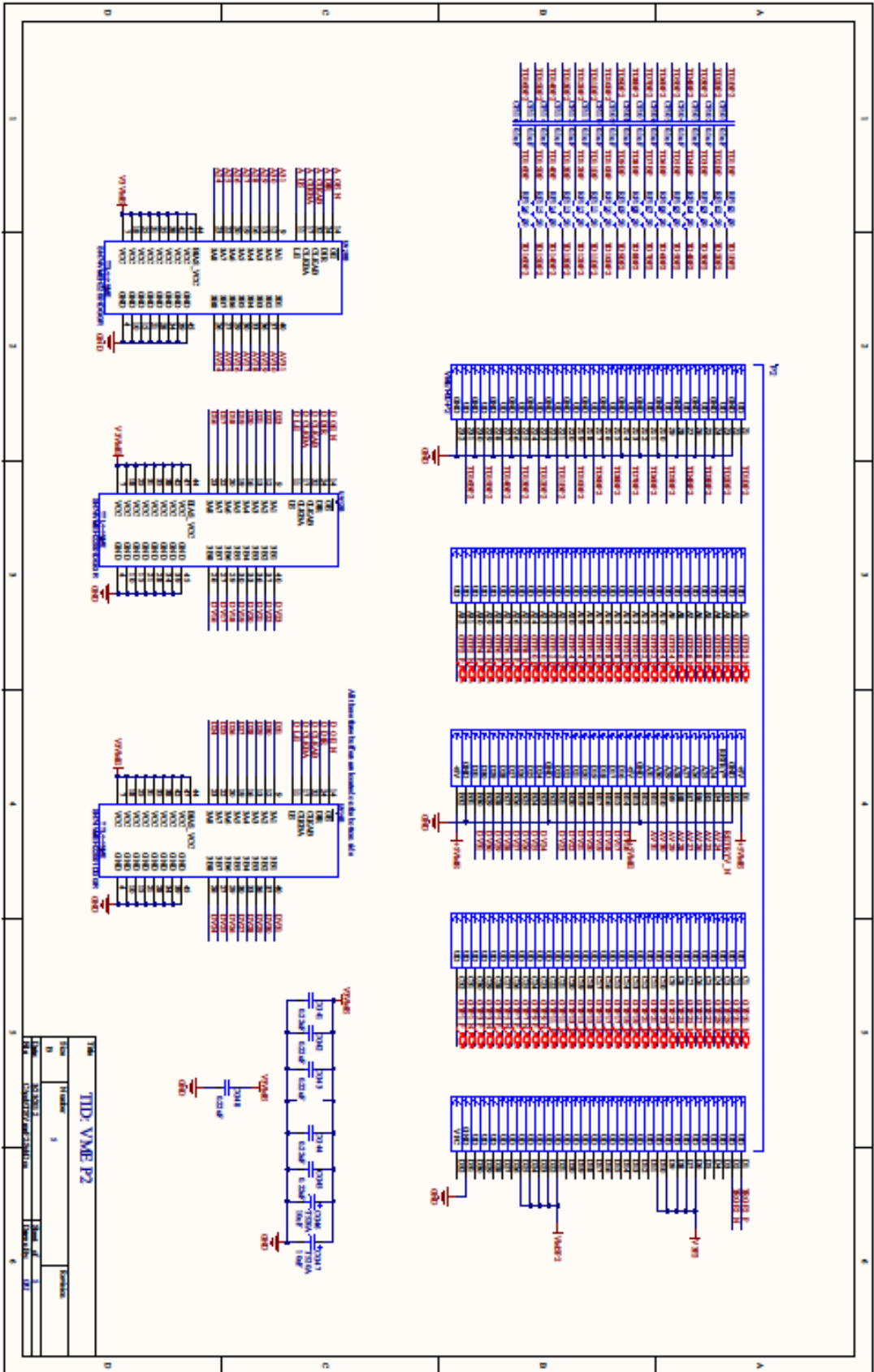
Appendix A: [TS schematics in PDF format](#)

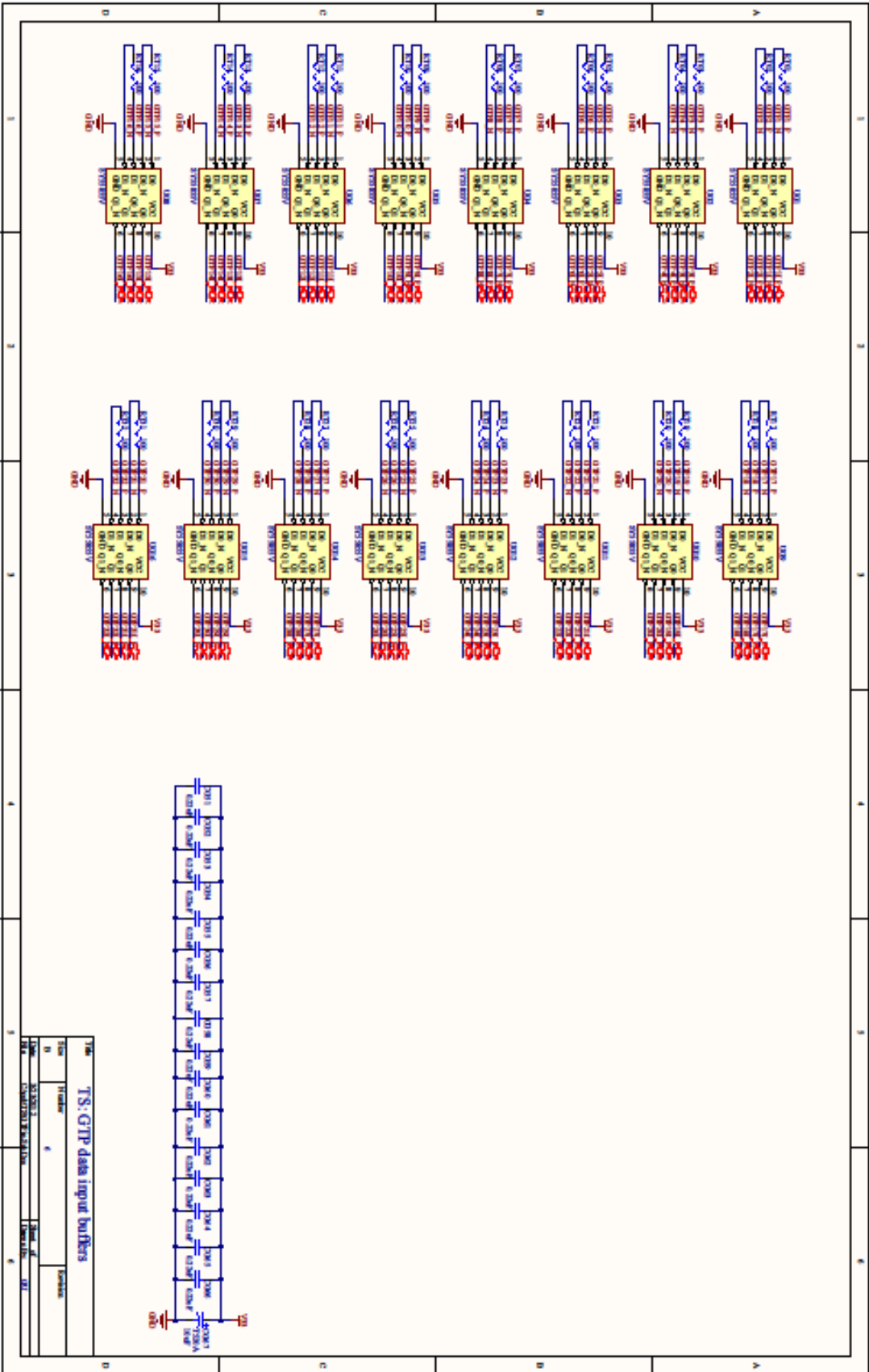




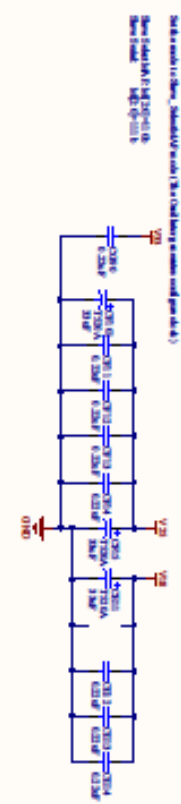
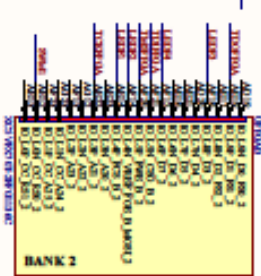
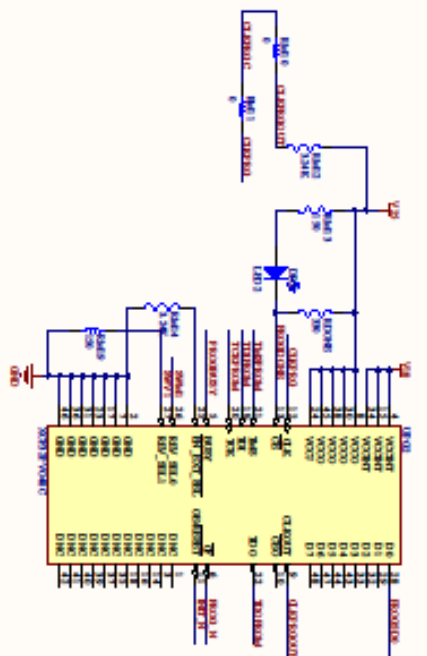
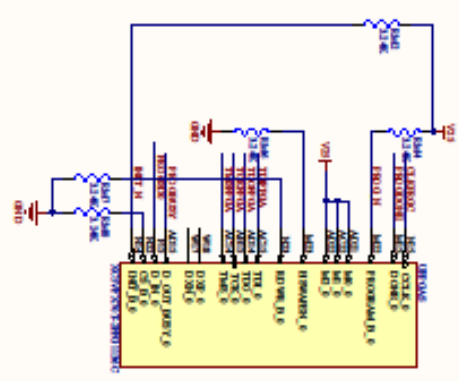
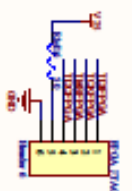
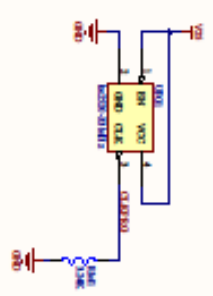












TS: FPGA Programming			
TS#	Header	TS#	TS#
1	TS: FPGA Programming	1	1
2	TS: FPGA Programming	2	2
3	TS: FPGA Programming	3	3
4	TS: FPGA Programming	4	4
5	TS: FPGA Programming	5	5
6	TS: FPGA Programming	6	6
7	TS: FPGA Programming	7	7
8	TS: FPGA Programming	8	8
9	TS: FPGA Programming	9	9
10	TS: FPGA Programming	10	10
11	TS: FPGA Programming	11	11
12	TS: FPGA Programming	12	12
13	TS: FPGA Programming	13	13
14	TS: FPGA Programming	14	14
15	TS: FPGA Programming	15	15
16	TS: FPGA Programming	16	16
17	TS: FPGA Programming	17	17
18	TS: FPGA Programming	18	18
19	TS: FPGA Programming	19	19
20	TS: FPGA Programming	20	20
21	TS: FPGA Programming	21	21
22	TS: FPGA Programming	22	22
23	TS: FPGA Programming	23	23
24	TS: FPGA Programming	24	24
25	TS: FPGA Programming	25	25
26	TS: FPGA Programming	26	26
27	TS: FPGA Programming	27	27
28	TS: FPGA Programming	28	28
29	TS: FPGA Programming	29	29
30	TS: FPGA Programming	30	30
31	TS: FPGA Programming	31	31
32	TS: FPGA Programming	32	32
33	TS: FPGA Programming	33	33
34	TS: FPGA Programming	34	34
35	TS: FPGA Programming	35	35
36	TS: FPGA Programming	36	36
37	TS: FPGA Programming	37	37
38	TS: FPGA Programming	38	38
39	TS: FPGA Programming	39	39
40	TS: FPGA Programming	40	40
41	TS: FPGA Programming	41	41
42	TS: FPGA Programming	42	42
43	TS: FPGA Programming	43	43
44	TS: FPGA Programming	44	44
45	TS: FPGA Programming	45	45
46	TS: FPGA Programming	46	46
47	TS: FPGA Programming	47	47
48	TS: FPGA Programming	48	48
49	TS: FPGA Programming	49	49
50	TS: FPGA Programming	50	50
51	TS: FPGA Programming	51	51
52	TS: FPGA Programming	52	52
53	TS: FPGA Programming	53	53
54	TS: FPGA Programming	54	54
55	TS: FPGA Programming	55	55
56	TS: FPGA Programming	56	56
57	TS: FPGA Programming	57	57
58	TS: FPGA Programming	58	58
59	TS: FPGA Programming	59	59
60	TS: FPGA Programming	60	60
61	TS: FPGA Programming	61	61
62	TS: FPGA Programming	62	62
63	TS: FPGA Programming	63	63
64	TS: FPGA Programming	64	64
65	TS: FPGA Programming	65	65
66	TS: FPGA Programming	66	66
67	TS: FPGA Programming	67	67
68	TS: FPGA Programming	68	68
69	TS: FPGA Programming	69	69
70	TS: FPGA Programming	70	70
71	TS: FPGA Programming	71	71
72	TS: FPGA Programming	72	72
73	TS: FPGA Programming	73	73
74	TS: FPGA Programming	74	74
75	TS: FPGA Programming	75	75
76	TS: FPGA Programming	76	76
77	TS: FPGA Programming	77	77
78	TS: FPGA Programming	78	78
79	TS: FPGA Programming	79	79
80	TS: FPGA Programming	80	80
81	TS: FPGA Programming	81	81
82	TS: FPGA Programming	82	82
83	TS: FPGA Programming	83	83
84	TS: FPGA Programming	84	84
85	TS: FPGA Programming	85	85
86	TS: FPGA Programming	86	86
87	TS: FPGA Programming	87	87
88	TS: FPGA Programming	88	88
89	TS: FPGA Programming	89	89
90	TS: FPGA Programming	90	90
91	TS: FPGA Programming	91	91
92	TS: FPGA Programming	92	92
93	TS: FPGA Programming	93	93
94	TS: FPGA Programming	94	94
95	TS: FPGA Programming	95	95
96	TS: FPGA Programming	96	96
97	TS: FPGA Programming	97	97
98	TS: FPGA Programming	98	98
99	TS: FPGA Programming	99	99
100	TS: FPGA Programming	100	100

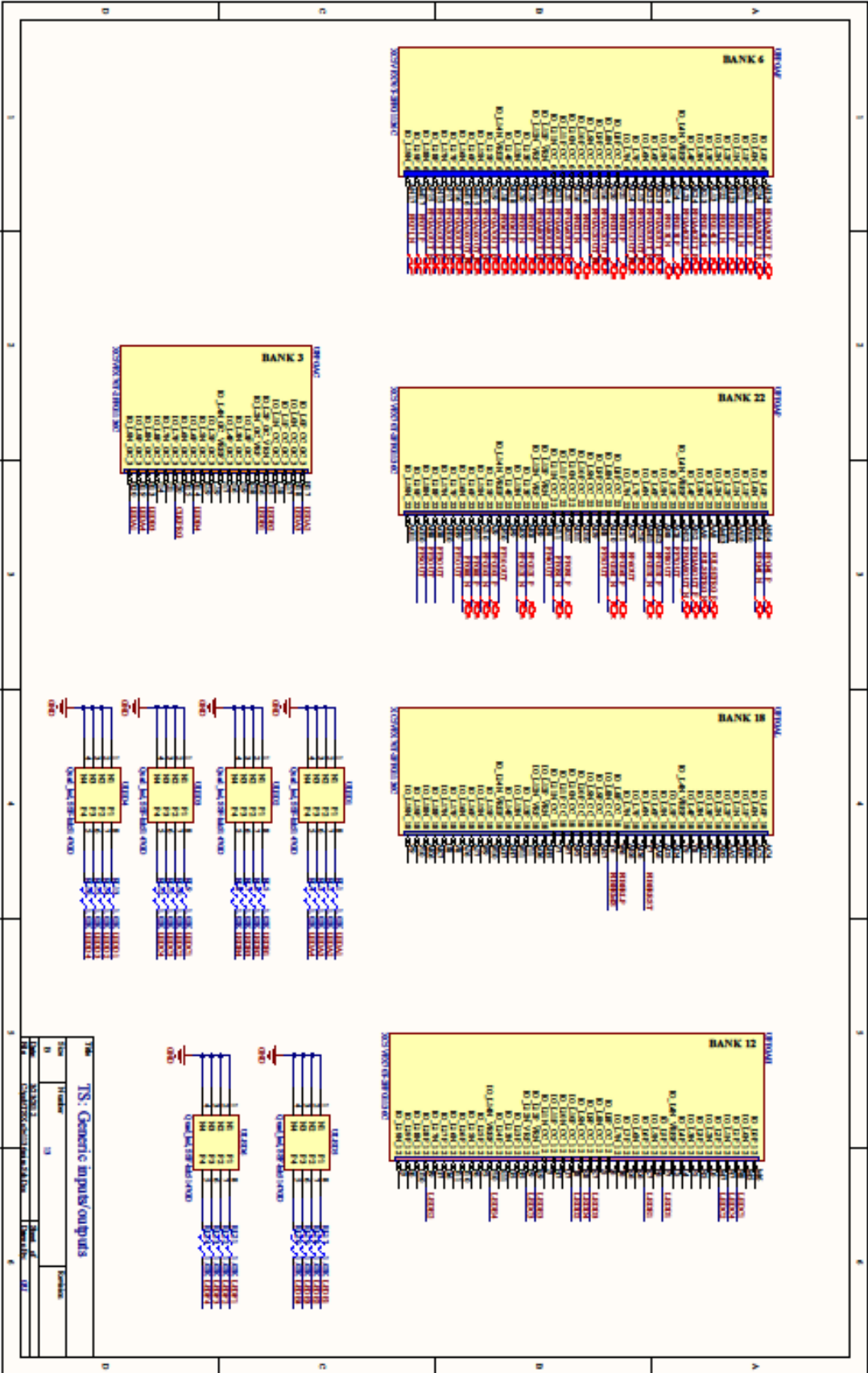


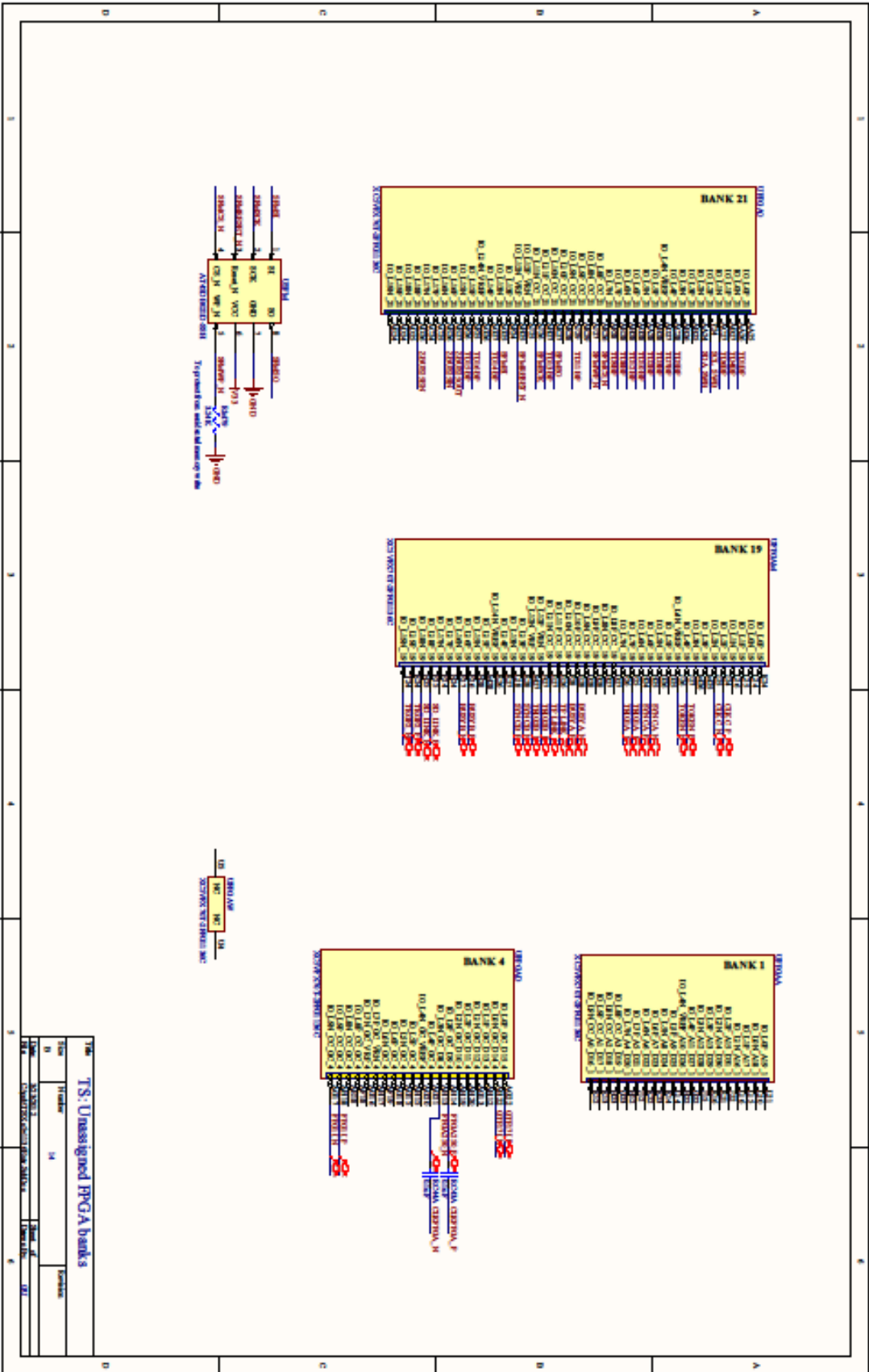


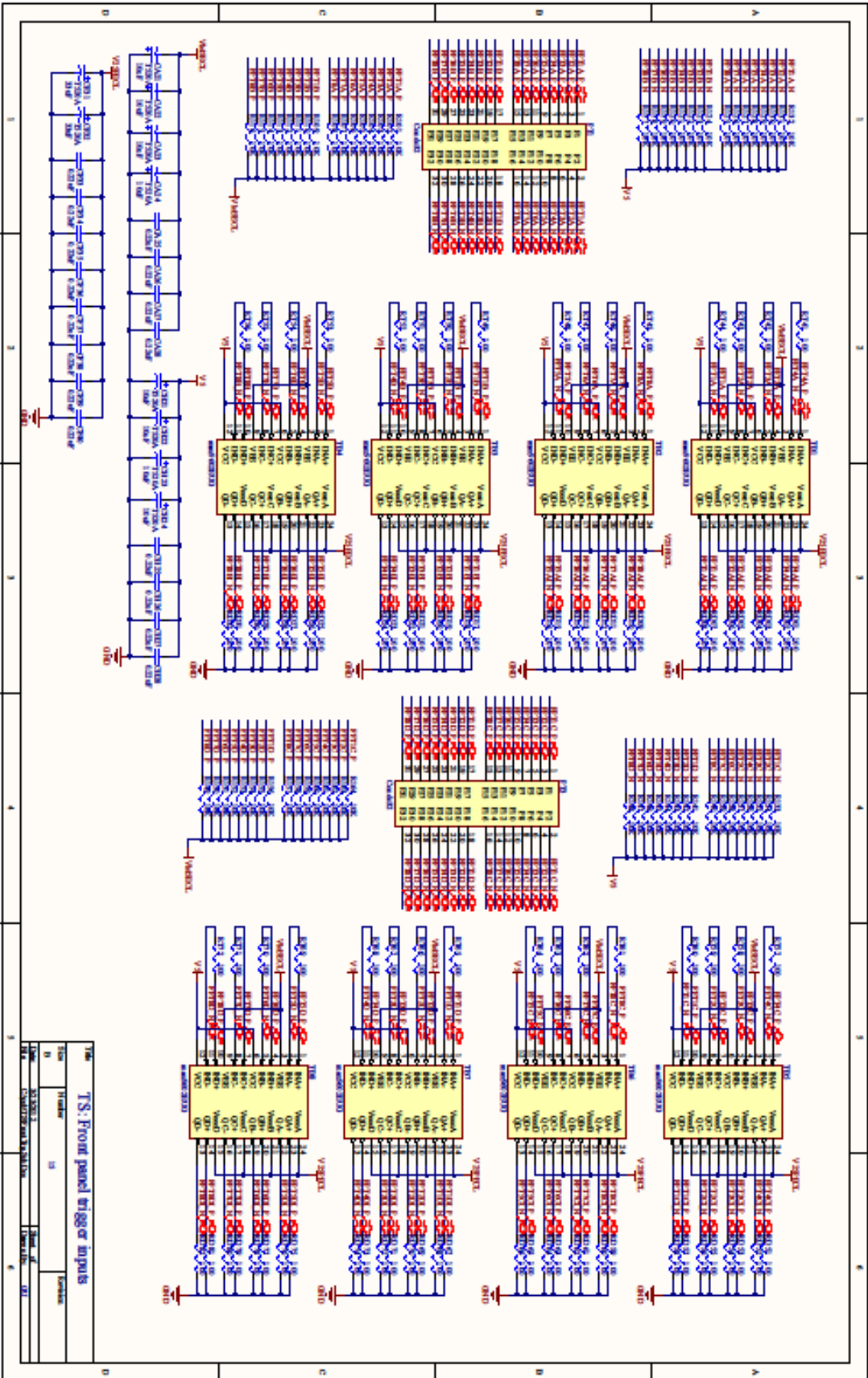






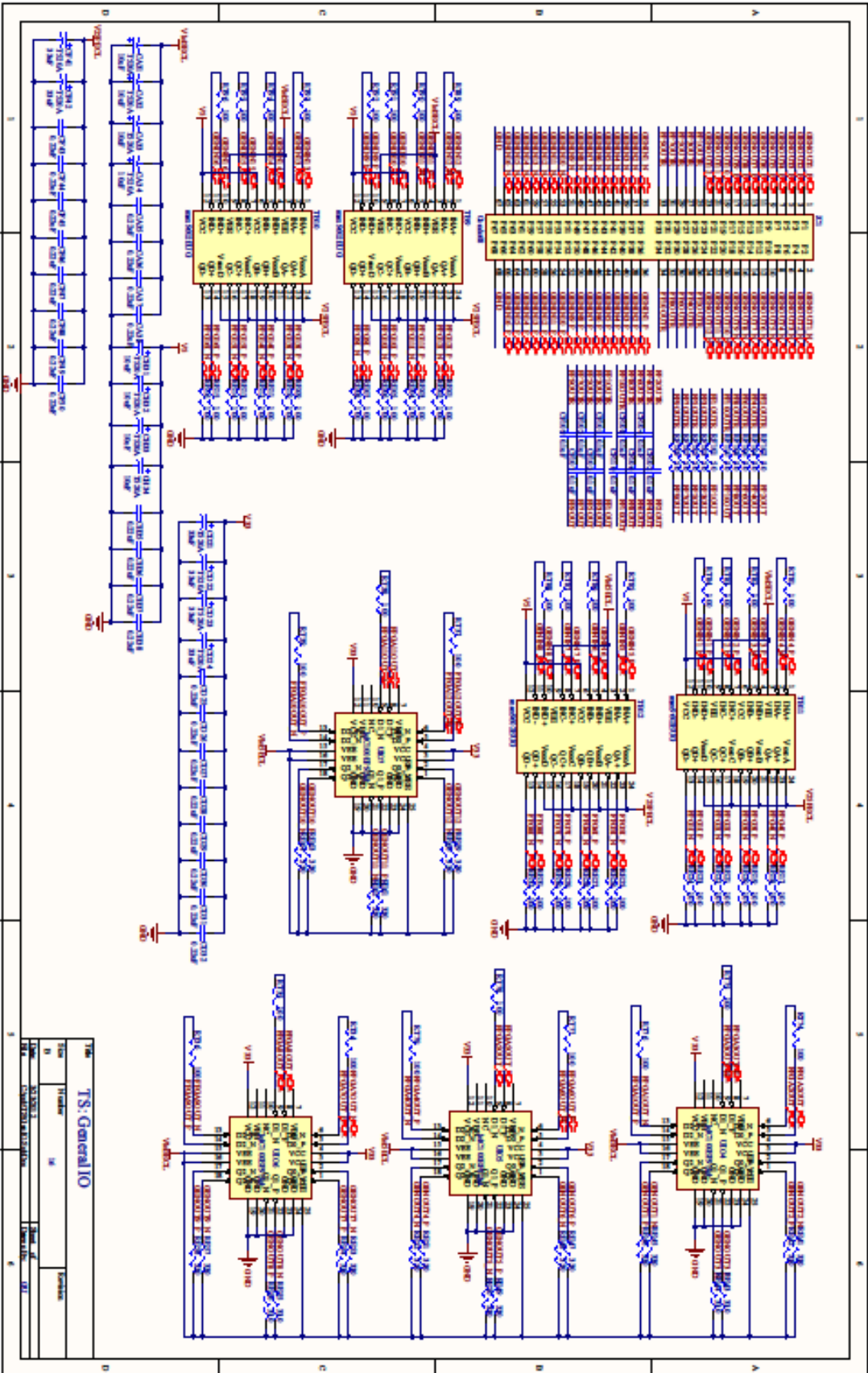




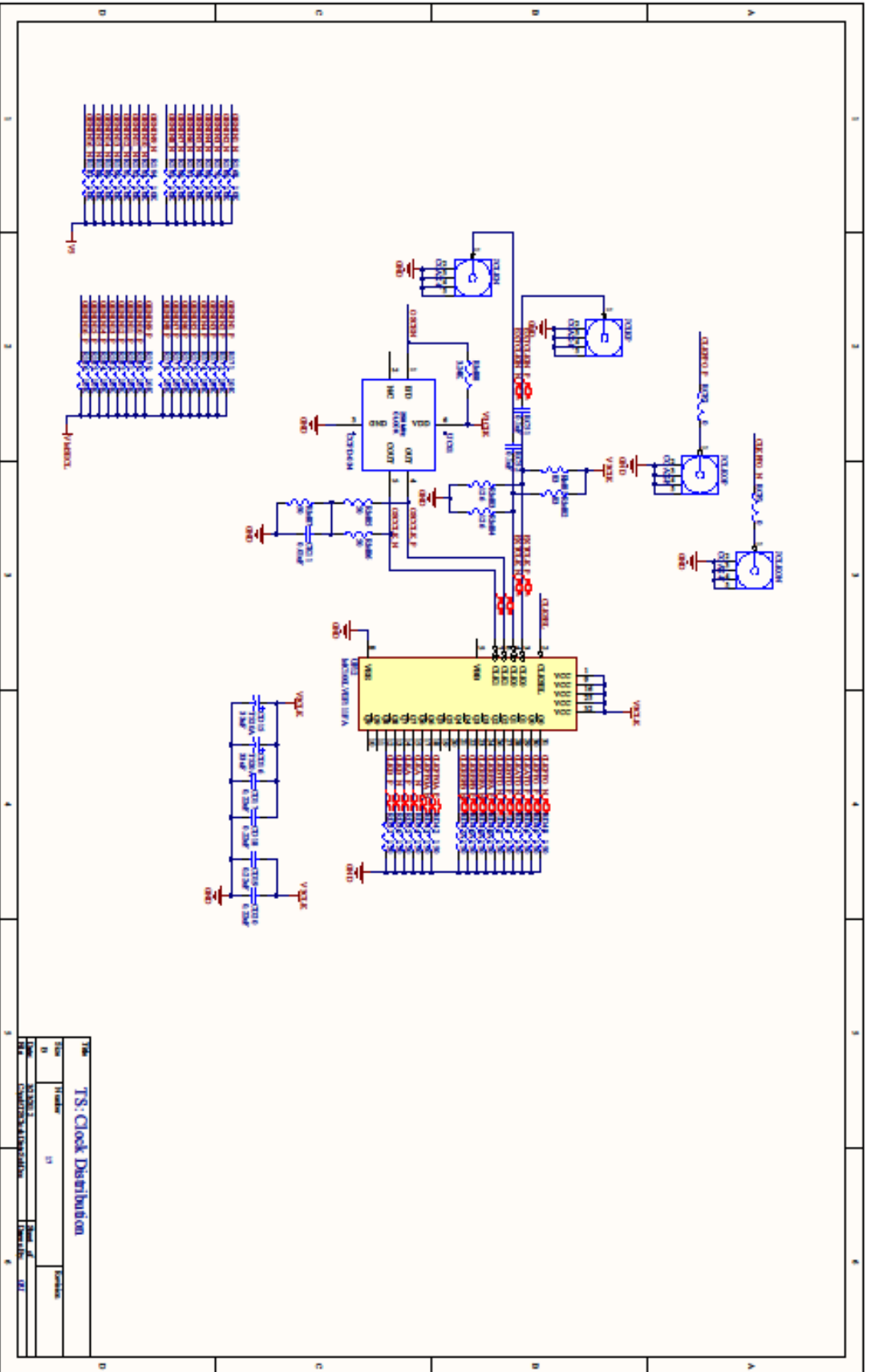


TS: Front panel trigger inputs

TS	Header	Pin	Location
TS	TS	1-16	TS







TS: Clock Distribution			
Item	Header	Pin	Location
1	74VHC04	1	U1
2	74VHC125	1	U2

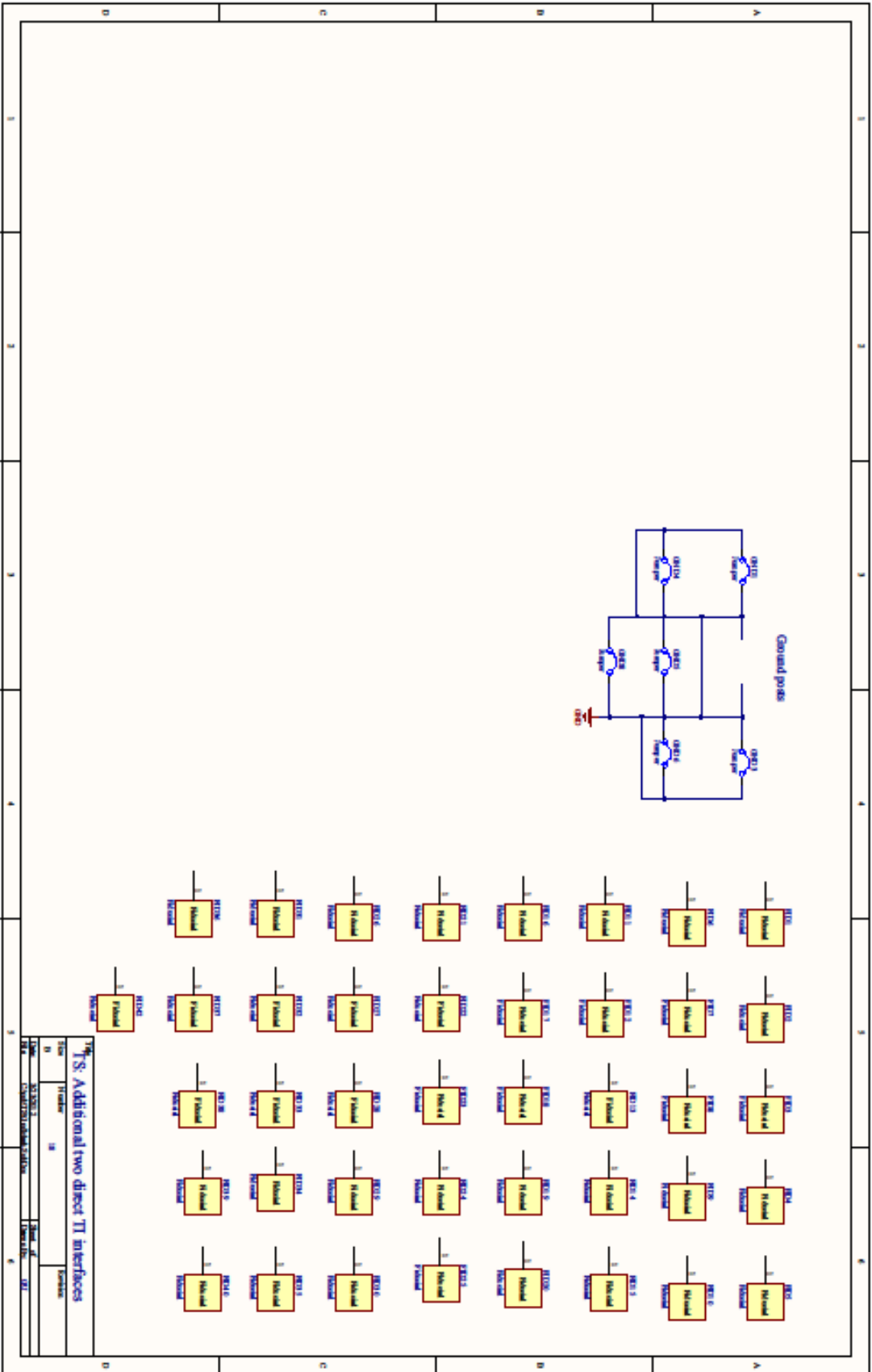
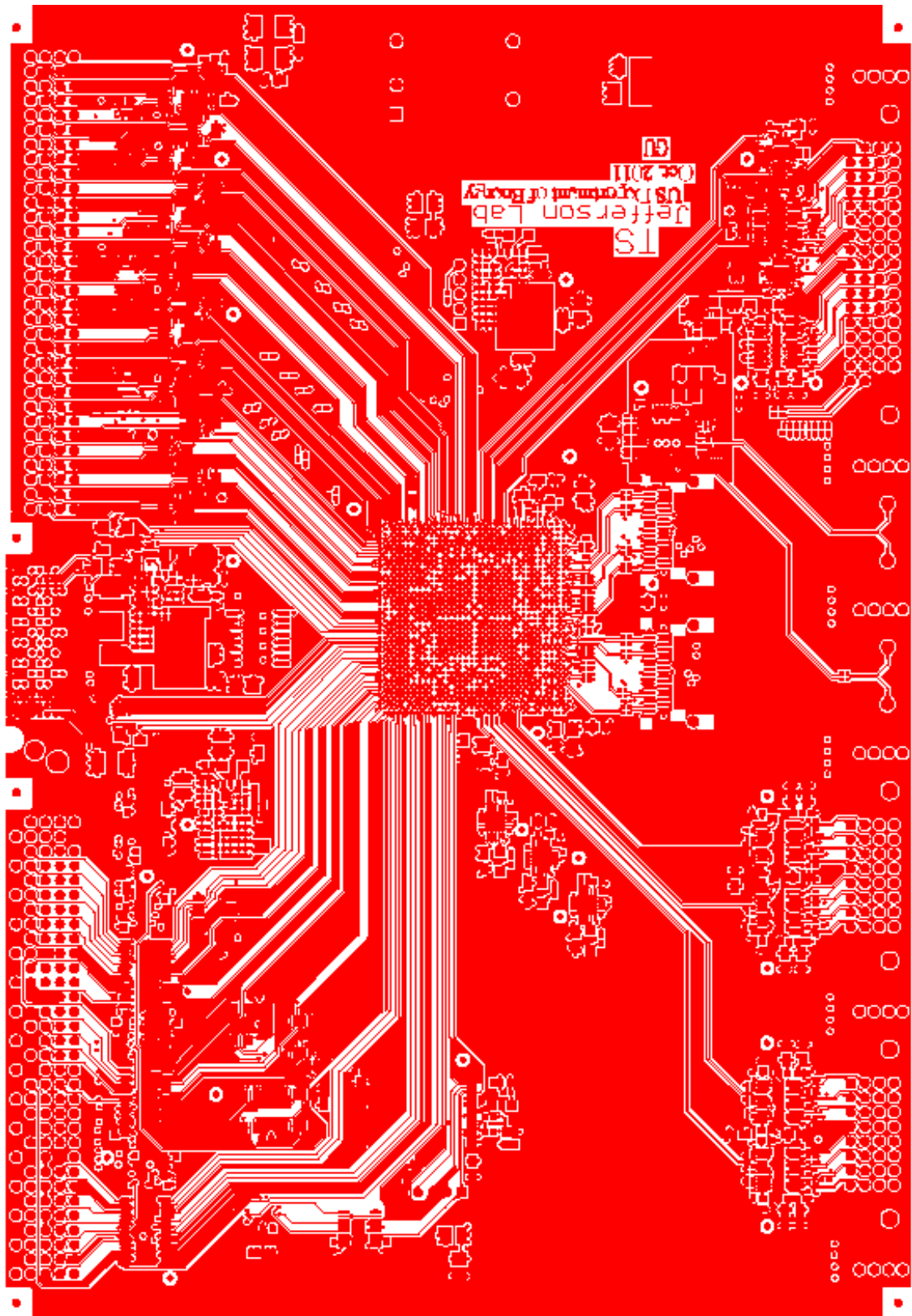


FIG. 18: Additional two direct TI interfaces

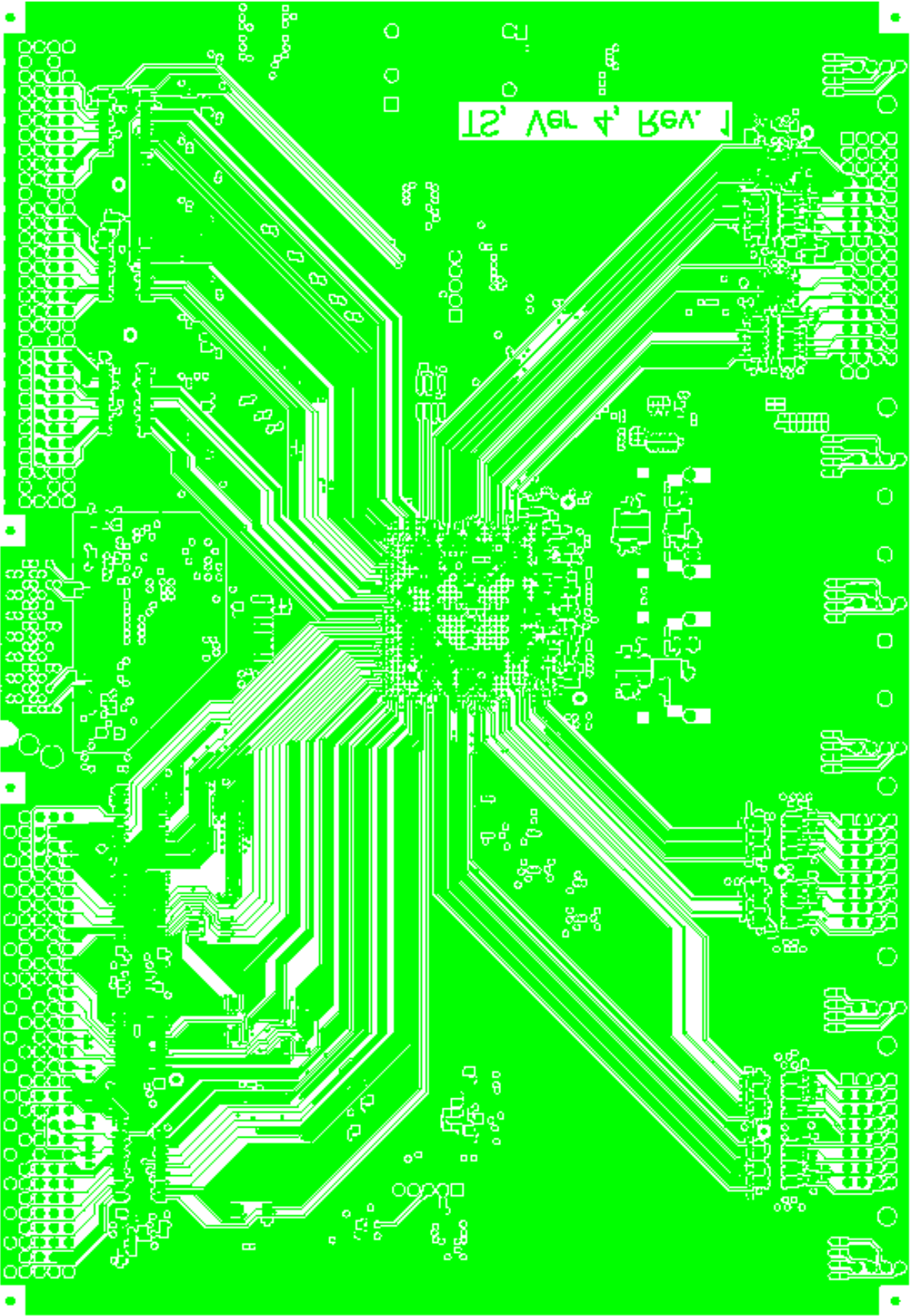
Item	TI number	Comment
1	18	

## Appendix B: TS Gerber files in PDF format



C:\naarct\dXP (TM) : ts.gtl

.gtl



## Appendix C: Bill of Material

Comment	Description	Footprint	Quantity	Manufacture Part Number
TS20A	Solid Tantalum Chip Capacitor, Standard T491 Series - Industrial Grade	A	78	Kemet TS20A336M006ATE070
0.22uF	Capacitor, Ceramic Chip Capacitor - Standard	0603	137	TDK C1608X7R1C224K
TS20B	Solid Tantalum Chip Capacitor, Standard T520 Series - Industrial Grade	B	12	Keme TS20B227M006ATE070
0.01uF	Capacitor (Semiconductor SIM Model)	0402	8	TDK C1005X7R1C103M
0.1uF	Capacitor	0402	40	TDK C1005X7R1C104K
LED2	Typical RED, GREEN, YELLOW/GaAs LED	3.2X1.6X1.1	8	Lite-On LTST-C150CKT
5.0A	FUSE 5A SLO BLO NANO 2 SMD	NANO_FUSE	6	TE Connectivity, 1206SFF500F/32-2
Fiducial		fiducial	41	
Header 6	Header, 6-Pin	HDR1X6	2	
Jumper	Jumper Wire	RAD-0.2	6	
Condo68		Condo68	1	3M, x3431-D302xx
COAX-F	RF Coaxial PCB Connector, MCX;	MCX5.08-H5	4	Amphenol, 132203-12
HC_3A	Inductor	1210	11	Taiyo Yuden, BRL3225T1R0M
HF_0.5A	Inductor	0603	21	
1.0uH	INDUCTOR 1.0UH 300MA 20% 0805	0805	6	
VME160-P1	VME160-P1	VME160	1	Harting, 02 01 160 1101
VME160-P2	VME160-P2	VME160	1	Harting, 02 01 160 1101
1410147-1	PD-PL-VXS	PD-105_PD-PL-VXS	1	Tyco, 1410147-1
Condo32		Condo32	2	3M, x3408-D302xx
QSF3P		QSF3P38	2	FCL 10099113-101LF socket; 10099114-001LF Cage; Avago, AFBR-79Q4Z Transceiver
10K	RES 0 OHM 1/16W 1% 0402 SMD	0402	96	
1.02K	Resistor	0603	48	
3.24K	Resistor	0603	56	
50	RES 0 OHM 1/16W 1% 0402 SMD	0402	32	
100	RES 0 OHM 1/16W 1% 0402 SMD, RES 100 OHM 1/16W 1% 0402 SMD	0402	201	
150	RES 0 OHM 1/16W 1% 0402 SMD, 5% Tolerance, 0402 Size, 0.063 W	0402	30	
0	RES 0 OHM 1/16W 1% 0402 SMD, 5% Tolerance, 0402 Size, 0.063 W	0402	44	
330	RES 0 OHM 1/16W 1% 0402 SMD	0402	26	
83	RES 0 OHM 1/16W 1% 0402 SMD	0402	2	
126	RES 0 OHM 1/16W 1% 0402 SMD	0402	2	
510	RES 0 OHM 1/16W 1% 0402 SMD	0402	5	
SW DIP-8	DIP Switch, 8 Position, SPST	SW16_L	1	
max9602EUG		MTC24_N	12	maxim, MAX9602EUG
SN74ABTE16246DGGR	IC 11-BIT I <sup>2</sup> S BUS TXRX 48-TSSOP	TSSOP50P810-48AL	1	
CCPD-034	CCPD-034	SO6-SX7	1	
SN74LVC10APW		TSSOP14	4	
SN74LVC86APW		TSSOP14	2	
74LVC27PW		TSSOP14	2	
SN74LVC04APW		TSSOP14	1	
SN74LVC240APW		TSSOP20	1	
MC100LVEP111FA	Low-Voltage 1:10 Differential LVPECL/LVPECL/HSTL Clock Driver	873A-02_N	1	
XCSVF70T-2FFG1136C	Xilinx Virtex-5 FXT Platform FPGA, 1136-Ball FFPGA, Speed Grade 2, 640 User I/Os, Commercial Grade, Pb-Free	FFG1136	1	Xilinx, XCSVF70T-2 FFG1136C
SY55855V	Dual LVPECL to LVDS translator	TSSOP50P490-10AN	16	
MC100EP91M	On Semiconductor, Any level positive to ECL translator	QFN50P400X400-24W4M	4	
Quad_led_SSF-147GD	LUMEX, Quad pack LEDs	LED14	6	
LTM4604EV	LTM4604EV	LGA-66_LTM4604EV(Primary)	3	Linear Technology, LTM4604AEV#PBF
TPS74401RGWT	IC LDO REG 3.0A W/S 20-VQFN	QFN-20	4	
SY55857L	Micrel dual Anylevel to LVPECL translator	TSSOP50P490-10AN	4	
EC5BE11		EC5BE17	1	
kc2520-33MHz	Miniature Oscillator	CFPX-5	1	
XCF32PVO48C	XCF00P Series, Platform Flash In-System Programmable Configuration 1.8V PROM, 48-Pin TSSOP, 32-Megabit, Commercial Grade	VO48	1	
AT45DB021D-SSH	Atmel flash memory	8S1_N	1	
SN74VMEH22501DGGR	IC UNIV BUS TXRX TRI-ST 48-TSSOP	TSSOP50P810-48AL	9	

## Appendix D: TS data format:

The TS data is formatted in blocks of events. Each Trigger\_1 is one event. A block of data contains a predefined number (the number is set by A24 register offset 0x14, it could be 1) of triggers. Each block has block header, block trailer, possible place holder, and event data. The data format is summarized here:

Block headers

Event#1 data

Event#2 data

.....

Event#N data (N is set by register 0x14, the block size)

Block trailer

Filler words for 2ESST transfer

Block Header#1:

bit(31:27): 10000, block header indicator;  
Bit(26:22): BoardID, the VME64x geographic address (slot number);  
Bit(21:18): 0101, ID for TS board;  
Bit(17:8): block number, lower ten bits;  
Bit(7:0): block size (as set by A24 register 0x14);

Block Header#2:

Bit(31:17): 1111,1111,0001,000X; or 0xFF1X;  
Bit(16): TimeStamp, 1 if timestamp is available, 0 if not;  
Bit(15:8): 0010,0000, or 0x20;  
Bit(7:0): Block size;

Block Trailer#1:

Bit(31:27): 10001, block trailer indicator;  
Bit(26:22): BoardID, VME64x geographic address (slot number);  
Bit(21:0): Word count; does not include block header or trailer.

Extra word:

Bit(31:27): "11111";  
Bit(26:22): BoardID, VME slot number, or VME64X geographic address  
Bit(21:16): "000000";  
Bit(15:0): Block Number;

Filler words:

Word pattern: 0x to make the word count an even number  
Word pattern: 0x to make the word count 'even' for 2ESST  
DataNotValid:

Bit(31:27): 11110;  
Bit(26:22): BoardID  
Bit(21:0): 00,0000,1011,1010,1101,0000, or 0x00BAD0;

Filler word:

Bit(31:27): 1111,1;  
Bit(26:22): BoardID;  
Bit(21:0): 00,1111,0001,0001,0001,0000; 0x0F1110 (0\_fill\_0)

Event data word1: (event header)

Bit(31:24): Trigger Type;  
0: filler events,  
1-32: GTP Physics trigger;  
33-64: Front panel physics trigger;  
250: multi-hit on GTP or Front panel physics trigger  
251: Multi-hit on GTP and Front panel physics trigger  
253: VME trigger;  
254: VME random trigger;  
For TImaster bit(31:26) represents the TS#6-1 inputs;  
Bit(23:16): 0000,0001, or 0x01;  
Bit(15:0): Event wordcount; Event header is excluded from the count

Event data word2:

Bit(31:0): trigger number; counting from 1 to be consistent with wrap around;

Event data word3: (enabled by data format register 0x18, bit#1)

Bit(31:0): trigger timing; 4ns step

Event data word4: (enabled by data format register 0x18, bit#2)

Bit(31:28): Data available for FASTBUS DAQ duplication branches

Bit(27:20): Local trigger number counter since last Readout Trigger.

Bit(19:16): trigger number bit(35:32), to form 36 bit trigger counter with word2;

Bit(15:0): trigger timing bit(47:32), to form 48 bit timing counter with word3;

Event data word5: (enabled by data format register 0x18, bit#3)

Bit(31:0): GTP trigger input, pattern after signal stretching;

Event data word6: (enabled by data format register 0x18, bit#4)

Bit(31:0): Ext. front panel trigger input, after signal stretching;

Event data word7: (enabled by data format register 0x18, bit#5)

Bit(31:16): "0000,0000,0000,0000";

Bit(15:0): front panel trigger input, after signal stretching;



## Appendix D.2: TS partition (subTS) data format:

The TS partition data is formatted in blocks of events. Each sub\_TS Trigger is one event. A block of data contains a predefined number (the number is set by A24 register offset 0x14, it could be 1) of triggers. Each block has block header, block trailer, possible place holder, and event data. The data format is summarized here:

### Block headers

Event#1 data

Event#2 data

.....

Event#N data (N is set by register 0x14, the block size)

### Block trailer

### Filler words for 2ESST transfer

#### Block Header#1:

bit(31:27): 10000, block header indicator;  
Bit(26:22): BoardID, the VME64x geographic address (slot number);  
Bit(21:18): 0101, ID for TS board;  
Bit(17:8): block number, lower ten bits;  
Bit(7:0): block size (as set by A24 register 0x14);

#### Block Header#2:

Bit(31:17): 1111,1111,0001,000X; or 0xFF1X;  
Bit(16): TimeStamp, 1 if timestamp is available, 0 if not;  
Bit(15:8): 0010,0000, or 0x20;  
Bit(7:0): Block size;

#### Block Trailer#1:

Bit(31:27): 10001, block trailer indicator;  
Bit(26:22): BoardID, VME64x geographic address (slot number);  
Bit(21:0): Word count; does not include block header or trailer.

#### Extra word (Filler words):

DataNotValid, read data when the data buffer is empty (no more data):

Bit(31:27): 11110;

Bit(26:22): BoardID

Bit(21:0): 00,0000,1011,1010,1101,0000, or 0x00BAD0;

Filler word #1 to make the word count an even number for D64 transfer:

Bit(31:27): 1111,1;

Bit(26:22): BoardID;

Bit(21:0): block number;

Filler word #2 to make the word count 'even' for 2ESST:

Bit(31:27): 1111,1;

Bit(26:22): BoardID;

Bit(21:0): 00,1111,0001,0001,0001,0000; 0x0F1110 (0\_fill\_0)

Event data word1: (event header)

Bit(31:24): Trigger Type;

0x00: filler events,

0x01: Physics trigger;

0x02: VME trigger;

0x03: Random trigger,

For TImaster bit(31:26) represents the TS#6-1 inputs;

Bit(23:16): 0000,0001, or 0x01;

Bit(15:0): Event wordcount; Event header is excluded from the count

Event data word2:

Bit(31:0): trigger number; counting from 1 to be consistent with wrap  
around;

Event data word3: (enabled by data format register 0x18, bit#1)

Bit(31:0): trigger timing; 4ns step

Event data word4: (enabled by data format register 0x18, bit#2)

Bit(31:16): "0000,0000,0000,0000", 32-bit for subTS event number;

Bit(15:0): trigger timing bit(47:32), to form 48 bit counter with word3;

## **Appendix E: Revision history:**

First version on Oct. 21<sup>st</sup>, 2010

Nov. 8, 2010: updated to reflect the recent discussions and comments

Sept. 30, 2011: Updated according to the PCB design

Mar. 21, 2012: Major updates to reflect the hardware/firmware development

May 9, 2012: minor updates on A24 registers

Aug. 23, 2012: Major updates, with the addition of TS partitioning (Sub\_TS#1, #2, #3 and #4)

Jan. 8, 2013: proofreading the manual, no major changes

Feb. 25, 2013: Updated for TSrev2 compatibility. (I2C addition, optic transceiver reset/monitor)

Mar. 20, 2013: Updated for front panel LED output, section 7.5 added.

Apr. 23, 2013: Updated register offsets: 0x24, 0x78, 0x98, 0xD4, 0xD8, 0x100. Added section 3.13  
“SyncEvent generation”

June 26, 2013: updated register offsets: 0x100, 0x134, 0x158, 0x334, 0x358, 0x534, 0x558, 0x734, 0x758,  
mainly the TS partition related registers.

July 22, 2013: updated register offsets: 0x18 for event data format.

July 31, 2013: updated register offset 0xD8 and 0xDC for event number readout.

Sept. 13, 2013: updated register 0x34, bit#28

Sept. 27, 2013: Added “Appendix D: TS Data format”

Nov. 14, 2013: Added the A24 offset 0x10BC - 0x1080 for the simplified trigger table loading.

Dec. 3, 2013: Added the A24 offset 0xE0 – 0xE8 for CrateID and trigger source enable read back.

Nov. 11, 2014: Added the simplified event type lookup table schematics as the last part of section3. And  
other minor changes.

Mar. 9, 2015: some VME register updates, mainly the readout;

Apr. 13, 2015: updated the front panel trigger input channel delay setting, Channel output pin assignments.  
Registers affected: 0x104-0x12C, 0x1C etc.

Apr. 30, 2015: Updated for HallA SBS local trigger addition.

May 7, 2015: Updated for Buffer/Block readout, Section#3.17; register 0x70, 0xA0, 0xA4, 0x118 and  
0x11C; readout dataformat