



Nuclear Physics Division
Data Acquisition Group

Description and Technical Information for the VME Trigger Interface (TI) Module

J. William Gu (jgu@jlab.org)

Updated on: April 15, 2025

Table of Contents

Section	Title	Page
1	Introduction	3
2	Purpose of TI Module	4
3	Function Description	7
4	Specifications	19
5	TI operation	21
6	Programming Requirements	24
7	VXS P0 and VME P2 Row-C Pin outs	39
8	Examples of individual test	42
9	Software setup for TI	43
10	Citation	43
Appendix A	Schematics	44
Appendix B	Fabrication Drawing	62
Appendix C	Bill of Materials	65
Appendix D	Glossary	71
Appendix E	TI data format	72
Appendix F	Revision history	74

1 Introduction

The Trigger Interface (TI) module is being designed for the Jefferson Lab 12GeV upgrade, mainly for HallD [(Collaboration G. , 2009)] and Hall-B [(Collaboration C. , 2009)], with other experimental Halls [(experiments, 1990)] compatibility. The TI boards are located in data acquisition frontend crates, and are responsible for providing a low-jitter system clock, sync signals and fixed latency trigger signals for the Front-end readout boards in the crates. The modules also merge the front-end crate status and generate a BUSY signal to request the Trigger Supervisor (TS) to pause the trigger. For the detailed description, refer to the Trigger Distribution system design document [Trigger Distribution]. Figure 1a shows the placement of the TI modules in the global trigger distribution scheme in experiment setup. Figure 1b shows the crate level diagram

Figure 1a:

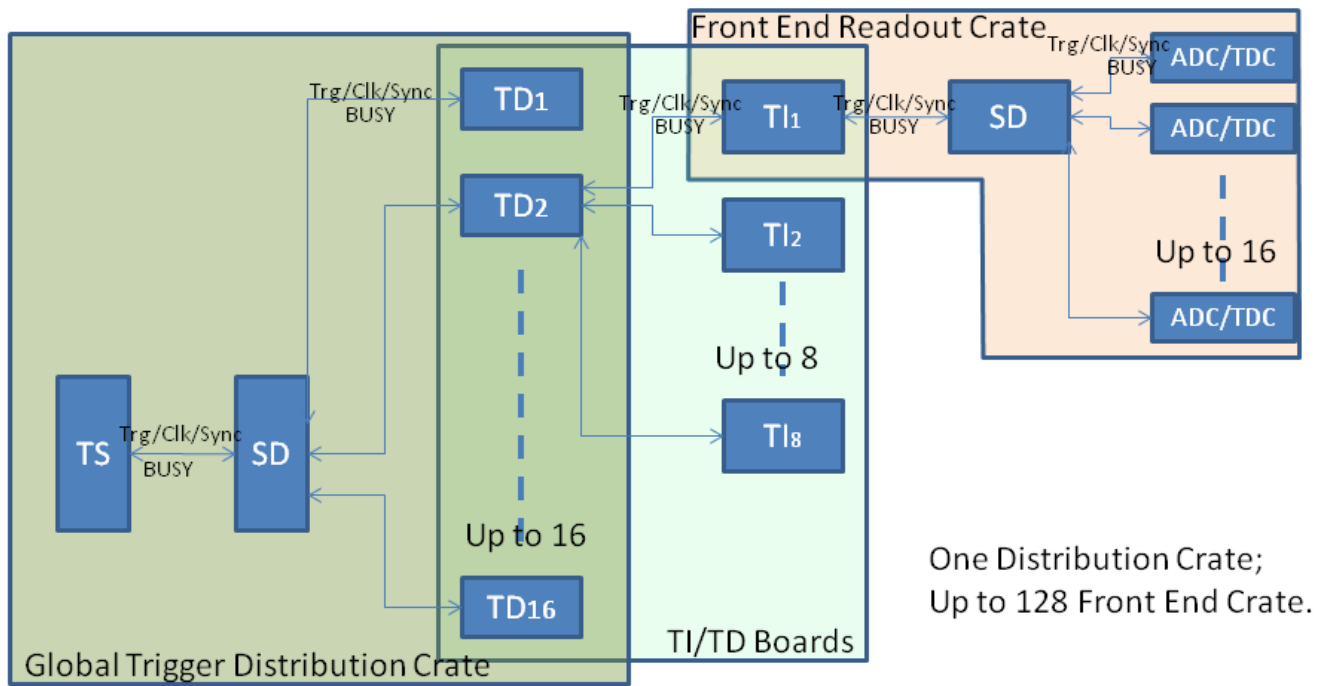
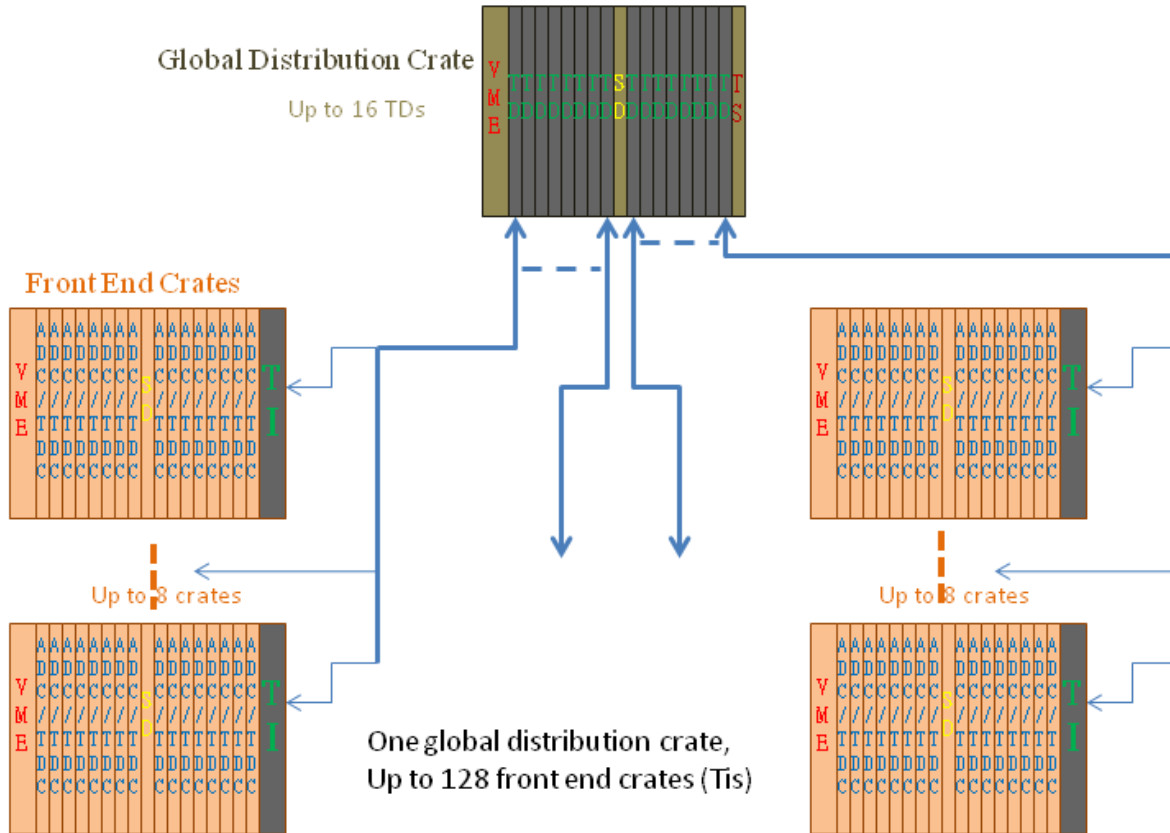


Figure 1b:



The TI shares the same PCB design as the Trigger Distribution (TD) board. The TI board has simple Trigger Supervisor (TS) functions built in. The TI can act as a subsystem trigger supervisor when used independently. The TI board can have optionally eight optic fibers populated, and distribute Trigger/Clock/Sync to another 8 (eight) TI boards. A subsystem with up to nine crates can be setup without a real TS module. The TI also has the flexibility to select the trigger and clock inputs from central trigger system or local (sub-system) trigger system.

2 Purpose of the module

TI board is positioned in the last slot of the front-end data acquisition crate, and connects to a TD module in the global trigger distribution crate. This is done using a four channels full-duplex fiber link, which provides a gigabit trigger link, global 250MHz clock and synchronization link to phase lock the trigger and initialize the system. The trigger link uses a reference clock derived from the 250MHz global pipeline clock allowing a trigger word to be distributed every 4 global pipeline clock cycles. Depending on the trigger word type the TI can issue a crate level trigger condition through the VXS switch port B. The trigger bits are sent to a Signal Distribution (SD) module that distributes these signals to all front-end modules in the crate. The TI also accepts status signals from the SD module, which is the logical OR of the status signals from all the other front-end modules. These status signals can be transmitted back to the TS through the fiber link to slow down or inhibit further distribution of triggers until the status has been resolved. For now, the status includes a BUSY only. When the BUSY is set, the front end boards are requesting for trigger inhibit. The assertion of a status signal will create a dead-time in the data acquisition system which should be a rare occurrence since the data acquisition is being designed to handle the full trigger rate that can occur from the physics events. The dead time will be monitored and recorded by the Trigger Supervisor board.

The TI can also perform simple Trigger Supervisor functions. In test setup or commissioning setup, the TI can perform as a TI Master (TM). In this case, it can take inputs from its front panel and generate trigger/clock like a TS, it can send the trigger, clock, and SYNC to the backplane (P0 and P2) like a TI, and it can send the trigger, clock and SYNC to the front panel fibers like a TD. With the TM, a test setup can include up to nine (9) crates. Figure 2a shows the setup. Figure 2b shows the crate level diagram.

Figure 2a

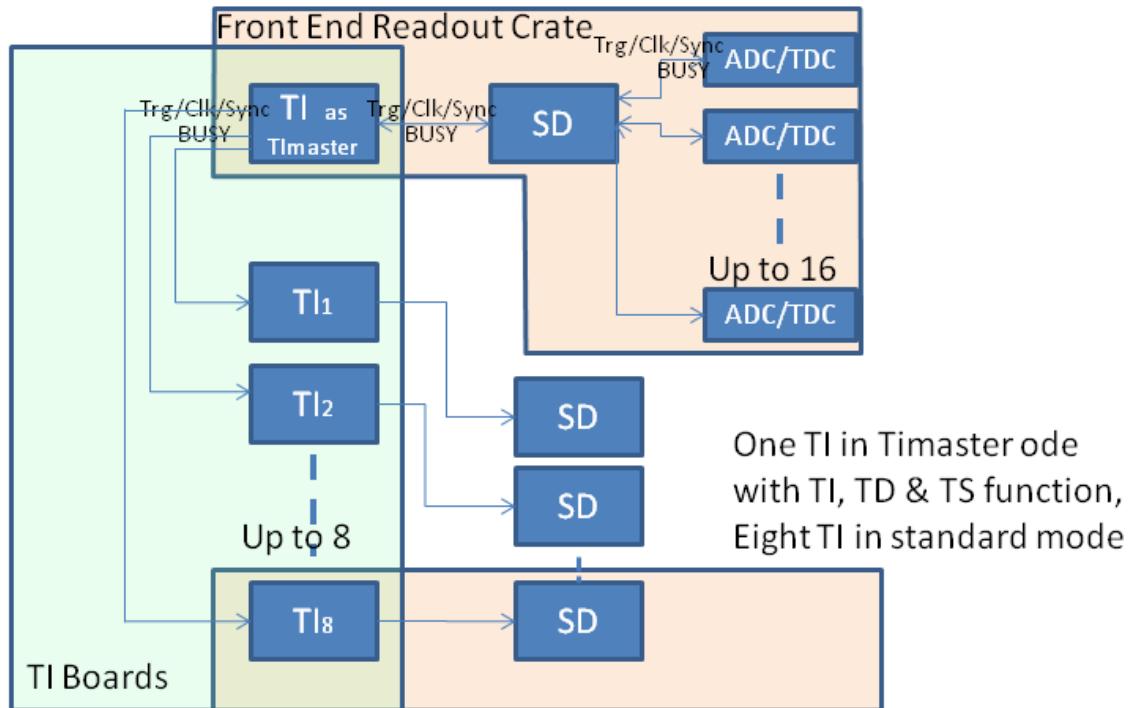
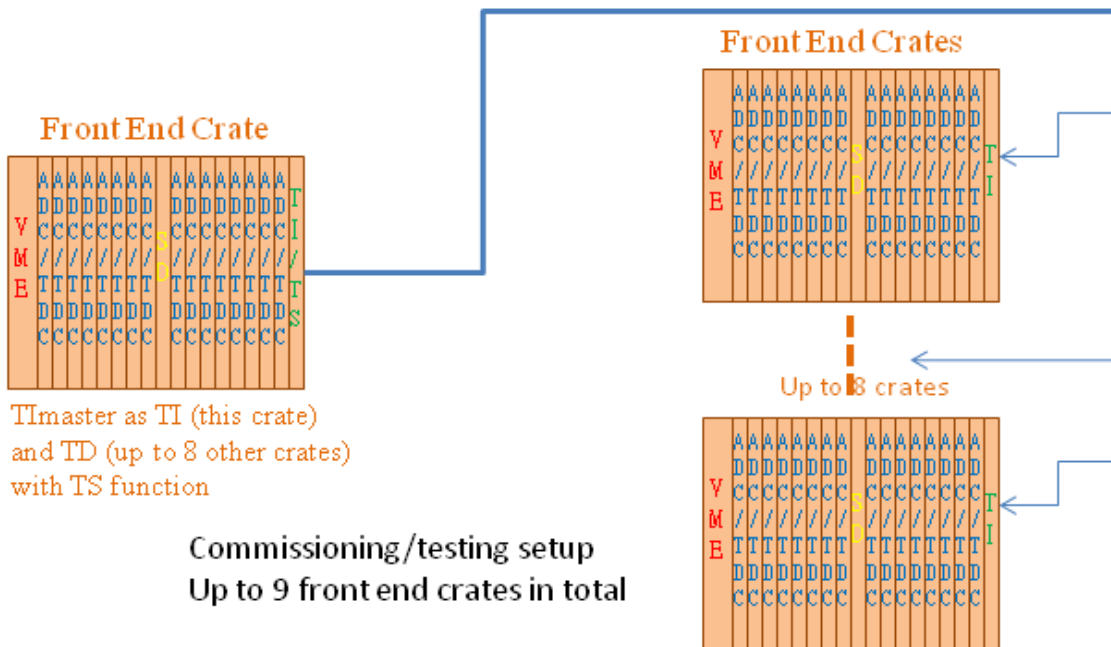


Figure 2b



The TI FPGA firmware is compatible with the TImaster mode and the standard TI mode. The firmware senses the TI clock setting. If the TI clock source is set to onboard oscillator or front panel ECL input, the TI is in TM mode, or else, the TI is in standard mode. For the TM mode, the TRG/SYNC used in the TM is similar to the TRG/SYNC sent in the optical transceiver. The TRG/SYN is looped back within the FPGA and decoded the same way as a standard TI board. Some registers are valid on the TM only, as these registers are specific to the TS function. The TM can be used by itself in the setup. It can also drive another up to eight TIs if the setup needs be expanded.

When the TI is in standard TI mode, it can have two HFBR-7924 modules stuffed. In this case, one can receive trigger from system TS, while the other can receive trigger from a sub-system TS. One TID can be located in any payload slots in a front-end crate, and configured as a subsystem TS. This configuration requires an extra optical transceiver on the TI module, and an extra TID board be configured as a subsystem TS with fiber outputs. Each optical transceiver module costs about \$400. For a subsystem with eight crates, the extra cost is about \$10K (extra optic fiber included). The benefit is that the subsystem can have independent control without affecting the other subsystem. That is a lot of flexibility: independent subsystem control, independent subsystem calibration etc. This is called a luxury option. Refer to figure 3a for the setup. Figure 3b shows the crate level diagram

Figure 3a:

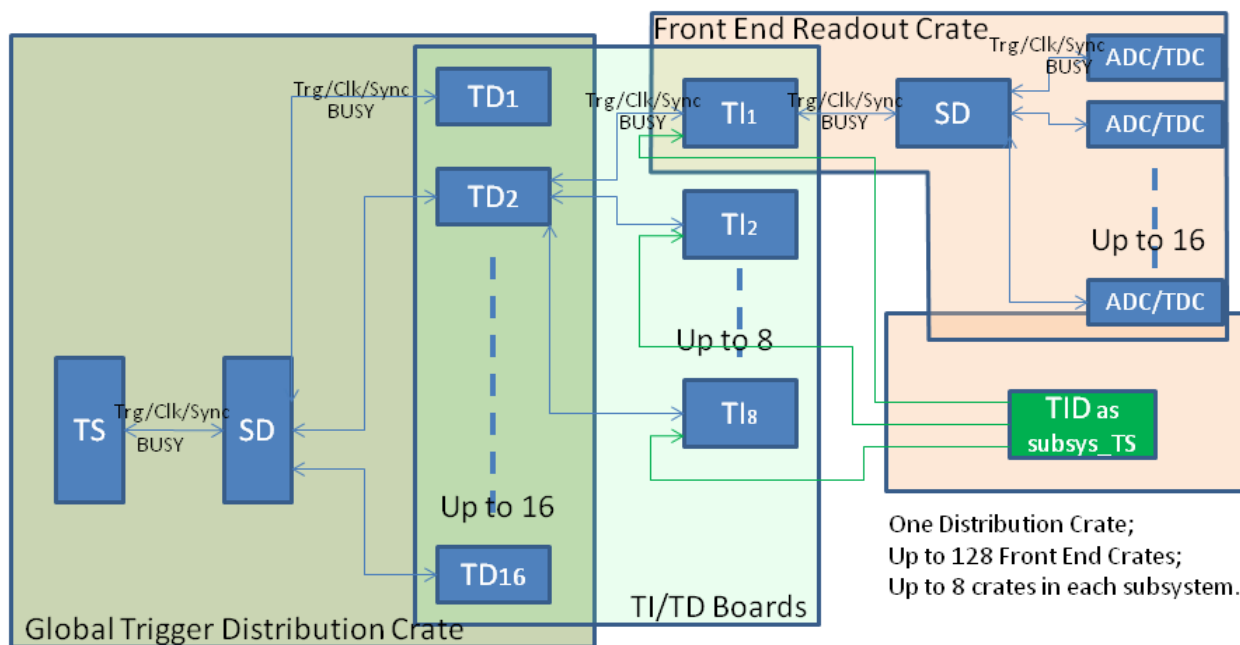
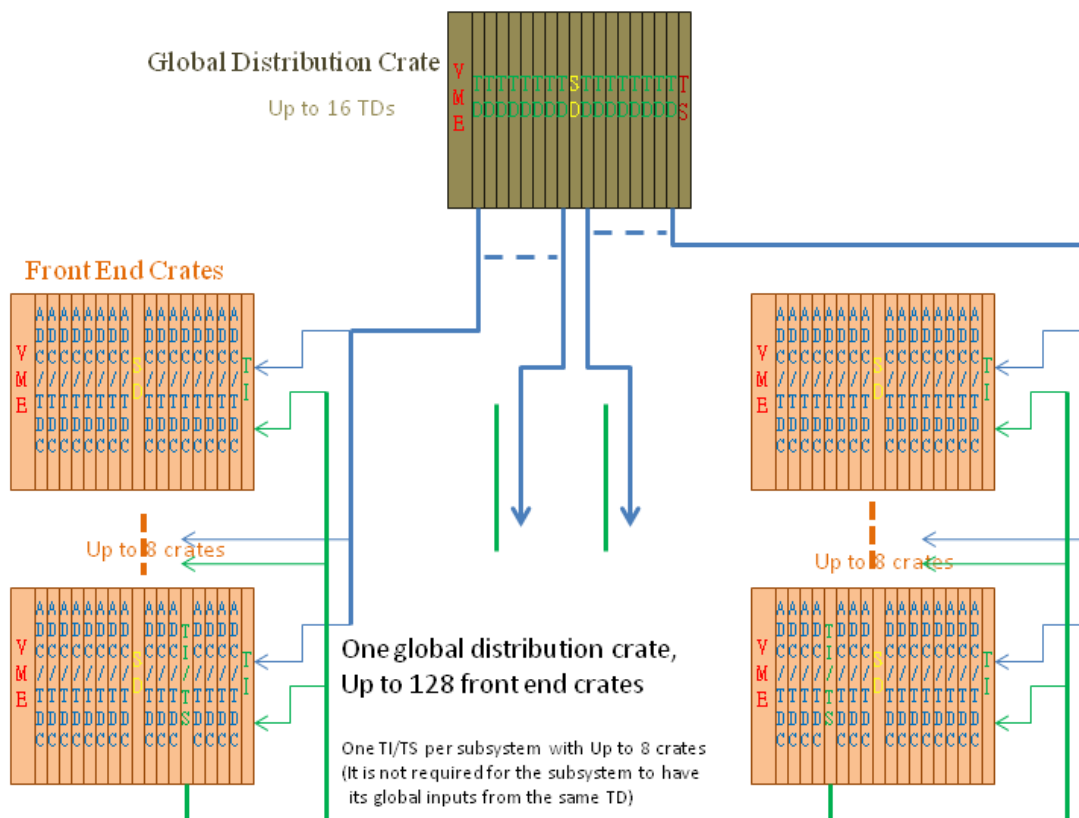


Figure 3b:



There is another direct connection between the TI and switch slot A board (CTP or GTP) and switch slot B board (SD). This link can be used to transfer data between the TI board and the switch slot boards. It is natural to implement the link at 250 Mbps using the system clock, or 500 Mbps with DDR technique. If the Data acquisition function is built in the SD, the SD can collect and assemble events at the trigger by trigger basis, which provides a redundant data readout path. The SD/CTP can send their data to the TI, and the TI combines the data with its own data event by event. The data can be readout through VME (there is no direct VME access to the switch slot A and switch slot B)

With a mezzanine board, the TI is backward compatible with Trigger Supervisor Rev2 module. The TI with the mezzanine card will behave the same way as TI_Rev2 (produced in the year of 2001) board with added functionality.

For non-VXS crate (or non-VXS data acquisition modules), the TI will send trigger/clock/Sync to Row-C on VME P2 connector. With a fan out board, the trigger/clock/SYNC are distributed to the frontend data acquisition modules (CAEN VX1290 TDC for example), and the BUSY from these modules are summed in the fan out module, and sent back to the TI.

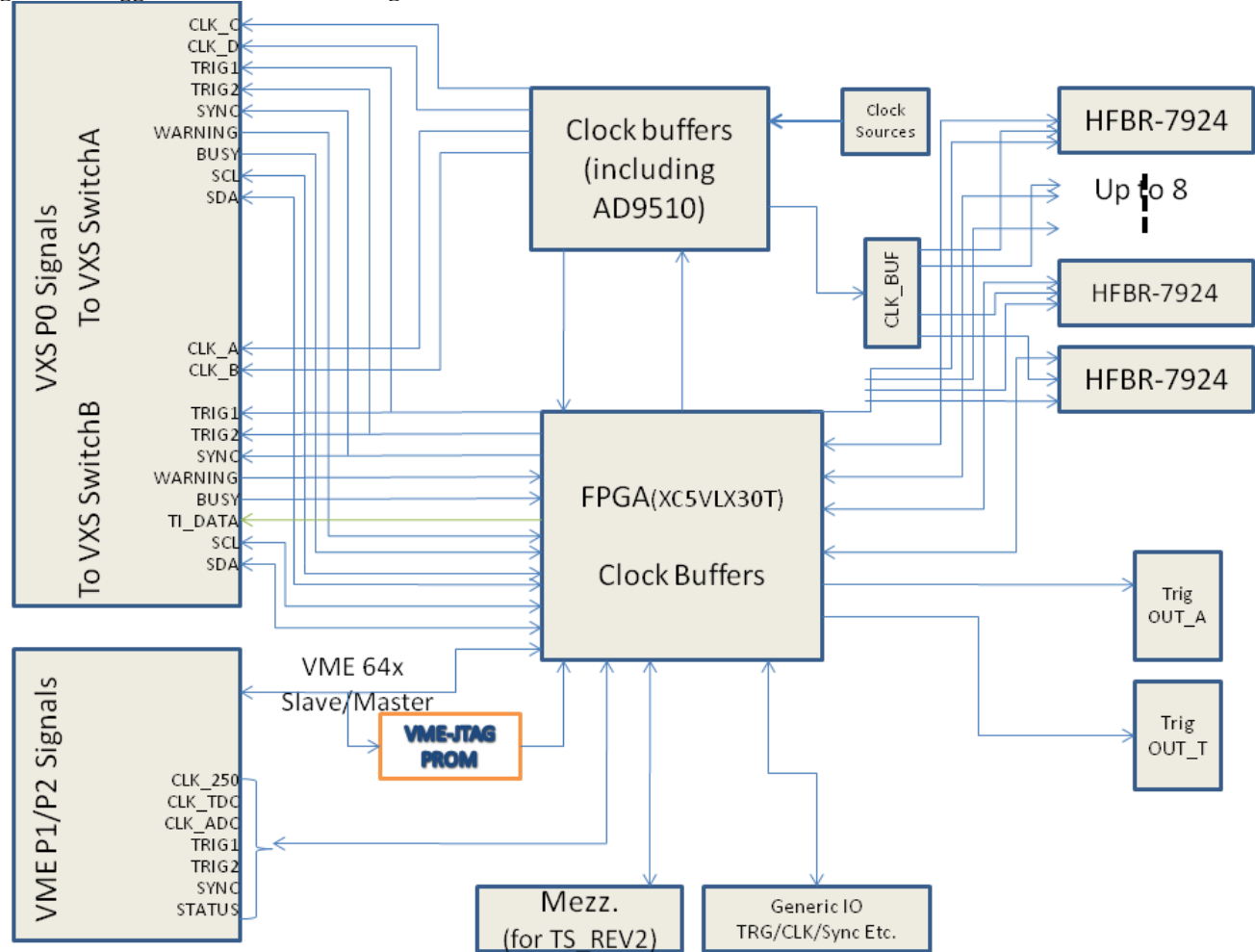
3 Functional Descriptions

3.1 General description

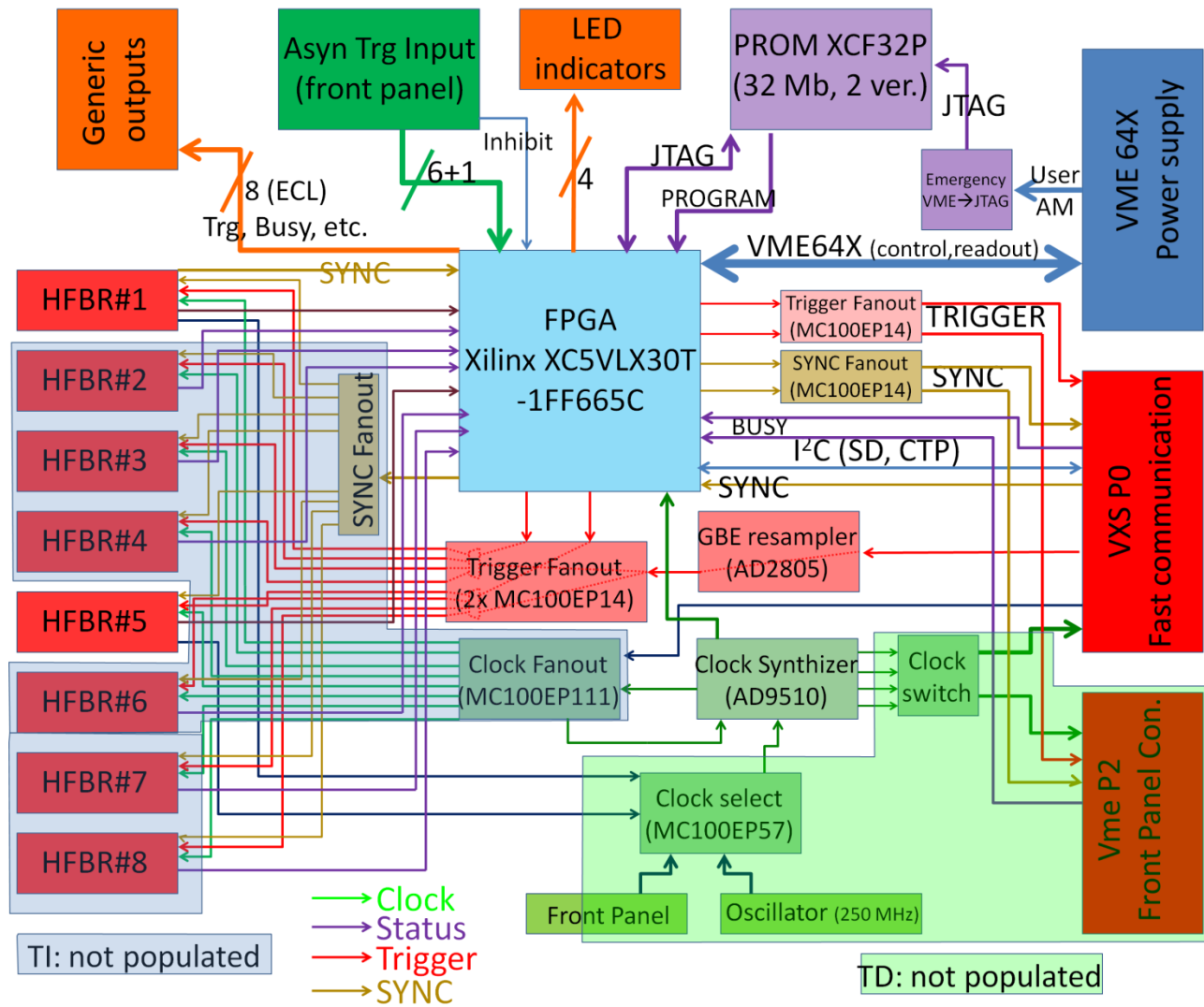
Figure 4 shows the block diagram of the TI module, indicating the major components used in the design. The HFBR-7924 is the multi-channel (4 Rx, 4 Tx) fiber link that the TI fans out/receives a low-jitter (<3ps RMS) 250MHz global pipeline clock, serialized 16bit trigger words, and a sync signal used in producing a synchronized trigger. The AD9510 is the main clock driver and gets synchronized lower frequency clocks. The Xilinx

XC5VLX30T is used to encode/decode the trigger words at 16ns, to interface with the VME, to control the working mode etc. The P0 is compatible with the VXS payload slots, which matches with GTP, CTP, SD positioned in switch slots.

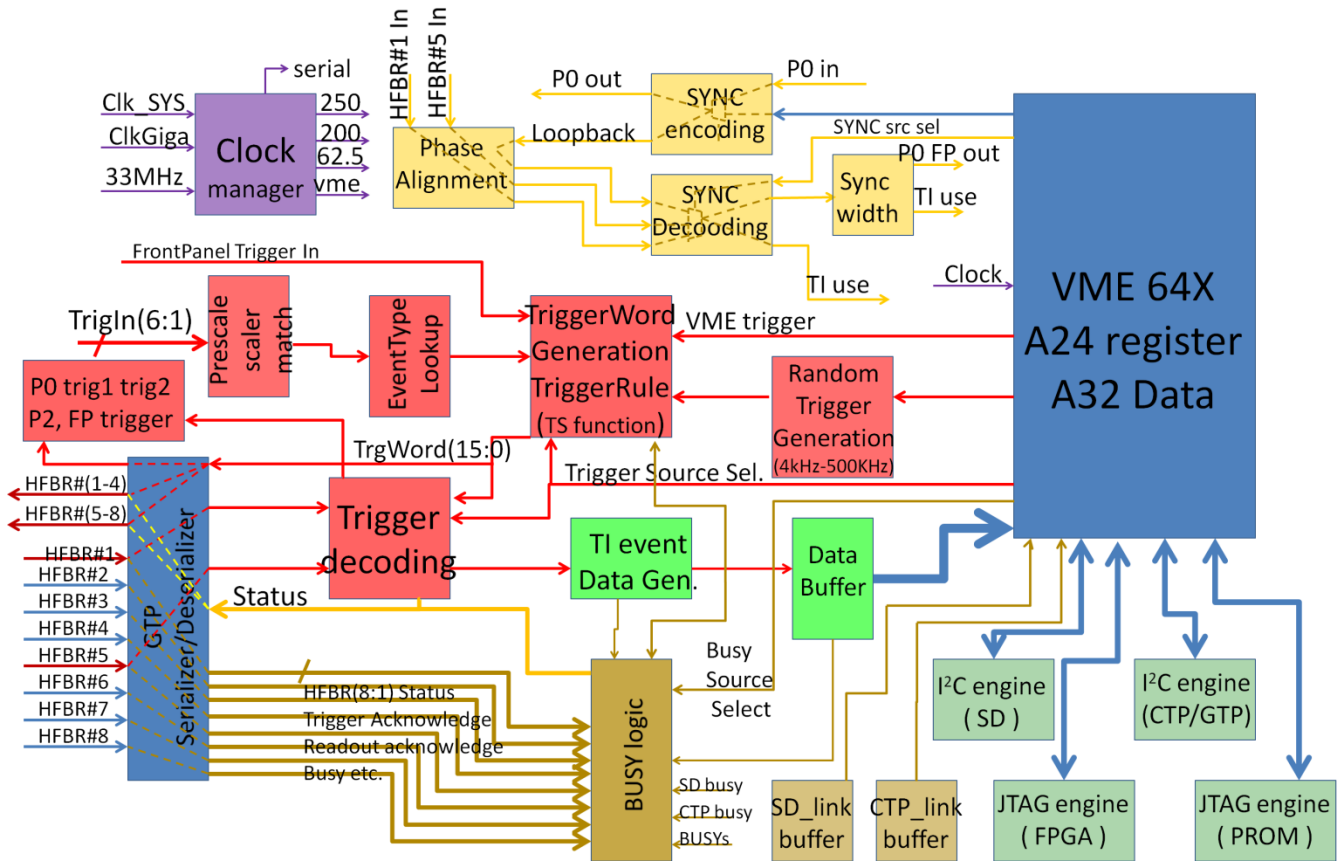
Figure 4: Trigger Interface Block Diagram



TI board block diagram



TI FPGA block diagram



3.2 Fiber links

The HFBR-7924 is the multi-channel (4 Rx, 4 Tx) fiber optic link for the TI. For some TImaster boards, all the eight HFBR-7924 transceivers will be installed, so each can support another eight front end data acquisition crates. For the standard TI in the front end crates and global trigger crate, One (maybe two) HFBR-7924 transceiver will be installed, to receive the trigger/clock/sync from central trigger (and subsystem trigger).

The HFBR_7924 is chosen over the HFBR-7934, because the HFBR-7924 is about \$100 cheaper per piece, and there is no visible performance degradation comparing with HFBR-7934. [GU, 2010]

The first pair (Tx/Rx) is used to transfer trigger words from TD to TI, and status from TI to TD. The second pair is used to transmit the 250MHz clock from TD to TI. The third pair is used to transmit the SYNC from TD to TI. The TI to TD links on second pair and third pair are not used.

The fourth pair (Tx/Rx) is looped back on the TD for fiber length (latency) measurement. The fiber numbers #2, #3, #4, #6, #7 and #8 on TImaster are looped back by a short trace on the PCB between the HFBR Tx and Rx pads. The fiber#1 and #5 are controlled by the FPGA. For TImaster, the FPGA loops back the signals; for standard TI, the FPGA generates a pulse and receives the loopback pulse. By measuring the delay, the fiber delay is measured.

3.3: Clock Distribution

One of the TI's major functions is the pipeline clock distribution. There are four possible sources for the 250MHz clock: 'onboard oscillator', 'external clock input from twisted pair cable', 'optical fiber input from central trigger HFBR#1' and 'optical fiber input from subsystem trigger HFBR#2'. The five clock sources will be multiplexed. There will be only one 250MHz clock running on the TI. The clock is multiplexed by a MC100EP57 chip and go to the AD9510 CLK1 input, and the output of the AD9510 drives all the outputs. In TImaster mode, The clock is fanned out by the MC100LVEP111 2:1:10 clock driver to the optic transceivers.

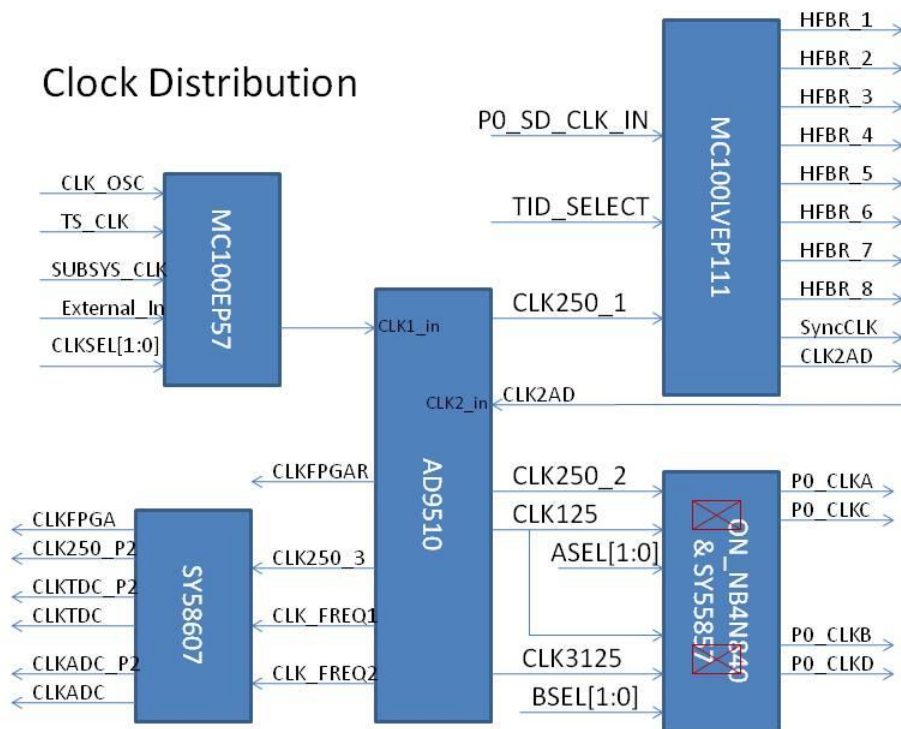
Three clocks (with frequencies of 250 MHz, 125 MHz, and 31.25 MHz) are distributed to the P0 backplane. One On-Semiconductors' NB4N840M and two NB6L72 are used to switch the clocks, so that, the switch slot #A and switch slot #B can get two clocks. Each clock could be any of the three frequencies. The NB6L72 is also used as a level shifter from CML to LVPECL.

The clock is also distributed to the front panel connectors via ECL on twisted pair cables and VME P2 backplane user defined pins for VME ADC or TDC modules, which are NOT VXS compatible.

The CAEN V1290 TDC will get a synchronized 41.67MHz clock through TID P2 connector via a fan out board in the crate, instead of the nominal 40MHz onboard oscillator frequency, because there will be too much effort involved to generate a system wide synchronized 40MHz clock, and the V1290 can be calibrated to accept the 41.67 MHz clock.

An eight-bit on board switch will be used to select the clock sources except for the AD9510, which will be set by the switch, but the setting can be over written by the FPGA firmware (and software).

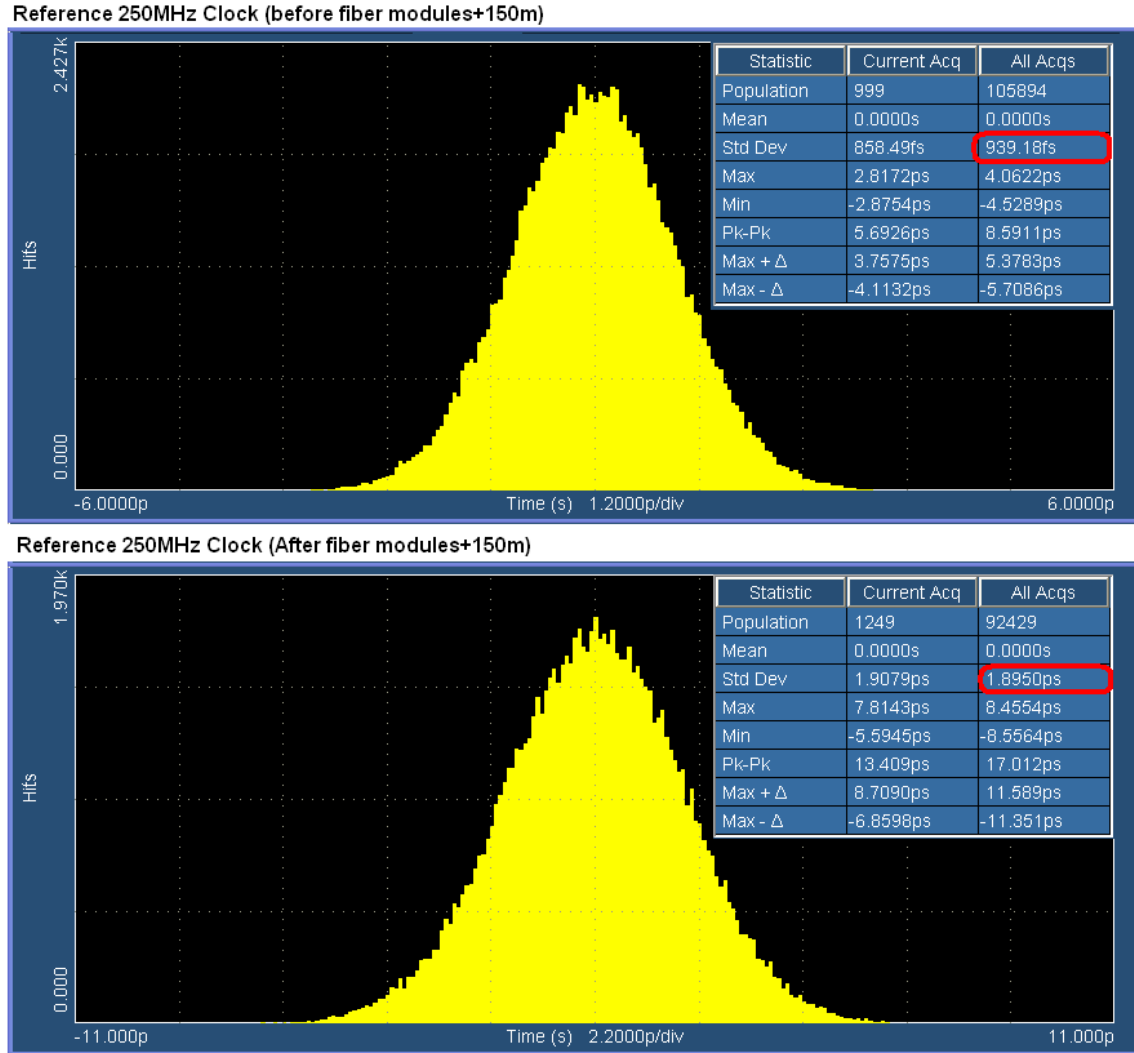
Figure 5 shows the TID clock distribution diagram:



The clock is distributed in (LVP)ECL levels to keep the low jitter and low skew (faster propagation). The LVDS clock is used to drive the FPGA for easy termination.

The global 250MHz clock signal received over the fiber runs the L1 trigger pipeline electronics and nearly all of the front-end modules. Several front-end modules require this clock to have low-jitter and so the clock signal is buffered with components that contribute low-jitter, including the fiber driver and receiver. Figure 3c shows a measurement of the jitter contribution from the fiber driver/receiver pair, including a small increase in clock jitter (roughly 1.6ps additive jitter from the fiber driver/receiver with 150m of fiber between them) by GU et al. [(GU, 2010)]. Careful component selection, signaling, and layout techniques are employed to minimize overall clock jitter. The SD board can further clean up the jitter by its PLL chips.

Figure 6: Global Clock (250 MHz) Jitter



3.4a TImaster trigger generation:

Figure 7a shows the trigger generation as TImaster. This logic is similar to the trigger generation on TS. The loopback trigger is used on TImaster, which is dealt similarly as the fiber inputs on TI boards.

TImaster trigger generation logic

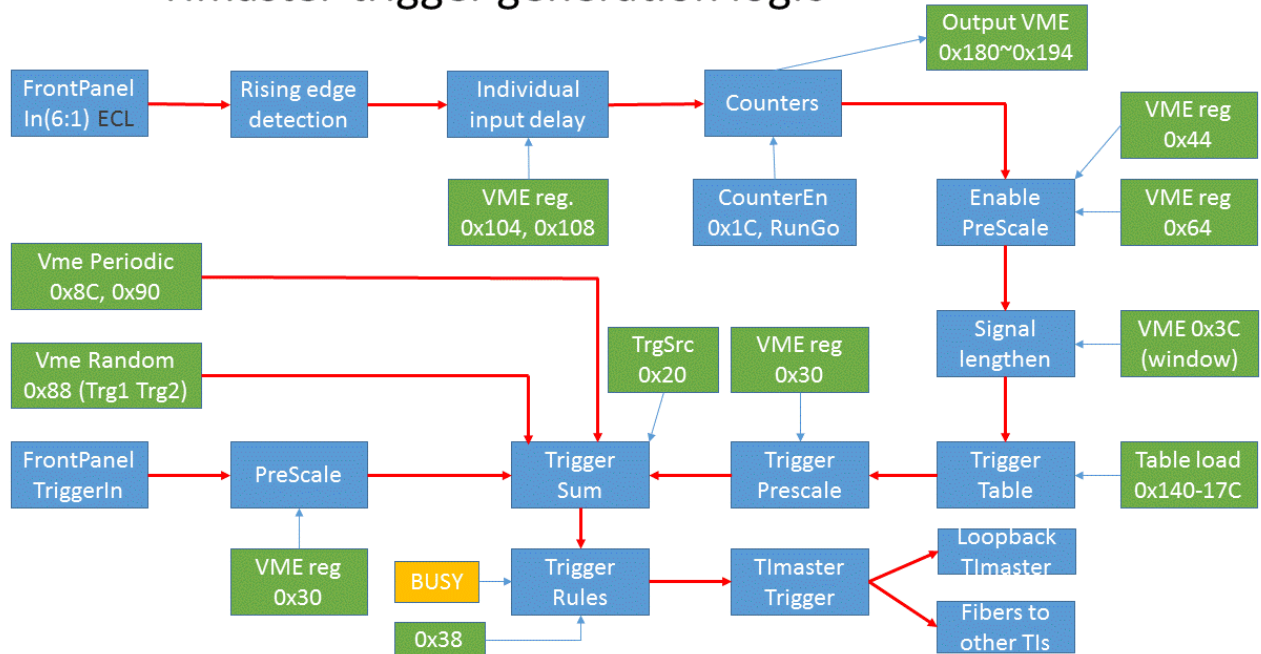
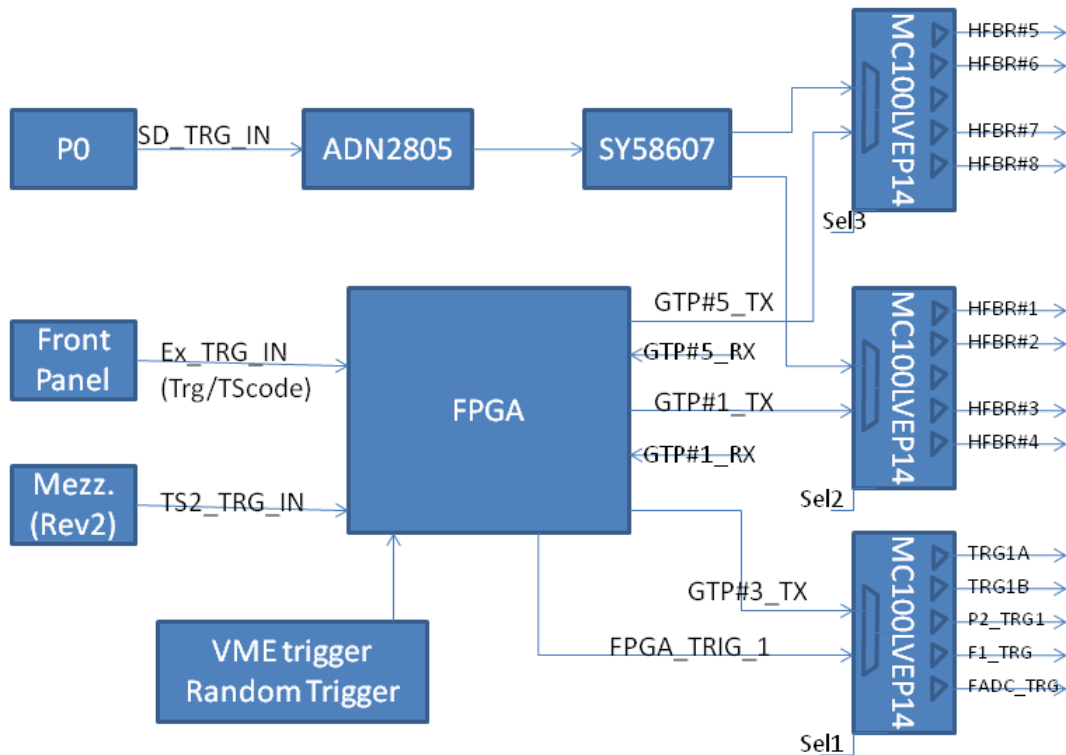


Figure 7a: TImaster trigger generation

3.4b Trigger distribution:

Figure 7 shows the trigger distribution on TI. The TI receives the trigger signal by the optical fiber and decoded by the FPGA. The trigger is sent to the fan out buffer then to the P0 backplane and other connectors. When the TI is in TImaster mode, the TM takes the trigger from the front panel input and send to the SD in gigabit serial mode (encoded) or pulse mode, and it also sent to the fiber in encoded format, though the HFBR#1, #2, #3 and #4 are identical, and HFBR#5, #6, #7 and #8 are identical. (Surely, these two groups can be the same too). The independent choices of the trigger enable the versatile functions of the TI.

Figure 7, Trigger distribution



3.5: Encoded Trigger Word

A 1.25Gbps serial link operating over the fiber is used for distributing a 16bit trigger word every 16ns. There is also a link going in the opposite direction, allowing status words to be sent back to the 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. Upon receipt, the TI drives prompt trigger signals to the front-end modules. The TS transmits these with fixed latency relative to the accepted trigger. This word is distributed every 16ns, so the trigger is distributed every 16ns. The timing information is added in the trigger word to distinguish which quadrant of the 16 ns period the trigger is generated to be fully compatible with the 4ns pipeline design architecture. The TI can distribute trigger signal in 4ns precision. This could potentially reduce the FADC readout window width, and reduce the event size.

Trigger content word – additional information about the trigger 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.

Control Word – request status, or other command (VME trigger for example). They can be queued in a FIFO and sent in any frame not used by a trigger strobe word or trigger content word.

Master Time Word – to fully use the trigger link, bits [13:2] of the TS time is transmitted whenever no other word types are available. By continuously receiving bits [13:2] of the TS time, each TID can promptly detect if its frontend crate 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.

This table shows the format of the trigger word.

Bit13	Bit12	Word Type	Meaning of Bit[11:0]
0	0	Time	Lower 12 bits of TS time

0	1	Control	Control or Command
1	0	Trigger Strobe	Trigger type/Trigger quadrant
1	1	Trigger Content	Additional Trigger data

Bits [15,14] are used on TS to further distinguish the trigger types and TS partitioning.

The following defines the data format for the opposite direction of the link (TI->TD->TS flow):

Bit15:14	Bit 13:12	Bit11:9	Bit8	Bit7:3	Bit2:0
parity	1 0	0 0 1	0	Crate_ID	
		0 1 0	active	Trigger Source(7:0)	
		0 1 1	0	New Block Level (15:8)	
		0 1 1	1	New Buffer Level (31:24)	
	0 1	Busy,AckS,SRreq,SRreq*		Busy, Trg1Ack, Trg2Ack, BlkRcv, ROCack	x x x

The table is valid since the production firmware. The VERY very early version of the firmware used the following encoding.

Bit [15] parity bit;

Bit [14:12] 000 right now;

Bit [11] TI (including the front end crate) BUSY: '1' = busy, '0' = not busy;

Bit [10] Trigger_1 acknowledge;

Bit [9] Trigger_2 acknowledge;

Bit [8] one block of triggers received;

Bit [7] ROC readout acknowledge (one block of trigger is readout and acknowledge by ROC);

Bit [6:0] status.

3.6: Fixed Latency SYNC

The SYNC signal is a 250Mbps serial line operating in synchronous mode. This serial link allows a 4bit command to be sent at chosen 4ns points in time. SYNC is synchronized to a slower clock (62.5 MHz) derived from the 250 MHz master clock and is sampled every 4 ns. 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 4bits that comprise the 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 (or the TS function of the TImaster). Since the TD distributes this serial line over the fiber module, additional encoding (Manchester) is performed to balance the 1's and 0's of the line and to keep the maximum run length of the signal below the requirements of the fiber module. The TID decodes the SYNC signal (Manchester and command). Careful design in minimizing SYNC to the distributed master CLK250 skew guarantees a fixed latency link.

The SYNC is fiber delay adjusted on each TI. The fiber latency is measured using the fourth pair of the fiber. The SYNC is delayed so that all the TI modules will execute the SYNC command at exactly the same time (within the skew of the global 250MHz clock distribution).

The SYNC link is used in conjunction with a synchronous FIFO to enforce a fixed latency on the serial trigger link. The TD uses LVPECL buffers to fan out the trigger signal and the Sync signal, which is encoded (Manchester encoding) in the FPGA, to the HFBR_7924 transceivers. In the TI, 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 TS starts transmission (time words) on the trigger link, and after a fixed number of 62.5MHz clock cycles issues a trigger start command on the SYNC line. In response to the trigger start command, the TI enables continuous readout of the FIFO. There must always be a non-zero number of words in the FIFO to maintain a fixed latency link. This will be true if the number of words pre-filled into the FIFO x 16 ns is greater than the latency uncertainty of the link. In the case of the MGT, several words (e.g. 3 or 4) are enough when the latency is set to minimum, as the elastic buffer is not

necessary as all the clocks are the same or derived from the same 250MHz clock (no clock frequency difference). The TS must always be transmitting valid data to maintain the fixed latency of the link. This is illustrated in figure 8:

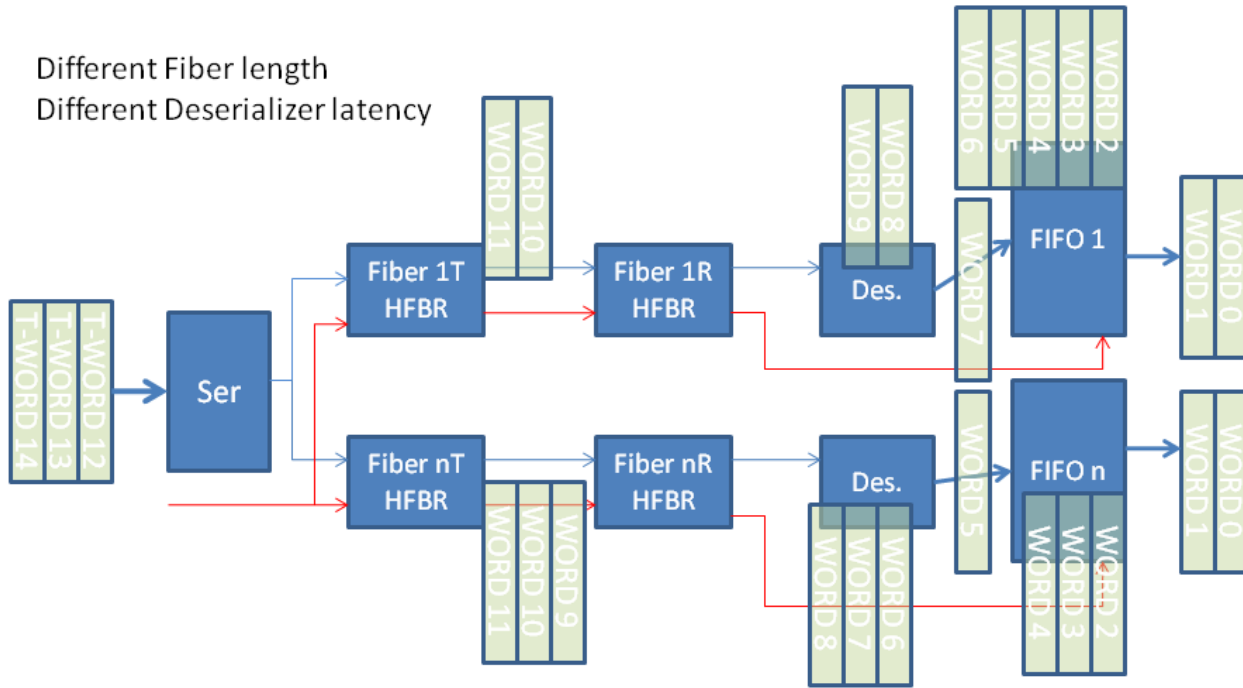
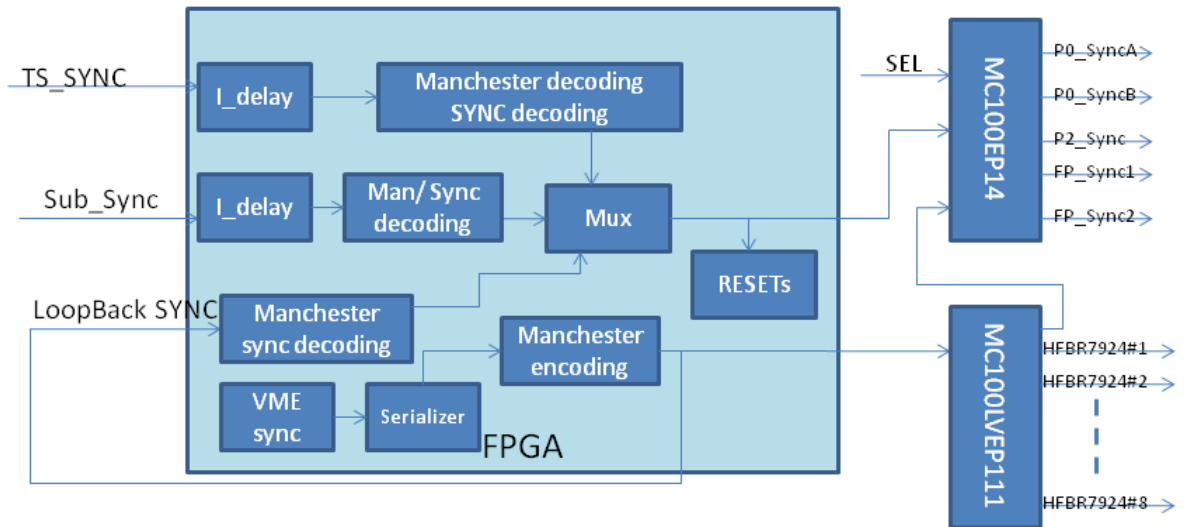


Figure 8: Illustration of the Trigger synchronization process

The SYNC link is also useful in ensuring that the lower frequency clocks derived in the TI from the distributed system clock (CLK250) have the same phase across all TI modules in the front-end data acquisition system. At startup, the TS issues a SYNC command CLKRESET. This resets the clock distribution chip (AD9510) in each TI on the same CLK250 edge, assuring that the lower frequency clocks (125MHz, 62.5MHz, 41.67MHz and 31.25MHz) are in phase across the entire system. This command is sent before the TI sync command. The delay between them is determined by the maximum MGT reset recovery time, as the MGT clock is changed (AD9510 clock reset) during clock reset. The time is several milliseconds.

Figure 9 shows the block diagram of the SYNC distribution on the TI board:



3.7 VME interface

The TI board is a VXS payload slot board. It is compatible with VME64x backplane. Normally, it is a VME slave board, with interrupt capability.

The TI can also be a master VME board. It supports single level bus request (BR3, level 3) only, as we do not expect many boards to be a VME master in the crate.

For simplicity, three kinds of VME address modifier codes are implemented. (1), User defined address modifier. (0x19, 0x1A, 0x1C and 0x1D) This is similar to the A24 address modifier. It is used to load the PROM by the emergency logic. (2), Standard A24 address modifier. This is used to readout the registers on the FPGA, slow controls of the TI peripherals, and to request data from other boards in the crate when in master mode. (3), A32 BLT data transfer. This is used to transfer data to the ROC (Read Out Controller). This is implemented the same way as other ADC/TDC board, so the ROC needs only one read to get all the front end boards' data out for higher efficiency.

3.8: The Xilinx PROM programming.

The Xilinx XCF32P PROM is used to program the FPGA. It can save two different versions of the FPGA (XC5VLX30T) firmware when used in non-compression mode. It can save four different versions of the firmware when bit stream compression is used, in which case, only the slave serial and slave SelectMAP modes are supported. The PROM is programmed using VME with emergency logic decoding. It can be addressed in the VME64x crates by its geographical address. If it is in the crate without geographical address, only one TI 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 A24 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 bits are unused. The emergency loading is tested. With M6100 controller, the 32 Mbit PROM (XCF32P) can be loaded in less than five minutes. The emergency loading provides VME remote firmware re-loading and broadcast firmware loading even if the FPGA is not working. One 33 MHz on-board oscillator is used to program the FPGA in slave mode, and used by the FPGA for slow control, for example, the VME to AD9510 serial control engine. The PROM is in master mode with external clock. This is set when producing the SVF download file.

The PROM can also be programmed by the on-board JTAG connector and VME-JTAG engine (after the FPGA is programmed and working) in the FPGA. The JTAG engine in the FPGA provides VME remote firmware loading when the FPGA is working with more efficient VME data transfer (32 bits versus 1 bit).

3.9: Status passing

The TI can merge the status together and pass on to upper level. Specifically, the TI can merge the status from the CTP and SD via P0 backplane and pass on to TD in the global trigger/clock distribution crate via optical links. Right now, only the BUSY is defined as status. We have the capability to add a less severe state, WARNING, to the status.

The TI in the front end crate can send its timing information back to the TD in the trigger distribution crate. This information can be used to check the system synchronization status. Here is how this will work.

On the start of the run, a global sync command will reset all the counters synchronously. Each TI will have its own clock counter, that count how many clock cycles since the sync reset. As the trigger is synchronized on all the TIs, all the TI should receive the same trigger at the same clock cycle. The TI in the front end crate will send the clock counter, recorded when it receives the trigger, to the TD in the distribution crate along with the trigger acknowledge. This signal may be received asynchronously as a status word (ref to section 3.5), but all the eight words should match on the TD. The TD check the difference between the trigger sent and the trigger acknowledge received. The TD also checks the trigger acknowledge timing. If the TD received un-matching numbers from front-crate, it is most likely a trigger loss if the timing difference is large; it is most likely a clock loss if the difference is small. In this way, the trigger synchronization can be checked at the hardware level.

3.10: Serial data communication with SD

The TI can send data to the SD via a 250Mbps link, (it is possible to increase it to 500 Mbps using DDR techniques). This can be implemented using the Xilinx SelectIO standard differential IO pin pair. In this case, the SD can be implemented as a data concentrator card to collect data from the VXS crate (the payload modules include FADC, FTDC, TI, etc.). The TI will send data to SD on every trigger (event). This is another data readout path in addition to the standard VME readout. The maximum data rate is 50 MB/sec per slot. The full crate can reach up to 900 MB/sec assuming all 18 payload slots are used. The data from SD is event based, and there is no need to re-format the data as needed by the VME readout (block readout is used for high efficient usage of the VME bus data transfer).

The SD can also send data to TI using this link, though there are not much useful per event data from SD anyway. The direction of the data link depends on the firmware (TI and SD) implementation.

3.11: legacy compatibility

A mezzanine board is used to interface with the TS_rev2 module directly for backward compatibility, which may still be used in the experiment halls. The Mezzanine board has the same connector as the TI_rev2 board (matching with Trigger Supervisor rev2). With the mezzanine board plugged in, the TI will behave like a TI_rev2 board (produced in 2001) to interface with TS_rev2 board.

3.12 Other functions

The TI has a generic IO 34 pins connector, which can accept one external trigger input, and six trigger input codes. It can also accept external clocks and trigger inhibit. Using these, some of the Trigger Supervisor functions can be implemented in the TI. The TI can serve as a subsystem trigger supervisor board. A nine crates setup can be implemented by TIs for (sub)system commissioning and subsystem test setups.

3.13: 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 two 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. It is also possible that the SyncEvent is 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 back 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 be continued.

3.14 Clock phase alignment inside of the FPGA

The TI FPGA gets 250 MHz clock and 31.25 MHz clock from AD9510. DCMs are used to generate the 62.5 MHz clock from 31.25 MHz clock, 250 MHz, 500 MHz and 200 MHz clocks from the 250 MHz inputs. There is no delay from the DCM input to output. This creates a problem for the clock phases, where the Clk_62.5MHz is about 1 (one) ns later than the Clk_250MHz. This 1 ns delay caused timing problems for signals transitioning between the two clock domains. This delay is caused by the AD9510 output. The 250 MHz is LVPECL output from AD9510, and the 31.25 MHz is LVDS output from AD9510. The LVDS is about 1ns slower than the LVPECL. (min 740ps, typical 920ps, max 1140ps). To compensate for this skew, the Clk1x270 is used for Clock_feedback on the 250MHz clock DCM, so all the clock outputs are delayed by 90° (or 1 ns) relative to the clock input. This way, all the clocks (250 MHz, 200 MHz, 500 MHz, 62.5 MHz) are phase aligned. The phase alignment is put in the firmware on Nov. 2, 2017 (tip79 and later).

For the TIpicie board, both the 250 MHz clock and the 31.25 MHz clock are from LVDS ports of the AD9510. The Clk1x0 should be used for the 250 MHz DCM. On TS board, the 250 MHz clock and the 62.5 MHz clock are from the same DCM, so there is no phase alignment issue on TS.

4. Specification Sheet

4.1 Mechanical

- Single width VITA 41 Payload Module. It will be positioned in PP18 in front-end data acquisition crate and global trigger crate; it can also be plugged into any slots in standard VME crates without VXS. In system commissioning, it can be in PP18 and support up to 9 crates with some TS functions. In luxury setup (optional), it can be in any payload slots (except PP18) of the subsystem crates, and serve as a subsystem Trigger Supervisor, which can group up to 8 crates together as a subsystem.

4.2 High speed serial P0 and P2 inputs:

- BUSY LVDS signals on P0;
- BUSY ECL signals on P2;
- I²C to VXS Switch A & B;

4.3 High speed serial P0 and P2 outputs:

- Any two of these clocks (250MHz, 125MHz, and 31.25MHz) to Switch slot #A and Switch slot #B.
- Trig 1 LVPECL Trigger Signal
- Trig 2 LVPECL Trigger Signal

- Sync LVPECL Trigger Signal
- 250/500 Mbps data (LVDS) to/from SD at per event basis

4.4 Front panel inputs and outputs:

- 250MHz ECL Clock Input & Output
- Trig 1 ECL Trigger Input & Output
- Trig 2 ECL Trigger Input & Output
- Sync ECL Trigger Input & Output
- Busy ECL Input & Output
- Up to eight HFBR-7924 transceivers for TImaster; up to two HFBR-7924 transceivers for standard TI in front-end data acquisition crate.

4.5 Fiber channel signals:

- SYNC Fixed Latency Link
 - 250Mbps Serial Communication
 - Manchester Encoded
 - SYNC to CLK skew variation adjusted at FPGA receiver.
- TRIGLINKTX/TRIGLINKRX
 - 1.25Gbps Trigger Word Line
 - Provides 16bit parallel data every 16ns
 - A BUSY status word in the opposite direction.
- CLK
 - 250MHz Clock <3ps RMS Jitter

4.6 LED Indicators: Front Panel:

- Bit 1 (close to the PCB): FPGA programmed and the clock (DCM locked) is ready;
- Bit 2: VME DTACK, VME activity;
- Bit 3: Trigger_1 is sent out;
- Bit 4: HFBR MGT Rx error;

On board:

- Power OK near each regulator or DC-DC converter (LED is OFF when the power is OK);
- FPGA program DONE (LED is OFF when programmed);
- Fiber optical transmitter FAULT and receiver SIGNAL_DETECTED near each HFBR-7924 module.

4.7 Programming:

- VME to JTAG A24D32 with user defined AM (Address Modifier) for remote loading with redundant On board JTAG connector;
- Custom VME to JTAG engine implemented in the FPGA using A24D32 for firmware loading;
- Up to four revisions of the firmware can be stored in the PROM simultaneously.

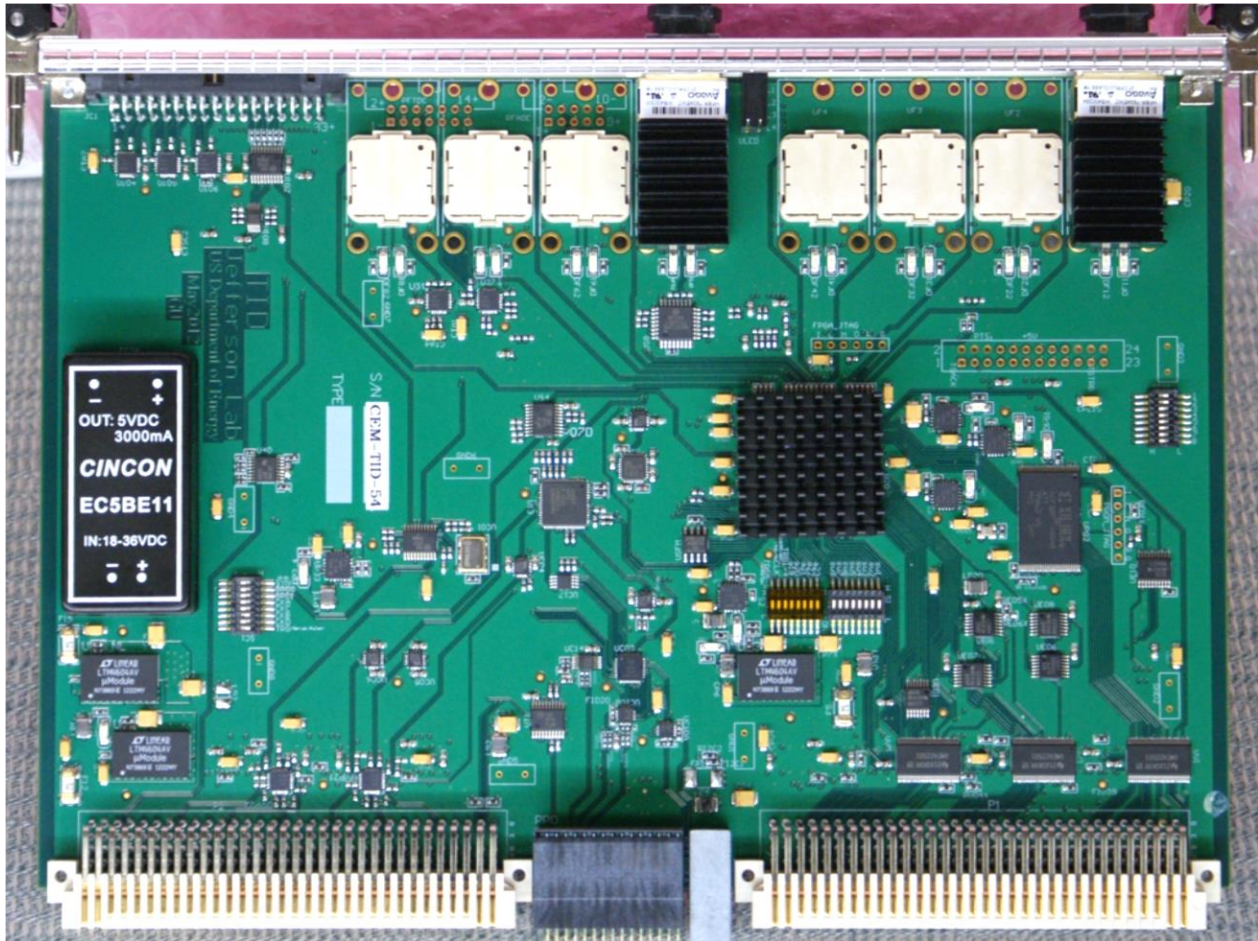
4.8 Power requirements:

- +5v @ 4 Amps; -12V @ 0.25 Amp; +12V @0.25 Amp (From Backplane)
- Optional DC-DC converters for +3.3V, and Local regulators for other required voltages: +1.0V, +1.2V, +1.8V, +2.5V, +3.3V, and -5V.

4.9 Environment:

- Forced air cooling: Weiner standard VME64x/VXS
- Commercial grade components (0-75 Celsius)

Figure 10 shows a typical TI board :



5 TI operation procedures:

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

5.1 TI Power supply:

The TI can use +3.3V directly from VME64x crate. It can also generate its own +3.3V supply by a DC-DC converter. Another dedicated +3.3V DC-DC converter can be used to power the optical transceivers (HFBR-7924). The HFBR_7924 can also be powered by the main +3.3V supply for the board. Proper settings are needed to avoid damage to the board or backplane.

If the dedicated +3.3V DC-DC converter is used to power the HFBR-7924:

- (1). Make sure that the DC-DC converter LTM4604, UP01, is stuffed.
- (2). Inductor, LP3 is removed.

If the TI main +3.3V power is used to power the HFBR-7924:

- (1). Make sure that UP1_NL is NOT stuffed,
- (2). Inductor LP3 is stuffed.

If the VME64x crate +3.3V power is used for the TI:

- (1). The fuse, FP1 is stuffed;
- (2). The DC-DC converter UP1 is removed.

If the VME64x crate +3.3V power is not used for the TI:

- (1). The fuse, FP1 is removed;
- (2). The UP1 is stuffed.

5.2 Hardware setting (Switch etc.):

The TI drives signals to the P0 connector. This is set by three pairs of resistors or capacitors. If the 0 Ohm resistors are used, the signals are DC coupled, if 0.047uF capacitors are used, the signals are AC coupled. When in Subsystem TS mode, stuff C0001/C0002, C0003/C0004, and C0007/C0008; when in TI mode (including TImaster), stuff C0011/C0012, C0013/C0014, and C0017/C0018. This will avoid driving conflict with SD.

There are four 8-bit switches on the TI, and marked as: S1, S2 (VME address space setting switches), SC01 (TI mode setting) and SC1 (TI clock setting). Some switches are LVTTL, some are LVPECL. When the switch is ON, the switch is LOW, (0V for LVTTL, and 1.4V for LVPECL); when the switch is OFF, the switch is HIGH, (3.3V for LVTTL, and 2.5V for LVPECL). Here are the details of the switch setting:

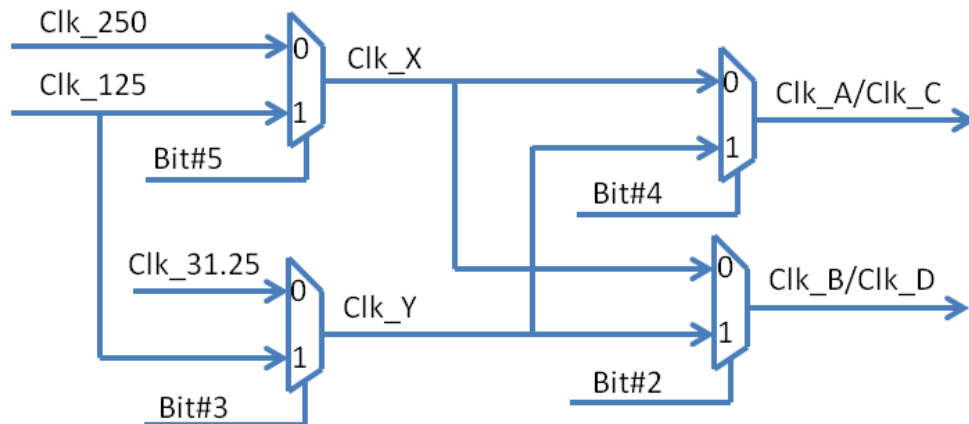
5.2.1. SC1 setting:

Bit1: LVPECL, open=high, trigger_1 source selection. When low, the trigger_1 source is FPGA MGT_112 output (serialized trigger data); when high, the trigger_1 source is standard FPGA differential output (pulse);

Bit2-5: LVTTL, open=high, P0 CLK_A, CLK_B, CLK_C, CLK_D selection. Clk_A is the same as CLK_C and CLK_B is the same as CLK_D. The clocks are selected in two stages:

Switch \ clock out	Clock_X	Clock_Y
Bit3=0; bit5=0	250	31.25
Bit3=1; bit5=0	250	125
Bit3=0; bit5=1	125	31.25
Bit3=1; bit5=1	125	125

Switch \ clock out	Clock_A, Clock_C	Clock_B, Clock_D
Bit2=0; bit4=0	Clock_X	Clock_X
Bit2=1; bit4=0	Clock_X	Clock_Y
Bit2=0; bit4=1	Clock_Y	Clock_X
Bit2=1; bit4=1	Clock_Y	Clock_Y



Bit[7:6]: LVPECL, open=HIGH, closed = LOW: TID 250MHz clock source selection. When bit7=0&bit6=0, on-board oscillator is selected; when bit7=1&bit6=0, TD input from optical transceiver

#1 is selected; when bit7=0&bit6=1: subsystem clock input from optical transceiver#5 is selected; when bit7=1&bit6=1: front panel generic connector inputs is selected; For FPGA control, refer to A24 register offset 0x2C bit(1:0).

Bit8: LVPECL, open=high, TI connectors' sync signal source selection. When low, Sync_reset is selected; when high, Manchester encoded SYNC signal is selected.

5.2.2 SC01 setting:

Bit[2:1]: LVTTL, open=high. Firmware revision selection. When Bit2=0&Bit1=0, select firmware Rev0; when Bit2=0&Bit1=1, select firmware Rev1; when Bit2=1&bit1=0, select firmware Rev2; when Bit2=1&Bit1=1, select firmware Rev3. If the program bits are not compressed, the PROM XCF32P can only fit two revisions of the firmware.

For TImaster5 firmware (not the latest TIFPGA firmware), the Bit(4:3) are decoupled from Bit(2:1). The Bit(2:1) controls the firmware revision, while the Bit(4:3) is a firmware variable. If Bit(4) = 1, the CLK1 input of AD9510 is selected, which is the mux of four independent 250MHz clocks. If Bit(4)=0, the CLK2 input of AD9510 is selected, which is the clock from P0 backplane. If Bit(4)=1 and Bit(3)=0, the FiberMode=1, which will power down the HFBR#2, #3, #4, #6, #7 and #8. When the FiberMode=1, the TID is in pure TI mode.

Bit[6:5]: LVTTL, open = high. Connector output clock frequency selection. When "11", the clocks are 41.667 MHz, used for CAEN TDC; when "00", the clocks are 250 MHz, used for FADC250 fan-out; when "01": the clocks are 125 MHz; when "10", the clocks are 31.25MHz (for flash TDC).

Bit[7,8]: LVTTL, open = high. This is to control the SD_link and CTP_link loopback test. When high, the driver is disabled, and the TI is the receiver for SD_link and CTP_link. When low, the driver is enabled, and the TI is the receiver and the driver. (loopback if no SD/CTP in the crate).

5.2.3 S1 setting:

Bit[1:3]: These three bits are used to control the OUTPUT(5:2) of the front panel 34-pin connector. The switches are used to multiplex the eight FPGA internal monitoring sources (32 signals) on the four output pins.

Bit[4]: Add an extra word if the total number of words in a block is odd. '1' to enable, '0' to disable;

Bit[8]: keep high, for TID_rev1, this pin is used as IACK input to the FPGA;

Bit[8:7]: keep high, for TID_rev2, these two pins are used as clock source selection output from the FPGA;

Bit[5:6]: not used.

5.2.4 S2 setting:

Bit[1:5]=A[23:19], VME address space in A32 or A24 mode. Be careful about the bit order. When in VME64x crate, these addresses should be set the same as its geographical address.

Bit6: LVPECL, open=high. The HFBR#1, #2, #3 and #4 trigger source selection. When low, the source is P0 (from SD), when high, the source is FPGA GTP_116.

Bit7: LVPECL, open=high. The HFBR#1_8 fan out clock source selection. When low, the fan out source is P0 (from SD), when high, the source is on-board clock manager AD9510. This should be consistent with the bit(4:3) of SC01 switch setting.

Bit8: LVPECL, open=high. The HFBR#5, #6, #7 and #8 trigger source selection. When low, the source is P0 (from SD), when high, the source is FPGA GTP_114.

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. Some parts of the board needs be powered down to reduce the power consumption. The FPGA GTP transceivers will be automatically powered depending on the mode setting (TI or TImaster).

The delay settings will be automatically loaded by fiber measurement. This information could be saved in the Serial Flash Memory (SFM). The parameters (delay etc.) can be set by the VME, and stored in the SFM. We expect the parameter to be stable, but adjustment may be necessary when moving the board (different fiber connection etc.)

In the test setup and commissioning test, some delay parameters (especially the TImaster) needs be manually set.

5.3.1 Trigger table loading:

There is a 6-bit inputs, 8-bits output look up table built in the TI(master). The six-bit inputs corresponds to the six front panel inputs (marked as TS#1, TS#2, ... TS#6). The eight-bit outputs defines the trigger/event type. Here is the eight output bits definition:

Output(7:6): 00: no trigger, Neither trigger 1 nor trigger 2;
01: trigger 1, no trigger2 (Event readout trigger);
10: Trigger 2, no trigger 1 (FADC250 playback);
11: Sync event.

Output(5:0): Event (trigger) type.

The table (64 bytes) is loaded by sixteen 32-bit VME words (A24 offset 0x140 – 0x17F)

5.3.1.1 More details about the trigger table:

The trigger table is a simple reloadable lookup table. The loading is Address(0x140-0x17C) sixteen 32-bit words. The look up is FPinut(6:1) (or TS#(6:1)) sixty-four bytes (8-bit).

FPinput(6:3) == VmeAdd(5:2);

FPinput(2:1) == VmeAdd(1:0); the byte position in the 32-bit word.

The 8-bit of the lookup table is defined as shown above (Output(7:0)), where the (5:0) is the event type.

Here are some example of the trigger tables:

- Example#1: Any of the six inputs (TS#6:1) generate TRIGGER, and the event type is the Input pattern. The sixteen words will be:

Word#15: 0x7f7e7d7c	Word#14: 0x7b7a7978	Word#13: 0x77767574	Word#12: 0x73727170
Word#11: 0x6f6e6d6c	Word#10: 0x6b6a6968	Word#9: 0x67666564	Word#8: 0x63626160
Word#7: 0x5f5e5d5c	Word#6: 0x5b5a5958	Word#5: 0x57565554	Word#4: 0x53525150
Word#3: 0x4f4e4d4c	Word#2: 0x4b4a4948	Word#1: 0x47464544	Word#0: 0x43424100

- Example#2: Any of the (lower) five inputs (TS#5:1) generate TRIGGER, the higher input (TS#6) generates TRIGGER2 (FADC playback). The SYNC event will be generated if TS#6 is true and any of the TS#5:1 is true. The event type is the same as the input pattern. The sixteen words will be:

Word#15: 0xfffffd7c	Word#14: 0xfbfaf9f8	Word#13: 0xf7f6f5f4	Word#12: 0xf3f2f1f0
Word#11: 0xdfdedddc	Word#10: 0xdbdad9d8	Word#9: 0xd7d6d5d4	Word#8: 0xd3d2d1a0
Word#7: 0x5f5e5d5c	Word#6: 0x5b5a5958	Word#5: 0x57565554	Word#4: 0x53525150
Word#3: 0x4f4e4d4c	Word#2: 0x4b4a4948	Word#1: 0x47464544	Word#0: 0x43424100

- Example#3: Any of the six inputs (TS#6:1) generate TRIGGER with event type defined as:
Any of TS#(2:1), event type '1';
Any of TS#(4:3), event type '2';
Any of TS#(6:5), event type '3'.

The trigger table is not fully defined. The trigger table will be fully defined if TS#(2:1) has the highest priority and the TS#(4:3) has the lowest priority. The sixteen words will be:

Word#15: 0x41414143	Word#14: 0x41414143	Word#13: 0x41414143	Word#12: 0x41414143
Word#11: 0x41414143	Word#10: 0x41414143	Word#9: 0x41414143	Word#8: 0x41414143
Word#7: 0x41414143	Word#6: 0x41414143	Word#5: 0x41414143	Word#4: 0x41414143
Word#3: 0x41414142	Word#2: 0x41414142	Word#1: 0x41414142	Word#0: 0x41414100

- Example#4: Any of the six inputs (TS#6:1) generate TRIGGER with event type defined as:

Any of TS#(2:1), event type '1';
Any of TS#(4:3), event type '2';
Any of TS#(6:5), event type '4'.

The trigger table is not fully defined. The trigger table will be fully defined if the event type is a 3-bit word (that is event type 3, 5, 6 and 7 are also valid) as defined below:

Any of TS#(2:1), enable event type bit#0;
Any of TS#(4:3), enable event type bit#1;
Any of TS#(6:5), enable event type bit#2.

The sixteen words will be:

Word#15: 0x47474746	Word#14: 0x47474746	Word#13: 0x47474746	Word#12: 0x454545 44
Word#11: 0x47474746	Word#10: 0x47474746	Word#9: 0x47474746	Word#8: 0x454545 44
Word#7: 0x47474746	Word#6: 0x47474746	Word#5: 0x47474746	Word#4: 0x454545 44
Word#3: 0x434343 42	Word#2: 0x434343 42	Word#1: 0x434343 42	Word#0: 0x 41414100

6. VME Programming Requirements (This part will be updated as the firmware develops)

There are three categories of Address Modifier codes are supported on the TI: the user-defined codes (A24) for emergency firmware loading; Standard A24 for FPGA register read/write and slow control; A32 block transfer for VME data readout.

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 TDI; data bit[0] is TMS.

For example, if the board is in slot#5 (that is ~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].

➤ Address offset: 0x00000: Board ID:

Bit 7-0 (R/W): Crate ID; Reset default 0x00;
 Bit 12-8 (R): A24 address, higher 5 bits; Reset default 000
 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 23-16 (R): PCB related setting, 0x01: production board, 0x00: prototype board;
 Bit 31-24 (R): Board type: 0x71: TI, 0x75: TS, 0x7D: TD.

➤ Address offset: 0x00004: Optic transceiver enable:

Bit 7-0 (R/W): on TI: HFBR#8, #7, ... #1 Enable, Reset default 0xFF all enabled.
 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;
Bit 7-0 (R/W): I2C selection for optic transceivers (TIDqsf). '0' to enable (select), '1' to disable (bypass);
 Bit8: Enable the P0 (SW#A, SW#B), Front panel connectors, and VmeP2 trigger and SyncReset signals, default is 1 to enable; When disabled, the trigger/SyncReset stay LOW.
 Bit 23-16 (R): if '1', the HFBR#8, #7, ... #1 is connected to a TI (serial Link synchronized);
 Bit 31-24 (R): if '1', the HFBR#8, #7,...#1 connected TI has its trigger source enabled.

➤ Address offset: 0x00008: Interrupt setting:

Bit 7-0 (R/W): Interrupt ID; Reset default 0xC8;
 Bit 10-8 (R/W): Interrupt level; Reset default 5;
 Bit 16 (R/W): IRQ enable. Reset default: 0;

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

Bit 7-0 (R/W): Trigger_1 delay, $4096 * \text{Bit7} + (\text{Bit}(6:0)+1) * 16 * (4^{**}\text{Bit7})$ ns; Reset default 0x00; That is when bit#7 = 0, the delay is "bit(6:0)+1" in steps of 16ns; when bit#7 =1, the delay is "bit(6:0) + 64" in steps of 64ns. The minimum delay for bit7=1 is about 4.1us.
 Bit 12-8 (R/W): Trigger_1 Pulse width (n+1)*4 ns; Reset default 0x07;
 Bit 23-16 (R/W): Trigger_2 delay, $4096 * \text{Bit23} + (\text{Bit}(22:16)+1) * 16 * (4^{**}\text{Bit23})$ ns; Reset default 0x00;
 Bit 28-24 (R/W): Trigger_2 Pulse width (n+1)*4 ns. Reset default 0x07;

➤ Address offset: 0x00010: A32 address space:

Bit 13-5 (R/W): Address Max; Reset default 0x1FF;
 Bit 22:14 (R/W): Address Min; Reset default 0x000;
 Bit 31-23 (R/W): Base Address. Reset default 0x100;

➤ 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. The ROC should readout these eight bits, and set them to the FADC250, FADC125, etc.

➤ Address offset: 0x00018: TI data format control: Reset default 011;

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

Bit 2-1: Event format control:

bit 1: '1' to enable the lower 32-bit trigger timing word;

bit 2: '1' to enable the higher 16-bit of event number and higher 16-bit of trigger timing;

Bit 3: if '1', the front panel trigger input pattern will be readout (before prescale).

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

Bit 31-24 (R): Buffer size set by TS (or TImaster). This is an intermediate value. The ROC should readout these eight bits, and set them to the FADC250, FADC125, etc.

➤ 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;

Bit 1 (R/W): '1' en_token_in is true, '0' en_token_in is false;

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

Bit 3 (R/W): '1' enable en_A32m, '0' disable en_A32m; assert to enable common A32 multi-board addressing of module;

Bit 4 (R/W): '1' enable en_A32, '0' disable en_A32;

Bit 7 (R/W): '1' enable VME bus interrupt for module error?

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

Bit 9 (R/W): '1' token_in high, '0' token_in low; If both bit 9 and bit 1 are set high, the Token_Out will be high (this is for SD test).

Bit 10 (R/W): '1' first_board true, '0' first_board false;

Bit 11 (R/W): '1' last_board true, '0' last board false;

Bit 15 (R/W): '1' disable data readout buffer full;

Bit 16/17 (R/W): assigned for TS to enable the front panel for scalar control;

Bit 20 (R/W): assigned for TS to enable VMEp2 output to TDIO;

Bit 21 (R/W): '1' to enable the instant block level (and buffer level) update;

Bit 22 (R/W): Buffer level select. '1' to select the 0x34 setting, '0' to select the TS broadcast setting.

Bit 23 (R/W): '1' to enable the buffer level threshold on TI.

Bit 24 (W): it is valid for one clock cycle only (to decrement the number of data blocks in the fifo?).

On Tlpcie: Bit 24 (W): '1' to set enable one data block to DMA. This is 'OR'ed with Bit 22.

Bit 25 (R/W): '1' to enable auto PCIexpress Tx/Rx reset when the EOF is longer than 256 clock cycles (stuck); '0' to disable the auto reset.

Bit (30:28) (R/W): when set to "101", the I2C is enabled for BAR1; otherwise, the JTAG is enabled for BAR1. The default is JTAG.

Bit 27: if set to '1', use the Trigger source enable (GO) to enable the trigger input counter (0x180...)

Bit 28: if set to '1', use the individual trigger input enable to enable the trigger input counter (0x180...)

Bit 27 and bit 28 to work together. When set to '0', the control is disabled.

Bit 31 (R/W): '1' to slow down the slow_clock for a factor of 32 for trigger rules.

➤ Address offset: 0x00020: Trigger source register:

Bit 15-0 (R/W): Trigger source enables: Reset default 0x0000;

Bit 0: P0 trigger input;

Bit 1: HFBR#1 trigger input;

Bit 2: TImaster loopback trigger input;

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 10: HFBR#5 trigger input

Bit 11: Enable trigger2 to generate trigger1 (valid for TS and TImaster)

Bit 12: SubTS#1 trigger enable, this is valid on HFBR#1 input only;

Bit 13: SubTS#2 trigger enable, this is valid on HFBR#1 input only;

Bit 14: SubTS#3 trigger enable, this is valid on HFBR#1 input only;

Bit 15: SubTS#4 trigger enable, this is valid on HFBR#1 input only;

When the TI is used for SBS, the Bit(15:12) indicates which duplication branch the crate is in.

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

Bit 23-16 (W): Force TrigSrc sending to TD if bit(23:16)=0xFC;

Bit 31-24 (W): RunGO. If bit(31:24) = 0x90, the TI front panel O#4 = '1', O#3, 2, 1 are still user controllable;
if Bit(31:24) is not 0x90, O#4, 3, 2, 1 are user controllable (assuming that the SW is "000")

➤ Address offset: 0x00024: Sync Source register:

Bit 15-0 (R/W): Sync Source enables: Reset default 0x02;

Bit 0: P0 sync input (in Subsystem TS mode);

Bit 1: HFBR#1 sync input;

Bit 2: HFBR#5 sync input;

Bit 3: Front panel trigger inhibit enable;

Bit 4: TImaster loopback SYNC enable

Bit 6: automatic SyncReset enable

Bit 7: Enable the option for RESET to be set high (sync code 0x99), and low (sync code 0xcc).

Bit(11-8) (R): last SYNC code from HFBR#1;

Bit(15-12) (R): last SYNC code from HFBR#5;

Bit(19:16) (R): last SYNC code from loopback;

Bit(20) (R): SYNC history fifo empty;

Bit(21) (R): SYNC history fifo has more than 512 entries;

Bit(22) (R): SYNC history fifo full (reached 1024 entries);

Bit 31-24 (R): Sync source monitoring.

➤ Address offset: 0x00028: Busy source registers:

Bit 15-0 (R/W): Busy source enables:

- 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;
- Bit 6: '1' BUSY if the trigger is out, but the acknowledge (from slave TI) has not been received;
- Bit 7: '1' enable TS feed_back 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 31-16 (R): BUSY source monitoring.

Bit 22: "OR" of all the 'trigger loss' for enabled (slave) TI.

(bit 6-15, 21-31 are meaningful for TI in master mode only)

➤ Address offset: 0x0002C: Clock source selection:

Bit 1-0 (R/W): software bit switch to control the clock source. Reset default 00;

Bit[1:0] = 00: oscillator clock;

Bit[1:0] = 01: HFBR#5 clock input;

Bit[1:0] = 10: HFBR#1 clock input;

Bit[1:0] = 11: Front panel 34-pin connector clock input.

Bit 19-16(R/W): "0101", to set the TI into bridge mode. (override the Bit(1:0) setting)

Bit 23-20(R/W): "0111", to set the TI into TS mode, that is: the Trigger2 is actually 20ns-wide trigger1.
(Firmware v12.1 and later, and not in the halls yet)

Bit 27-24(R): switch SC1 bit(5:2) readout, available for TIqsf only.

➤ Address offset: 0x00030: Trigger_1 pre-scale:

Bit 15-0 (R/W): pre-scale factor: $\text{Rate} = \text{Rate}_0 / (\text{Bit}(15:0) + 1)$.

Bit 19-16 (R/W): Multiply factor for FPTrgIn only (not the TS_codeIn, $2^{**}(\text{bit}(19:16))$).

Bit 22-20 (R/W): Multiply enable, "101" (integrating Adc) to switch to multiply mode from pre-scale mode

Bit 31-23 (R/W): Preload the counter for multiplying logic. (Counter(12:4) & "0000").

Bit 31-16 valid for firmware V12.2 and later

➤ Address offset: 0x00034: Trigger block inhibit:

Bit 7-0 (R/W): Trigger inhibit threshold (in the unit of event blocks); Reset default 0x01;

Bit 15-8 (R): Number of blocks in the DAQ ready to be readout.

Bit 20-16 (R): Number of events before the block is formed. Set to "1111" if more than 30 (and <256).

Bit 23-21 (R): *When the TI is used in SBS, the number of events which have Fastbus TDC data or ADC data.*
Set to "111" if more than 6 (and <16).

Bit 26-24 (R): Number of IRQ waiting, set to "111" if more than 6 (and <256).

Bit 27 (R): if '1', At least one readout trigger is dumped (no trigger data) because of the FIFO is overflowing
(to conserve the TI data integrity).

Bit 28 (R): if '1', the RUN is stopped because the block number (readout) has reached. (set by 0xFC)

Bit 29 (R): if '1', the event block is being filled by FillTrg;

Bit 30 (R): SyncReset Request set. If '1', SyncReset is required. To be issued by TImaster/TS.

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 $(\text{Bit}(6:0) * (16/500 \text{ ns}))$; Bit7 determines 16ns or 500ns step. Reset default 0x03;

Bit 15-8 (R/W): no more than 2 trigger in $(\text{Bit}(14:8) * (16/500\text{ns}))$; Bit15 determines 16ns or 500ns step. Reset default 0x03;

Bit 23-16 (R/W): no more than 3 triggers in $(\text{Bit}(22:16) * (16/500 \text{ ns}))$; Bit23 determines 16ns or 500ns step. Reset default 0x03;

Bit 31-24 (R/W): no more than 4 triggers in $(\text{Bit}(30:24) * (16/500 \text{ ns}))$. Bit31 determines 16ns or 500ns step. Reset default 0x03;

➤ Address offset: 0x0003C: Trigger coincidence window:

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

Bit24-16 (R/W): Set the delay between TRIGGER2 and the (TRIGGER2 generated) TRIGGER1. The delay is in 4ns step with an offset (minimum setting) of ~2.6us;

Bit 31 (R/W): '1' to enable the TriggerInput (front panel 6:1) pattern LEVEL readout (TI format data word control bit#3).

➤ Address offset: 0x00044: Front panel trigger input enable (in TIp firmware):

Bit 5-0 (R/W): Front panel trigger input enable (TScore). Reset default 0x00000000;

➤ Address offset: 0x00048: Front panel generic trigger input enable (before TIp firmware):

Bit 15-0 (R/W): Front panel generic trigger input enable (34-pin TI). Reset default 0x00000000; The lower 6 bits are used for front panel TS_code input control.

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

Bit 3-0 (R/W): 4-bit output to the front panel generic output connector; Reset default 0x0000;

Bit 7-4 (R): set to '0000';

Bit 15-8 (R): Number of data blocks in the FIFO in VME block (VME readout), same as reg0x34 bit(15:8) if Number of data blocks in FIFO is less than 256.

Bit 23-16 (R): Number of data blocks ready for Interrupt Request, bit(19:16) should be the same as reg0x34 bit(27:24).

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

Changed to "extra Readout Acknowledgement" count from tip (Ver7.1);

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;

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

Bit 15-8 (R/W): Beam Pass delay, for TIbridgeTS mode only; 'b1011,1xxx, xxx=001,010,011,100,101 for beam pass 1, 2, 3, 4, and 5 respectively;

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

Bit 23-16 (R): on TI: SYNC phase of HFBR#5 input;

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

- Address offset: 0x00054: ROC readout, number of blocks to be readout (this is a difference counter between the number_of_blocks_sent_out and the number_of_blocks_acknowledged):

Bit 7-0 (R): Number of blocks to be readout on VME_ROC;

Bit 15-8 (R): Number of blocks to be readout on ROC#2 (VTP_ROC);

Bit 23-16 (R): Number of blocks to be readout on ROC#3;

Bit(31:24 (R): Number of blocks to be readout on ROC#4.

- Address offset: 0x00058: ROC readout, number of blocks to be readout (ROC#3-8 have not been implemented yet): (before V11.3)

Bit 7-0 (R): Number of blocks to be readout on ROC#5;

Bit 15-8 (R): Number of blocks to be readout on ROC#6;

Bit 23-16 (R): Number of blocks to be readout on ROC#7;

Bit(31:24 (R): Number of blocks to be readout on ROC#8.

- Address offset: 0x00058: ROC readout (tip113 and later): (0x54 and 0x58 should be consistent, 0x58(7:0) – 0x58(23:16) == 0x54(7:0); 0x58(15:8) – 0x58(23:16) == 0x54(15:8))

Bit 7-0 (R): Number of ReadOut Acknowledge from TI (itself, reg0x100, data 0x80);

Bit 15-8 (R): Number of ReadOut Acknowledge from VTP

Bit 31-16 (R): Number of Readout Acknowledge to be sent to TD.

- Address offset: 0x00060(R): ChipScope readout (TIp firmware V14.2):

Bit 31-0 (R): ChipScope Channel#(127:96);

- Address offset: 0x00064: Front panel generic trigger input prescale (TIp firmware):

Bit 3-0 (R/W): FP Generic trigger input #1;

Bit 7-4 (R/W): FP Generic trigger input #2;

Bit 11-8 (R/W): FP Generic trigger input #3;

Bit 15-12 (R/W): FP Generic trigger input #4;

Bit 19-16 (R/W): FP Generic trigger input #5;

Bit 23-20 (R/W): FP Generic trigger input #6;

- Address offset: 0x00068(W): ChipScope trigger control (TIp firmware V14.1):

Bit 15-0 (W): ChipScope Channel(95:80) disable, '1' to disable, '0' to enable;

Bit 31:16 (/W): ChipScope Channel(95:80) logic level requirement, '1' for HIGH, '0' for LOW.

The ChipScope trigger is the AND of all these 16 channels. For example: if the trigger is Ch#94 HIGH and Ch#86 LOW, the bit#14 and bit#6 need be set to 0 and all the other 14 bits (in 15:0) need be set to '1' to bypass; bit#30 needs be set to '1', and bit#22 needs be set to '0', and all the other 14 bits (in 31:16) are 'do not care'; so this setting will work 4000BFBF.

- Address offset: 0x00068(R): ChipScope readout (TIp firmware V14.1):

Bit 31-24 (R): ChipScope Channel#(31:24);
 Bit 23-20 (R): Fixed pattern, 0xA in V14.1, ChipScope Ch#(23:20) in v14.2;
 Bit 19-16 (R): Fixed pattern, 0x5 in v14.1, 0x6 in v14.2;
 Bit 15-0 (R): Number of read from the ChiScope FIFO (1024 deep);

- Address offset: 0x0006C(R): ChipScope readout (TIp firmware V14.1):

Bit 31-0 (R): ChipScope Channel#(63:32);

- Address offset: 0x00070(R): ChipScope readout (TIp firmware V14.1):

Bit 31-0 (R): ChipScope Channel#(95:64);

Comment: The ChipScope Ch#(95:60), Ch#(59:24), Ch#(127:96;23:20) are saved in three separate FIFOs respectively. The register 0x68 (only this one) read will increase the fifo_read pointer after presenting the data to the VME bus. Because of the data available latency from Fifo_read command, the proper sequence is to read 0x68 the last (for example 0x60, 0x70, 0x6c, then 0x68).

If the 0x68 is read first, the ChipScope Ch#(31:24) will be one clock cycle off.

- Address offset: 0x00074: VME event type setting (TIp firmware):

Bit 23-16 (R/W): VME periodic VME event type, default 0xFD;

Bit 31-24 (R/W): VME pseudo random trigger event type, default 0xFE.

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

Bit 3-0 (R/W): FP Generic trigger input #1;

Bit 7-4 (R/W): FP Generic trigger input #2;

Bit 11-8 (R/W): FP Generic trigger input #3;

Bit 15-12 (R/W): FP Generic trigger input #4;

Bit 19-16 (R/W): FP Generic trigger input #5;

Bit 23-20 (R/W): FP Generic trigger input #6;

- Address offset: 0x00078: VME Sync Load

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;

0xAA: reset the TIs slave trigger ready register;

0xBB: Reset the event number, and trigger input scalars (the 0xDD will not do);

0xDD: (SyncReset), FPGA logic and counter reset, this reset does not go 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 for ~4 us.

0x66, 0x88: 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
- Address offset: 0x00080: Reset pulse width: Reset default 0000,0111
 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);
- 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:
 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; (There is NO requirement that it match with bit 3)
 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:
 Bit 15-0: Number of trigger_1s to be generated; If 0xFFFF, the number of event will not be limited.
 Bit 31-16: (trigger rate control) Time between triggers. $T = (120+30*\text{Bit}(30:16))*2048^{\text{Bit}(31)}$ ns. (Assuming that the ClkVme=33MHz or 30ns period).
- Address offset: 0x00090(R/W): VME Trigger_2 Generation:
 Bit 15-0: Number of trigger_2s to be generated;
 Bit 31-16: (trigger rate control) Time between triggers. $T = (120+30*\text{Bit}(30:16))*2048^{\text{Bit}(31)}$ ns. (Assuming that the ClkVme=33MHz or 30ns period).
- Address offset: 0x00094 (R): Number of Blocks in the DAQ system:
 Bit 31-24: Number of events before a full data block;
 Bit 23-0: Number of full data blocks the TI has ever generated (since last reset or SyncEvt);
- Address offset: 0x00098 (R): SYNC history fifo, The fifo status is in register offset 0x24.
 Bit 31-21: time stamp of the SYNC command, in steps of ~4us;
 Bit(20): whether the time stamp has overflowed since previous SYNC code
 Bit (19): TImaster generated sync code valid;
 Bit (18:15): TImaster generated sync code;
 Bit (14): Loopback sync valid;
 Bit (13:10): Loopback sync code;

- Bit (9): HFBR#5 sync valid;
- Bit (8:5): HFBR#5 sync code;
- Bit (4): HFBR#1 sync valid;
- Bit (3:0): HFBR#1 sync code;
- Address offset: 0x0009C (R/W): The FPGA running mode;
 - Bit 7-0: TS in running mode if set to 0x5A; 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.
 - Bit 23-16 (R): GTP_ROC_ID (since firmware V8.3)
 - Bit 24 (R): GTP_ROC_ID has been updated
- Address offset: 0x000A0 (R): Fiber latency measurement result:
 - Bit 31-23: latency data in 4ns steps
 - Bit 22-16: Delay in the IODelay, in $5000/64=78.125$ ps steps
 - Bit 15:0: Delay in the carry chain, two bits per slice, (or two mux per bit)
- Address offset: 0x000A4 (R): Fiber latency measurement result from HFBR#5
 - Bit 31-23: latency data in 4ns steps
 - Bit 22-16: Delay in the IODelay, in $5000/64=78.125$ ps steps
 - Bit 15:0: Delay in the carry chain, two bits per slice, (or two mux per bit)
- Address offset: 0x000A8 (R): Trigger live timer:
 - Bit 31-0 (r): board live time counter. The real time is $\text{Bit}(31:0)*256*30\text{ns}$. (ScalarLatch is required.)
- Address offset: 0x000AC (R): Trigger busy (trigger dead) timer:
 - Bit 31-0 (r): TID busy (cannot accept trigger, or trigger dead) time counter. The real time is $\text{Bit}(31:0)*256*30\text{ns}$. 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:
 - Bit 7-0: MGT[7:0] reset_done;
 - Bit 11-8: MGT PLL lock detected (two MGTs per PLL lock);
 - Bit 15-12: fixed at 0x9
 - Bit 22-16: loopback trigger buffer length, this is the similar register as bit(9:0) of register offset 0xB8.
 - Bit 23: TI(master) is ready.
 - Bit 31-24: if '1', the HFBR#8, #7,...#1 is receiving idle word (K byte plus non-K byte);
- Address offset: 0x000B4 (R): MGT STATUS_B registers:
 - 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:

Bit 9-0: Global trigger data buffer length (to be minimized to 0 for the longest fiber);
 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 26: IO_DELAY_CONTROL ready (from firmware tip82 v8.2 and later)
 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:

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:

Bit 7-0: Number of blocks to be readout on HFBR#1 (blocks need ROCack);
 Bit 15-8: Number of blocks is still missing on HFBR#1 (blocks need TRGack);
 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:

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:

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: FrontPanel Trigger counter, Number of triggers from FP trigger before lookup table.

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

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;

- 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);

- Bit 9-5: A24 address set by the onboard hardware switch;
- Bit 14-10: GA(4:0), VME64x geographic address;
- Bit 15: parity of GA(4:0);
- Bit 27-16: Number of blocks to be readout on TM itself (loopback);
- Bit 31-28: Number of blocks is still missing on TM itself (loopback);
- Address offset: 0x000D4 (R/W): Periodic Sync Event register
 - Bit 19-0: Number of data blocks to assert a sync event (periodic SyncEvent);
 - Bit 31-20: not used. When the Bit(19:0) is not zero, the Sync event is enabled.
- Address offset: 0x000D8 : Event number register
 - Bit 15-0 (R/W): Prompt Trigger Width, width = (bit(6:0) + 3) * 4ns;
 - Bit 31-16: higher 16-bit (bit 47-32) of event number counter;
- Address offset: 0x000DC (R): Event number register
 - Bit 31-0: lower 32-bit (bit 31-0) of event number counter.
- Address offset: 0x000E0 (R): FPGA input clock frequency measurement
 - Bit 31-16: 250 MHz input clock frequency, should be decimal ~25000;
 - Bit 15-0: ClkAux input frequency (from this, the 62.5 MHz clock is generated), should be decimal ~3125.
- Address offset: 0x000E4 (R): Collision check in MGT block
 - Bit 31-24: The TIstatus enabled (BUSY, Trigger_Ack, Readout Ack, etc), but the MGT is being reset;
 - Bit 15-0: The trigger enabled (only useful as TI master) but the MGT is being reset.
- Address offset: 0x000E8 (R): Acknowledge counters
 - Bit 31-24: Number of trigger acknowledge (trigger received);
 - Bit 23-16: Number of Readout acknowledge (it should be the same as reg0x58(23:16), though it is from a different counter in the firmware)
 - Bit 15-8: Number of data blocks received;
 - Bit 7-0: Any acknowledge (trigger received, readout, block received) during the TI being BUSY.
- Address offset: 0x000EC: ROC enable, GTP_ROC_ID
 - Bit 7-0 (R/W): ROC 8:1 enable, the default is 00000001
 - Bit 18-10: SyncResetRequest enable. For TI master and TD, this corresponds to the eight TI slaves (bit#18:11) and the loopback (bit#10).
 - Bit 28-20: Monitor of the SyncResetRequest corresponding to the bit#18:10.
- Address offset: 0x000F0 (R): valid for TM (or with TS function)
 - Bit 31-0: Number of valid code from Front Panel Async trigger inputs
- Address offset: 0x000F8 (R): valid for TI bridge (since V13.2)
 - Bit 15-0: Sync Beam Pass delay (number of Clk250 between FIFO write and read);
 - Bit 31-0: Trigger (fiber#5) Beam Pass delay (number of ClkUsr2 between FIFO write and read)

- Address offset: 0x000FC (R/W): End_of_Run number of block setting

Bit 31-0: number of block to End the run. If this is set to 0, there is no limit (disabled).

Bit 0: Enable the front panel (external input) signal to latch the trigger input scalars;

Bit 1: 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.**

- Address offset: 0x00100 (W): Reset and one-shot registers. The signal will be one ClkVme cycle. If the ClkVme is 25 MHz, the one-shot will be 40ns wide. Positive logic.

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 the optic transceiver, (the QSFP module on board).

Bit 6: if '1', reset the SYNC history fifo (clear the fifo);

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', generate a SyncEvent (forced SyncEvent). This can be used as end_of_run etc.

Bit 21: if '1', clock reset for the ClkVme, and all other clocks (similar to Sync_code 0x11), Be cautious.

Bit 22: if '1', MGT Rx_CDR reset, which also includes RxReset.

Bit 23: if '1', generate Sync_Reset_Request.

Bit 24: if '1', all the trigger input scalars are latched (ready for read out), the BusyTimer and LiveTimer are also latched;

Bit 25: if '1', all the trigger input scalars are reset. (Bit 24 and Bit 25 can be set simultaneously). The event number is also reset by this.

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: 0x00104 (R/W): FP input trigger (TS Code) delay (in 4 ns steps):

Bit 8-0: TScore#1 trigger input delay (Channel#1);

Bit 18-10: TScore#2 trigger input delay (Channel#2);

Bit 28-20: TScore#3 trigger input delay (Channel#3);

- Address offset: 0x00108 (R/W): FP input trigger (TS Code) delay (in 4 ns steps):

- Bit 8-0: TScore#4 trigger input delay (Channel#4);
- Bit 18-10: TScore#5 trigger input delay (Channel#5);
- Bit 28-20: TScore#6 trigger input delay (Channel#6);
- Address offset: 0x00110 (R): SW#A (CTP) busy counter
 - Bit 31-0: 32-bit Busy timer of switch slot#A (same steps as live/busy timer).
- Address offset: 0x00114 (R): SW#B (SD) busy counter
 - Bit 31-0: 32-bit Busy timer of switch slot#B (SD) (same steps as live/busy timer).
- Address offset: 0x00118 (R): VME P2 Busy input
 - Bit 31-0: 32-bit Busy timer of Vme P2 busy input (same steps as live/busy timer).
- Address offset: 0x0011C (R): Front panel TDC busy counter
 - Bit 31-0: 32-bit Busy timer of front panel TDC input (same steps as live/busy timer).
- Address offset: 0x00120 (R): Front panel ADC busy input counter
 - Bit 31-0: 32-bit Busy timer of front panel ADC busy input (same steps as live/busy timer).
- Address offset: 0x00124 (R): Front panel inhibit input counter
 - Bit 31-0: 32-bit Busy timer of front panel inhibit (busy) (same steps as live/busy timer).
- Address offset: 0x00128 (R): TImaster trigger inhibit counter
 - Bit 31-0: 32-bit Busy timer of TImaster trigger generation (same steps as live/busy timer).
- Address offset: 0x138 (R/W): Extra hold off after trigger rule (minimum busy width)
 - Bit 31: '1' to enable the minimum busy width for trigger rule#4;
 - Bit(30:24): Minimum busy width for rule#4. $\text{MinimumWidth} = \text{bit}(30:24) * \text{Tclock}$. $\text{Tclock} = 480\text{ns}$ if bit#31 of Reg#1C is set to '0'; $\text{Tclock} = 480 * 32\text{ns}$ if bit#31 of Reg 0x1C is set to '1';
 - Bit 23: '1' to enable the minimum busy width for trigger rule#3;
 - Bit(22:16): Minimum busy width for rule#3. $\text{MinimumWidth} = \text{bit}(22:16) * \text{Tclock}$. $\text{Tclock} = 480\text{ns}$ if bit#31 of Reg#1C is set to '0'; $\text{Tclock} = 480 * 32\text{ns}$ if bit#31 of Reg 0x1C is set to '1';
 - Bit 15: '1' to enable the minimum busy width for trigger rule#2;
 - Bit(14:8): Minimum busy width for rule#2. $\text{MinimumWidth} = \text{bit}(14:8) * 16\text{ns}$. The maximum width setting is $\sim 2\mu\text{s}$.
 - Bit 7-0: Minimum busy width for trigger rule#1, not used, and no need for this.
- Address offset: 0x00140-0x0017C (W): event type lookup table loading: (the Address offset was 0x008CX "0x8C0 – 0x8FC" in earlier version ModTI. To use common design modules as Tlpcie, the offset was moved to lower address in TIP##)
 - 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), Refer to section 5.3.1 for details.
- Address offset: 0x00170 (R): Fiber#1 non-idle data history buffer

Bit 31: The history fifo empty status, '1' means EMPTY;

Bit 30-16: The time stamp of the history buffer, in 16-ns steps;

Bit 15-0: the received 16-bit 'trigger word', valid data, same as the 2-byte fiber interface.

Bit15:14	Bit 13:12	Bit11:9	Bit8	Bit7:3	Bit2:0
parity	1 0	0 0 1	0	Crate_ID	
		0 1 0	active	Trigger Source(7:0)	
		0 1 1	0	New Block Level (15:8)	
		0 1 1	1	New Buffer Level (31:24)	
	0 1	Busy,AckS,SRreq,SRreq*		Busy, Trg1Ack, Trg2Ack, BlkRcv, ROCack	x x x

- Address offset: 0x00174 (R): Fiber#5 non-idle data history buffer

Bit 31: The history fifo empty status, '1' means EMPTY;

Bit 30-16: The time stamp of the history buffer, in 16-ns steps;

Bit 15-0: the received 16-bit 'trigger word', valid data, same as the 2-byte fiber interface. Same decoding as the table in register 0x00170.

- Address offset: 0x00180 (R): FP input trigger scalar for #1: (the scalars are before pre-scale etc.)

Bit 31-0: Number of input#1 counter. The meaning of Bit (6:0) depends on Bit#31. If Bit#31 is 0, Bit(6:0) is the counter(6:0); if Bit#31 is 1, Bit(6:0) is the counter(37:31). This is basically a 38 bit counter readout. The same is true for registers 0x184~0x194.

- 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): FP input trigger scalar for #5:

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

- Address offset: 0x00194 (R): FP input trigger scalar for #6:

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

- Address offset: 0x0019C (R): TIM loopback readout busy counter

Bit 31-0: 32-bit Busy timer of TIM loopback (same steps as live/busy timer).

- Address offset: 0x001A0 (R): TI#1 readout busy counter

Bit 31-0: 32-bit Busy timer of TI#1 (same steps as live/busy timer).

- Address offset: 0x001A4 (R): TI#2 readout busy counter

Bit 31-0: 32-bit Busy timer of TI#2 (same steps as live/busy timer).

- Address offset: 0x001A8 (R): TI#3 readout busy counter
 - Bit 31-0: 32-bit Busy timer of TI#3 (same steps as live/busy timer).
- Address offset: 0x001AC (R): TI#4 readout busy counter
 - Bit 31-0: 32-bit Busy timer of TI#4 (same steps as live/busy timer).
- Address offset: 0x001B0 (R): TI#5 readout busy counter
 - Bit 31-0: 32-bit Busy timer of TI#5 (same steps as live/busy timer).
- Address offset: 0x001B4 (R): TI#6 readout busy counter
 - Bit 31-0: 32-bit Busy timer of TI#6 (same steps as live/busy timer).
- Address offset: 0x001B8 (R): TI#7 readout busy counter
 - Bit 31-0: 32-bit Busy timer of TI#7 (same steps as live/busy timer).
- Address offset: 0x001BC (R): TI#8 readout busy counter
 - Bit 31-0: 32-bit Busy timer of TI#8 (same steps as live/busy timer).
- Address offset: 0x001D0 (R): optic transceiver HFBR#1 TI ID (in master mode)
 - Bit 7-0: Trigger Source Enable of the TI connected to fiber#1;
 - Bit 15-8: Crate ID of the TI connected to fiber#1.
 - Bit 23-16: Block level set on TI connected to fiber#1.
 - Bit 31-24: Buffer level set on TI connected to fiber#1.
- Address offset: 0x001D4 (R): optic transceiver HFBR#2 TI ID (in master mode)
 - Bit 7-0: Trigger Source Enable of the TI connected to fiber#2;
 - Bit 15-8: Crate ID of the TI connected to fiber#2.
 - Bit 23-16: Block level set on TI connected to fiber#2.
 - Bit 31-24: Buffer level set on TI connected to fiber#2.
- Address offset: 0x001D8 (R): optic transceiver HFBR#3 TI ID (in master mode)
 - Bit 7-0: Trigger Source Enable of the TI connected to fiber#3;
 - Bit 15-8: Crate ID of the TI connected to fiber#3.
 - Bit 23-16: Block level set on TI connected to fiber#3.
 - Bit 31-24: Buffer level set on TI connected to fiber#3.
- Address offset: 0x001DC (R): optic transceiver HFBR#4 TI ID (in master mode)
 - Bit 7-0: Trigger Source Enable of the TI connected to fiber#4;
 - Bit 15-8: Crate ID of the TI connected to fiber#4.
 - Bit 23-16: Block level set on TI connected to fiber#4.
 - Bit 31-24: Buffer level set on TI connected to fiber#4.
- Address offset: 0x001E0 (R): optic transceiver HFBR#5 TI ID (in master mode)
 - Bit 7-0: Trigger Source Enable of the TI connected to fiber#5;

Bit 15-8: Crate ID of the TI connected to fiber#5.
Bit 23-16: Block level set on TI connected to fiber#5.
Bit 31-24: Buffer level set on TI connected to fiber#5.

- Address offset: 0x001E4 (R): optic transceiver HFBR#6 TI ID (in master mode)

Bit 7-0: Trigger Source Enable of the TI connected to fiber#6;
Bit 15-8: Crate ID of the TI connected to fiber#6.
Bit 23-16: Block level set on TI connected to fiber#6.
Bit 31-24: Buffer level set on TI connected to fiber#6.

- Address offset: 0x001E8 (R): optic transceiver HFBR#7 TI ID (in master mode)

Bit 7-0: Trigger Source Enable of the TI connected to fiber#7;
Bit 15-8: Crate ID of the TI connected to fiber#7.
Bit 23-16: Block level set on TI connected to fiber#7.
Bit 31-24: Buffer level set on TI connected to fiber#7.

- Address offset: 0x001EC (R): optic transceiver HFBR#8 TI ID (in master mode)

Bit 7-0: Trigger Source Enable of the TI connected to fiber#8;
Bit 15-8: Crate ID of the TI connected to fiber#8.
Bit 23-16: Block level set on TI connected to fiber#8.
Bit 31-24: Buffer level set on TI connected to fiber#8.

- Address offset: 0x001F0 (R): The TI master (itself) ID (in master mode)

Bit 7-0: Trigger Source Enable of the TI itself, should be the same as bit7-0 of register 0x20;
Bit 15-8: Crate ID of the TI itself, should be the same as bit7-0 of register 0x00.
Bit 23-16: Block level set on TI connected to loopback (itself).
Bit 31-24: Buffer level set on TI connected to loopback (itself).

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

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), Refer to section 5.3.1 for details.

- Address offset: 0x02000~0x021FC (R): Mirror registers for VXS switch slot #B on TI board

Bit 31-0: 32-bit wide SD status (or data).

- Address offset: 0x02800~0x029FC (R): Mirror registers for VXS switch slot #A on TI board

Bit 31-0: 32-bit wide CTP/GTP status (or 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, and VME to I2C engine for the P2 connector (could be used for anything). In the I2C engines design, only the lower one-byte or 2-byte of the 32-bit data word is used. The higher bytes are not used.

Address offset: 0x1XXXX: JTAG for PROM; Refer to the programming manual for VME to JTAG design for details.

Address offset: 0x2XXXX: JTAG for FPGA;

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

Address offset: 0x4XXXX: I2C for VXS switch slot #B;

Address offset: 0x5XXXX: I2C for optic transceivers (AFBR-79EIDZ);

Address offset: 0x6XXXX: Serial Flash memory interface.

Address offset: 0x0NXXX: SFM memory readout: (to be implemented)

0x07X0: IO_Delay #X reset;

0x07X4: IO_Delay #X delay increment (by 1);

0x07X8: IO_Delay #X de-serialized data readout, idle='0xFFFF';

0x07Xc: IO_Delay #X automatic delay increment by a number stored in SFM.

6.4 VME data acquisition:

For data acquisition, the A32 block reads are used. The base address is set by the upper 9 bits of A24 register 0x00010, that is A[31:23] = RegData[31:23] of A24=0x00010.

7 Backplane pin out and Front panel pinout tables:

7.1 VXS P0 Pinout Table

Payload slot#18, TI or TImaster			
Pin name	Signal Description	Signal Level	Direction
DP1 (A1+, B1-)	CLOCK_C	LVPECL(DP)	PP18 → SWA
DP2 (D1+, E1-)	CLOCK_D	LVPECL(DP)	PP18 → SWA
DP3 (B2+, C2-)	Not Used		
DP4 (E2+, F2-)	SYNC	LVPECL(DP)	PP18 → SWA
DP5 (A3+, B3-)	TRIG1	LVPECL(DP)	PP18 → SWA
DP6 (D3+, E3-)	TRIG2	LVPECL(DP)	PP18 → SWA
DP7 (B4+, C4-)	TP_Data_LINK	LVDS(DP)	PP18 ↔ SWA
DP8 (E4+, F4-)	BUSY	LVDS	PP18 ← SWA
SE1 (G1)	SCL	I2C (+3.3V)	PP18 → SWA
SE2 (G3)	SDA	I2C (+3.3V)	PP18 ↔ SWA
DP23 (B12+, C12-)	CLOCK_A	LVPECL(DP)	PP18 → SWB
DP24 (E12+, F12-)	CLOCK_B	LVPECL(DP)	PP18 → SWB
DP25 (A13+, B13-)	TOKEN_OUT	LVDS(DP)	PP18 → SWB
DP26 (D13+, E13-)	SYNC	LVPECL(DP)	PP18 → SWB
DP27 (B14+, C14-)	TRIG1	LVPECL(DP)	PP18 → SWB
DP28 (E14+, F14-)	TRIG2	LVPECL(DP)	PP18 → SWB
DP29 (A15+, B15-)	SD/GTP_Data_Link	LVDS, 250Mbps	PP18 ↔ SWB
DP30 (D15+, E15-)	BUSY	LVDS	PP18 ← SWB
SE7 (G13)	SCL	I2C(+3.3V)	PP18 → SWB

SE8 (G15)	SDA	I2C(+3.3V)	PP18 \leftrightarrow SWB
-----------	-----	------------	----------------------------

7.2 Other payload slots for subsystem TS:

DP23 (B12+, C12-)	TRIG_LINK	LVPECL(DP)	PP \leftarrow SWB
DP24 (E12+, F12-)	SYNC	LVPECL(DP)	PP \leftarrow SWB
DP25 (A13+, B13-)	CLK250MHz	LVPECL(DP)	PP \leftarrow SWB
DP26			
DP27			
DP28			
DP29	SD_Data_link	LVDS	PP \leftrightarrow SWB
DP30	BUSY	LVDS	PP \rightarrow SWB
SE7			
SE8			

7.3 VME P2 User-defined pin table

Similar to this, but Row-C is used. The two adjacent pins are used as a pair for differential signals.

Pin name	Signal Name	Signal Level
C01	SCL	I2C (LVCMOS)
C02	SDA	I2C (LVCMOS)
C05	CLK250_P	ECL
C06	CLK250_N	ECL
C09	CLK_A_P	ECL
C10	CLK_A_N	ECL
C13	CLK_B_P	ECL
C14	CLK_B_N	ECL
C17	TRIG1_P	ECL
C18	TRIG1_N	ECL
C21	TRIG2_P	ECL
C22	TRIG2_N	ECL
C25	SYNC_P	ECL
C26	SYNC_N	ECL
C29	BUSY+	ECL
C30	BUSY-	ECL

7.4 Front panel ADC connector pin table (not available with 8-port TImaster)

Pin name	Signal Name	Signal Level
1 / 2	FADC BUSY_in + / -	Differential
3 / 4	Trigger_out + / -	ECL
5 / 6	SyncReset_out + / -	ECL
7 / 8	GND	Ground
9 / 10	Clock_out + / -	ECL

7.5 Front panel TDC connector pin table (not available with 8-port TImaster)

Pin name	Signal Name	Signal Level
1 / 2	F1TDC BUSY_in - / +	Differential
3 / 4		Floating
5 / 6	Trigger_out - / +	ECL
7 / 8	SyncReset_out - / +	ECL
9 / 10		Floating
11 / 12		Floating
13 / 14	Clock_out - / +	LVPECL

* The pin polarity is different from other front panel connectors, and the Clock_out is LVPECL, not ECL.

** The BUSY_in can be any differential signals (LVDS, LVPECL, or ECL) as the receiver claims 'any differential inputs'

7.6 Front panel lower 34-pin connector (generic IO) pin table

Pin name	Signal Name	Signal Level
1 / 2	BUSY_out + / -	ECL
3 / 4	Trigger_out + / -	ECL
5 / 6	UserControlled_out1 + / -	ECL
7 / 8	UserControlled_out2 + / -	ECL
9 / 10	UserControlled_out3 + / -	ECL
11 / 12	UserControlled_out4 + / -	ECL
13 / 14	OT1 + / -	ECL
15 / 16	PromptTrigger_out + / -	ECL
17 / 18	Inhibit_in + / -	ECL
19 / 20	Trigger_in + / -	ECL
21 / 22	Trigger_ch#1_in + / -	ECL
23 / 24	Trigger_ch#2_in + / -	ECL
25 / 26	Trigger_ch#3_in + / -	ECL
27 / 28	Clock_in + / -	ECL
29 / 30	Trigger_ch#4_in + / -	ECL
31 / 32	Trigger_ch#5_in + / -	ECL
33 / 34	Trigger_ch#6_in + / -	ECL

* The Trigger_ch#n_in are going through a trigger lookup table, which can generate trigger_1 and trigger_2, or sync_events depending on the trigger lookup table. The Trigger_in generates trigger_1 only, and ORs with the lookup table output.

** The Inhibit_in is basically the busy input, which will disable the trigger generation when the TI is in master mode (generating event readout triggers).

8 TID Operation examples:

The following is some operating procedures at VxWorks interactive mode. This is just a memo for the quick test of the TID board. They may change as the TID debug proceeds. First, one needs to login the VME controller. Here is the sequence:

Xming to PHECDA, linux server computer

From any xterm (or PUTTY), telnet to DAVW1 (neither username, nor password is needed). Only one telnet process is supported for the MVME6100 module). The address mapping for A24 is 0x90xxxxxx.

For DAVW8, which is a MVME5100 module, the address mapping for A24 is 0xFA000000. The M6100 has twice as much as M5100 in A32 memory. The M5100 is 0x08000000-0x0FFFFFFF; while the M6100 is 0x08000000-0x17FFFFFFF.

The following commands are assuming that the TI is in slot#21 VME64x compatible crate (Geographic address available). 0xA8 = 10101xxx, that is the Geographic Address GA=21 or 10101.

8.1 A24 register echoing (write and read):

→ *(0x90A80008)=0x5566aa99; the same register should be read out

8.2 Readout the FPGA user_ID:

The FPGA user code is readout through VME_to_JTAG engine. The FPGA user_ID is firmware specific. If this code matches, it is TID and the right version of the firmware is loaded.

- *(0x90a80100)=0x04: VME_to_JTAG engine logic reset
- *(0x90aa003C)=0x0: Reset FPGA JTAG to 'reset_idle' state
- *(0x90aa092C)=0x3C8: enable user_ID readback
- *(0x90aa1F1C)=0x00: shift in 32-bit data, and the readback is user_ID. The user_ID should be 710xnmmm: n is the major version of the firmware, mmm is the revision of the firmware.

8.5 SW#A I2C operation:

The VME_to_I2C engine hides the serialization, and the two-bytes write (and/or read) can be finished by a single VME write (and/or read) command. For SW#A, the I2C chain is 011; for device 1101000, the I2C device at the FPGA is 000; assuming the byte address is 0xAD (10101101).

- *(0x90a80100)=0x01: VME_to_I2C engine reset
- *(0x90ab1EB4)=0xDADA: VME_to_I2C write
- m 0x90ab1EB4, 4: VME to I2C read. ignore the higher two bytes
- *(0x90a80004)=0x100: Set the I2C address to 0b1101xxx
- *(0x90ab1eb4)=0xdata: VME to I2C write

9 TID Operation procedures (software setup):

The following is the operating procedures at VxWorks interactive mode. This is just a memo for the quick test of the TI board. The procedure will be optimized as the test goes along. This assumes that the hardware switch is set properly (as TI, TD, TS/TD/TI etc.)

9.1 Fiber latency measurement:

- *(0x90a80100)=0x4000 //reset the digital IODELAY logic/clock
- *(0x90a80100)=0x2000 //phase alignment of the latency measurement signal
- *(0x90a80100)=0x8000 //Fiber latency measurement
- M 0x90a800a0,4 //Fiber latency readout
- *(0x90a80100)=0x800 //Sync phase alignment relative to the 250MHz clock
- *(0x90a80050)=0xAAxxBBxx // load the Sync delay (AA for HFBR#5, BB for HFBR#1). The BB should be based on the previous measurement.
- M 0x90a80050,4 //check the SYNC delay setting and phase alignment.

9.3 Trigger link startup:

- *(0x90a80078)=0x77
- *(0x90a80078)=0x77 //WriteStart, buffer_write_address reset and trigger data transmit disable
- *(0x90a8007c)=0x54 //Sync latency, delay before SYNC being serialized. This delay should cover the trigger word sterilizer/deserializer delay.
- *(0x90a80080)=0x7 // The Reset pulse width is (7+1)*4ns;

- ➔ `*(0x90a80078)=0x55` //Start the trigger link, and this SYNC command will enable the trigger link buffer readout.

9.7 Trigger/DAQ monitoring:

- ➔ `M 0x90a80034, 4` //polling the register to see if there is data block to read
- ➔ `M 0x90a800a8, 4` //TID trigger live timer (live time)
- ➔ `M 0x90a800ac, 4` //TID trigger busy timer (dead time)

9.8 Data Readout:

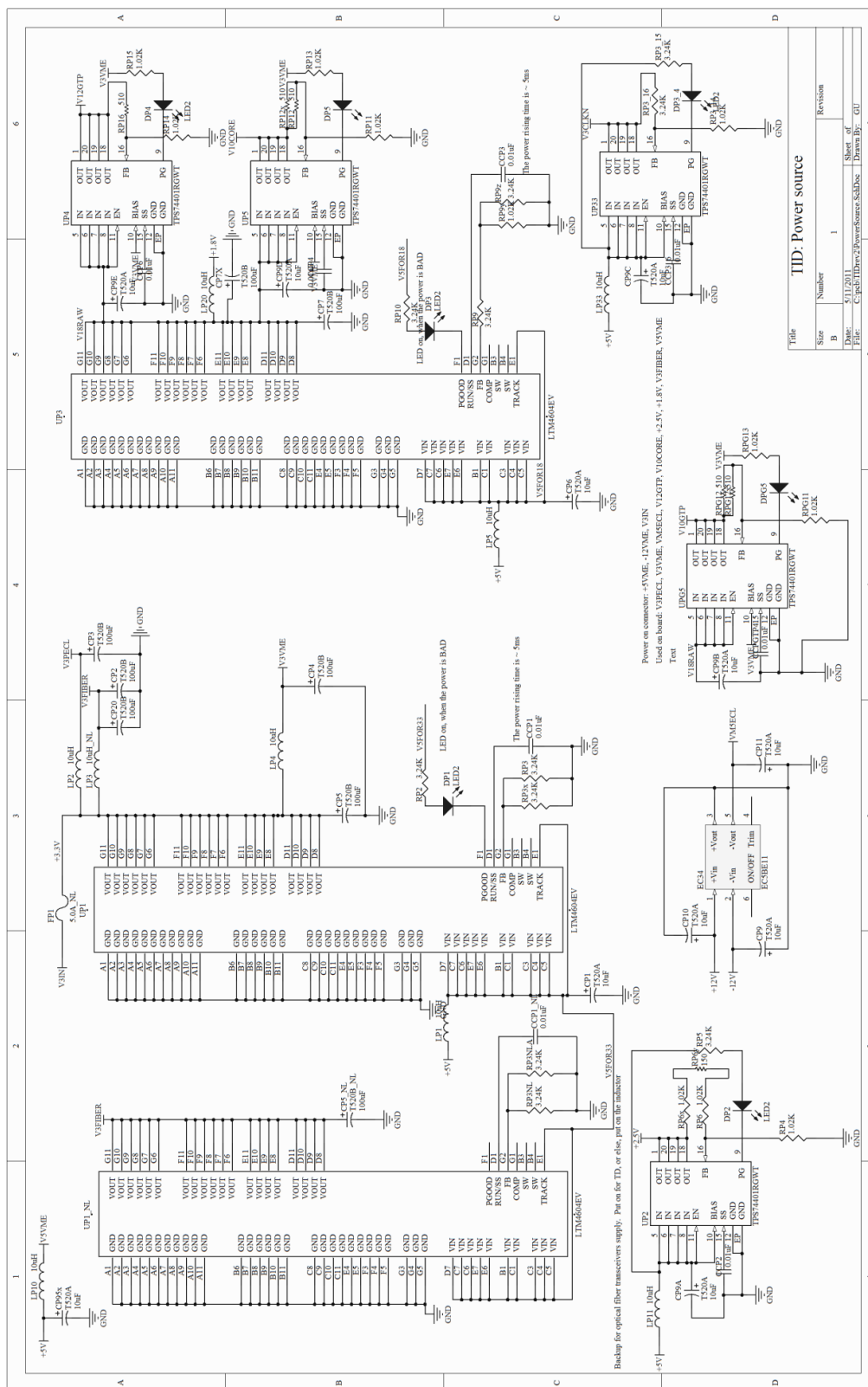
- ➔ `M 0x80300100,4` //A32 readout; or
- ➔ `tidRead(slot, nwords)` //readout nwords from TI; or
- ➔ `tidBERead(slot)` // readout a block, and using `Bus_error` to terminate the read; or
- ➔

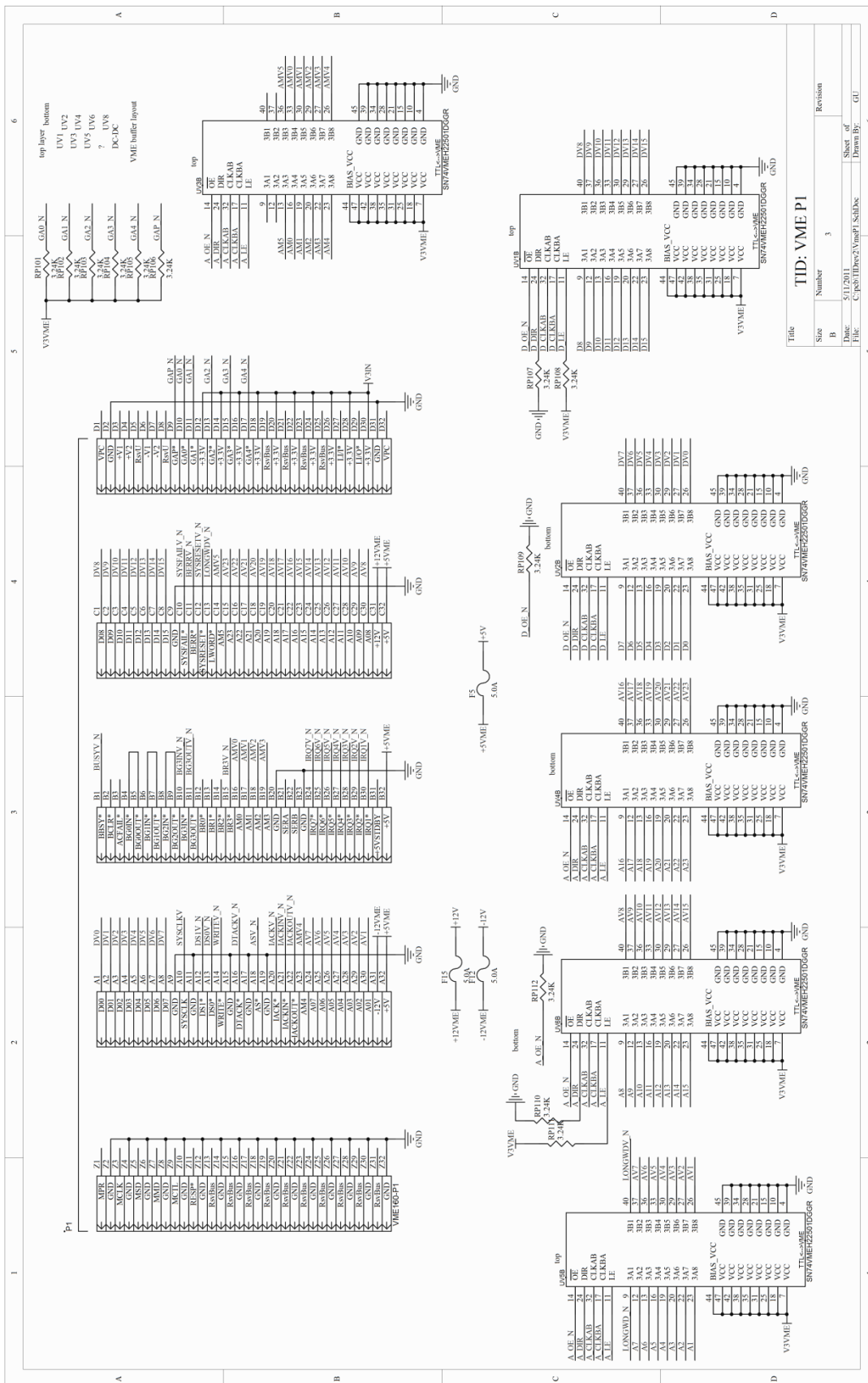
10. Citations:

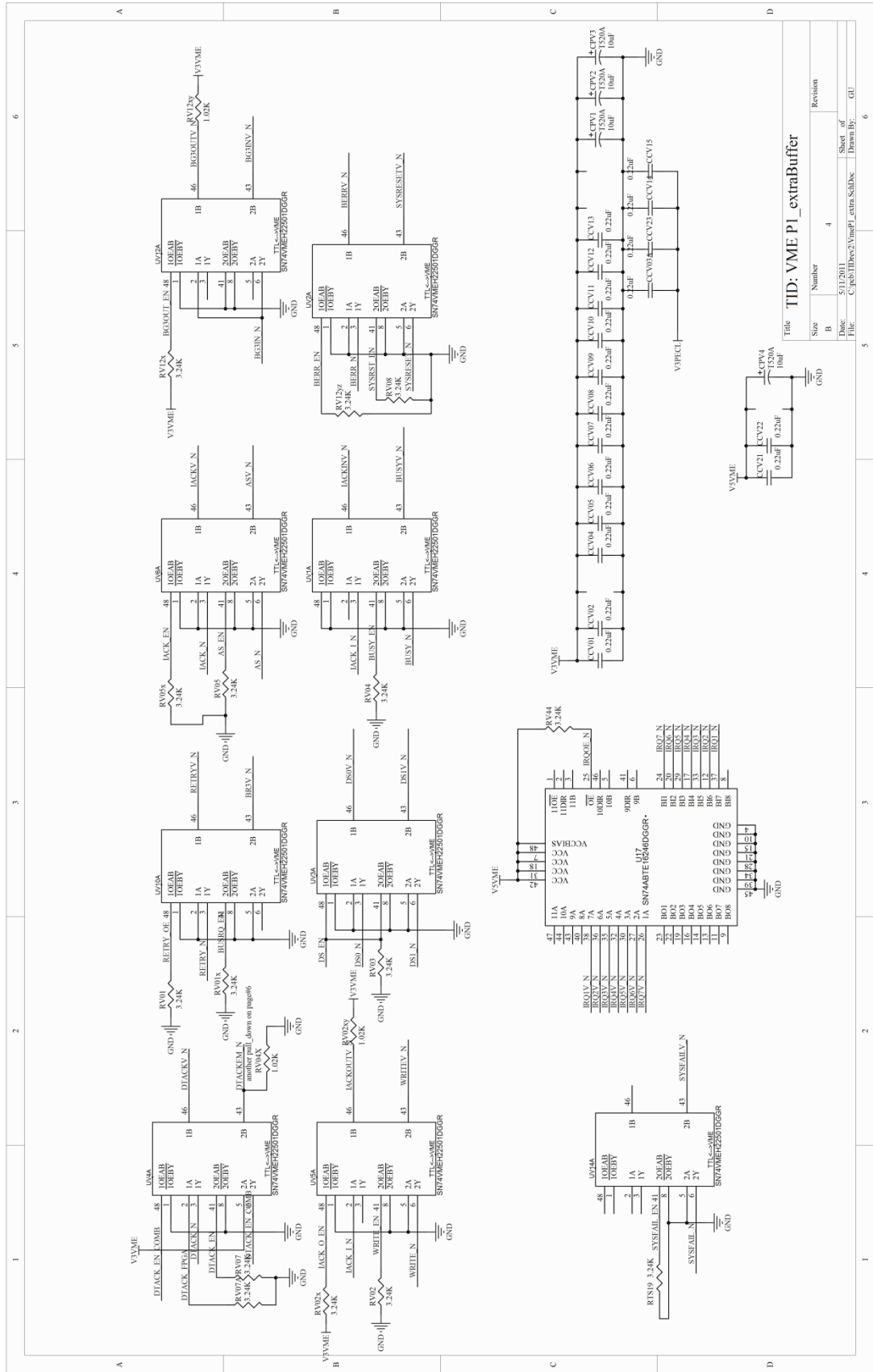
Works Cited

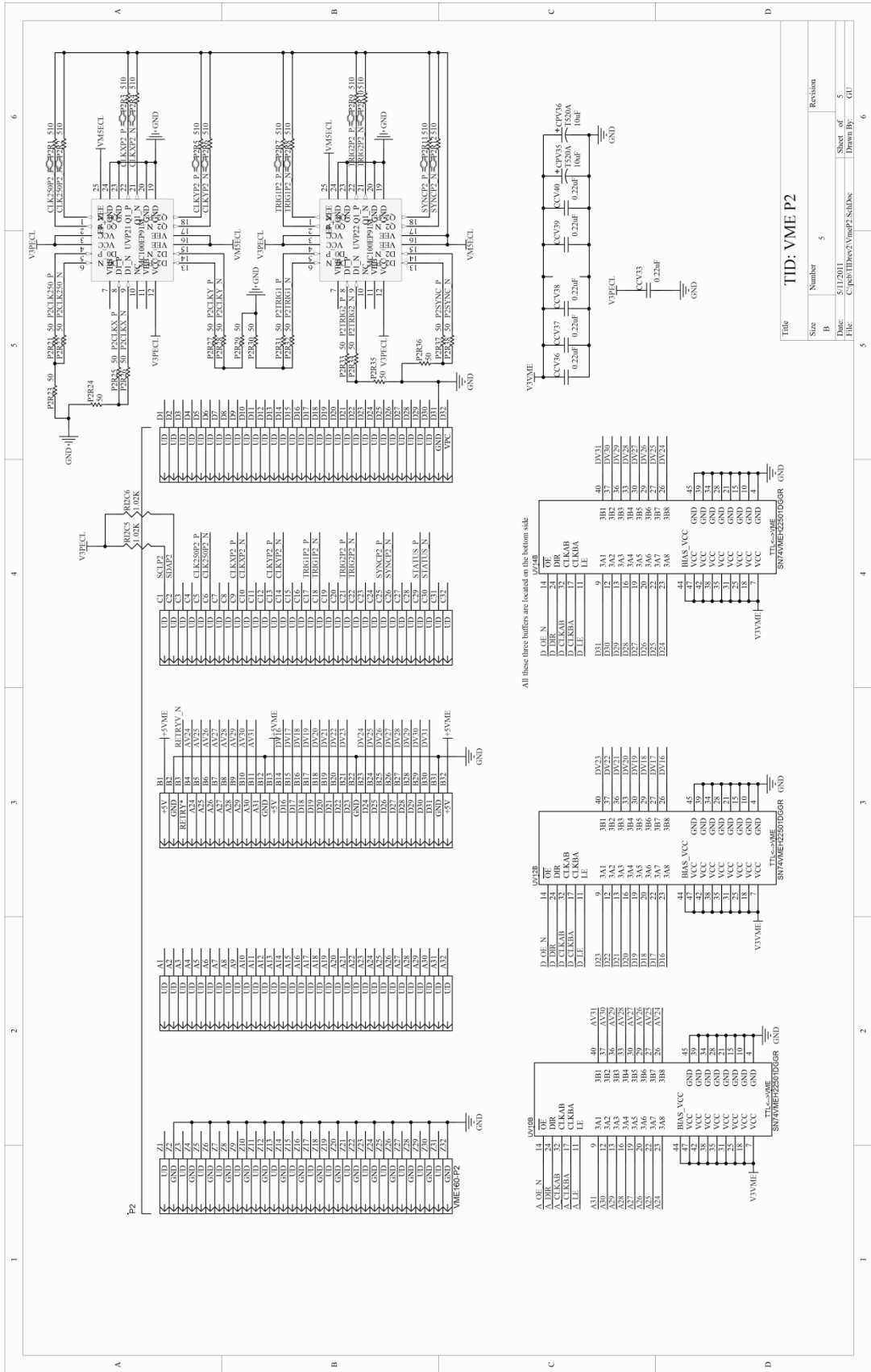
Chris. (2009). Global Trigger Processor., (pp. 1-14).
 Chris. (2009). Hall_D trigger layout., (pp. 20-25). Virginia.
 Collaboration, C. (2009). CLAS12 experiment. *Journal* , 1-25.
 Collaboration, G. (2009). GlueX experiment. *journal* , 1-20.
 Cuevas, C. (2008). Signal Distribution Module., (pp. 1-10).
 Ed. (2010). Trigger Supervisor Module., (pp. 1-19).
 experiments, C. (1990). CEBAF experiments. *journal* , 1-10.
 GU. (2010). Optical transceiver jitter measurement., (pp. 1-12).
 Raydo, B. (2008). Trigger Distribution board., (pp. 1-10).
 GU, (2010). VME to JTAG implementation
 GU, (2010). VME to I2C implementation

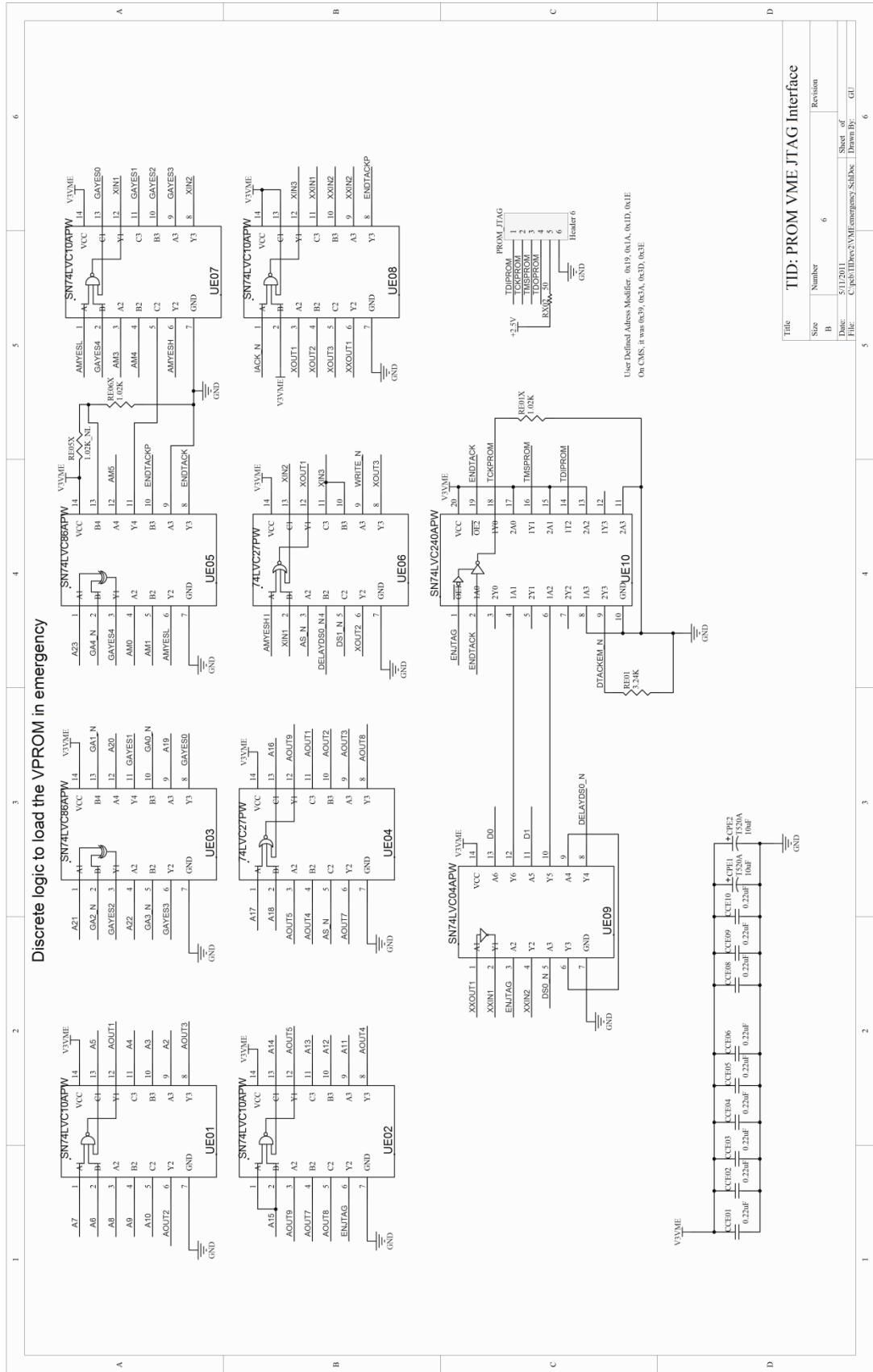
Appendix A: TID Schematics:

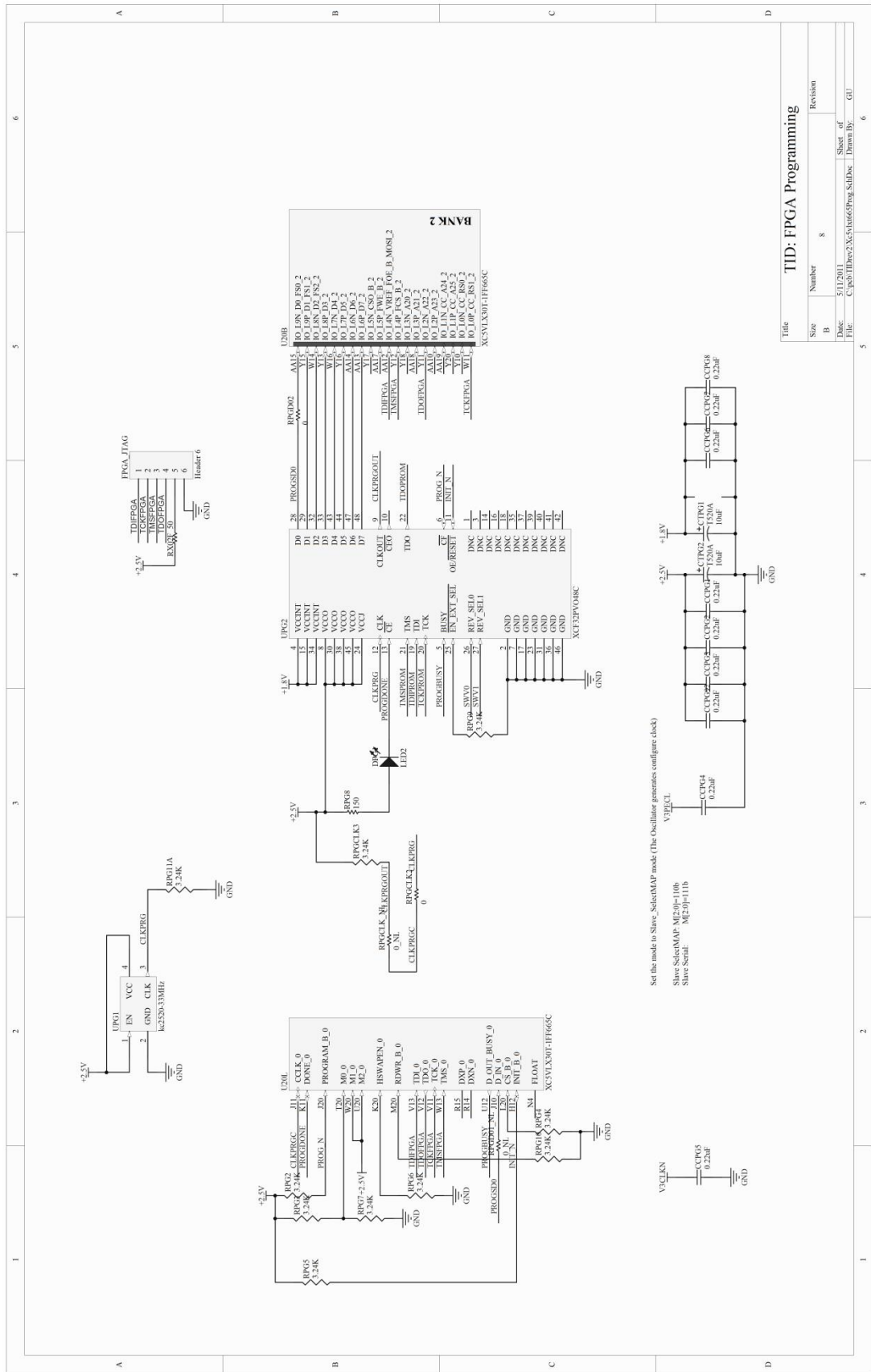






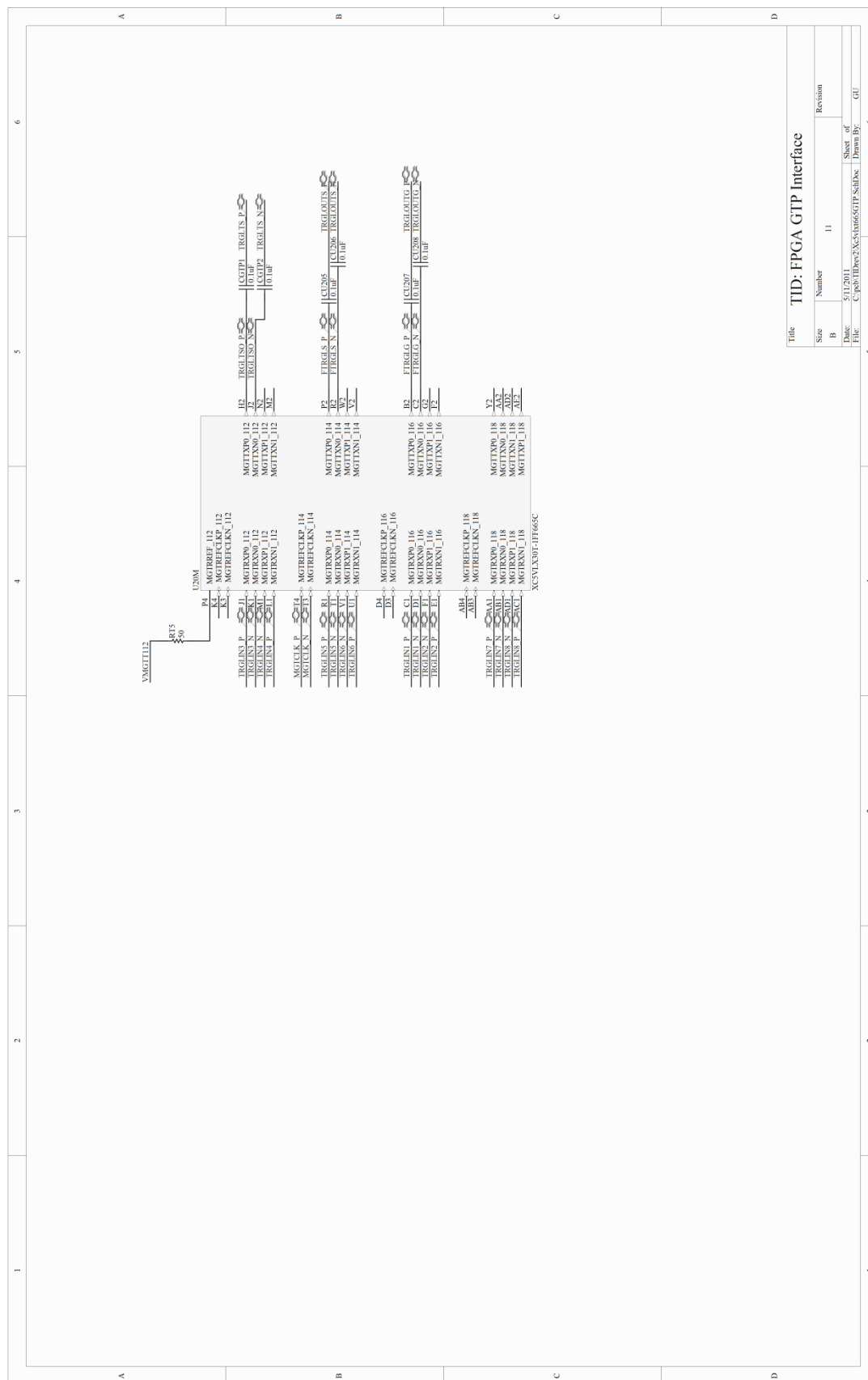


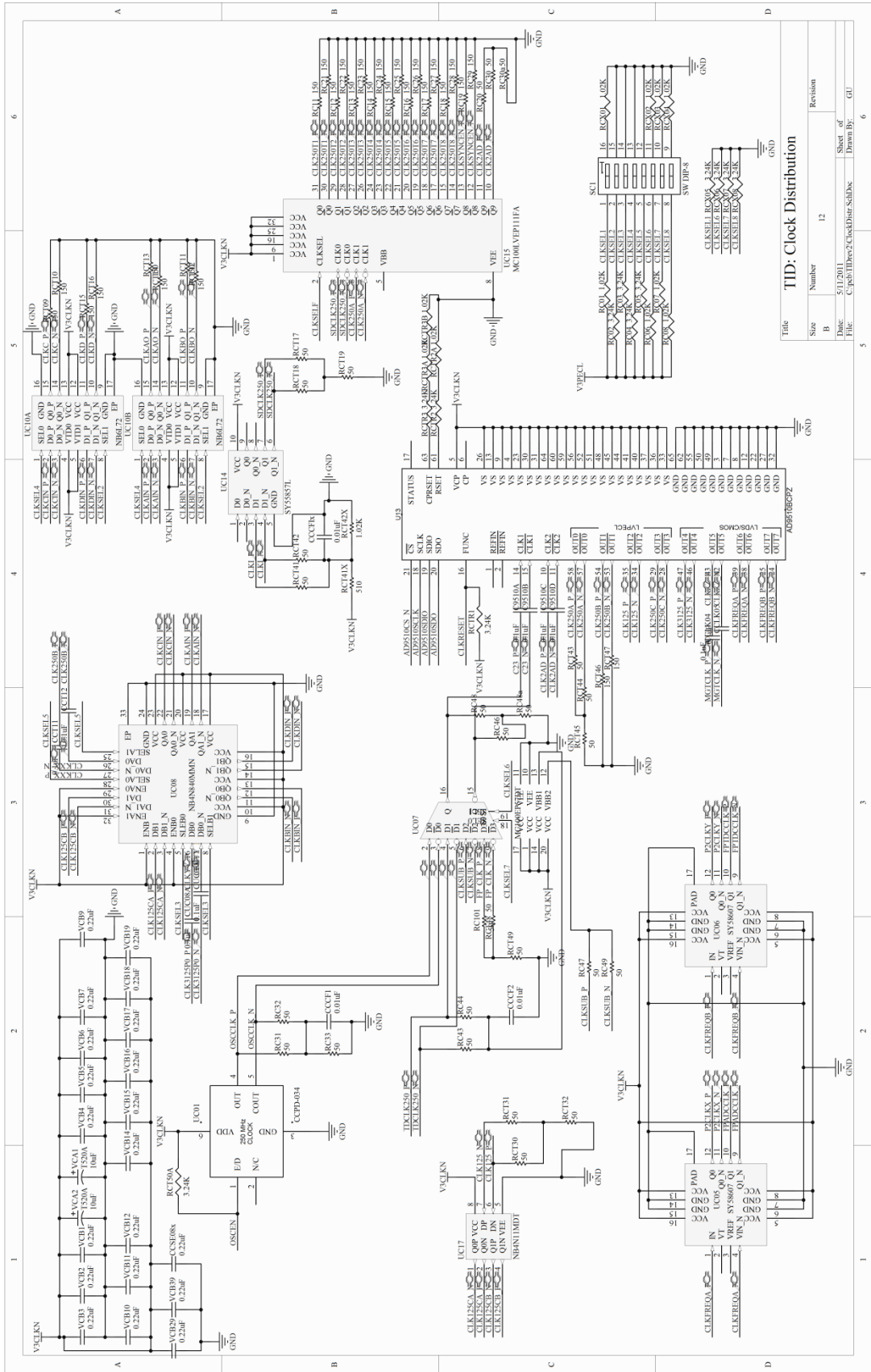




TID: FPGA Programming

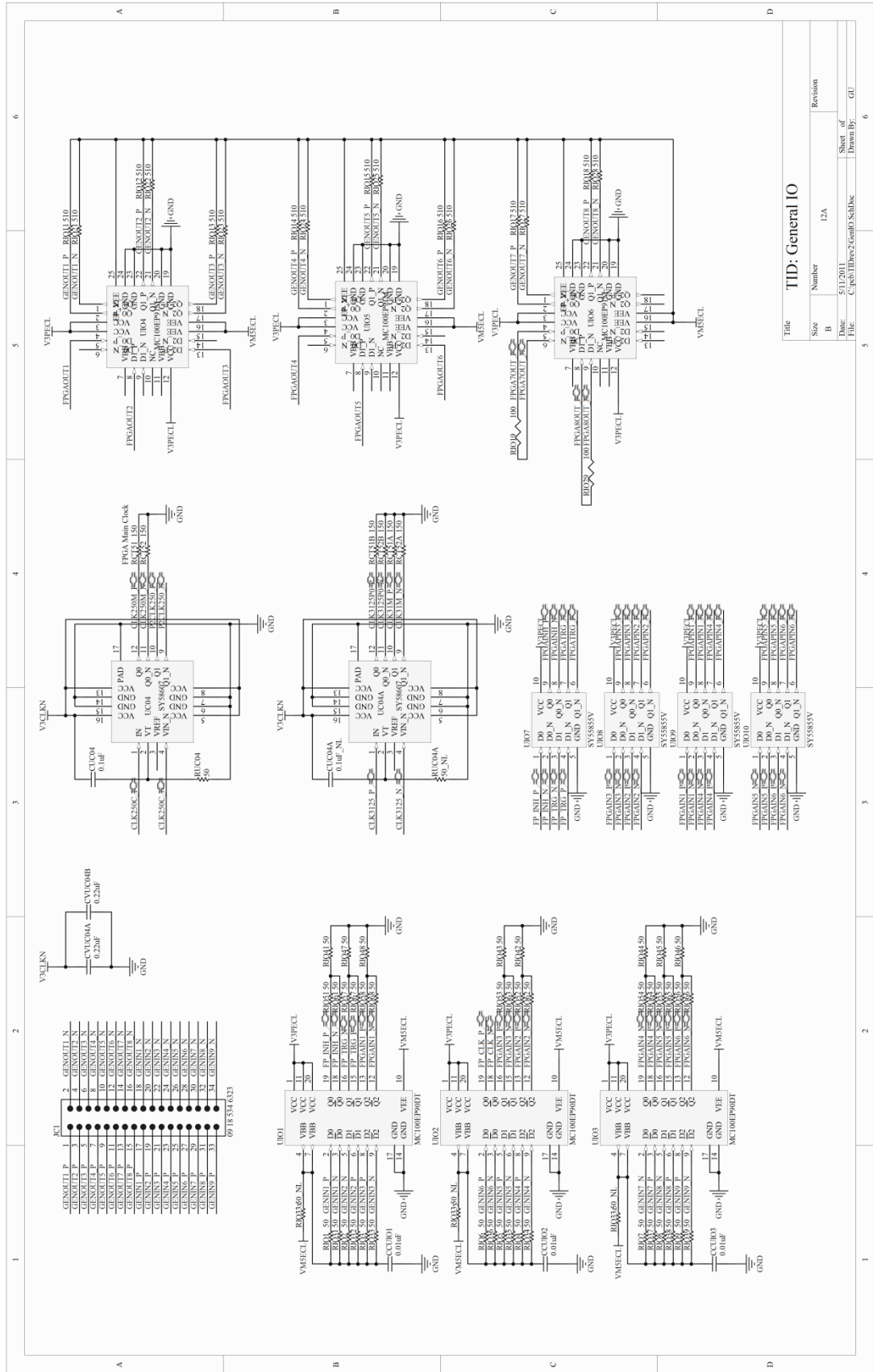
Title	Size	Number	Revision
B		8	
Date:	5/1/2011	Sheet of	
File:	C:\P3\1106\3\5\5\1066\Prog SchDoc	Sheet of	
		Sheet of	61

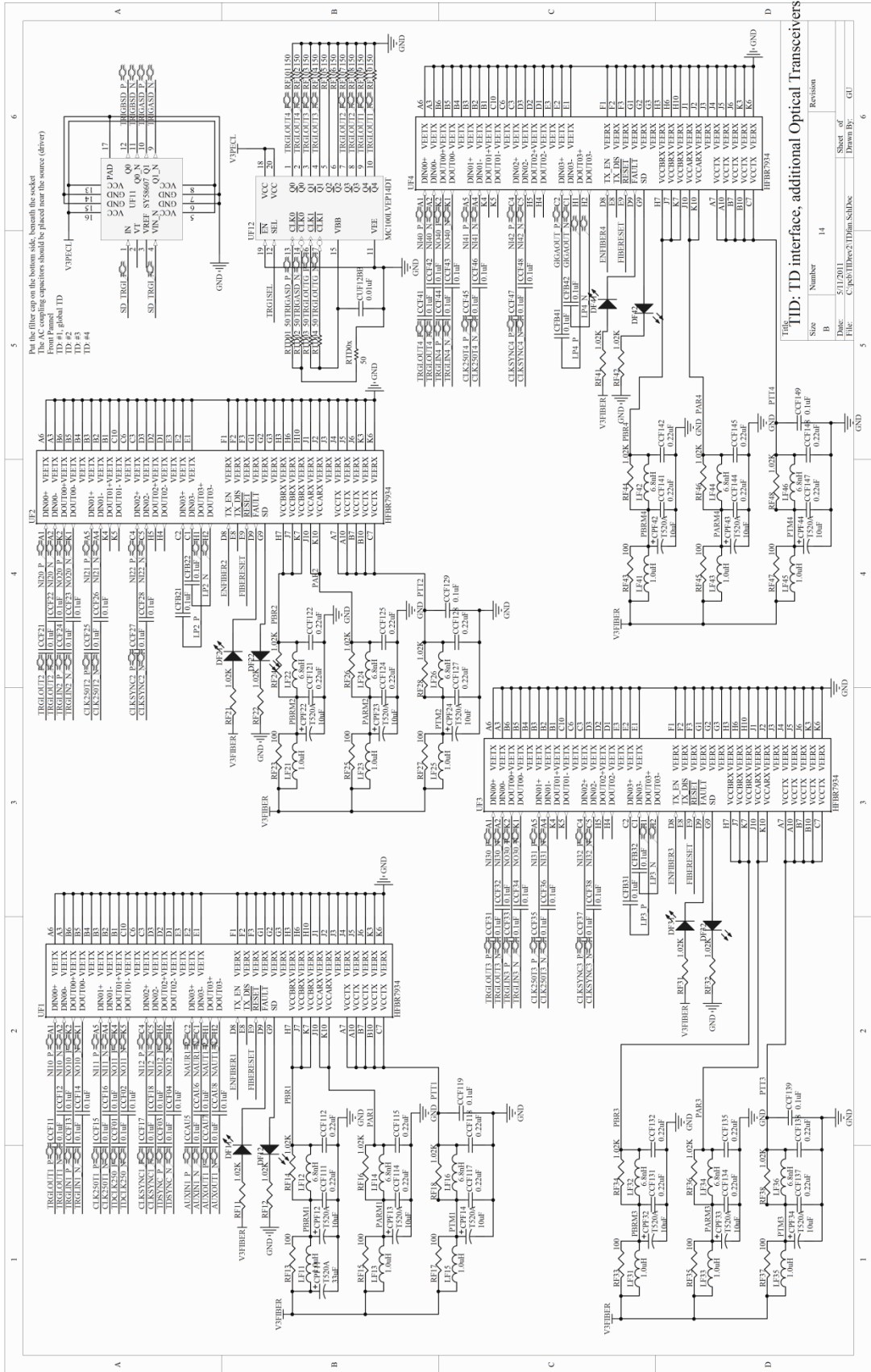


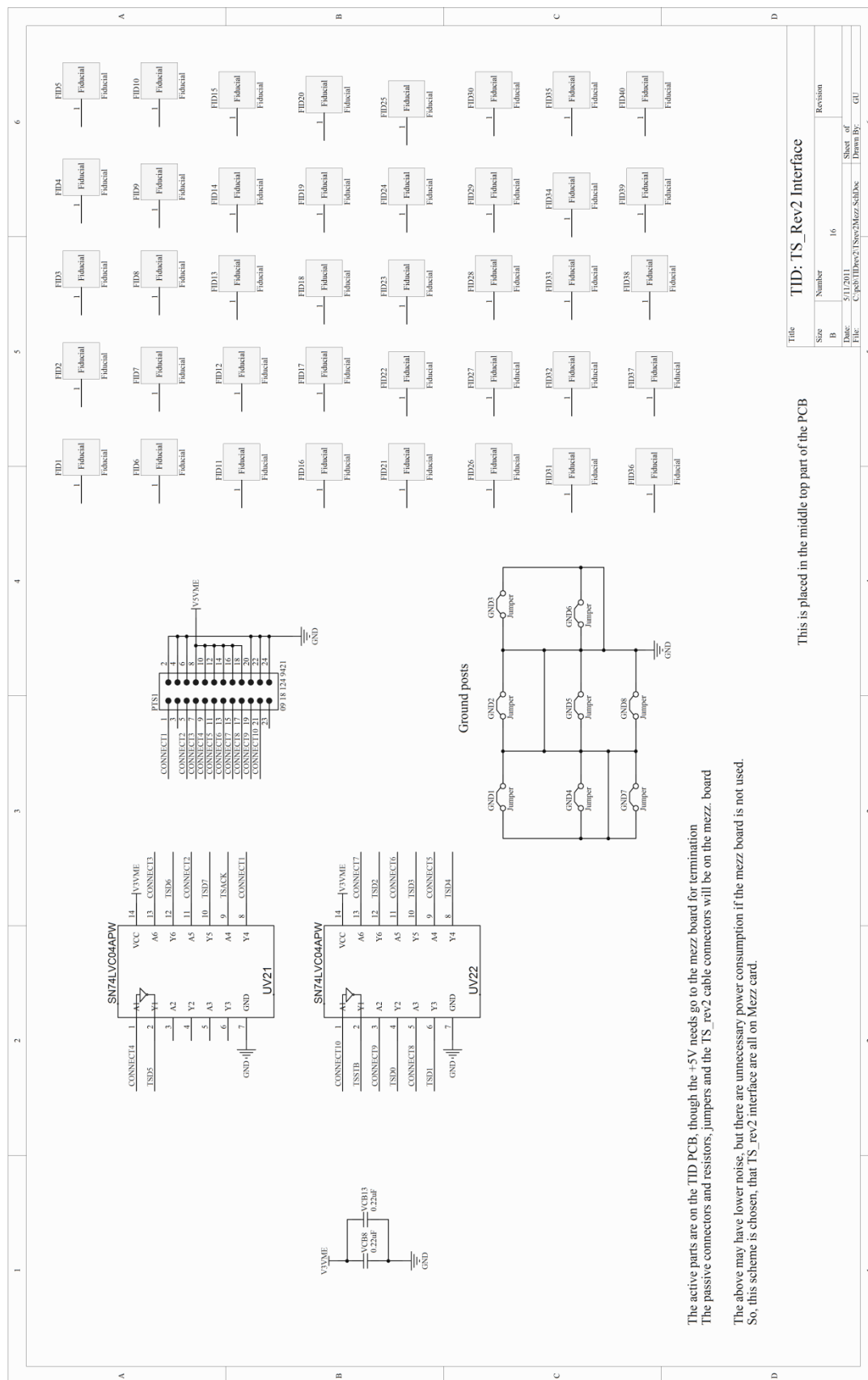


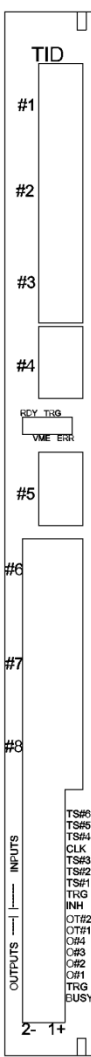
TID: Clock Distribution

Title	Size	Number	Revision
B	12		
Time	3/1/2011	Sheet of	6
File	C:\P1\Brow2\ClockDist\SubAs	Drawn By	GH







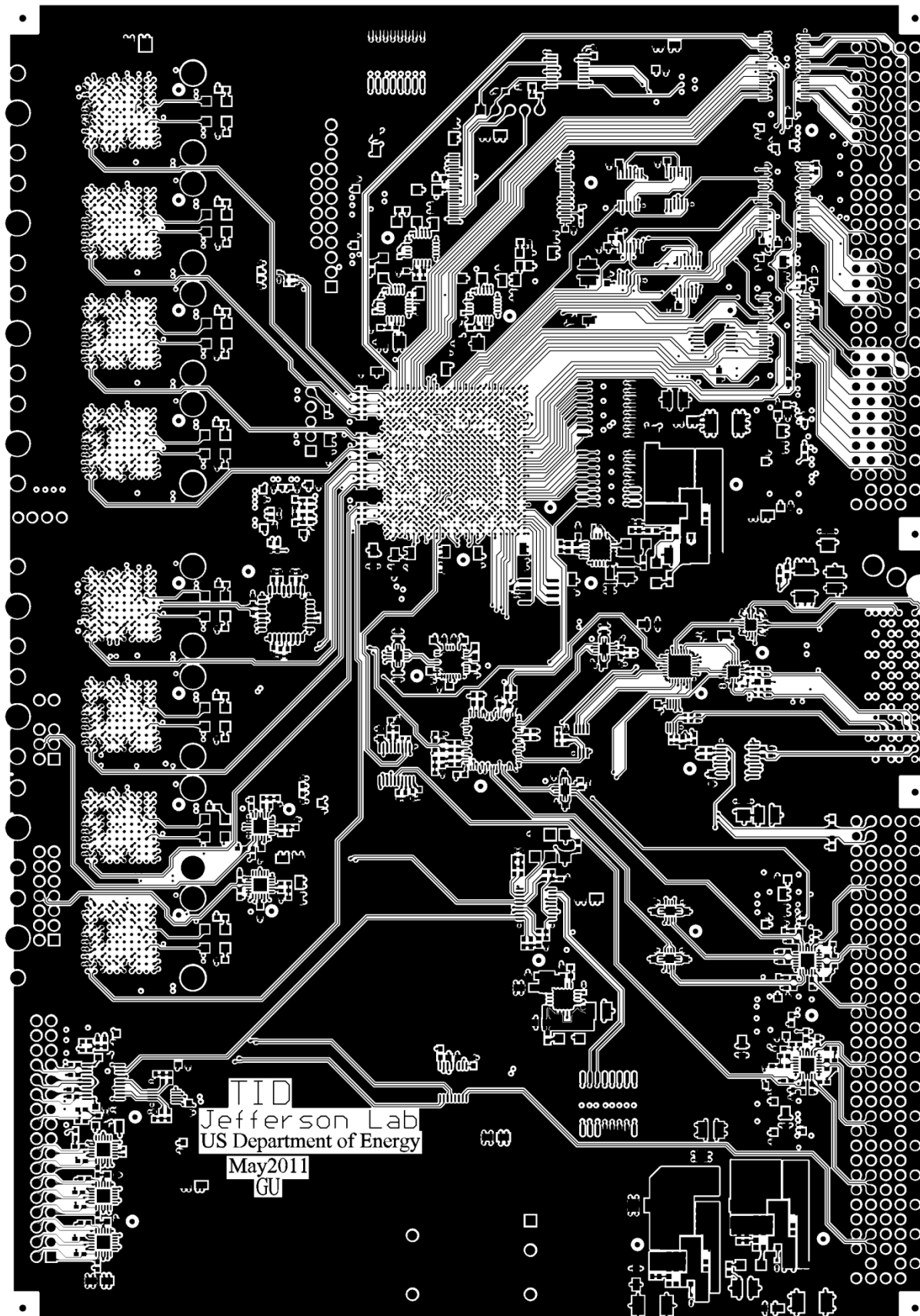
CAMtasticDXP[®] (TM):[illegible]

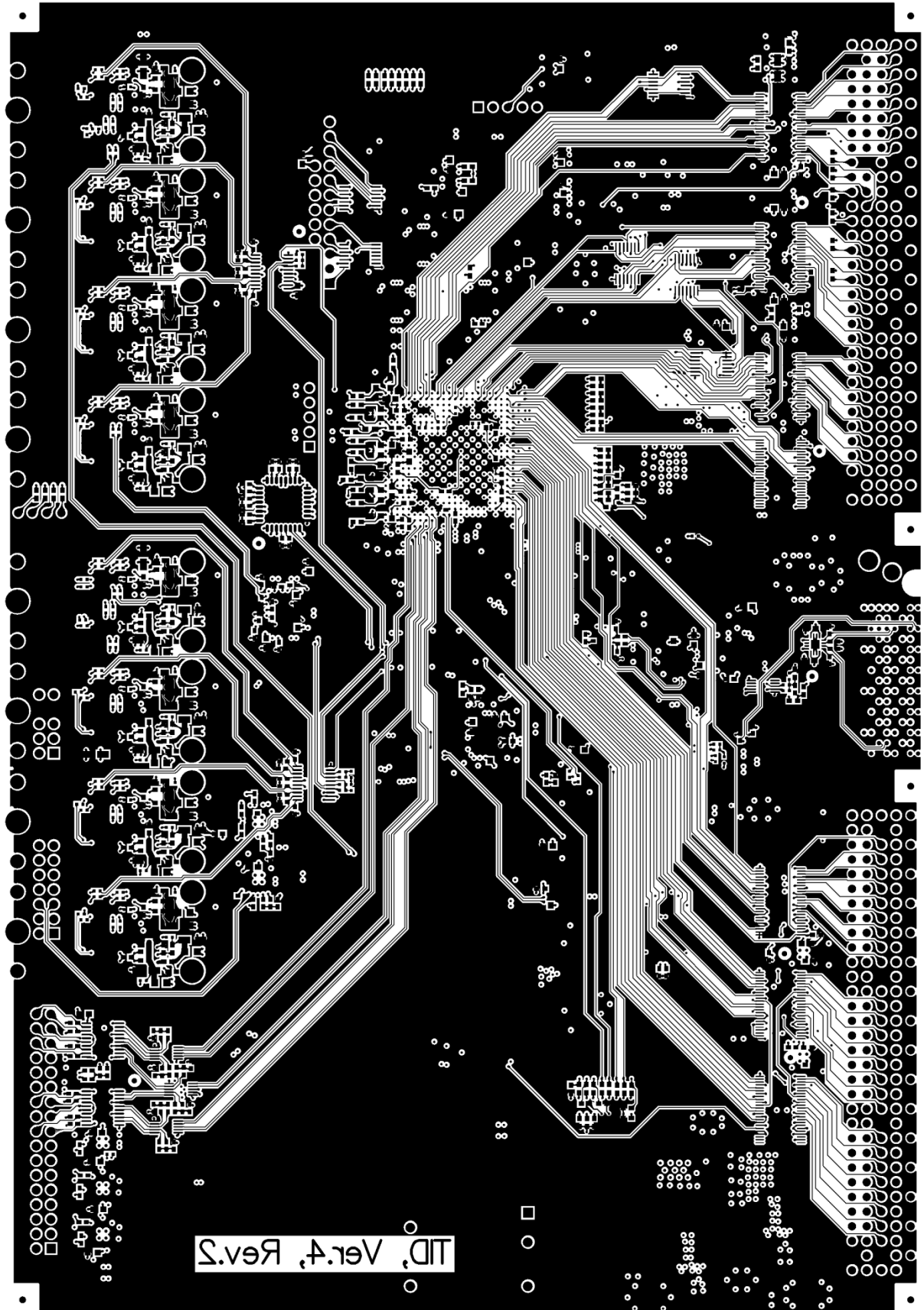
DRILL FILES

- (1) TIDev2.drv = Drill Report File
- (2) TIDev2-plated.txt = NC Drill file for PLATED holes
- (3) TIDev2-machined.txt = NC Drill file for MACHINED holes

Symbol	Hit Count	Tool Size	Plated	Hole Type
●	2903	0.3mm (11.81mm)	PTH	Round
◆	115	0.33mm (12.992mm)	PTH	Round
◇	109	0.56mm (22.047mm)	PTH	Round
▽	16	0.7mm (27.559mm)	PTH	Round
◻	12	0.9mm (35.433mm)	PTH	Round
◼	320	0.9652mm (38mm)	PTH	Round
⊗	82	1mm (39.37mm)	PTH	Round
○	8	1.016mm (40mm)	PTH	Round
⊗	6	1.27mm (50mm)	PTH	Round
⊗	16	1.7018mm (67mm)	PTH	Round
⊗	24	2.6924mm (106mm)	PTH	Round
⊗	6	2.794mm (110mm)	PTH	Round
△	1	2.794mm (110mm)	NPTH	Round
⊗	1	3.2mm (125.954mm)	NPTH	Round
	3619 Total			

Drilling Details.





Appendix C: Bill of materials:

Component list

Source Data From:

Project:

Variant:

Report

Date: 5/12/2011

Print Date: 12-May-11

Description	Footprint	Quantity	Designator
Capacitor	0402	11	C0001, C0002, C0003, C0004, C0007, C0008, C0013, C0014, C0017, C0018, CUC04A
Capacitor (Semiconductor SIM Model)	1608[0603]	28	C0005, C0006, CCL01, CCL02, CCL03, CCL04, CCL05, CCL06, CCL07, CCL08, CCL09, CCL11, CCL12, CCL13, CCL14, CCL15, CCL16, CCL17, CCL18, CP01, CP02, CP03, CP04, CP05, CP06, CP07, CUP07A, CUP07B
Capacitor, Capacitor (Semiconductor SIM Model)	0402	121	C9510A, C9510B, C9510C, C9510D, CCAU1, CCAU2, CCAU3, CCAU4, CCAU5, CCAU6, CCAU7, CCAU8, CCF01, CCF02, CCF03, CCF04, CCF11, CCF12, CCF13, CCF14, CCF15, CCF16, CCF17, CCF18, CCF21, CCF22, CCF23, CCF24, CCF25, CCF26, CCF27, CCF28, CCF31, CCF32, CCF33, CCF34, CCF35, CCF36, CCF37, CCF38, CCF41, CCF42, CCF43, CCF44, CCF45, CCF46, CCF47, CCF48, CCF51, CCF52, CCF53, CCF54, CCF55, CCF56, CCF57, CCF58, CCF61, CCF62, CCF63, CCF64, CCF65, CCF66, CCF67, CCF68, CCF71, CCF72, CCF73, CCF74, CCF75, CCF76, CCF77, CCF78, CCF81, CCF82, CCF83, CCF84, CCF85, CCF86, CCF87, CCF88, CCF101, CCF102, CCF103, CCF104, CCF119, CCF129, CCF139, CCF149, CCF159, CCF169, CCF179, CCF189, CCLK04, CCLK05, CCT11, CCT12, CFB21, CFB22, CFB31, CFB32, CFB41, CFB42, CFB61, CFB62, CFB71, CFB72, CFB81, CFB82, CGTP1, CGTP2, CU201, CU202, CU203, CU204, CU205, CU206, CU207, CU208, CUC04, CUC08A, CUC08B

Capacitor, Ceramic Chip Capacitor - Standard	0603	143	CCA01, CCA02, CCA03, CCA04, CCA05, CCA06, CCA07, CCA08, CCA09, CCA10, CCA11, CCA12, CCA13, CCA23, CCA24, CCA25, CCA26, CCE01, CCE02, CCE03, CCE04, CCE05, CCE06, CCE08, CCE09, CCE10, CCF111, CCF112, CCF114, CCF115, CCF117, CCF118, CCF121, CCF122, CCF124, CCF125, CCF127, CCF128, CCF131, CCF132, CCF134, CCF135, CCF137, CCF138, CCF141, CCF142, CCF144, CCF145, CCF147, CCF148, CCF151, CCF152, CCF154, CCF155, CCF157, CCF158, CCF161, CCF162, CCF164, CCF165, CCF167, CCF168, CCF171, CCF172, CCF174, CCF175, CCF177, CCF178, CCF181, CCF182, CCF184, CCF185, CCF187, CCF188, CCPG1, CCPG2, CCPG3, CCPG4, CCPG5, CCPG6, CCPG7, CCPG8, CCPG92, CCSE01, CCSE02, CCSE03, CCSE04, CCSE05, CCSE06, CCSE07, CCSE08, CCSE08x, CCSE09, CCSE11, CCSE12, CCSE13, CCV01, CCV02, CCV03A, CCV04, CCV05, CCV06, CCV07, CCV08, CCV09, CCV10, CCV11, CCV12, CCV13, CCV14, CCV15, CCV21, CCV22, CCV23, CCV33, CCV36, CCV37, CCV38, CCV39, CCV40, CVUC04A, CVUC04B, VCB1, VCB2, VCB3, VCB4, VCB5, VCB6, VCB7, VCB8, VCB9, VCB10, VCB11, VCB12, VCB13, VCB14, VCB15, VCB16, VCB17, VCB18, VCB19, VCB29, VCB39
Capacitor, Capacitor (Semiconductor SIM Model)	0402	18	CCCF1, CCCF2, CCCFix, CCCFS, CCGT, CCP1, CCP1_NL, CCP2, CCP3, CCP3_6, CCP4, CCP6, CCPGTP4, CCUIO1, CCUIO2, CCUIO3, CUF12BB, CUF13BB
Solid Tantalum Chip Capacitor, Standard T491 Series - Industrial Grade	A	71	CP1, CP6, CP9, CP9A, CP9B, CP9C, CP9D, CP9E, CP10, CP11, CP95x, CPE1, CPE2, CPF11, CPF12, CPF13, CPF14, CPF22, CPF23, CPF24, CPF32, CPF33, CPF34, CPF42, CPF43, CPF44, CPF51, CPF52, CPF53, CPF54, CPF62, CPF63, CPF64, CPF71, CPF72, CPF73, CPF74, CPF82, CPF83, CPF84, CPL02, CPL03, CPL04, CPL05, CPL06, CPL07, CPL08, CPL09, CPL10, CPL11, CPL12, CPL13, CPL14, CPL15, CPV1, CPV2, CPV3, CPV4, CPV35, CPV36, CTA1, CTA2, CTA3, CTA4, CTPG1, CTPG2, CTSC1, CTSC2, CTSC3, VCA1, VCA2
Solid Tantalum Chip Capacitor, Standard T491 Series - Industrial Grade, Solid Tantalum Chip Capacitor, Standard T520 Series - Industrial Grade	B	9	CP2, CP3, CP4, CP5, CP7, CP7X, CP20, CPL01, CPL20
Solid Tantalum Chip Capacitor, Standard T491 Series - Industrial Grade	B	1	CP5_NL
Typical RED, GREEN, YELLOW, AMBER GaAs LED	3.2X1.6X1 .1	24	DF11, DF12, DF21, DF22, DF31, DF32, DF41, DF42, DF51, DF52, DF61, DF62, DF71, DF72, DF81, DF82, DP1, DP2, DP3, DP3_4, DP4, DP5, DPG1, DPG5
	EC5BE17	1	EC34
FUSE 5A SLO BLO NANO 2 SMD	NANO_F USE	3	F5, F12, F15
	fiducial	40	FID1, FID2, FID3, FID4, FID5, FID6, FID7, FID8, FID9, FID10, FID11, FID12, FID13, FID14, FID15, FID16, FID17, FID18, FID19, FID20, FID21, FID22, FID23, FID24, FID25, FID26, FID27, FID28, FID29, FID30, FID31, FID32, FID33, FID34, FID35, FID36, FID37, FID38, FID39, FID40

FUSE 5A SLO BLO NANO 2 SMD	NANO_F USE	1	FP1
Header, 6-Pin	HDR1X6	2	FPGA_JTAG, PROM_JTAG
Jumper Wire	RAD-0.2	8	GND1, GND2, GND3, GND4, GND5, GND6, GND7, GND8
Flat Cable Connector (IDC), Low-Profile Male Header, Angled Solder Pin, 34 Contacts, Performance Level 2	918534x3 23	1	JC1
INDUCTOR 1.0UH 300MA 20% 0805	0805	24	LF11, LF13, LF15, LF21, LF23, LF25, LF31, LF33, LF35, LF41, LF43, LF45, LF51, LF53, LF55, LF61, LF63, LF65, LF71, LF73, LF75, LF81, LF83, LF85
INDUCTOR 6.8NH 10% 0603 SMD	0603	24	LF12, LF14, LF16, LF22, LF24, LF26, LF32, LF34, LF36, LF42, LF44, LF46, LF52, LF54, LF56, LF62, LF64, LF66, LF72, LF74, LF76, LF82, LF84, LF86
Inductor	L0603	17	LL01, LL02, LL03, LL04, LL05, LL06, LL07, LL08, LL09, LL11, LL12, LL13, LL14, LL15, LL16, LL17, LL18
Inductor	1210	9	LP1, LP2, LP3, LP4, LP5, LP10, LP11, LP20, LP33
VME160-P1	VME160	1	P1
VME160-P2	VME160	1	P2
Precision Thick Film Chip Resistor, 10 Ohm to 1M Ohm Range, 1% and 0.5% Tolerance, 0402 Size, 0.063 W, Thick Film Chip Resistor, 1 Ohm to 2.2M Ohm Range, 5% Tolerance, 0402 Size, 0.063 W	2-0402	45	P2R1, P2R2, P2R3, P2R4, P2R5, P2R6, P2R7, P2R8, P2R9, P2R10, P2R11, P2R12, RAT10, RAT11, RAT12, RAT13, RAT14, RAT15, RAT18, RAT19, RAT20, RAT21, RCT41X, RIO11, RIO12, RIO13, RIO14, RIO15, RIO16, RIO17, RIO18, RIO21, RIO22, RIO23, RIO24, RIO25, RIO26, RIO27, RIO28, RP12, RP12x, RP16, RPG12, RPG12x, RST99X
Precision Thick Film Chip Resistor, 10 Ohm to 1M Ohm Range, 1% and 0.5% Tolerance, 0402 Size, 0.063 W, Thick Film Chip Resistor, 1 Ohm to 2.2M Ohm Range, 5% Tolerance, 0402 Size, 0.063 W	2-0402	112	P2R21, P2R22, P2R23, P2R24, P2R25, P2R26, P2R27, P2R28, P2R29, P2R30, P2R31, P2R32, P2R33, P2R34, P2R35, P2R36, P2R37, P2R38, RAT01, RAT02, RAT03, RAT16, RAT16A, RAT16B, RAT24, RAT24A, RAT24B, RC20, RC30, RC30a, RC31, RC32, RC33, RC43, RC44, RC46, RC47, RC48, RC48a, RC49, RC101, RCT17, RCT18, RCT19, RCT30, RCT31, RCT32, RCT43, RCT44, RCT45, RCT48, RCT49, RIO1, RIO2, RIO3, RIO4, RIO5, RIO6, RIO7, RIO8, RIO9, RIO31, RIO32, RIO33, RIO34, RIO35, RIO36, RIO37, RIO38, RIO39, RIO41, RIO42, RIO43, RIO44, RIO45, RIO46, RIO47, RIO48, RIO51, RIO52, RIO53, RIO54, RIO55, RIO56, RIO57, RIO58, RIO61, RIO62, RIO63, RIO64, RIO65, RIO66, RIO67, RIO68, RST31, RST31A, RST32, RST155, RST156, RST157, RST158, RST159, RST160, RT1, RT2, RT3, RT4, RT5, RUC04, RVP0, RX02, RX02F
Flat Cable Connector (IDC), Low-Profile Male Header, Angled Solder Pin, 10 Contacts, Performance Level 2	918510x3 23	1	PFADC
Flat Cable Connector (IDC), Low-Profile Male Header, Angled Solder Pin, 14 Contacts, Performance Level 2	918514x3 23	1	PFTDC

P0-PL-VXS	P0-105_P0-PL-VXS	1	PP0
Flat Cable Connector (IDC), PCB Transition Connector, 2 Rows, Kinked Solder Pin, 24 Contacts	91812494 21	1	PTS1
Precision Thick Film Chip Resistor, 10 Ohm to 1M Ohm Range, 1% and 0.5% Tolerance, 0402 Size, 0.063 W, Thick Film Chip Resistor, 1 Ohm to 2.2M Ohm Range, 5% Tolerance, 0402 Size, 0.063 W	2-0402	80	RAT22, RAT23, RC11, RC12, RC13, RC14, RC15, RC16, RC17, RC18, RC19, RC21, RC22, RC23, RC24, RC25, RC26, RC27, RC28, RC29, RCT46, RCT47, RCT51, RCT51A, RCT51B, RCT52, RCT52A, RCT52B, RF90, RF91, RF92, RF93, RF94, RF95, RF96, RF97, RF98, RF99, RF101, RF102, RF103, RF104, RF105, RF106, RF107, RF108, RF109, RF110, RP6y, RPG8, RST33, RST34, RST35, RST36, RST37, RST38, RST39, RST40, RST41, RST42, RST43, RST44, RST45, RST46, RST47, RST48, RST49, RST50, RST101, RST102, RST103, RST104, RST151, RST152, RST153, RST154, RUP06A, RUP06B, RUP06C, RUP06D
Precision Thick Film Chip Resistor, 10 Ohm to 1M Ohm Range, 1% and 0.5% Tolerance, 0402 Size, 0.063 W, RES 100 OHM 1/16W 1% 0402 SMD, Resistor	0603	85	RC01, RC06, RC07, RC08, RCT42X, RCTR2A, RCTR3A, RCTR3B, RCX01, RCX02, RCX03, RCX04, RCX09, RCX10, RCX11, RE01X, RE06X, RF11, RF12, RF14, RF16, RF18, RF21, RF22, RF24, RF26, RF28, RF31, RF32, RF34, RF36, RF38, RF41, RF42, RF44, RF46, RF48, RF51, RF52, RF54, RF56, RF58, RF61, RF62, RF64, RF66, RF68, RF71, RF72, RF74, RF76, RF78, RF81, RF82, RF84, RF86, RF88, RI2C1, RI2C2, RI2C3, RI2C4, RI2C5, RI2C6, RP3_14, RP4, RP6, RP6x, RP9y, RP11, RP13, RP14, RP15, RPG11, RPG13, RS16, RS17, RS18, RS21A, RS22A, RS23A, RS24A, RST99D, RV02xy, RV04X, RV12xy
Resistor	0603	88	RC02, RC03, RC04, RC05, RCT50A, RCTR1, RCTR2, RCTR3, RCX05, RCX06, RCX07, RCX08, RCX12, RCX13, RCX14, RE01, RP2, RP3, RP3_15, RP3_16, RP3NL, RP3NLA, RP3x, RP5, RP9, RP9z, RP10, RP101, RP102, RP103, RP104, RP105, RP106, RP107, RP108, RP109, RP110, RP111, RP112, RPG2, RPG3, RPG4, RPG5, RPG6, RPG7, RPG9, RPG10, RPG11A, RPGCLK3, RS1, RS2, RS3, RS4, RS5, RS6, RS7, RS8, RS11, RS12, RS13, RS14, RS15, RS21, RS22, RS23, RS24, RS25, RS26, RS27, RS28, RTS19, RU40, RU40G, RUSFM, RV01, RV01x, RV02, RV02x, RV03, RV04, RV05, RV05x, RV07, RV07A, RV08, RV12x, RV12yz, RV44
Precision Thick Film Chip Resistor, 10 Ohm to 1M Ohm Range, 1% and 0.5% Tolerance, 0402 Size, 0.063 W	0402	8	RCT09, RCT10, RCT11, RCT12, RCT13, RCT14, RCT15, RCT16

Precision Thick Film Chip Resistor, 10 Ohm to 1M Ohm Range, 1% and 0.5% Tolerance, 0402 Size, 0.063 W, RES 100 OHM 1/16W 1% 0402 SMD, Thick Film Chip Resistor, 1 Ohm to 2.2M Ohm Range, 5% Tolerance, 0402 Size, 0.063 W	0402	16	RCT41, RCT42, RGT1, RGT2, RST99B, RST99C, RTD0x, RTD01, RTD02, RTD03, RTD04, RTD05, RTD05X, RTD06, RTD07, RTD08
Resistor	0603	1	RE05X
RES 100 OHM 1/16W 1% 0402 SMD, Thick Film Chip Resistor, 1 Ohm to 2.2M Ohm Range, 5% Tolerance, 0402 Size, 0.063 W	0402	32	RF13, RF15, RF17, RF23, RF25, RF27, RF33, RF35, RF37, RF43, RF45, RF47, RF53, RF55, RF57, RF63, RF65, RF67, RF73, RF75, RF77, RF83, RF85, RF87, RIO19, RIO29, RST16, RST99, RST99A, RT4A, RT4B, RT4C
Thick Film Chip Resistor, 1 Ohm to 2.2M Ohm Range, 5% Tolerance, 0402 Size, 0.063 W	0402	6	RID01, RID02, RID03, RID04, RPGCLK_NL, RPDG01_NL
Thick Film Chip Resistor, 1 Ohm to 2.2M Ohm Range, 5% Tolerance, 0402 Size, 0.063 W	0402	8	RID11, RID12, RID13, RID14, RID15, RID16, RPGCLK2, RPDG02
Thick Film Chip Resistor, 1 Ohm to 2.2M Ohm Range, 5% Tolerance, 0402 Size, 0.063 W	2-0402	4	RIO33x, RIO33y, RIO33z, RUC04A
DIP Switch, 8 Position, SPST	SW16_L	4	S1, S2, SC01, SC1
AD9510	CP-64-1_N	1	U13
IC 11-BIT I-WS BUS TXRX 48-TSSOP	TSSOP50 P810-48AL	1	U17
Xilinx Virtex-5 LXT Platform FPGA, 665-Ball FFBGA, Speed Grade 1, 360 User I/Os, Commercial Grade	FF665	1	U20
On Semiconductor, Any level positive to ECL translator	QFN50P4 00X400-24W4M	7	U27, U31, UIO4, UIO5, UIO6, UVP21, UVP22
4 Channel ECL/PECL/LVDS - > LVTTTL	TSSOP-16	1	U40
CCPD-034	SO6-5X7	1	UC01
Any level in, LVPECL out 1:2 fanout	QFN50P3 00X300-16V7N	6	UC04, UC04A, UC05, UC06, UF11, UP06
4:1 Differential Multiplexer	948E-02_L	1	UC07
ON-Semi Dual 2 X 2 CrossSwitch	RHB32_N	1	UC08

2x2 crosspoint switch with LVPECL output	QFN50P3 00X300-16V4N	2	UC10A, UC10B
Micrel dual Anylevel to LVPECL translator	TSSOP50 P490-10AN	1	UC14
Low-Voltage 1:10 Differential LVECL/LVPECL/LVEPECL/H STL Clock Driver	873A-02_L	2	UC15, US8
Differential to CML buffer	846A-02	1	UC17
	TSSOP14	4	UE01, UE02, UE07, UE08
	TSSOP14	2	UE03, UE05
	TSSOP14	2	UE04, UE06
	TSSOP14	3	UE09, UV21, UV22
	TSSOP20	1	UE10
HFBR7934	84512-102LF	8	UF1, UF2, UF3, UF4, UF5, UF6, UF7, UF8
Low-Voltage 1:5 Differential LVECL/LVPECL/LVEPECL/H STL Clock Driver	948E-02_N	4	UF12, UF13, US4, UT1
Triple ECL to PECL Translator	948E-02_N	3	UIO1, UIO2, UIO3
Dual LVPECL to LVDS translator	TSSOP50 P490-10AN	5	UIO7, UIO8, UIO9, UIO10, US1
LUMEX, Quad pack LEDs	LED14	1	ULED
LTM4604EV	LGA-66_LTM4 604EV(Pri mary)	3	UP1, UP1_NL, UP3
IC LDO REG 3.0A W/SS 20-VQFN	QFN-20	5	UP2, UP4, UP5, UP33, UPG5
Analog 1.25Gbps Clock and Data Recovery IC	QFN50P5 00X500-32W2N	1	UP07
Miniature Oscillator	CFPX-5	1	UPG1
XCF00P Series, Platform Flash In-System Programmable Configuration 1.8V PROM, 48-Pin TSSOP, 32-Megabit, Commercial Grade	VO48	1	UPG2
Atmel flash memory	8S1_N	1	USFM
IC UNIV BUS TXRX TRI-ST 48-TSSOP	TSSOP50 P810-48AL	9	UV1, UV2, UV3, UV4, UV5, UV6, UV10, UV12, UV14

Appendix D: Glossory:

TID: Trigger Interface and Distribution module; a PCB design can be configured as TI, TD, TS or TM;

TI: Trigger Interface module; It seats in payload slot#18 in front end crates, interfaces the trigger and the DAQ system; It is one stuffing variation of the TID.

TD: Trigger Distribution module; It seats in payload slot#1-16 in the global trigger distribution and fans out the TRIGGER/CLOCK/SYNC to eight TIs; it is one stuffing variation of the TID.

TS: Trigger Supervisor; It seats in payload#18 in the global trigger distribution crate; It is the interface between DAQ and trigger system; A simplified (pre-prototype) TS can be stuffed from a TID;

TM: TID Master. It is used in the subsystem test or commissioning setup; It generates TRG/CLK/SYNC as a TS, sends TRG/CLK/SYNC to P0 and P2 like a TI, and fans out TRG/CLK/SYNC through fiber to other TI like a TD.

SD: Signal Distribution module; It fans out TRG/CLK/SYNC from payload slot#18 to payload slots#1-16; It has clock jitter cleanup capability.

GTP: Global Trigger Processor module.

CTP: Crate Trigger Processor module.

DAQ: Data Acquisition.

ROC: Readout Controller; A VME CPU module used to readout the front end data through VME bus.

VXS: VME Switched Serial; A VME extension with dual-star serial switch slots.

MGT: Multiple Gbps Transceiver; A builtin transceiver module in Xilinx FPGA. In XC5VLX30T FPGA, it supports up to 3.125 Gbps.

Appendix E: TID data format:

The TID data is formatted in blocks of events. Each Trigger_1 is one event. A block of data contains a predefined number (the number 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

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): 0000, ID for TI 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): SyncEventSet
Bit(20:0): Word count; does not include block header or trailer.

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,
0x01-0x40: Physics trigger; if the trigger is from TS, 0x01-0x20 indicates one bit from the GTP, 0x21-0x40 indicates one bit from TS front panel.
0xfc: multiple trigger inputs to TS;
0xfd: VME trigger;
0xfe: 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): trigger number bit(47:32), to form 48 bit counter with word2;
 Bit(31:28): Which Fastbus duplication has data (firmware tip##)
 Bit(27:20): number of local triggers since last readout trigger; (firmware tip##)
 Bit(19:16): trigger number bit(35:32) (firmware tip##)
 Bit(15:0): trigger timing bit(47:32), to form 48 bit counter with word3;

Event data word5: (enabled by data format register 0x18, bit#3)
 Bit(5:0): The TI front panel TSinput(6:1), before prescale;
 Bit(31:16): fixed as 0xDA56

Appendix F: Document revision history:

Updated on Apr. 20th, 2010

Updated on May 13th, 2010

Updated on May 19th, 2010: add the trigger acknowledge in the status word;

Updated on July 6th, 2010: Define the TI/TD mode in section 5.2.2;

Updated on Sept. 20th, 2010: Re-define the A24 address space;

Updated on Oct. 15th, 2010: Added Busy_Input_Enable (A24, 0x0004);

Updated on Oct. 18th, 2010: updated the Emergency loading after tests;

Updated on Jan. 13th, 2011: Add examples for VME trigger word loading;

Updated on Feb. 2nd, 2011: clarify the A24 0x00-0x1C registers;

Updated on Mar. 21st, 2011: further updates on registers to match with the firmware;

Updated on Mar. 25th, 2011: Added the software setup procedure section, with some updates on the registers;

Updated on Apr. 5th, 2011: Register offset 0x14 and 0x2C update.

Updated on Apr. 18th, 2011: Enables for each GTP input bits, CLK250 source selection

Updated on Apr. 20th, 2011: added the Sync decoding, and two more A24 read registers;

Updated on May 12th, 2011: Added the PCB schematics, fab layers, and BOM as appendix A, appendix B and appendix C.

Updated on June 6th, 2011: Added A24 offsets 0x38 and 0x3C for extra readout (v10)

Updated on June 22nd, 2011: Redefined A24, 0x08 and 0x30 registers;

Updated on Aug. 18th, 2011: Overall revision to add the TM description, Glossary and data format;

Updated on May 31st, 2012: expanded the trigger rule bits from 5 to 7 bits;

Updated on Oct. 10th, 2012: Put in the new register map; Specifically for TI/TImaster;

Updated on Feb. 14th, 2013: Add the A24 offset for switch slot mirror registers;

Updated on Mar. 12th, 2013: Add the switch usage of SC01

Updated on Apr. 3rd, 2013: updated offset register 0x24, 0x78 and 0x98

Updated on Apr. 23rd, 2013: updated offset registers: 0xD4, 0x100; added section 3.13 SyncEvent generation.

Updated on July 26th, 2013: clarify the trigger timing words (register 0x18);

Updated on Aug. 15th, 2013: updated A24 register 0x78, 0x100, 0xD8, 0xDC;

Updated on Sept. 13, 2013: updated A24 registers: 0x34, 0x04, 0x94 and 0xB0;

Updated on Sept. 27, 2013: updated the Appendix E, data format;

Updated on Oct. 17, 2013: updated register 0x14, 0x78 and 0x84;

Updated on Dec. 3, 2013: Added register offsets 0x1D0 – 0x1F0 for slave TI IDs;

Updated on Feb. 28, 2014: fixed the register 0xA4 to reflect the firmware, which is the fiber#5 measurement;

Updated on May 20, 2014: added register 0xEC, modified register 0x98.

Updated on Sept. 29, 2014: added the trigger lookup table description (section 5.3.1).

Updated on Oct. 16, 2014: updated A24 register offsets 0x20 and 0x3C for trigger2 generated trigger1; 0xD4;

Updated on Jan. 6, 2015: use A24 offset 0x28 bit 6 to enable trigger acknowledge BUSY. Bit#22 is used to monitor the ‘trigger loss’ (trigger is sent out, but the trigger acknowledge has not been received)

Updated on Apr. 14, 2015: Remove the **RED register offsets** as no firmware is used that old. Added the 0x104, 0x108 registers for front panel input delays; FP input register redefine (TIproduction firmware).

Updated on Nov. 18, 2015: Added register 0x138 and modified register 0x0C.

Updated on Feb. 22, 2016: Updated the busy timers (0x110-0x128 and 0x19C- 0x1BC). I2C for the AFBR-79EIDZ.

Updated on May 31, 2016: Updated the register 0x1C (bit31), 0x100 (bit#21 and bit#22), and 0x138 (Minimum busy_width for trigger rules).

Updated on June 14, 2016: updated the 0xEC register for SyncResetRequest enable and 0x9C bit(24:16).

Updated on Aug. 4, 2016: Updated the 0x3C register, so bit#31 controls the trigger input LEVEL readout.

Updated on Jan. 4, 2017: Added the P0 Clock selection diagram on page#21 (~).

Updated on Apr. 12, 2017: Updated registers 0x18, 0x1c, 0x170(r), 0x174(r), 0x1d0-0x1f0

Updated on May 5, 2017: added the table for acknowledge/busy word in page 13 and register 0x170 readout.

Updated on July 28, 2017: Updated the 0x20 (trigger source) register write. Added the RunGo, force update.

Updated on Sept. 29, 2017: Added VME register 0x1C, bit#27 and #28 for trigger input counters.

Updated on Nov. 3, 2017: Added section 3.14 about the clock phase alignment.

Updated on Jan. 19, 2018: Added section 3.4a about trigger generation as TImaster; updated the scalar readout (0x180-0x194, it was implemented that way, the document was not updated).

Updated on Mar. 27, 2018: Added section 7.4, 7.5, and 7.6 for the front panel connector pin IO.

Updated on July 19, 2019: Register 0x34 register description, and added bit#27 for trigger dump.

Updated on Nov. 4, 2019: Added the Register 0x54 and 0x58 description, They were in the firmware, but not the manual.

Updated on Mar. 26, 2021: Updated the register 0x2C, switch SC01 setting, Data block trailer bit#21

Updated on Sept. 27, 2021: Updated register 0x58, 0xE0-0xE8

Updated on Oct. 13, 2022: Updated register 0x2C and 0x30, and the data format (Appendix E)

Updated on Oct. 13, 2023: Updated register 0x50, bit(15:8) for beam pass delay on the TIbridgeTS (only), 0xF8 readout the delay setting

Updated on Feb. 13, 2024: Updated register 0x68, 0x6C, and 0x70 for the ChipScope replacement available in firmware version 14.1.

Updated on Apr. 15, 2025: Added subsection 5.3.1.1, detailed trigger_table loading, and some examples.