Nuclear Physics Division
Data Acquisition Group

Description and Technical Information for the PCI_express Trigger Interface (TIpcie) Module

William Gu (jgu@jlab.org)

Updated: June 14, 2016

Table of Contents

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 VXS 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]. The PCIexpress TI (TIpcie) board is based on the VXS TI board. Instead of VME64x, the TIpcie will be plugged into a PCIexpress slot in a computer, communicates through PCIexpress. Figure 1 shows the placement of the TI modules in the global trigger distribution scheme in experiment setup.
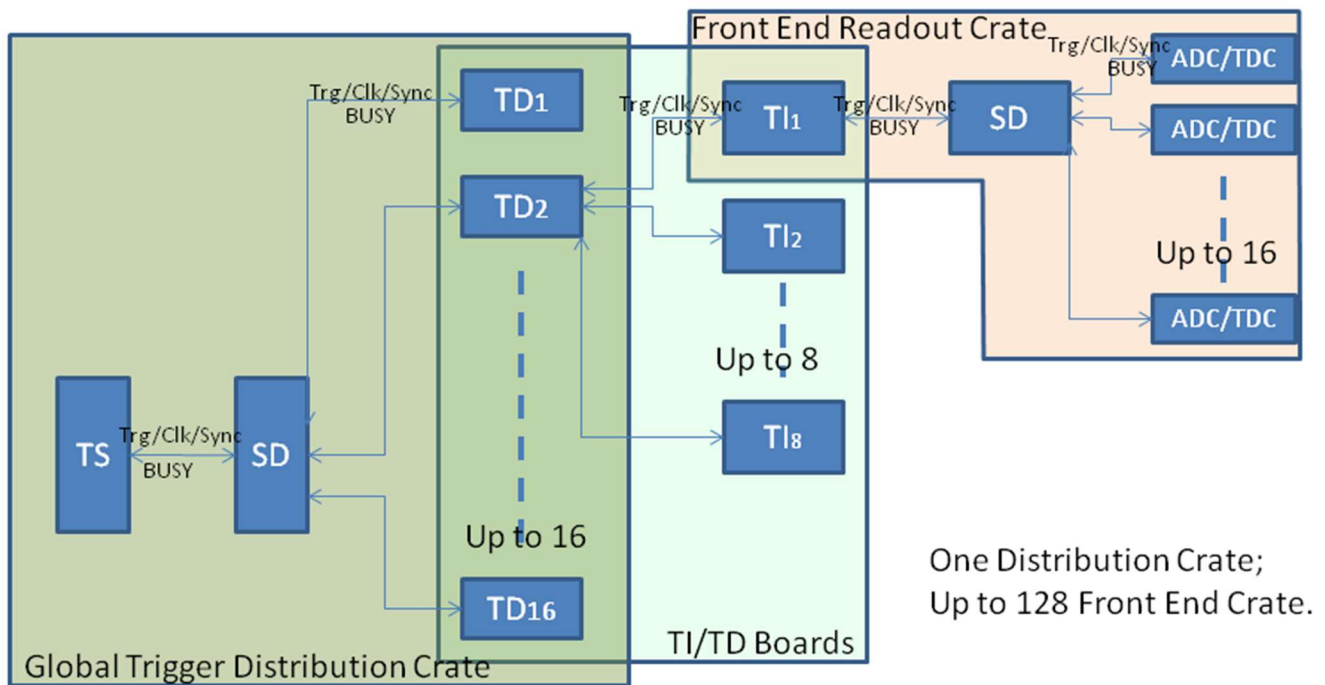

Figure 1 Trigger/Clock/Sync distribution system

The TIpcie board has simple Trigger Supervisor (TS) functions built in. The TIpcie can act as a trigger supervisor when used independently. The TIpcie board can be used in master mode and distributes Trigger/Clock/Sync to another TI board. The TIpcie also has the flexibility to select the trigger and clock inputs from central trigger system (used in slave mode) or onboard oscillator and loopback(used in master mode).

2 Purpose of the module

TIpcie board can be plugged in any PCIexpress x4 compatible slots. It can connect to a Trigger Distribution (TD) module in the global trigger distribution crate, sit in standalone mode, or drive another TI board as a master. This is done using a four channels full-duplex fiber link, which provides a gigabit trigger link, 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 pipeline clock allowing a trigger word to be distributed every 4 global pipeline clock cycles. Figure 2 is a picture of the TIpcie board with a low profile bracket.

Figure 2 TIpcie PCB picture

The TIpcie can also perform simple Trigger Supervisor functions, as a TI Master (TM). In this case, it can take inputs from its front panel and generate trigger/clock like TS, and it can send the trigger, clock and SYNC to the front panel fibers like a TD.

The TIpcie FPGA firmware is compatible with the master mode and the standard mode. The firmware senses the TI clock setting. If the TIpcie clock source is set to onboard oscillator input, the TIpcie 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 TI (or TIpcie) if the setup needs be expanded.

The TIpcie has a generic 16-channel LVDS input/output connector and a 32-channel LVCMOS/LVTTL input/output connector. One can directly connect two 34-pin cables to these connectors, or have a mezz card as a cable adaptor to connect to these two connectors. The advantage of the mezz card is that the mezz card can hold an fan to actively cool the TIpcie board, and one 3M P50E-068P1-SR1 connector can be used for better appearance (neat and convenient inside the computer). With the mezz board, the TIpcie looks like a double width card. One application of the generic 48-channel IO could be a high precision TDC.

3 Functional Descriptions

3.1 General description

Figure 2 shows the block diagram of the TIpcie module, indicating the major components used in the design. The AFBR-79EIDZ 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 computer PCIexpress bridge/root master.
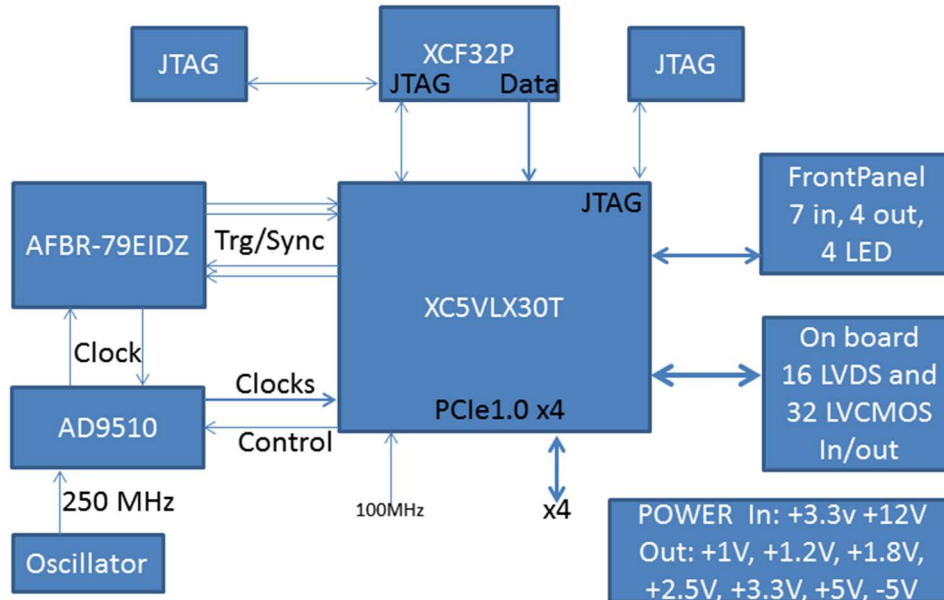
Figure 3 TIpcie block diagram

3.2 Fiber links

The AFBR-79EIDZ is the multi-channel (4 Rx, 4 Tx) fiber optic link for the TIpcie. The AFBR-79EIDZ has the same optic interface to replace the optic transceivers HFBR-7924 used on VXS TI and TD boards. When the TIpcie is in standalone mode, the optic transceiver is not used.

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

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

The fourth pair (Tx/Rx) is looped back on the TIpcie when the the TIpcie is in master mode. Otherwise, the pair is used to send the FPGA generated pulse and receive the loopback (from TD) to measure the fiber latency.

3.3: Clock Distribution

One of the TIpcie's major functions is the pipeline clock distribution. There are two possible sources for the 250MHz clock: 'onboard oscillator' and 'optical fiber input from TD/TS'. When the TIpcie is in master mode, or standalone mode, the on board oscillator will be used, or else the optic fiber input will be used. There will be only one 250MHz clock running on the TIpcie. The clock source is selected by the FPGA using AD9510 clock driver, which also generate slower clocks (31.25MHz, 62.5 MHz, and 125 MHz).

3.4 Trigger distribution:

When the TIpcie is in master mode (or standalone mode), it can takes the trigger from the front panel inputs and do a quick lookup table and form a trigger word. It can also combine with the software trigger (which

includes periodic trigger up to 7 MHz (from ~16 Hz) and random trigger up to 466 KHz (from ~15Hz) and send to the MGT serialized mode (encoded) to the fiber. The trigger word is looped back to trigger the TIpcie itself.

When the TIpcie is in standard (slave) mode, it receives the trigger signal through the AFBR optical fiber and decoded by the FPGA. The trigger is used to generate TIpcie event data which includes the event type, trigger time stamp, and event number.

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 → TIpcie, or TImaster → TIslave):

Trigger strobe word – generated by the TS (or TIpcie in master mode) in response to the acceptance of a level 1 trigger. Upon receipt, the TIpcie aligns the trigger and start to assemble trigger event. 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.

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. So far, the trigger content words has not been used.

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 TI(pcie) 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.

Bit(15:14)	Bit(13:12)	Word Type	Bit(11:10)	Bit(9:8)	Bit(7:0)
0 1	0 0	Time	Lower 12 bits of TS time		
0 1	0 1	Control	Control or Command		
0 1	1 0	Trigger Strobe	quadrant	Type	Event type
0 1	1 1	Trigger Content	Additional Trigger data		
1 0	0 1	GTP trigger	Quadrant	Type	Event type
1 0	1 0	FP trigger	Quadrant	Type	Event type
1 0	1 1	TS partition	TS4(2:0)	TS3(2:0)	TS2(2:0)	TS1(2:0)

- Bit(9:8) Type: 01: Trigger1, 10: Trigger2, 11: Sync trigger, 00: no trigger;
- TS partition trigger: TSn(2:0), 000: no trigger;

The data coming back to the TImaster or TD is defined into two categories. The first category is the status and acknowledgements. The second category is the register data. The first category has higher priority than the second category.

The following defines the data format for the opposite direction of the link (TIpcie → TD, or TI → TImaster flow):

Bit(15:14)	Bit(13:12)	Word Type	Bit(11:8)				Bit(7:0)					
Parity	0 1	Ack/Busy	Busy	Trg Ack	Sync RstReq	#Sync RstReq	Busy	Trg1 Ack	Trg2 Ack	Blk Rcvd	Blk Ack	xxx
"0 0" or "1 1"	1 0	Register	0 0 1 0				Crate ID (7:0)					
			0 1 0		Board Ready		Trigger Source (7:0)					
			0 1 1		0		NewBlockSize					

3.6: Fixed Latency SYNC

The SYNC signal is a 250Mbps serial line operating in synchronous mode. This serial link allows a 4-bit command to be sent at chosen 4ns points in time. SYNC is synchronized to a slower clock (250/24 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 TIpcie, 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 4:
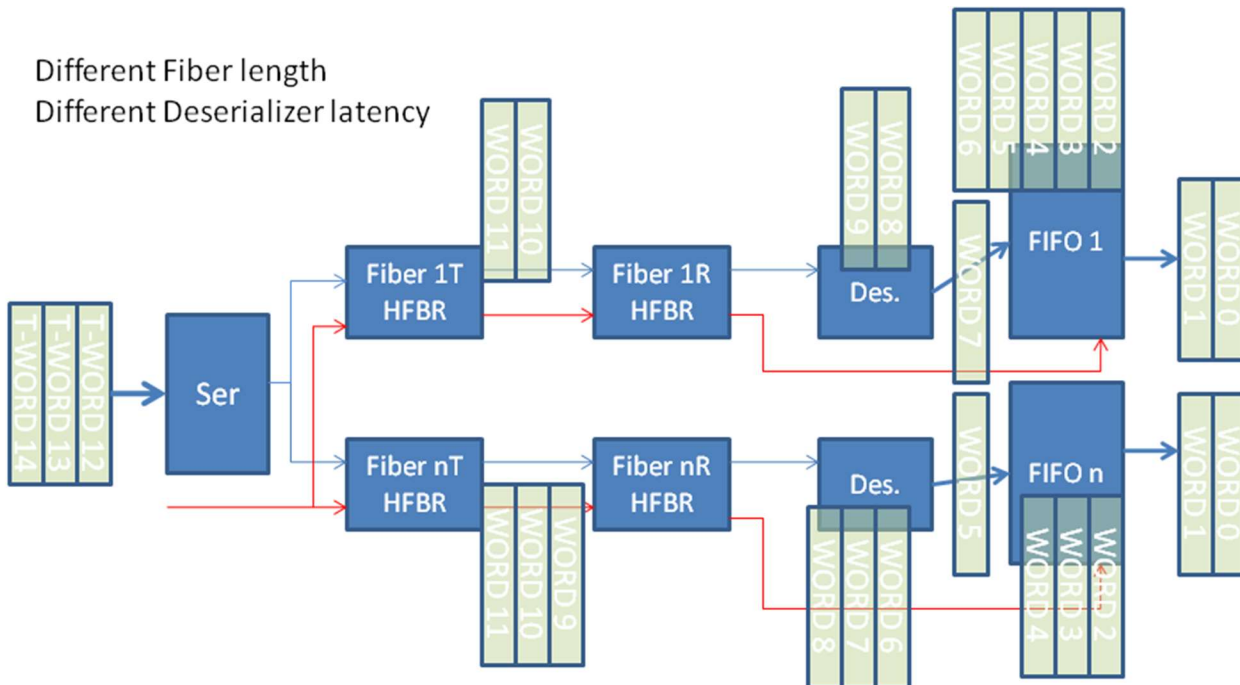
Figure 4 Illustration of the trigger synchronization process

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(pcie) 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.
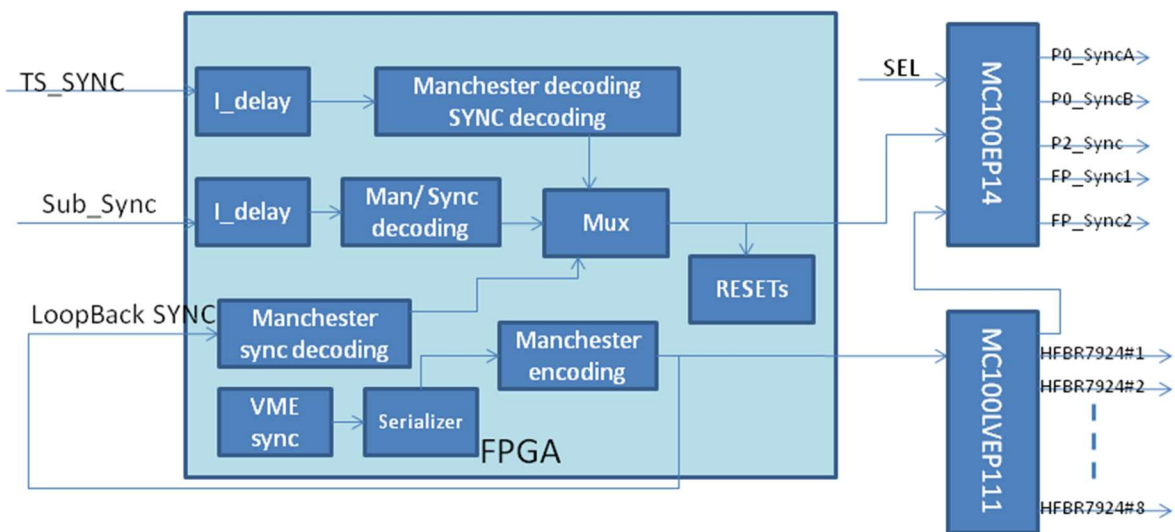
Figure 5 The block diagram of the SYNC distribution on the TI board. On TIpcie, there is no MC100EP14 buffers, nor Sub_sync input.

3.7 PCI express interface

The TIpcie board is a PCI express endpoint device. It is capable of PCIe 1.0 x4 (2.5 Gbps, 4 lanes). It is compatible with PCI express x4 or wider physical connector, though it is compatible with PCI express x1 connection.

The TIpcie has two 32-bit BA spaces. The BA#0 has 512 bytes, and it is compatible with regular TI VME setting registers. The BA#2 has 4 kBytes, which is for two JTAG engines, one for the FPGA and the other for PROM. The TIpcie trigger data can be readout word by word (32-bit) through a BA#0 memory word. The data can also be written to the computer memory via DMA.

The TIpcie support MSI for interrupt.

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. The PROM can be programmed through the on-board JTAG connector. It can also be programmed by the PCI express directly (remotely) via the FPGA translation (FPGA PCI express to JTAG engine One 50 MHz on-board oscillator is used to program the FPGA in slave mode, In slave SelectMAP (default, the PROM is in master mode) mode, the FPGA can be loaded in less than 30 ms.

3.9 Other functions

The TIpcie has a generic differential connector with four outputs and 7 inputs. This can be designed as regular TI compatible input/outputs. It also has two 34-pin connectors on board, which further expand the TIpcie IO capability.

3.10: 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 PCI express to the TIpcie if the TIpcie is in master mode or standalone mode). 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 (or PCI express BA#0 memory byte space 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(pcie) will generate BUSY, and set the SyncEvent marker in the ROC polling register offset 0x34. The BUSY of TI(pcie) will propagate back to the TD 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(pcie). 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(pcie). After all the TI(pcie) boards clear their BUSY, the TS will distribute trigger again. Depending on the data acquisition mode and the amount of data backed up on the frontend, this process may take tens of microseconds to many milliseconds.

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

4. Specification Sheet

4.1 Mechanical
- Single width low profile PCI express 1.0 4x module. It will be positioned in any PCI express x4 compatible slot.

4.2 PCI express:
- PCI express 1.0 4x compatible, that is 4 lanes with 2.5 Gbps full duplex;
- Reference Clock at 100 MHz;
- JTAG on PCI express slot is not used, as most computers do not support (supply) JTAG;

4.3 Onboard connectors:
- 32-channels of LVTTL or LVCMOS either input or output depending on the FPGA configuration.
- 16-channels LVDS input or output depending on the FPGA configuration.

4.4 Front panel inputs and outputs:
- 7-channels Any (differential) level inputs for trigger
- 3-channel ECL output
- 1-channel 2.5V LVPECL Output

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: (farther away from the motherboard) FPGA programmed and the clock (DCM locked) is ready;
- Bit 2: PCI express activity
- Bit 3: Trigger_1 is sent out / received;
- Bit 4: AFBR optic transceiver 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);

4.7 Programming:
- PCI express to JTAG engine using BA#1 for remote loading with redundant On board JTAG connector;
- Two revisions of the firmware can be stored in the PROM simultaneously, selected by an on-board switch.

4.8 Power requirements:

- +3.3V @ 3 Amps; +12V @ 0.25 Amp; (From PCI express connector)
- Local regulators for other required voltages: +1.0V, +1.2V, +1.8V, +2.5V, +3.3V, +5V and -5V.

4.9 Environment:
- Well ventilated desktop computers;
- Commercial grade components (0-75 Celsius)

5 TIpcie operation procedures:
The TIpcie needs be properly set, and plugged into the PCI express slot. Damage may happen to the TIpcie, computer, or other components in the computer if the right procedure is not followed.

5.1 TIpcie Power supply:
The TIpcie uses +3.3V and +12V from the PCI express connector. It generate its own +5V, -5V supply by a DC-DC converter from +12V. It generates +1.8V by a DC-DC converter. It generates +1.0V, +1.2V +2.5V supplies by LDO linear regulators.

5.2 Hardware setting (Switch etc.):
There is one 8-bit switches on the TIpcie marked as SC1. This switch is used to set the FPGA firmware revision, the sources of the four front panel outputs.

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.

Bit[4:3]: LVTTL, open high. Source selection for the four differential outputs (one LVPECL2.5V, three ECL). When "00", the outputs are controlled by software BR#0, offset 0x4C, bit(3:0).
Bit[7:5]: LVCMOS from FPGA, not used
Bit[8]: Connected to the RESET# on the PCI express slot.

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.

5.3.1 Computer boot process:
The TIpcie card is recognized at the computer BIOS boot. During the boot process, the PCI express enumeration will set the proper working parameters to TIpcie configuration registers. If the FPGA firmware is reloaded after the computer boot, the configuration registers will be lost. In that case, the configuration registers needs be reloaded. The easiest way is to restart the computer without power cycle the computer.

5.3.2 Driver loading:
The TIpcie card is set as manufacture "D0E1", for DOE lab), device "71E0" (TI pci Express). As superuser, the driver (pci_skel.ko) can be reloaded ~jgu/new_pti/linux_driver/reload_driver.sh.

5.3.3 TIpcie setup:
The TIpcie card needs be setup properly:
Main Clock selection: BR#0, offset 0x2C: 0x0 for onboard oscillator, 0x2 for fiber input;
SYNC source selection: BR#0, offset 0x24: 0x10 for loopback; 0x02 for fiber input

DMA setting: BR#0, offset 0x54: TLP packet format, Max packet size, DMA size, higher address; offset 0x58: 32-bit DMA physical address;

Trigger source etc: BR#0, offset 0x20, offset 0x28, 0x18 etc.

5.3.4 Trigger table loading:

There is a 6-bit inputs, 8-bits output look up table built in the TIpcie(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. The eight output bits are defined by the data (table) loaded.

The table (64 bytes) is loaded by sixteen 32-bit VME words (BR#0, offset 0x1C0 – 0x1FC)

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

There are three 32-bit Base Address are defined on the TIpcie card.

The BA#0 has 512 bytes addressable in 32-bit word only. These memory spaces are equivalent to the VME A24D32 registers on TI board.

The BA#1 has 4 kbyte addressable in 32-bit word only. These memory spaces are equivalent to the VME JTAG engine on TI board.

The BA#2 has 1 MB read only (in 32-bit word only). This is designed to read out the TIpcie event data in block read mode. Right now, the pci_skel driver seems to send the single word read only, so this function is not used, nor the function has been debugged.

The default data readout process is the DMA write. When a readout data block is ready, the TIpcie will write the data to DMA as a ring buffer in 4kbyte blocks. The kernel can set 1MB to 4MB as DMA buffer in 32-bit addressing mode (the default, though the linux is 64-bit operating system).

6.1 BA#0 memory:

BA#0 memory spaces are used for register (VME A24D32 equivalent) read/write. They are addressed in 32-bit word only (not byte addressable, to be simple).

➢ Address offset: 0x00000: Board ID:

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

Bit 13-8 (R): not defined yet;

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

Bit 19-16 (R): PCB related setting, 0x1: production board, 0x0; prototype board;

Bit 31-20 (R): Board type: 0x71: TI, 0x75: TS, 0x7D: TD, 0x71E: TIpcie.

➢ Address offset: 0x00004: Optic transceiver I2C enable:

Bit 0 (R/W): AFBR optic transceiver I2C Enable, Reset default 0x1, disabled.

Bit 0: '0' enable AFBR#1 I2C, '1' disable AFBR#1 I2C; ('disable' means 'bypass in I2C')

Bit 31-16 (R): number of 64-bit word available in FIFO ready for PCI express DMA write.

➢ 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, (n+1)*4 ns; Reset default 0x07;
Bit 15-8 (R/W): Trigger_1 Pulse width (n+1)*4 ns; Reset default 0x07;
Bit 23-16 (R/W): Trigger_2 delay, (n+1)*4 ns; Reset default 0x07;
Bit 31-24 (R/W): Trigger_2 Pulse width (n+1)*4 ns. Reset default 0x07.

➢ Address offset: 0x00010 (R): DMA address:

Bit 31-22: DMA address;
Bit 21-20: DMA address, determined by 1/2/4 Mbyte allocation;
Bit 19-12: 4 KB block address;
Bit 11-7: 128 Byte address;
Bit 6-0: always '0', the minimum addressable block is 128 bytes.

➢ 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 other front DAQ electronics (FADC250, FADC125, etc).

➢ Address offset: 0x00018: TI data format control: Reset default 0011;

Bit 0: if '1', two block placeholder words are enabled; no longer used, default to no placeholder.
Bit 3-1: 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: '1' to enable the front panel input bit pattern readout (before prescale).

➢ Address offset: 0x0001C: (Vme) PCIe setting:

Bit 21 (R/W): Enable instant block level update;
Bit 23 (R/W): '1' to enable Interrupt, '0' to disable Interrupt to CPU;
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 26 (R/W): '1' to enable data to DMA, '0' to block (forbid) the data to DMA;
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.
On TI: 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 Trigger_rule by a factor of 32 (to get long trigger rule).

➢ Address offset: 0x00020: Trigger source register:

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

Bit 1: AFBR optic transceiver 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 7: Random Trigger.

Bit 11: Enable trigger2 to generate trigger1 (valid for TIpcie in master mode)

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

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

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

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

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

➢ Address offset: 0x00024: Sync Source register:

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

Bit 1: AFBR optic transceiver input;

Bit 4: Loopback SYNC enable, when TIpcie is in master or standalone mode;

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 AFBR;

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 Data busy (that is, when the data is ready to be readout (at least one block), set the TIpcie to busy.

Bit 2: '1' enable the TI_Fifo_Full busy. (that is, if the FIFO is full, it is BUSY)

Bit 4: '1' enable the front panel (bracket) busy input for TI_BUSY to propagate to 'TS' for TS_BUSY;

Bit 5: '1' enable the front panel (bracket) busy input for TS_BUSY directly when in master mode;

Bit 6: '1' BUSY if the trigger is out, but the acknowledge (from slave TI) has not been received;

Bit 7: '1' enable TPpcie feed_back BUSY, '0' disable the busy. (used in master or standalone mode)

Bit 8: AFBR BUSY enables: '1' enable the AFBR BUSY input, '0' disable;

Bit 31-16 (R): BUSY source monitoring.

Bit 22: 'trigger loss' for enabled (slave) TI(pcie).

➢ 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] = 10: AFBR clock input;

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

    Bit 15-0 (R/W): pre-scale factor: Rate = Rate_0 / (Bit(15:0)+1).

- ➢ Address offset: 0x00034: Trigger block inhibit:

    Bit 7-0 (R/W): TIpcie 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 23-16 (R): Number of events before a block is formed.
    Bit 27-24 (R): Number of missing block acknowledge;
    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 (Bit(6:0)*(16/160 ns)); Bit7 determines 16ns or 160ns step.
        Reset default 0x03;
    Bit 15-8 (R/W): no more than 2 trigger in (Bit(14:8)*(16/320ns)); Bit15 determines 16ns or 320ns step.
        Reset default 0x03;
    Bit 23-16 (R/W): no more than 3 triggers in (Bit(22:16)*(32/640 ns)); Bit23 determines 32ns or 640ns step.
        Reset default 0x03;
    Bit 31-24 (R/W): no more than 4 triggers in (Bit(30:24)*(64/1280 ns)). Bit31 determines 64ns or 1280ns
        step. Reset default 0x03;
    The slower clocks can be even slowed down by a factor of 32 when the register 0x1c bit 31 is set to 1 !

- ➢ 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;

- ➢ Address offset: 0x00044: Front panel generic trigger input enable:

    Bit 5-0 (R/W): Front panel trigger code input enable (6 of the 7 input). Reset default 0x00.

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

    Bit 3-0 (R/W): 4-bit output to the front panel generic output connector;
    Bit 15-8 (R): Number of data blocks in the FIFO for readout;
    Bit 23-16 (R): Number of data blocks ready for Interrupt Request;
    Bit 31-24 (R): on TI: Number of events of a partial block (or, before the block is formed)

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

    Bit 7-0 (R): on TI: SYNC phase of AFBR input;
    Bit 15-8 (R/W): AFBR SYNC input delay; Reset default 0x00;
    Bit 23-16 (R/W): TIpcie (internal loopback) SYNC delay; Reset default 0x00;

➤ Address offset: 0x00054: DMA setting:

Bit 15-0 (R/W): DMA physical address bit(47:32), used in 4-header TLP mode;
Bit 20 (R/W): Enable the Dynamic DMA physical Address setting. '0': one DMA block only. (Apr. 2018)
Bit 25-24 (R/W): DMA size, '01': 1048576 byte, '10': 2097152, '11': 4194304;
Bit 30-28 (R/W): TLP Maximum Packet Size. '001': 128 byte; '010': 256 byte; '100': 512 byte;
Bit 31 (R/W): DMA TLP address mode: '1': 64-bit/4 header mode; '0': 32-bit/3 header mode

➤ Address offset: 0x00058: DMA address setting:

Bit 31-0 (R/W): DMA lower 32-bit physical address.

➤ Address offset: 0x0005C: PCI express Configuration Link:

Bit 15-0 (R): PCI express Configuration Link Command: link control register from the PCI express extended capacity structure.
Bit 31-16 (R): Configuration Link Status: link status register from the PCI express extended capacity structure.

➤ Address offset: 0x060 (R): PCI express configure status:

Bit 0: Configuration To Turnoff: notify the user that a PME_TURN_OFF message has been received ans the main power will soon be removed (negative logic).
Bit 1: Configuration Read Write Done (negative logic).
Bit 2: Configuration Error Completion Ready (negative logic)
Bit 3: Configuration Interrupt Ready (negative logic)
Bit 4: Configuration Interrupt MSI Enabled. Indicates that the Message Signalling Interrupt (MSI) messaging is enabled. If 0, then only legacy (INTx) interrupts can be sent.
Bit 12-5: Configuration Interrupt Data Out (7:0);
Bit 15-13: Configuration Interrupt Multiple Message Enable (2:0);
Bit 31-16: Configuration Completer ID (15:0)

➤ Address offset: 0x00064: Front Panel generic trigger input pre-scale: Reset default 0x00000000

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: 0x0006C (R): PCI express Configuration:

Bit 15-0: Configuration Command (15:0), command register from the configuration space header
Bit 31-16: Configuration Status(15:0), status register from the configuration space header.

➤ Address offset: 0x00070 (R): PCI express Device Configuration:

Bit 15-0: Configuration Device Command(15:0), Device control register from the PCI express extended capacity structure;

Bit 31-16: Configuration Device Status(15:0), Device status register from the PCI express extended capacity structure.

➢ Address offset: 0x00074 (R/W): Event type for some special event:

Bit 7-0: Multiple hits by GTP 'or' External trigger inputs (default 0xFA, for TS only)
Bit 15-8: Multiple hits by GTP 'and' External trigger inputs (default 0xFA, for TS only)
Bit 23-16: Software generated periodic trigger event type (default 0xFD);
Bit 31-24: Software generated random trigger event type (default 0xFE).

➢ 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: PCI express clock DCM reset, and full reset; (do not use)
0x22: CLK250 resync (AD9510, DCM resync and MGT reset for trigger link);
0x33: AD9510 re-sync (slower clock phase adjustment), part of 0x22 functions;
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 TIslave trigger ready register;
0xBB: Reset the event number, and trigger input scalars (the 0xDD will not do);
0xDD: (SyncReset), FPGA logic and counter reset;
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: generate an ~ 4us SyncReset signal.
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 00,0111

Bit 7-0 (R/W): Reset (generated by Sync code 0xdd) pulse width. Pulse width is (Bit(6:0)*(4/32 ns)), Bit(7) determines the steps (4ns or 32ns);

➢ Address offset: 0x00084: software 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): software Random Trigger Command Register:

Bit 3-0: Random trigger_1 rates: $466 \text{ KHz}/(2^{Bit(3:0)})$;
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: $466\ KHz/(2^{Bit(11:8)})$;

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): software periodic Trigger Generation:

Bit 15-0: Number of trigger_1 to be generated; if 0xFFFF, the number of event will not be limited.

Bit 31-16: (trigger rate control) Time between triggers. $T = (32+8*Bit(30:16)*2048^{Bit(31)})$ ns. (Assuming that the PCI express reference clock is 100MHz, and the interface clock is 125 MHz or 8ns period. As the bit(17:16) is not used for higher frequency mode when Bit(31)=0, it might be better to write $T = (32+4*8*Bit(30:18)*2048^{Bit(31)})$ ns)

➢ 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. $T = (32+8*Bit(30:16)*2048^{Bit(31)})$ ns. (Assuming that the PCI express reference clock of 100 MHz, and the interface clock is 125 MHz or 8 ns 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): TIpcie master generated sync code valid;

Bit (18:15): TIpcie master generated sync code;

Bit (14): Loopback sync valid;

Bit (13:10): Loopback sync code;

Bit (4): AFBR fiber sync valid;

Bit (3:0): AFBR fiber 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.

➢ 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=78.125ps 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 Bit(31:0)*256*8ns. (Scalar Latch is required.)

➢ Address offset: 0x000AC (R): Trigger busy (trigger dead) timer:

Bit 31-0 (r): TI(pcie) busy (cannot accept trigger, or trigger dead) time counter. The real time is Bit(31:0)*256*8ns. 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 0: Trigger Link (AFBR) MGT reset_done;
Bit 5-4: Trigger link received Data Error;
Bit 7-6: Trigger link received data Not_in_table;
Bit 8: Trigger link MGT PLL lock detected;
Bit 15-13: PCI Express Link State (2:0): One-hot encoded bus that reports the PCI express link state to the user: 110 → PCI express link state is "L0"; 101 → PCI express link state is "L0s"; 011 → PCI express link state is "L1"; 111 → PCI express link state is "in transition"
Bit 23-16: trn_rfc_ph_av(7:0): Receive Posted Header Flow Control Credits Available: the number of Posted Header FC credits available to the remote link partner.
Bit 31-24: trn_rfc_nph_av(7:0): Receive Non-posted Header Flow Control Credit Available: Number of Non-posted Header FC Crefits available to the remote link partner.

➢ Address offset: 0x000B4 (R): MGT STATUS_B registers (not assigned on TIpcie):

Bit 31-0: The current DMA physical address. (Apr. 2018)
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 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: PCI express interface CLK (125 MHz, not the 100 MHz reference) 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 TIpcie in master mode:

Bit 7-0: Number of blocks to be readout on AFBR (blocks need ROCack);
Bit 15-8: Number of blocks is still missing on AFBR (blocks need TRGack);
Bit 23-16: Number of blocks to be readout on TIpcie itself (loopback);

Bit 31-24: Number of blocks is still missing on TIpcie 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: 0x000EC (R/W): ROC enable

Bit 7-0: ROC 8:1 enable, the default is 00000001

Bit 11-10: SyncResetRequest enable. This corresponds to the TI slaves (bit#11) and the loopback (bit#10).
Bit 21-20: Monitor of the SyncResetRequest corresponding to the bit#11:10.

➢ 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: 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 PCI express interface cycle. If the ClkPcie is 125 MHz, the one-shot will be 8ns wide. Positive logic.

Bit 0: not used;
Bit 1: if '1', reset the PCIe_to_I2C engine (not used);
Bit 2: if '1', RESET signal to reset the PCIe_to_JTAG engine;
Bit 3: if '1', RESET signal to reset the PCIe_to_SFM engine (not used);
Bit 4: if '1', RESET signal to reset the BR#0 registers (TIpcie settings) to their default values;
Bit 6: if '1', reset the SYNC history FIFO (clear the FIFO);
Bit 7: if '1', this register will generate a BUSY reset, and ROC_Ack pulse;
Bit 8: if '1', Reset the CLK250/Clk200 DCM.
Bit 9: if '1', Reset the CLK125 DCM.
Bit 10: if '1', Reset the trigger MGT (MultiGigabit Transceiver,) inside the FPGA.
Bit 11: if '1', Auto alignment of SYNC phase from AFBR fiber;
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 17: if '1', the available number of data blocks will decrease by 1 without real readout,

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 (0xAC) and LiveTimer (0xA8) 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): front panel input trigger (TS Code) delay (in 4 ns steps):

Bit 8-0: TScode#1 trigger input delay (Channel#1);
Bit 18-10: TScode#2 trigger input delay (Channel#2);
Bit 28-20: TScode#3 trigger input delay (Channel#3);

➢ Address offset: 0x00108 (R/W): front panel input trigger (TS Code) delay (in 4 ns steps):

Bit 8-0: TScode#4 trigger input delay (Channel#4);
Bit 18-10: TScode#5 trigger input delay (Channel#5);
Bit 28-20: TScode#6 trigger input delay (Channel#6);

➢ 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. MinimumWidth = bit(30:24)*Tclock. Tclock = 480ns if bit#31 of Reg#1C is set to '0'; Tclock = 480*32ns 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. MinimumWidth = bit(22:16)*Tclock. Tclock = 480ns if bit#31 of Reg#1C is set to '0'; Tclock = 480*32ns 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. MinimumWidth = bit(14:8)*16ns. The maximum width setting is ~2us.
Bit 7-0: Minimum busy width for trigger rule#1, not used, and no need for this.

➢ Address offset: 0x140 – 0x17C (W): Trigger table loading:

Bit 31-0: 32-bit wide table loading.
Address bits(5-2) are used to load 16 32-bit words;

➢ Address offset: 0x00180 (R): FP input trigger scalar for #1:

Bit 31-0: 32-bit scalar, number of input#1 counter

➢ Address offset: 0x00184 (R): FP input trigger scalar for #2:

Bit 31-0: 32-bit scalar, number of FP input#2 counter

➢ Address offset: 0x00188 (R): FP input trigger scalar for #3:

Bit 31-0: 32-bit scalar, number of FP input#3 counter

➢ Address offset: 0x0018C (R): FP input trigger scalar for #4:

Bit 31-0: 32-bit scalar, number of FP input#4 counter

➢ Address offset: 0x00190 (R): 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: 0x001D0 (R): optic transceiver AFBR TI ID (in master mode)

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

➢ 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.

6.2 BAR#1 (PCI express to JTAG engine, PCIe to I2C engine):

➢ When the BAR0 register 0x1C bit (30:28) is anything but "101", the BAR#1 is a PCIe to JTAG engine.

BAR#1 4 Kbyte spaces are used for PCI express to JTAG engines. The engines include: PCI express to JTAG engine for the FPGA and PCI express to JTAG engine for the PROM. Not all the 32 bits of data may be used. In that case, the lower N bits (specified by the address) are used.

Register offset Address(11:0): 0xXXX

Address Bit(11): JTAG engine selection, '1' for PROM, '0' for FPGA;

Address Bit(10:6): Number of bits to shift (Instruction register or Data register), n=A(10:6)+1, that is the minimum number of bits to shift is 1, and maximum number of bits to shift is 32, the lower order of bits in the 32-bit data are used.

Address Bit(5:4): Specific JTAG command. "01": JTAG data register shift; "10": JTAG instruction register shift; "11": JTAG reset, (set the JTAG to RESET_IDLE state).

Address Bit(3): JTAG TRAILER_ENABLE. This will enable several extra JTAG clocks (after register shift) to move the JTAG state machine from register shift (either data register or instruction register) to RESET_IDLE.

Address Bit(2): JTAG HEADER_ENABLE. This will enable several extra JTAG clocks (before register shift) to move the JTAG state machine from RESET_IDLE to register shift (either data register or instruction register depending on the shift type A(5:4))

Address Bit(1:0): "00" only.

The PCI express read to the device will return the data currently stored in the TDO shift register (32-bit). The read address is Address(11). Address(10:0) are DONOT care bits. 32-bit data read only

➢ When the BAR0 register 0x1C bit (30:28) are set as "101", the BAR#1 is a PCIe to I2C engine.

BAR#1 4 Kbyte spaces are used for PCI express to I2C engines. PCIe A32 operation only.

Register offset Address(11:0): 0xXXX definition:

Address(11): Lowest bit of the device address, for Avago AFBR-79EIDZ, the address is "1010_000", that is this bit is set to '0'.

Address(10): '1' for two bytes operation; '0' for single byte operation.

Address(9:2): Eight bits of I2C (starting) byte address.

Address(1:0): always "00".

The readout data is always 32-bit, but only the lower 8 bits (16 bits) are used for one bytes (two bytes) read and write. The PCI read data is the data in the engine register, so there is a one-read lag (behind).

6.4 PCI express data acquisition:

For data acquisition, the DMA data writes are used. The DMA physical address is set by the BR#0 register (0x54 and 0x58). The TIpcie writes to the DMA as a ring buffer. The buffer unit is 4 Kbyte. For extra large events, two buffer units may be used (when the block level is set to more than 240, and the event size is set to maximum).

7 Backplane pin out tables:

7.1 PCI express Pinout Table (copied from Wikipedia)

Pin	Side B	Side A	Description
1	+12 V	PRSNT1#	Must connect to farthest PRSNT2# pin
2	+12 V	+12 V	
3	+12 V	+12 V	
4	Ground	Ground	
5	SMCLK	TCK	
6	SMDAT	TDI	
7	Ground	TDO	SMBus and JTAG port pins
8	+3.3 V	TMS	
9	TRST#	+3.3 V	
10	+3.3 V aux	+3.3 V	Standby power
11	WAKE#	PERST#	Link reactivation; fundamental reset
		Key notch	
12	CLKREQ#	Ground	Request running clock
13	Ground	REFCLK+	Reference clock differential pair
14	HSOp(0)	REFCLK−	Lane 0 transmit data, + and −
15	HSOn(0)	Ground	
16	Ground	HSIp(0)	Lane 0 receive data, + and −
17	PRSNT2#	HSIn(0)	
18	Ground	Ground	

PCI Express ×1 cards end at pin 18

19	HSOp(1)	Reserved	Lane 1 transmit data, + and −
20	HSOn(1)	Ground	
21	Ground	HSIp(1)	Lane 1 receive data, + and −
22	Ground	HSIn(1)	
23	HSOp(2)	Ground	Lane 2 transmit data, + and −
24	HSOn(2)	Ground	
25	Ground	HSIp(2)	Lane 2 receive data, + and −
26	Ground	HSIn(2)	
27	HSOp(3)	Ground	Lane 3 transmit data, + and −
28	HSOn(3)	Ground	
29	Ground	HSIp(3)	Lane 3 receive data, + and −
30	Reserved	HSIn(3)	
31	PRSNT2#	Ground	
32	Ground	Reserved	

7.3 Front panel (PCI express bracket) User-defined pin table

11 pairs of the 12-pair connector are used:

Pin name	Signal Name	Signal Level
Pin 1/2	Not used	NA
Pin 3+/4-	Output #1,	ECL
Pin 5+/6-	Output #2	ECL
Pin 7+/8-	Output #3	ECL
Pin 9+/10-	Output #4	LVPECL25
Pin 11+/12-	Input, front panel BUSY in	Any level differential
Pin 13+/14-	Input TS#1	Any level differential
Pin 15+/16-	Input TS#2	Any level differential
Pin 17+/18-	Input TS#3	Any level differential
Pin 19+/20-	Input TS#4	Any level differential
Pin 21+/22-	Input TS#5	Any level differential
Pin 23+/24-	Input TS#6	Any level differential

* for HallB HPS (and later), Output#1 = Reset; Output#2 = Trigger; Output#3 = Clk250/2; Output#4 = Clk125, Pin1/2 is Clk125 from AD9510.

7.4 JTAG connector pin table

6-pin connector pin definition, fly wire to Xilinx programmer (USBII):

Pin name	Signal Name	Signal Level
Pin 1	JTAG TDI	LVTTL
Pin 2	JTAG TCK	LVTTL
Pin 3	JTAG TMS	LVTTL
Pin 4	JAG TDO	LVTTL
Pin 5	Xilinx programmer Vref	2.5V

Pin 6	GND	0 V

8 TID Operation examples:

The following is some operating procedures on Dell T5500 machine.

GU_PC > cd ~jgu/new_pti/linux_driver/

GU_PC > su

GU_PC > reload_driver.sh

GU_PC > cd ~jgu/new_pti/test

8.1 BA#0 register (0x00) write/read operation:

GU_PC > pti_test w 0 0x00 0xdadadada -- memory 0x00 write

GU_PC > pti_test r 0 0x00   -- memory 0x00 read

8.2 BA#1 access: Readout the FPGA user_ID:

The FPGA user code is readout through PCI express to JTAG engine by BA#1 memory.

GU_PC > pti_test w 1 0x 3c 0   -- reset the JTAG engine

GU_PC > pti_test w 1 0x26c 0x3c8 -- 10-bit instruction register

GU_PC > pti_test w 1 0x7dc 0   -- 32-bit data register

GU_PC > pti_test r 1 0x0 -- 32-bit data register read

9. Citations:

Works Cited

Chris. (2009). Gloabal 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 experiements. *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:

UP3

A1	GND	VOUT	G11	V18RAW
A2	GND	VOUT	G10	
A3	GND	VOUT	G9	
A4	GND	VOUT	G8	
A5	GND	VOUT	G7	
A6	GND	VOUT	G6	
A7	GND			
A8	GND			
A9	GND	VOUT	F11	
A10	GND	VOUT	F10	
A11	GND	VOUT	F9	
		VOUT	F8	
		VOUT	F7	
B6	GND	VOUT	F6	
B7	GND			
B8	GND	VOUT	E11	LP20 10uH V18
B9	GND	VOUT	E10	
B10	GND	VOUT	E9	
B11	GND	VOUT	E8	

UP4 TPS74401RGWT

V12GTP
V3FOR18
RP16 510
RP15 1.02K
DP4
RP14 1.02K
LED2
V10CORE

UP5 TPS74401RGWT

CP16 T520A 10uF
V3FOR18
0.01uF
CS12
RP12x 510 V3FOR18
RP12 510
RP13 1.02K
DP5
RP11 1.02K
GND

C8	GND
C9	GND
C10	GND
C11	GND
E4	GND
E5	GND
F3	GND
F4	GND
F5	GND
G3	GND
G4	GND
G5	GND

GND

VOUT D11
VOUT D10
VOUT D9
VOUT D8

CP15 T520A 10uF
V3FOR18
0.01uF
CS13

RP10 3.24K V3FOR18
LED on, when the power is BAD
DP3
LED2

D7	VIN
C7	VIN
C6	VIN
E7	VIN
E6	VIN
B1	VIN
C1	VIN
C3	VIN
C4	VIN
C5	VIN

PGOOD F1 D1
RUN/SS G2
FB G1
COMP B3
SW B4
SW E1
TRACK

RP9 3.24K
RP9y 1.02K RP9z 3.24K CS11 0.01uF
The power rising time is ~ 5ms
GND

V3IN LP5 10uH

V3FOR18
LTM4604EV

CP14 T520A 10uF
GND

PCIe4Edge

B1	+12V	PRSNT1#	A1	PRESENT
B2	+12V		A2	+12V
B3	+12V	+12V	A3	
B4	GND	+12V	A4	
B5	SMCLK	GND	A5	TCKPROM
B6	SMDAT	TCK	A6	TDIPROM
B7	GND	TDI	A7	TDOPROM
B8	+3.3V	TDO	A8	TMSPROM
B9	TRST#	TMS	A9	
B10	+3.3AUX	+3.3V	A10	+3.3V
B11	WAKE#	+3.3V	A11	
		PERST#		
B12	Reserved	GND	A12	
B13	GND	GND	A13	PCIECLK P
B14	HSOp0	RefClk+	A14	PCIECLK N
B15	HSOn0	RefClk-	A15	
B16	GND	GND	A16	PCIETX0 P
B17	PRSNT2#	HSIp0	A17	PCIETX0 N
B18	GND	HSIn0	A18	
B19	HSOp1	GND	A19	
B20	HSOn1	Reserved	A20	
B21	GND	GND	A21	PCIETX1 P
B22	GND	HSIp1	A22	PCIETX1 N
B23	HSOp2	HSIn1	A23	
B24	GND	GND	A24	
B25	GND	GND	A25	PCIETX2 P
B26	GND	HSIp2	A26	PCIETX2 N
B27	HSOp3	HSIn2	A27	
B28	HSOn3	GND	A28	
B29	GND	GND	A29	PCIETX3 P
B30	Reserved	HSIp3	A30	PCIETX3 N
B31	PRSNT2#	HSIn3	A31	
B32	GND	GND	A32	
		Reserved		

PCIe4

+12V
+3.3V
PCIERX0 P B13
PCIERX0 N B15
PRESENT B17
PCIERX1 P B19
PCIERX1 N B21
PCIERX2 P B23
PCIERX2 N B25
PCIERX3 P B27
PCIERX3 N B29
PRESENT B31

GND GND

Pfront

	1	2	
GENOUT1 P	3	4	GENOUT1 N
GENOUT2 P	5	6	GENOUT2 N
GENOUT3 P	7	8	GENOUT3 N
GENOUT4 P	9	10	GENOUT4 N
GENIN0 P	11	12	GENIN0 N
GENIN1 P	13	14	GENIN1 N
GENIN2 P	15	16	GENIN2 N
GENIN3 P	17	18	GENIN3 N
GENIN4 P	19	20	GENIN4 N
GENIN5 P	21	22	GENIN5 N
GENIN6 P	23	24	GENIN6 N

09 18 124 9621

GENIN0 N R123 10K
GENIN1 N R122 10K
GENIN2 N R121 10K
GENIN3 N R120 10K
GENIN4 N R119 10K
GENIN5 N R118 10K
GENIN6 N R117 10K
V5

GENIN0 P R110 10K
GENIN1 P R111 10K
GENIN2 P R112 10K
GENIN3 P R113 10K
GENIN4 P R114 10K
GENIN5 P R115 10K
GENIN6 P R116 10K
VM5

TB1 — max9602EUG

RT41 100 GENIN3_P	1 INA+ VccoA 24
GENIN3_N	2 INA- QA+ 23
	3 VEE QA- 22 FPGAPIN3_P RD01 100
VM5 GENIN4_P	4 INB+ 21 FPGAPIN3_N RD02 100
RT42 100 GENIN4_N	5 INB- VccoB 20 FPGAPIN4_P RD03 100
	6 VCC QB+ 19 FPGAPIN4_N RD04 100
GENIN5_P	7 INC+ QB- 18
RT43 100 GENIN5_N	8 INC- VccoC 17 FPGAPIN5_P RD05 100
	9 VEE QC+ 16 FPGAPIN5_N RD06 100
RT44 100 GENIN6_P	10 IND+ QC- 15
GENIN6_N	11 IND- VccoD 14 FPGAPIN6_P RD07 100
V5	12 VCC QD+ 13 FPGAPIN6_N RD08 100

V25 GND

TB2 — max9602EUG

RT34 100 FPGAOUT4_P	1 INA+ VccoA 24
FPGAOUT4_N	2 INA- QA+ 23 GENOUT4_P RD09 100
VM5	3 VEE QA- 22 GENOUT4_N RD10 100
GENIN0_P	4 INB+ 21
RT35 100 GENIN0_N	5 INB- VccoB 20 FPGAPIN0_P RD11 100
	6 VCC QB+ 19 FPGAPIN0_N RD12 100
GENIN1_P	7 INC+ QB- 18
RT47 100 GENIN1_N	8 INC- VccoC 17 FPGAPIN1_P RD13 100
	9 VEE QC+ 16 FPGAPIN1_N RD14 100
RT48 100 GENIN2_P	10 IND+ QC- 15
GENIN2_N	11 IND- VccoD 14 FPGAPIN2_P RD15 100
V5	12 VCC QD+ 13 FPGAPIN2_N RD16 100

V25 GND

MC100EP91 — EP_VEE

RT31 100 FPGAOUT3_P	GENOUT3_P RIO11 510
FPGAOUT3_N	GENOUT3_N RIO12 510
VBB 7	25
GND 24
RT32 100 FPGAOUT2_P 8	GND 23
FPGAOUT2_N 9	GENOUT2_P RIO13 510
10	22
V33 11	21 GENOUT2_N RIO14 510
VCC 12	GND 20
	GND 19 GND
RT33 100 FPGAOUT1_P	GENOUT1_P RIO15 510
FPGAOUT1_N 13	GENOUT1_N RIO16 510

VM5

Decoupling capacitors:
VM5 CI1 T520A 10uF, CI2 T520A 10uF, CI3 T520A 10uF, CI11 0.22uF, CI12 0.22uF, CI13 0.22uF, CI14 0.22uF, CI15 0.22uF, CI16 0.22uF
V5 CI4 T520A 10uF, CI5 T520A 10uF, CI17 0.22uF, CI18 0.22uF, CI19 0.22uF, CI20 0.22uF
V33 CI6 T520A 10uF, CI21 0.22uF, CI22 0.22uF
V25 CI7 T520A 10uF, CI8 T520A 10uF, CI23 0.22uF, CI24 0.22uF, CI25 0.22uF, CI26 0.22uF, CI27 0.22uF, CI28 0.22uF, CI29 0.22uF, CI30 0.22uF
GND

SC1 — SW DIP-8

RC01 3.24K CLKSEL1	1	16	
RC02 3.24K CLKSEL2	2	15	
RC03 3.24K CLKSEL3	3	14	
RC04 3.24K CLKSEL4	4	13	
RC05 1.02K CLKSEL5	5	12	RCX01 1.02K
RC06 1.02K CLKSEL6	6	11	RCX02 1.02K
RC07 1.02K CLKSEL7	7	10	RCX03 1.02K
RC08 1.02K CLKSEL8	8	9	RCX04 1.02K

GND

FPGASEL5 PSW5 510 CLKSEL5 RCX05 3.24K
FPGASEL6 PSW6 510 CLKSEL6 RCX06 3.24K
FPGASEL7 PSW7 510 CLKSEL7 RCX07 3.24K
FPGASEL8 PSW8 510 CLKSEL8 RCX08 3.24K
GND

V33 CC1 T520A 100uF, CC2 T520A 100uF, CC11 0.22uF, CC12 0.22uF, CC13 0.22uF, CC14 0.22uF, CC15 0.22uF
GND

UC01 — CCPD-034 160 MHz VCLOCK

RCT50A 3.24K V33	6
OSCEN 1	E/D OUT 4 OSCCLK_P
2	N/C COUT 5 OSCCLK_N
	3 GND

GND
RC31 50, RC32 50, RC33 50, CC32 0.01uF
GND
RC43 100
TDCLK250_P, TDCLK250_N
GND

U13 — AD9510BCPZ

AD9510CS_N 21	CS STATUS 17
AD9510SCLK 18	SCLK
AD9510SDIO 19	SDIO CPRSET 63 RCTR3 3.24K
AD9510SDO 20	SDO RSET 61 RCTR2 3.24K
	VCP 5
CLKRESET 16	FUNC CP 6
RCTR1 3.24K V33	
REFIN 1	REFIN VS 26, 13, 9, 4, 23, 30, 31, 64, 60, 59, 56, 52, 51, 48, 45, 44, 41, 40, 37, 36, 33, 65
REFIN 2	REFIN
CC47 0.1uF CLK1 14	CLK1
CC48 0.1uF 15	CLK1
CC46 0.1uF CLK2 10	CLK2
CC45 0.1uF 11	CLK2
RCT44 150 CLK250B_N 58	OUT0
RCT45 150 CLK250B_P 57	OUT0
RCT46 150 CLK250A_N 54	OUT1
RCT47 150 CLK250A_P 53	OUT1
35	OUT2
34	OUT2
29	OUT3
28	OUT3
CLK3125_P CC46C 0.1uF CLKSLOW_P 47	OUT4
CLK3125_N CC45 0.1uF CLKSLOW_N 46	OUT4
CLK250C_P CC48C 0.1uF CLKFREQB_P 43	OUT5
CLK250C_N CC47C 0.1uF CLKFREQB_N 42	OUT5
PCI2CLK_P CC42 0.1uF CLKFREQA_P 39	OUT6
PCI2CLK_N CC41 0.1uF CLKFREQA_N 38	OUT6
MGTCLK_P CC44 0.1uF CLK_P 25	OUT7
MGTCLK_N CC43 0.1uF CLK_N 24	OUT7
	GND 62, 55, 50, 49, 3, 7, 8, 12, 22, 27, 32

GND

U20L — XC5VLX30T-1FF665C

Net	Pin	Signal
V25		
RPG2 3.24K	J11	CLKPRGC — CCLK_0
	K11	PROGDONE — DONE_0
RPG5 3.24K	J20	PROG_N — PROGRAM_B_0
GND	T20	M0_0
	W20	M1_0
V25	U20	M2_0
	K20	HSWAPEN_0
	M20	RDWR_B_0
RPG6 3.24K	V13	TDIFPGA — TDI_0
	V12	TDOFPGA — TDO_0
	V11	TCKFPGA — TCK_0
	W13	TMSFPGA — TMS_0
	R15	DXP_0
	R14	DXN_0
PROGBUSY	U12	D_OUT_BUSY_0
RPGD01_NL 0	J10	D_IN_0
PROGSD0	L20	CS_B_0
INIT_N	H12	INIT_B_0
RPG10 3.24K / RPG4 3.24K	N4	FLOAT

GND

UPG1 — kc2520-33MHz
- 1 EN — VCC 4
- 2 GND — CLK 3 → CLKPRG
- V25, GND

FPGA_JTAG — Header 6
- V25
- TDIFPGA 1
- TCKFPGA 2
- TMSFPGA 3
- TDOFPGA 4
- 5
- 6
- 50 RX02F
- GND

PROM_JTAG — Header 6
- V25
- TDIPROM 1
- TCKPROM 2
- TMSPROM 3
- TDOPROM 4
- 5
- 6
- 50 RX02
- GND

RPG11A 3.24K
GND

UPG2 — XCF32PVO48C

Pin	Signal		Pin	Signal
4	VCCINT		28	D0 — PROGSD0 (RPGD02 0) AA15
15	VCCINT		29	D1 — Y15
34	VCCINT		32	D2 — W15
8	VCCO		33	D3 — Y13
30	VCCO		43	D4 — W16
38	VCCO		44	D5 — Y16
45	VCCO		47	D6 — AA14
24	VCCJ		48	D7 — Y14
12	CLK — CLKPRG		9	CLKOUT — CLKPRGOUT
13	CE — PROGDONE		10	CEO
21	TMS — TMSPROM		22	TDO — TDOPROM
19	TDI — TDIPROM			
20	TCK — TCKPROM			
5	BUSY — PROGBUSY		6	CF — PROG_N
25	EN_EXT_SEL		11	OE/RESET — INIT_N
26	REV_SEL0 — CLKSEL1		1	DNC
27	REV_SEL1 — CLKSEL2		3	DNC
2	GND		14	DNC
7	GND		16	DNC
17	GND		35	DNC
23	GND		37	DNC
31	GND		39	DNC
36	GND		40	DNC
46	GND		41	DNC
			42	DNC

V18, V25
RPGCLK3 3.24K — RPGCLK_N — CLKPRGOUT 0
RPG8 150 — RDONE 510 — DPG2 LED2
CLKPRGC — RPGCLK2 — CLKPRG 0
RPG9 3.24K
GND

U20B — XC5VLX30T-1FF665C — BANK 2

Pin	Signal
AA15	IO_L9N_D0_FS0_2
Y15	IO_L9P_D1_FS1_2
W15	IO_L8N_D2_FS2_2
Y13	IO_L8P_D3_2
W16	IO_L7N_D4_2
Y16	IO_L7P_D5_2
AA14	IO_L6N_D6_2
Y14	IO_L6P_D7_2
AA13	IO_L5N_CSO_B_2
AA16	IO_L5P_FWE_B_2
Y18	IO_L4N_VREF_FOE_B_MOSI_2
Y19	IO_L4P_FCS_B_2
AA18	IO_L3N_A20_2
AA19	IO_L3P_A21_2
AA16	IO_L2N_A22_2
AA19	IO_L2P_A23_2
AA17	IO_L1N_CC_A24_2
Y20	IO_L1P_CC_A25_2
Y19	IO_L0N_CC_RS0_2
W19	IO_L0P_CC_RS1_2

TDIFPGA, TMSFPGA, TDOFPGA, TCKFPGA, CLKSEL1, CLKSEL2

QSFPA — QSFP

Pin	Left	Right	Pin
1	GND	GND	38
2	TX2N — TXA2_N (CCF17)	TX1N — TXA1_N (CCAU5)	37
3	TX2P — TXA2_P (CCF18)	TX1P — TXA1_P	36
4	GND	GND	35
5	TX4N — TXA4_N (CCF03)	TX3N — TXA3_N (CCAU7)	34
6	TX4P — TXA4_P (CCF04)	TX3P — TXA3_P	33
7	GND	GND	32
8	ModSel1L — TIAMSEL	LPMode	31
9	ResetL — FIBERRST	Vcc1	30
10	VccRx — QSFPRXA	VccTx — QSFPTXA	29
11	SCL — FIBERICK	IntL — MODINTA	28
12	SDA — FIBERIDATA	ModPrsL — MODPRSTA	27
13	GND	GND	26
14	RX3P — SYNCAR_P	RX4P — AUXOUT1_P	25
15	RX3N — SYNCAR_N	RX4N — AUXOUT1_N	24
16	GND	GND	23
17	RX1P — TRGLIN1_P	RX2P — TDCLK250_P	22
18	RX1N — TRGLIN1_N	RX2N — TDCLK250_N	21
19	GND	GND	20

CLK250A_N / CLK250A_P (CCF17/CCF18 0.1uF)
AUXIN1_N / AUXIN1_P (CCF03/CCF04 0.1uF)
TRGLOUTG_N / TRGLOUTG_P (CCAU6 0.1uF)
CLKSYNCA_N / CLKSYNCA_P (CCAU8 0.1uF)
SYNCAR_P / SYNCAR_N
TRGLIN1_P / TRGLIN1_N
GND

MODINTA — RFMI 1.02K — V33
MODPRSTA — RFMP 1.02K — V33
FIBERICK — RFCK 1.02K — V33
FIBERIDATA — RFDA 1.02K — V33

V33
RM41 100 — LGI1 1.0uH — CF1 10uF — QSFPTXA — T520A 10uF — CF2, CF11 0.22uF
RM42 100 — LGI2 1.0uH — CF3 10uF — QSFPRXA — T520A 10uF — CF12 0.22uF
RM43 100 — LGI3 1.0uH — CF4 10uF — QSFPVA — T520A 10uF — CF13 0.22uF
GND

U20N

V33
V25
V25

Pin	Signal		Pin	Signal
F11	VCCO_0		U8	VCCAUX
J12	VCCO_0		K9	VCCAUX
B13	VCCO_1		M9	VCCAUX
E14	VCCO_1		P9	VCCAUX
AA16	VCCO_2		T9	VCCAUX
AD17	VCCO_2		W10	VCCAUX
D17	VCCO_3		M19	VCCAUX
G18	VCCO_3		P19	VCCAUX
AB13	VCCO_4		T19	VCCAUX
AE14	VCCO_4		V19	VCCAUX
J22	VCCO_11		Y19	VCCAUX
M23	VCCO_11		N20	VCCAUX
H25	VCCO_11			

V10GTP
V25
V25

CPL02 T520A 22uF
CPL03 T520A 22uF
CPL04 T520A 22uF
CPL05 T520A 22uF
GND

V10CORE

CPL01 T520A 22uF
CPL11 T520A 22uF
GND

CPL10 T520A 22uF
CPL07 T520A 22uF
CPL06 T520A 22uF
CPL08 T520A 22uF
CPL09 T520A 22uF
GND

CPL12 T520A 22uF
CPL13 T520A 22uF
CPL14 T520A 22uF
CPL15 T520A 22uF
GND

VCCINT pins: L10, N10, R10, U10, M11, P11, T11, L12, N12, R12, K13, M13, P13, T13, J14, L14, U14, H15, K15, T15, V15, J16, L16, N16, R16, U16, H17, K17, M17, P17, T17, V17, J18, L18, N18, R18, U18, W18, K19

Pin	Signal
H5	VCCO_12
L6	VCCO_12
P7	VCCO_12
W22	VCCO_13
R24	VCCO_13
V25	VCCO_13
F21	VCCO_15
B23	VCCO_15
E24	VCCO_15
D7	VCCO_16
G8	VCCO_16
C10	VCCO_16
AC20	VCCO_17
AB23	VCCO_17
AE24	VCCO_17
AA6	VCCO_18
AD7	VCCO_18
AC10	VCCO_18

VMGTR18 CCL14 0.22uF GND
VMGTR16 0.22uF CPL13 GND
VMGTR14 0.22uF CPL12 GND
VMGTR12 0.22uF CPL11 GND

VMGT112 0.22uF CPL01 GND
VMGT114 0.22uF CPL02 GND
VMGT116 0.22uF CPL03 GND
VMGT118 0.22uF CPL04 GND

V10GTP

Pin	Signal
L3	MGTAVCC_112
L4	MGTAVCC_112
U3	MGTAVCC_114
U4	MGTAVCC_114
E3	MGTAVCC_116
E4	MGTAVCC_116
AC3	MGTAVCC_118
AC4	MGTAVCC_118
M3	MGTAVCCPLL_112
V3	MGTAVCCPLL_114
F3	MGTAVCCPLL_116
AD3	MGTAVCCPLL_118
P5	MGTAVTTRXC

VMGT112 LL01 10uH
VMGT114 LL02 10uH
VMGT116 LL03 10uH
VMGT118 LL04 10uH
VPLL12 LL05 10uH
VPLL14 LL06 10uH
VPLL16 LL07 10uH
VPLL18 LL08 10uH
VMGTC LL10 10uH

V12GTP

Pin	Signal	Signal
J3	MGTAVTTRX_112	VMGTR12
R3	MGTAVTTRX_114	VMGTR14
C3	MGTAVTTRX_116	VMGTR16
AA3	MGTAVTTRX_118	VMGTR18
H3	MGTAVTTTX_112	VMGTT112
N3	MGTAVTTTX_112	VMGTT112
P3	MGTAVTTTX_114	VMGTT114
W3	MGTAVTTTX_114	VMGTT114
B3	MGTAVTTTX_116	VMGTT116
G3	MGTAVTTTX_116	VMGTT116
Y3	MGTAVTTTX_118	VMGTT118
AE3	MGTAVTTTX_118	VMGTT118

LL11 10uH
LL12 10uH
LL13 10uH
LL14 10uH
LL15 10uH
LL16 10uH
LL17 10uH
LL18 10uH

VPLL12 CCL05 0.22uF GND
VPLL14 0.22uF CPL06 GND
VMGTC 0.22uF CPL09 GND
VPLL16 0.22uF CPL07 GND
VPLL18 0.22uF CPL08 GND

CPL20 T520B 330uF
GND

V25

Pin	Signal		Pin	Signal
J19	VBATT_0		M14	AVSS_0
M15	AVDD_0		N14	VREFN_0
P15	VREFP_0		P14	VN_0
N15	VP_0			

V12GTP RVP0 50

XC5VLX30T-1FF665C

VMGTR12 0.22uF CPL15 GND
VMGTT114 0.22uF CPL16 GND
VMGTT116 0.22uF CPL17 GND
VMGTT118 0.22uF CPL18 GND

Some of the GTP_TXs will be powered down, so the power can be connected this way (incorrectly)

U20O

GND GND GND GND GND GND GND GND GND GND GND GND GND GND GND GND GND GND GND GND GND GND GND GND GND GND GND GND GND GND GND GND GND GND GND GND GND GND GND GND GND GND GND GND GND GND GND GND GND GND GND GND GND GND GND GND GND GND GND GND GND GND GND GND GND GND GND GND GND GND RSVD RSVD

A2 D2 E2 K2 L2 T2 U2 AB2 AC2 AF2 C4 F4 J4 M4 R4 V4 AA4 AE4 C5 N5 V5 AC5 A6 F6 T6 AF6 J7 W7 B8 M8 AB8 E9 L9 N9 R9 U9 Y9 AE9 H10 K10 M10 P10 T10 V10 A11 L11 N11 R11 U11 AA11 AF11 D12 K12 M12 P12 T12 W12 AD12 G13 J13 L13 N13 R13 U13 H14

K14 T14 V14 Y14 C15 J15 L15 U15 W15 AC15 A16 F16 H16 K16 M16 P16 T16 V16 AF16 J17 L17 N17 R17 U17 W17 B18 K18 M18 P18 T18 V18 AB18 E19 L19 N19 R19 U19 W19 AE19 C20 H20 V20 A21 L21 T21 AA21 AF21 D22 P22 AD22 G23 U23 K24 Y24 C25 N25 AC25 A26 F26 L26 T26 AA26 AF26 P20 R20

GND

XC5VLX30T-1FF665C

U20P

NC   M5

XC5VLX30T-1FF665C

VMGTT112

RT5
50

U20M

P4   MGTRREF_112
K4   MGTREFCLKP_112
K3   MGTREFCLKN_112

PCI2CLK_P
PCI2CLK_N

J1   MGTRXP0_112   MGTTXP0_112   H2   MGTTX20_P   CG21   PCIETX2_P
K1   MGTRXN0_112   MGTTXN0_112   J2   MGTTX20_N   0.1uF CG22 PCIETX2_N
M1   MGTRXP1_112   MGTTXP1_112   N2   MGTTX21_P   CG23   0.1uF PCIETX3_N
L1   MGTRXN1_112   MGTTXN1_112   M2   MGTTX21_N   0.1uF CG24 PCIETX3_P
                                                               0.1uF
PCIERX2_P
PCIERX2_N
PCIERX3_N
PCIERX3_P

T4   MGTREFCLKP_114
T3   MGTREFCLKN_114

R1   MGTRXP0_114   MGTTXP0_114   P2
T1   MGTRXN0_114   MGTTXN0_114   R2
V1   MGTRXP1_114   MGTTXP1_114   W2
U1   MGTRXN1_114   MGTTXN1_114   V2

PCIECLK_P   D4   MGTREFCLKP_116
PCIECLK_N   D3   MGTREFCLKN_116

C1   MGTRXP0_116   MGTTXP0_116   B2   MGTTX30_P   CG25   PCIETX0_P
D1   MGTRXN0_116   MGTTXN0_116   C2   MGTTX30_N   0.1uF CG26 PCIETX0_N
F1   MGTRXP1_116   MGTTXP1_116   G2   MGTTX31_P   CG27   0.1uF PCIETX1_N
E1   MGTRXN1_116   MGTTXN1_116   F2   MGTTX31_N   0.1uF CG28 PCIETX1_P
                                                               0.1uF
PCIERX0_P
PCIERX0_N
PCIERX1_N
PCIERX1_P

MGTCLK_P   AB4   MGTREFCLKP_118
MGTCLK_N   AB3   MGTREFCLKN_118

TRGLIN1_P   AA1   MGTRXP0_118   MGTTXP0_118   Y2   TRGLOUTG_N
TRGLIN1_N   AB1   MGTRXN0_118   MGTTXN0_118   AA2 TRGLOUTG_P
            AD1   MGTRXP1_118   MGTTXN1_118   AD2
            AC1   MGTRXN1_118   MGTTXP1_118   AE2

XC5VLX30T-1FF665C

U20F

BANK 12

XC5VLX30T-1FF665C

Pin label	Pin
IO_L0P_12	Y6
IO_L0N_12	Y5
IO_L1P_12	G6
IO_L1N_12	H6
IO_L2P_12	Y4
IO_L2N_12	W4
IO_L3P_12	G5
IO_L3N_12	F5
IO_L4P_12	W5
IO_L4N_VREF_12	V6
IO_L5P_12	G4
IO_L5N_12	H4
IO_L6P_12	V6
IO_L6N_12	V7
IO_L7P_12	J5
IO_L7N_12	J6
IO_L8P_CC_12	U7
IO_L8N_CC_12	T8
IO_L9P_CC_12	K5
IO_L9N_CC_12	L5
IO_L10P_CC_12	K6
IO_L10N_CC_12	K7
IO_L11P_CC_12	U6
IO_L11N_CC_12	U5
IO_L12P_VRN_12	K8
IO_L12N_VRP_12	L7
IO_L13P_12	T5
IO_L13N_12	R5
IO_L14P_12	M7
IO_L14N_VREF_12	L8
IO_L15P_12	R6
IO_L15N_12	T7
IO_L16P_12	P6
IO_L16N_12	N6
IO_L17P_12	M6
IO_L17N_12	N7
IO_L18P_12	N8
IO_L18N_12	P8
IO_L19P_12	R8
IO_L19N_12	R7

U20K

BANK 18

XC5VLX30T-1FF665C

Pin label	Pin	Signal
IO_L0P_18	AF12	TESTS25
IO_L0N_18	AE12	TESTS26
IO_L1P_18	V8	
IO_L1N_18	V9	
IO_L2P_18	AE11	TESTS24
IO_L2N_18	AD11	TESTS20
IO_L3P_18	W9	TESTS11
IO_L3N_18	W8	
IO_L4P_18	AD10	TESTS19
IO_L4N_VREF_18	AE10	TESTS23
IO_L5P_18	Y7	
IO_L5N_18	Y8	
IO_L6P_18	AF9	TESTS22
IO_L6N_18	AF10	TESTS12
IO_L7P_18	AA7	
IO_L7N_18	AA8	
IO_L8P_CC_18	AF7	TESTS15
IO_L8N_CC_18	AF8	TESTS21
IO_L9P_CC_18	AA5	
IO_L9N_CC_18	AB5	
IO_L10P_CC_18	AB6	
IO_L10N_CC_18	AB7	TESTS9
IO_L11P_CC_18	AE8	TESTS6
IO_L11N_CC_18	AE7	TESTS4
IO_L12P_VRN_18	AC6	
IO_L12N_VRP_18	AD5	TESTS14
IO_L13P_18	AE6	
IO_L13N_18	AF5	
IO_L14P_18	AE5	
IO_L14N_VREF_18	AD4	
IO_L15P_18	AF4	
IO_L15N_18	AF3	TESTS13
IO_L16P_18	AD6	
IO_L16N_18	AC7	TESTS8
IO_L17P_18	AC8	TESTS16
IO_L17N_18	AD8	TESTS18
IO_L18P_18	AD9	TESTS10
IO_L18N_18	AC9	TESTS17
IO_L19P_18	AB9	
IO_L19N_18	AA9	

D1
RS21A 1.02K LEDV0
LED2
D2
RS22A 1.02K LEDV1
LED2
D3
RS23A 1.02K LEDM1
LED2
D4
RS24A 1.02K LEDM2
LED2
GND

U20J

BANK 17

XC5VLX30T-1FF665C

Pin label	Pin	Signal
IO_L0P_17	AC26	TEST1_P
IO_L0N_17	AD26	TEST1_N
IO_L1P_17	AD25	TEST7_P
IO_L1N_17	AD24	TEST7_N
IO_L2P_17	AE25	TEST8_P
IO_L2N_17	AE26	TEST8_N
IO_L3P_17	AF25	TEST9_P
IO_L3N_17	AF24	TEST9_N
IO_L4P_17	AF23	TEST2_P
IO_L4N_VREF_17	AE23	TEST2_N
IO_L5P_17	AE22	TEST3_P
IO_L5N_17	AD23	TEST3_N
IO_L6P_17	AC24	
IO_L6N_17	AC23	
IO_L7P_17	AC22	
IO_L7N_17	AB22	
IO_L8P_CC_17	AF22	TEST10_P
IO_L8N_CC_17	AE21	TEST10_N
IO_L9P_CC_17	AE20	TEST4_P
IO_L9N_CC_17	AD20	TEST4_N
IO_L10P_CC_17	AD19	TEST12_P
IO_L10N_CC_17	AC21	TEST12_N
IO_L11P_CC_17	AD21	TEST11_P
IO_L11N_CC_17	AF19	TEST11_N
IO_L12P_VRN_17	AF18	TEST13_P
IO_L12N_VRP_17	AE18	TEST13_N
IO_L13P_17	AD18	
IO_L13N_17	AE17	
IO_L14P_17	AF17	TEST5_P
IO_L14N_VREF_17	AE16	TEST5_N
IO_L15P_17	AD16	TEST14_N
IO_L15N_17	AD15	TEST14_P
IO_L16P_17	AE15	
IO_L16N_17	AF15	TEST15_N
IO_L17P_17	AF14	TEST15_P
IO_L17N_17	AF13	TEST6_P
IO_L18P_17	AE13	TEST6_N
IO_L18N_17	AD13	TEST16_P
IO_L19P_17	AD14	TEST16_N
IO_L19N_17		

U20D

BANK 4

XC5VLX30T-1FF665C

Pin label	Pin	Signal
IO_L0P_GC_D15_4	Y21	
IO_L0N_GC_D14_4	AA20	
IO_L1P_GC_D13_4	AB10	TESTS1
IO_L1N_GC_D12_4	AB11	
IO_L2P_GC_D11_4	AB21	TESTS32
IO_L2N_GC_D10_4	AB20	TESTS31
IO_L3P_GC_D9_4	AC11	TESTS2
IO_L3N_GC_D8_4	AB12	TESTS3
IO_L4P_GC_4	AB19	TESTS29
IO_L4N_GC_VREF_4	AC19	TESTS30
IO_L5P_GC_4	AC12	
IO_L5N_GC_4	AC13	
IO_L6P_GC_4	AC18	
IO_L6N_GC_4	AB17	
IO_L7P_GC_VRN_4	AB14	
IO_L7N_GC_VRP_4	AC14	TESTS5
IO_L8P_GC_CC_4	AC17	TESTS28
IO_L8N_CC_4	AB16	
IO_L9P_CC_GC_4	AB15	TESTS7
IO_L9N_CC_GC_4	AC16	TESTS27

U20A

BANK 1

XC5VLX30T-1FF665C

Pin label	Pin
IO_L0P_A19_1	G15
IO_L0N_A18_1	G16
IO_L1P_A17_1	H13
IO_L1N_A16_1	G14
IO_L2P_A15_D31_1	G17
IO_L2N_A14_D30_1	F17
IO_L3P_A13_D29_1	F15
IO_L3N_A12_D28_1	F14
IO_L4P_A11_D27_1	F18
IO_L4N_VREF_A10_D26_1	G19
IO_L5P_A9_D25_1	G12
IO_L5N_A8_D24_1	H18
IO_L6P_A7_D23_1	H19
IO_L6N_A6_D22_1	G11
IO_L7P_A5_D21_1	G20
IO_L7N_A4_D20_1	J21
IO_L8P_CC_A3_D19_1	J20
IO_L8N_CC_A2_D18_1	G10
IO_L9P_CC_A1_D17_1	H9
IO_L9N_CC_A0_D16_1	

V33
CM1
T520A
10uF
CMI1
0.22uF
CMI2
0.22uF
GND

USFM

Pin		Pin	
SFMSI 1	SI	SO	8 SFMSO
SFMSCK 2	SCK	GND	7 GND
SFMRESET_N 3	Reset_N	VCC	6 V33
SFMCS_N 4	CS_N	WP_N	5 SFMWP_N

AT45DB021D-SSH

RUSFM
3.24K
GND

To protect from accidental memory write

U20I — BANK 16 — XC5VLX30T-1FF665C

Pin name	Pad
IO_L0P_16	H7
IO_L0N_16	G7
IO_L1P_16	F7
IO_L1N_16	F8
IO_L2P_16	F9
IO_L2N_16	G9
IO_L3P_16	H8
IO_L3N_16	J8
IO_L4P_16	A9
IO_L4N_VREF_16	A8
IO_L5P_16	E8
IO_L5N_16	E7
IO_L6P_16	B9
IO_L6N_16	C8
IO_L7P_16	E6
IO_L7N_16	D6
IO_L8P_CC_16	C9
IO_L8N_CC_16	D8
IO_L9P_CC_16	C7
IO_L9N_CC_16	A7
IO_L10P_CC_16	B7 — FPGAOUT1
IO_L10N_CC_16	D9 — FPGAOUT1
IO_L11P_CC_16	D10 — FPGAOUT2
IO_L11N_CC_16	B6 — FPGAOUT2
IO_L12P_VRN_16	A5
IO_L12N_VRP_16	B10 — FPGAOUT3
IO_L13P_16	A10 — FPGAOUT3
IO_L13N_16	A4
IO_L14P_16	A3
IO_L14N_VREF_16	B11
IO_L15P_16	A12
IO_L15N_16	B4
IO_L16P_16	B5
IO_L16N_16	B12
IO_L17P_16	C12
IO_L17N_16	D5
IO_L18P_16	E5
IO_L18N_16	C11 — FPGAOUT4_P
IO_L19P_16	D11 — FPGAOUT4_N
IO_L19N_16	

U20H — BANK 15 — XC5VLX30T-1FF665C

Pin name	Pad	Net
IO_L0P_15	C13	FPGAPIN0_P
IO_L0N_15	C14	FPGAPIN0_N
IO_L1P_15	B14	FPGAPIN1_P
IO_L1N_15	A13	FPGAPIN1_N
IO_L2P_15	A14	FPGAPIN2_P
IO_L2N_15	A15	FPGAPIN2_N
IO_L3P_15	B15	
IO_L3N_15	C16	
IO_L4P_15	B16	FPGAPIN3_P
IO_L4N_VREF_15	C17	FPGAPIN3_N
IO_L5P_15	B17	FPGAPIN4_P
IO_L5N_15	A17	FPGAPIN4_N
IO_L6P_15	A18	FPGAPIN5_P
IO_L6N_15	A19	FPGAPIN5_N
IO_L7P_15	B19	
IO_L7N_15	C18	
IO_L8P_CC_15	A20	AUXIN5_P
IO_L8N_CC_15	B20	AUXIN5_N
IO_L9P_CC_15	C19	FPGAPIN6_P
IO_L9N_CC_15	D19	FPGAPIN6_N
IO_L10P_CC_15	D21	
IO_L10N_CC_15	D20	
IO_L11P_CC_15	B21	
IO_L11N_CC_15	C21	
IO_L12P_VRN_15	D23	
IO_L12N_VRP_15	C22	
IO_L13P_15	B22	SYNCBR_P
IO_L13N_15	A22	SYNCBR_N
IO_L14P_15	A23	AUXOUT1_P
IO_L14N_VREF_15	A24	AUXOUT1_N
IO_L15P_15	B24	AUXIN1_P
IO_L15N_15	C23	AUXIN1_N
IO_L16P_15	D24	CLKSYNCB_P
IO_L16N_15	C24	CLKSYNCB_N
IO_L17P_15	B25	SYNCAR_P
IO_L17N_15	A25	SYNCAR_N
IO_L18P_15	B26	CLKSYNCA_P
IO_L18N_15	C26	CLKSYNCA_N
IO_L19P_15	D26	AUXOUT5_P
IO_L19N_15	D25	AUXOUT5_N

U20C — BANK 3 — XC5VLX30T-1FF665C

Pin name	Pad	Net
IO_L0P_CC_GC_3	D15	CLK250C_P
IO_L0N_CC_GC_3	E15	CLK250C_N
IO_L1P_CC_GC_3	D16	
IO_L1N_CC_GC_3	E16	
IO_L2P_GC_VRN_3	D14	
IO_L2N_GC_VRP_3	D13	
IO_L3P_GC_3	E17	CLK3125_P
IO_L3N_GC_3	D18	CLK3125_N
IO_L4P_GC_3	E13	
IO_L4N_GC_VREF_3	E12	
IO_L5P_GC_3	F18	
IO_L5N_GC_3	F19	
IO_L6P_GC_3	F12	
IO_L6N_GC_3	F11	
IO_L7P_GC_3	E20	
IO_L7N_GC_3	E21	
IO_L8P_GC_3	E10	
IO_L8N_GC_3	F10	
IO_L9P_GC_3	F20	
IO_L9N_GC_3	G21	

U20E — BANK 11 — XC5VLX30T-1FF665C

Pin name	Pad	Net
IO_L0P_11	E26	MODINTA
IO_L0N_11	E25	TIAMSEL
IO_L1P_11	F25	MODPRSTA
IO_L1N_11	G26	TCKPROM
IO_L2P_11	H26	TMSPROM
IO_L2N_11	G25	TDIPROM
IO_L3P_11	F24	FIBERICK
IO_L3N_11	G24	
IO_L4P_11	E23	AD9510SDIO
IO_L4N_VREF_11	E22	TIBMSEL
IO_L5P_11	F23	FIBERIDATA
IO_L5N_11	F22	FIBERRST
IO_L6P_11	G22	AD9510CS_N
IO_L6N_11	H22	
IO_L7P_11	H23	
IO_L7N_11	J23	
IO_L8P_CC_11	J21	
IO_L8N_CC_11	K21	
IO_L9P_CC_11	K22	
IO_L9N_CC_11	K23	
IO_L10P_CC_SM15P_11	J23	
IO_L10N_CC_SM15N_11	L22	
IO_L11P_CC_SM14P_11	M21	
IO_L11N_CC_SM14N_11	N21	
IO_L12P_VRN_11	H24	
IO_L12N_VRP_11	J24	
IO_L13P_11	J25	
IO_L13N_11	J26	TDOPROM
IO_L14P_11	K26	
IO_L14N_VREF_11	L25	
IO_L15P_SM13P_11	L24	
IO_L15N_SM13N_11	K25	
IO_L16P_SM12P_11	N26	CLKRESET
IO_L16N_SM12N_11	M26	
IO_L17P_SM11P_11	M25	AD9510SCLK
IO_L17N_SM11N_11	M24	
IO_L18P_SM10P_11	N24	AD9510SDO
IO_L18N_SM10N_11	N23	
IO_L19P_SM9P_11	N22	
IO_L19N_SM9N_11	M22	

U20G — BANK 13 — XC5VLX30T-1FF665C

Pin name	Pad	Net
IO_L0P_SM8P_13	P26	SFMSCK
IO_L0N_SM8N_13	R26	SFMCS_N
IO_L1P_SM7P_13	P25	SFMSI
IO_L1N_SM7N_13	R25	
IO_L2P_SM6P_13	T24	
IO_L2N_SM6N_13	T23	
IO_L3P_SM5P_13	R23	
IO_L3N_SM5N_13	R22	
IO_L4P_13	U26	SFMWP_N
IO_L4N_VREF_13	V26	LEDM1
IO_L5P_SM4P_13	U25	LEDV1
IO_L5N_SM4N_13	T25	SFMSO
IO_L6P_SM3P_13	T24	
IO_L6N_SM3N_13	T23	
IO_L7P_SM2P_13	U24	LEDV0
IO_L7N_SM2N_13	V24	LEDM2
IO_L8P_CC_SM1P_13	W26	CLKSEL3
IO_L8N_CC_SM1N_13	W25	CLKSEL4
IO_L9P_CC_SM0P_13	W24	
IO_L9N_CC_SM0N_13	V23	
IO_L10P_CC_13	AA22	
IO_L10N_CC_13	Y22	
IO_L11P_CC_13	Y23	
IO_L11N_CC_13	W23	
IO_L12P_VRN_13	Y26	FPGASEL5
IO_L12N_VRP_13	Y25	FPGASEL6
IO_L13P_13	AA25	
IO_L13N_13	AB26	FPGASEL7
IO_L14P_13	AB25	FPGASEL8
IO_L14N_VREF_13	AA24	
IO_L15P_13	AB24	
IO_L15N_13	AA23	
IO_L16P_13	P21	
IO_L16N_13	R21	
IO_L17P_13	T22	
IO_L17N_13	U22	
IO_L18P_13	U21	
IO_L18N_13	V22	
IO_L19P_13	V21	
IO_L19N_13	W21	

Appendix B: TID fabrication drawing:
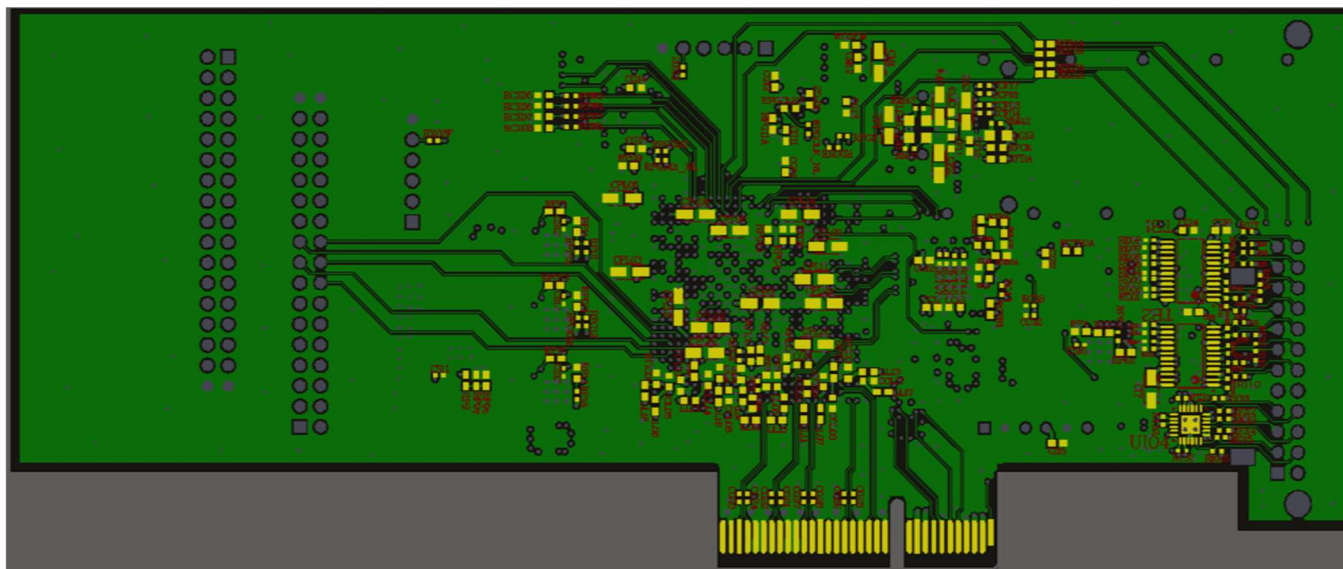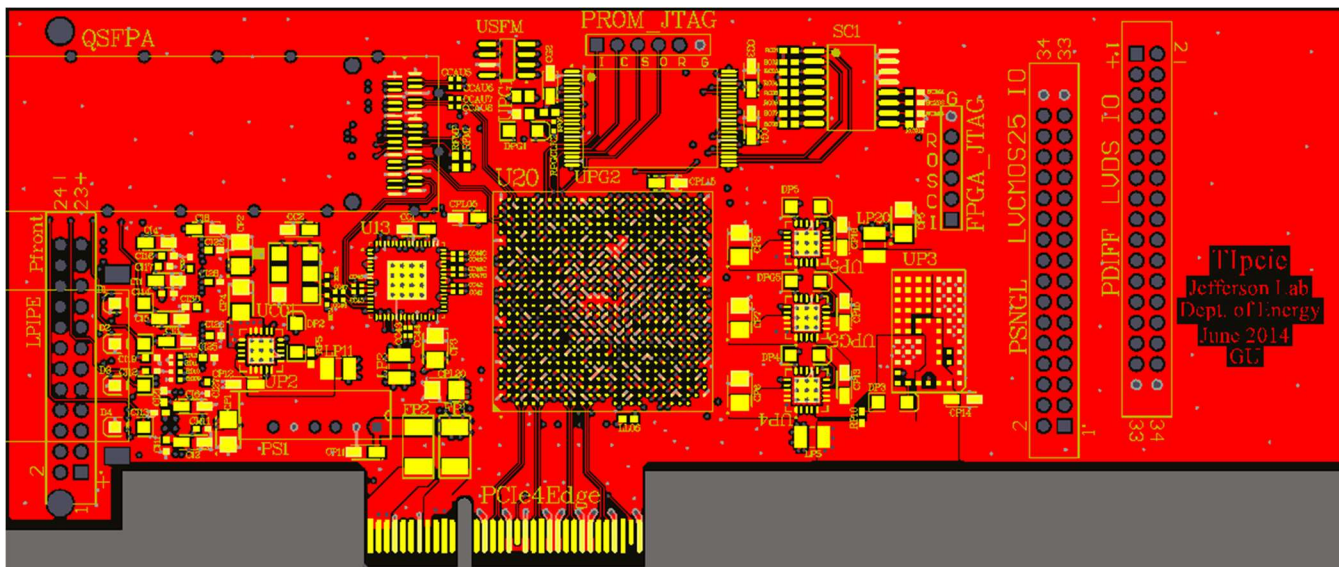
Appendix C: Bill of materials:

Comment	Description	Designator	Quantity
T520A	Solid Tantalum Capacitor,	CC1, CC2, CF1, CF2, CF3, CF4, CG1, CG2, CG3, CI1, CI2, CI3, CI4, CI5, CI6, CI7, CI8, CM1, CP11, CP12, CP13, CP14, CP15, CP16, CPL01, CPL02, CPL03, CPL04, CPL05, CPL06, CPL07, CPL08, CPL09, CPL10, CPL11, CPL12, CPL13, CPL14, CPL15	39
0.22uF	Ceramic Chip Capacitor - Standard	CC11, CC12, CC13, CC14, CC15, CCL01, CF11, CF12, CF13, CG11, CG12, CG13, CG14, CG15, CG16, CI11, CI12, CI13, CI14, CI15, CI16, CI17, CI18, CI19, CI20, CI21, CI22, CI23, CI24, CI25, CI26, CI27, CI28, CI29, CI30, CM11, CM12, CCL02, CCL03, CCL04, CCL05, CCL06, CCL07, CCL08, CCL09, CCL11, CCL12, CCL13, CCL14, CCL15, CCL16, CCL17, CCL18	53
0.01uF	Ceramic Chip Capacitor - Standard	CC32, CS11, CS12, CS13, CS14, CS15	6
0.1uF	Ceramic Chip Capacitor - Standard	CC41, CC42, CC43, CC44, CC45, CC45C, CC46, CC46C, CC47, CC47C, CC48, CC48C, CCAU5, CCAU6, CCAU7, CCAU8, CCF03, CCF04, CCF17, CCF18, CG21, CG22, CG23, CG24, CG25, CG26, CG27, CG28	28
T520B	Solid Tantalum Chip Capacitor, Standard T520 Series	CP1, CP2, CP3, CP4, CP5, CP6, CP7, CP8, CPL20	9
LED2	Typical RED GaAs LED	D1, D2, D3, D4, DP2, DP3, DP4, DP5, DPG1, DPG5	10
5.0A	FUSE 5A FAST BLOW NANO 2 SMD	FP1, FP2	2
Header 6	Header, 6-Pin, 0.1" pitch	FPGA_JTAG, PROM_JTAG	2
1.0uH	INDUCTOR 1.0UH 300MA 20% 0805	LG11, LG12, LG13	3
HF_0.5A	Inductor	LL01, LL02, LL03, LL04, LL05, LL06, LL07, LL08, LL09, LL11, LL12, LL13, LL14, LL15, LL16, LL17, LL18	17
HC_3A	Inductor	LP2, LP5, LP11, LP20	4
LIGHTPIPE		LPIPE	1
PCIe4		PCIe4Edge	1
09 18 134 9621	Flat Cable Connector (IDC), PCB Transition Connector, 2 Rows, Solder Pin, 34 Contacts	PDIFF, PSNGL	2
09 18 124 9621	Flat Cable Connector (IDC), PCB Transition Connector, 2 Rows, Solder Pin, 24 Contacts	Pfront	1
NMH1205SC		PS1	1
510	RES 510 OHM 1/16W 1% 0402 SMD	PSW5, PSW6, PSW7, PSW8, RDONE, RIO11, RIO12, RIO13, RIO14, RIO15, RIO16, RP12, RP12x, RP16, RPG12, RPG12x	16
QSFP	QSFP 38-pin connector	QSFPA	1
10K	RES 10K OHM 1/16W 1% 0402 SMD	R110, R111, R112, R113, R114, R115, R116, R117, R118, R119, R120, R121, R122, R123	14
3.24K	Resistor, chip 3.24K	RC01, RC02, RC03, RC04, RCT50A, RCTR1, RCTR2, RCTR3, RCX05, RCX06, RCX07, RCX08, RP5, RP9, RP9z, RP10, RPG2, RPG4, RPG5, RPG6, RPG9, RPG10, RPG11A, RPGCLK3, RUSFM	25
1.02K	Resistor, chip 1.02K	RC05, RC06, RC07, RC08, RCTR2A, RCTR3A, RCTR3B, RCX01, RCX02, RCX03, RCX04, RFCK, RFDA, RFMI, RFMP, RP4, RP6, RP6x, RP9y, RP11, RP13, RP14, RP15, RPG11, RPG13, RS21A, RS22A, RS23A, RS24A	29
50	RES 50 OHM 1/16W 1% 0402 SMD	RC31, RC32, RC33, RT5, RVP0, RX02, RX02F	7
100	RES 100 OHM 1/16W 1% 0402 SMD	RC43, RD01, RD02, RD03, RD04, RD05, RD06, RD07, RD08, RD09, RD10, RD11, RD12, RD13, RD14, RD15, RD16, RM41, RM42, RM43, RT31, RT32, RT33, RT34, RT35, RT41, RT42, RT43, RT44, RT47, RT48	31
150	RES 150 OHM 1/16W 1% 0402 SMD	RCT44, RCT45, RCT46, RCT47, RP6y, RPG8	6
0	Chip Resistor, 0 Ohm, 0402 Size, 0.063 W	RPGCLK2, RPGD01_NL, RPGD02	3

SW DIP-8	DIP Switch, 8 Position, SPST	SC1	1
max9602EUG	Differential discriminator	TB1, TB2	2
AD9510BCP	AD9510 clock	U13	1
XC5VLX30T-1FF665C	Xilinx Virtex-5 LXT Platform FPGA, 665-Ball FFBGA, Speed Grade 1, 360 User I/Os, Commercial Grade	U20	1
CCPD-034	CCPD-034, 250MHz low jitter	UC01	1
MC100EP91M	On Semiconductor, Any level positive to ECL translator	UIO4	1
TPS74401RGWT	IC LDO REG 3.0A W/SS 20-VQFN	UP2, UP4, UP5, UPG5	4
LTM4604EV	LTM4604AEV	UP3	1
kc2520-33MHz	Miniature Oscillator, 33MHz	UPG1	1
XCF32PVO48C	XCF00P Series, Platform Flash In-System Programmable Configuration 1.8V PROM, 48-Pin TSSOP, 32-Megabit, Commercial Grade	UPG2	1
AT45DB021D-SSH	Serial flash memory	USFM	1

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.

TIpcie: Trigger Interface with PCI express x4 format.

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: TIpcie 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 (block level, this number could be 1) of triggers. Each block has two block headers, one block trailer, possible filler words, 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 64-bit alignment

```
Block Header#1:
  bit(31:27): 10000, block header indicator;
  Bit(26:22): BoardID;
  Bit(21:18): 0000, ID for TI board;
  Bit(17:8): block number, lower ten bits;
  Bit(7:0): block size (or block level, as set by 0x84 trigger command);

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;
  Bit(21:0): 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 (64-bit aligned):
    Bit(31:27): 1111,1;
    Bit(26:22): BoardID;
    Bit(21:0): block number;

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

The data is written to DMA by TIpcie through PCI express. The data are written in the DMA as a ring buffer with 4 Kbyte cells. Each readout data may use 1 or 2 cells, but one cell can have only one readout block. In each buffer cell, the data is further divided into subcells. Each subcell has MPS (for PCI express, the MPS can be 128 byte, 256 byte, … 4 Kbyte) byte of data. Here is the data structure in one cell (in 32-bit word):

Status word:

Bit(31:28): free timer (11:8);

Bit(27:16): 64-bit word count of remaining data

Bit(15:8): free timer(7:0);

Bit(7:0): status of some internal signals, right now, they are: DataIn(34), DataIn(35), DataIn(70), DataIn(71), WriteEn, CountLoad, ReadyForRead, WriteBusy.

Data word #1:   -- readout data word

……

Data word #n :   -- last word in the TLP packet. If the remaining data is more than MPS, n = (MPS/4)-2. To read more data, go to the next subcell. If the remaining data is less than the MPS, the Data word#n is the end of the readout data. To read more data, go to next cell (4 Kbyte) for more events.

……

Dataword # (MPS/4-1) -- 0x00000000, filler word for the packet

The above ring buffer is for 3-header readout. For 4-header readout, there will be no status word. The software has to look for data word count to determine the end of the readout.

Appendix F: Document revision history:

Feb. 2, 2015: First release, based on TI.docx;

Nov. 25, 2015: updated the 0x04, 0x104, 0x108, 0x138 registers

Dec. 8, 2015: updated the 0x34, 0x4c, 0x94, and 0xc0 registers

Jan. 21, 2016: updated the 0x1c and 0x38 registers.

May 4, 2016: updated the 0x28 register with bit(4) and bit(5).

May 31, 2016: Updated the register 0x138 and 0x100 (bit 21 and bit22).

June 2, 2016: Updated the BAR#1 for the optic transceiver I2C engine, and BAR#0 register 0x1C.

June 14, 2016: Updated the 0xEC register to enable the SyncResetRequst

Apr. 13, 2018: Added bit#20 for reg 0x54, and assign 0xB4 for current DMA physical address.