



Nuclear Physics Division  
*Fast Electronics Group*

**Description and Technical Information for the TIpicieUS (PCI  
Express Trigger Interface with Xilinx UltraScale+ FPGA)**

J. William Gu ([jgu@jlab.org](mailto:jgu@jlab.org))

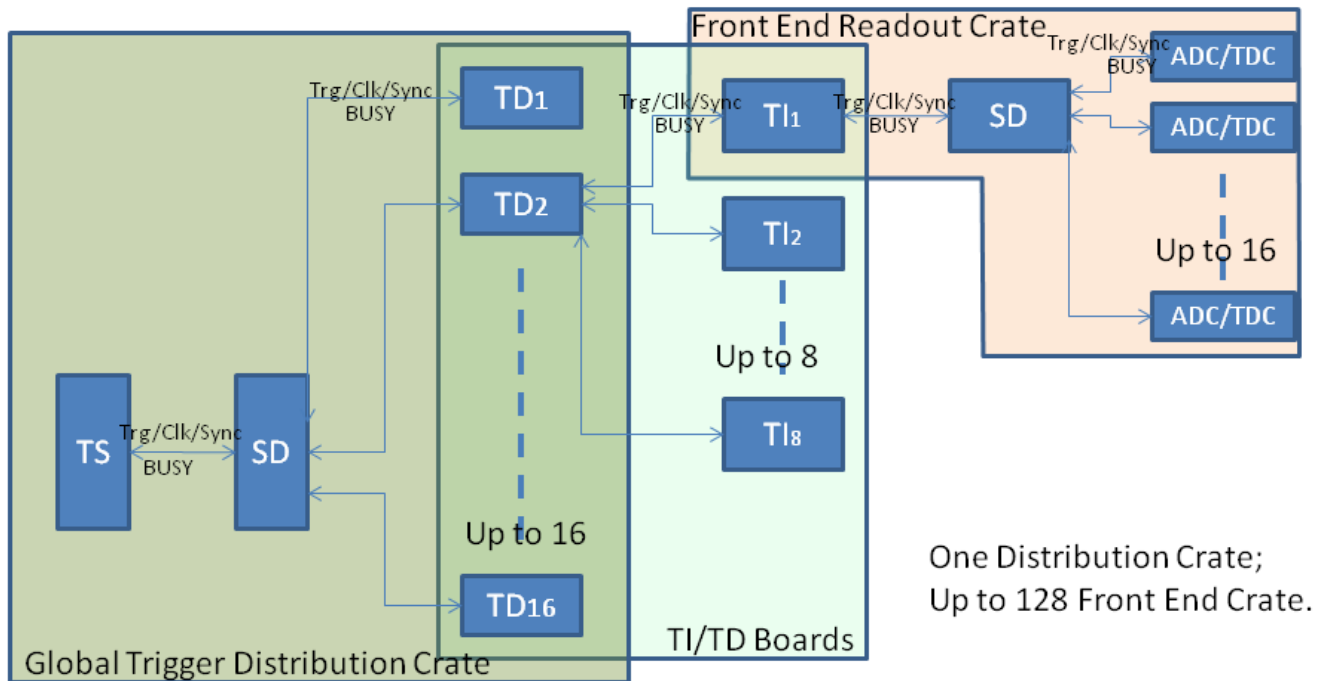
First release: Nov. 2, 2020

## Table of Contents

<b>Section</b>	<b>Title</b>	<b>Page</b>
1	Introduction	3
2	Purpose of TIpicieUS Module	4
3	Function Description	5
4	Specifications	10
5	TIpicieUS operation	11
6	Programming Requirements	13
7	Connector Pin out tables	25
8	TIpicieUS Operation examples	28
9	Citation	29
Appendix A	Schematics	30
Appendix B	Fabrication Drawing	43
Appendix C	Bill of Materials	44
Appendix D	Glossary	47
Appendix E	TIpicieUS data format	48
Appendix F	Revision history	50

# 1 Introduction

The Trigger Interface (TI) module was designed for the Jefferson Lab 12GeV upgrade, mainly for Hall-D and Hall-B, with other experimental Halls compatibility. The VXS TI boards are located in data acquisition frontend crates, and are responsible for providing a low-jitter system clock, system 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 PCI express TI (Tlpcie) board is based on the VXS TI board. Instead of VME64x, the Tlpcie board is plugged into a PCI express slot in a computer, communicates through PCI express. 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 TlpcieV5 was produced in 2014 based on the Xilinx XC5VLX30T FPGA, the same as the VXS/VME TI board, with just one QSFP optic link. The TlpcieV5 was successfully used in various experiments and test stands, but with some limitations, which also motivated the TlpcieUS design. The major limitations are, first, the Virtex-5 PCI express IP core is not compatible with the Intel CPU (Xeon) PCI express, so that the TlpcieV5 can only fit in some certain computers/servers, not the direct link with the Xeon CPU; second, there is only one optic transceiver port, which severely limited its functionality and usefulness.

The TlpcieUS board uses the Xilinx UltraScalePlus Kintex FPGA, which is four generations newer than the virtex-5 FPGA, and capable of PCI express gen3. Two QSFP optic transceiver ports are implemented on the TlpcieUS, and still in the low profile and half-size PCI express format. The TlpcieUS is designed with the full functionalities of the VXS/VME TI with two optic transceivers, which corresponding to VME TI fiber#1 and fiber#5.

## 2 Purpose of the TIpicieUS module

TIpicieUS board can be plugged in any PCI express x8 (8-lanes) compatible slots. It can connect to a Trigger Distribution (TD) module in the global trigger distribution system and connect to a Trigger Interface (TI) in master mode configuration, sit in standalone mode, drive another two TI boards as a master, or daisy chaining the Trigger distribution (using fiber#1 as slave TI, and fiber#5 as a master TI). This is implemented using two quad-channels full-duplex fiber links (QSFP). Each QSFP link provides a gigabit trigger link, a 250MHz clock distribution, a synchronization link to phase lock the trigger and initialize the system, and a dedicated link to measure the fiber loopback delays. The trigger link uses a reference clock derived from the 250MHz system (pipeline) clock allowing a trigger word to be distributed every four global system clock cycles. Figure 2 is a picture of the TIpicieUS board with a low profile bracket.



Figure 2 TIpicieUS PCB picture

The TIpicieUS can also perform simple Trigger Supervisor functions, as a TI Master (TM). In this case, it can take inputs from its front panel (, which connects to the 40-pin on-board connector at the very right edge of Figure2) and generate trigger/clock like TS, and it can send the trigger, clock and SYNC to the front panel fibers like a TD.

The TIpicieUS FPGA firmware is compatible with the master mode and the standard (slave) mode. The firmware senses the TI clock settings. If the TIpicieUS is set to master mode or standalone mode, the TIpicieUS uses the on-board oscillator or the copper input as clock source. The TIpicieUS generates triggers either by random and periodic sequences internally or by the front panel copper inputs. The TIpicieUS generates SYNC signal, and set the fiber measurement pair to loopback in FPGA (, which is different from the TD, where the loopback is at the transceiver via a pair of resistors). The TIpicieUS sends the trigger/Clock/Sync to both QSFP optic transceivers.

If the TIpicieUS is set to daisy chain (FANOUT) mode, it uses Fiber port#1 clock as clock source and fan out the clock to fiber#5. It uses Fiber#1 trigger input, and fan-out the trigger to fiber#5. It uses the Fiber#1 SYNC input, and fan-out the SYNC to fiber#5. It measures the fiber#1 latency, and set fiber#5 fiber measurement to loopback mode in FPGA, so that the TI modules connects to the Fiber#5 can do the fiber latency measurement.

If the TIpicieUS is set to slave mode, it uses clock from fiber#1 or Fiber#5 depending on the DAQ system setting. It also uses the same fiber for trigger, SYNC and fiber measurement. It is customary to use fiber#1 for global trigger (usually connects to TS with TD), and fiber#5 for subsystem trigger (usually connects to TI master).

Some registers are valid on the TI master only, as these registers are specific to the TS function.

The TIpicieUS has two generic 68-pin high density connectors with 64-channel LVDS input/output directly connected to the FPGA. The TIpicieUS also has one 40-pin connector to mimic the 34-pin connector on the VME TI boards. This 40-pin connector connects directly to the second PCI express front panel, and pin#1-34 is compatible with the VME TI 34-pin connector. The pin#35/36 and pin#37/38 are two extra any-level differential signal inputs, and pin#39/40 are LVPECL33 clock output directly from the clock driver (AD9510), the frequency is (250/n) MHz, where n=1, 2, ..., 8 set by the FPGA firmware.

### 3 TIpicieUS Functional Descriptions

#### 3.1 General description

Figure 3 shows the block diagram of the TIpicieUS module, indicating the major components used in the design. The AFBR-79EIDZ is the quad-channel (4 Rx, 4 Tx) fiber link that the TI fans out/receives a low-jitter (<3ps RMS) 250MHz system 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 XCKU3P is used to encode/decode the trigger words at 16ns, to interface with the computer PCI express bridge/root master.

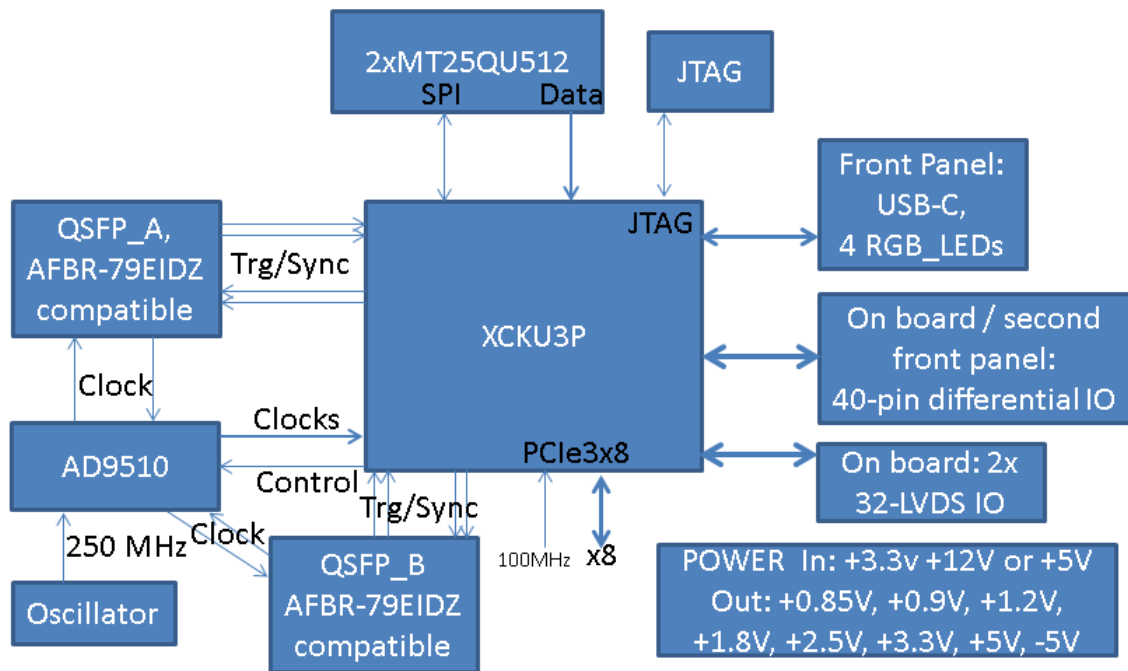


Figure 3 TIpicieUS block diagram

#### 3.2 Fiber links

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

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

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

The fourth pair (Tx/Rx) is looped back on the TIpcieUS FPGA when the TIpcieUS 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 TIpcieUS' major functions is the system (pipeline) clock distribution. There are five possible sources for the 250MHz clock: 'onboard oscillator', 'QSFP#A input from TD/TS', 'QSFP#B input from TD/TI master', 'copper input from front panel/on-board 40-pin connector', and 'recovered clock from trigger MGT link'.

When the TIpcieUS is in master mode, or standalone mode, the on board oscillator or the copper input is used. Or else the optic fiber input is used. There will be only one 250MHz clock running on the TIpcieUS. The clock source is selected by the FPGA using a 4:1 multiplexer for the first four clock sources, and the AD9510 clock driver SPI interface for the recovered clock. The AD9510 also generates slower clocks (31.25MHz, 41.667 MHz, 62.5 MHz, and 125 MHz).

The TIpcieUS has a dedicated LVPECL33 clock output to the 40-pin connector (or the second front panel). This clock is a low jitter clock directly from the AD9510, with a frequency of 250/n MHz, where n is an integer with the value of 1 to 8, and set by the FPGA.

### 3.4 Trigger distribution:

When the TIpcieUS is in master mode (or standalone mode), it can take the trigger from the front panel inputs and form a trigger word like the VXS/VME TI master. It can also combine with the software trigger (which includes periodic trigger with a frequency of ~16 Hz to 7 MHz and random trigger with an average of frequency of ~15 Hz to 460 KHz) and send to the MGT serialized mode (encoded) to the fiber. The trigger word is looped back to trigger the TIpcieUS itself. The TIpcieUS has two more any-level differential inputs than VXS/VME TI boards.

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

When the TIpcieUS is in daisy chain mode, it receives the trigger signal through QSFP#A, and decoded by the FPGA as a TI slave. It also send this trigger out to QSFP#B as a TI master.

### 3.5: Encoded Trigger Word

A 1.25Gbps serial link operating over the fiber is used for distributing a 16-bit 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 16-bit trigger words are decoded as following (TS → TD → TIpcieUS, or TI-master → TI-slave):

Trigger strobe word – generated by the TS (or TIpcieUS in master mode) in response to the acceptance of a level-1 trigger. Upon receipt, the TIpcieUS aligns the trigger and start to assemble trigger data (one 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 fine 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 use the trigger link fully, 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 (or TIpcie) 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: TS#n(2:0), when “000”: no trigger;

The data coming back to the TI-master or TD are divided 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 (TIpcieUS → TD, or TI → TI master 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				

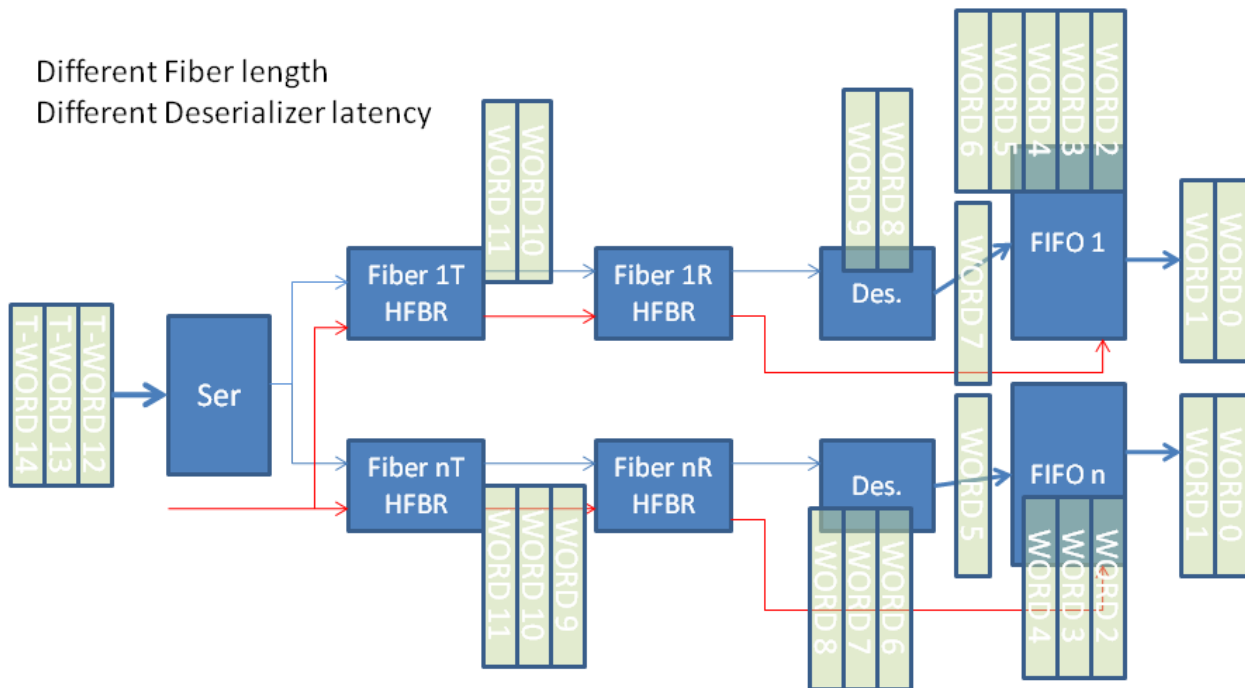
### 3.6: Fixed Latency SYNC

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), which is aligned with the 32.25 MHz and 41.667 MHz clocks, 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 as ‘1’. A command is sent between idle times by sending first a ‘0’ followed by the 4 bits 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 TI master). 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 timing is adjusted by the fiber latency 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 optic transceivers (Avago HFBR\_7924, Avago AFBR-79EIDZ, or other QSFP compatible). In the TIpcieUS, 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 transmit 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 phase aligned 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 the clock (phase re-alignment) reset. The time for the MGT reset is several milliseconds.

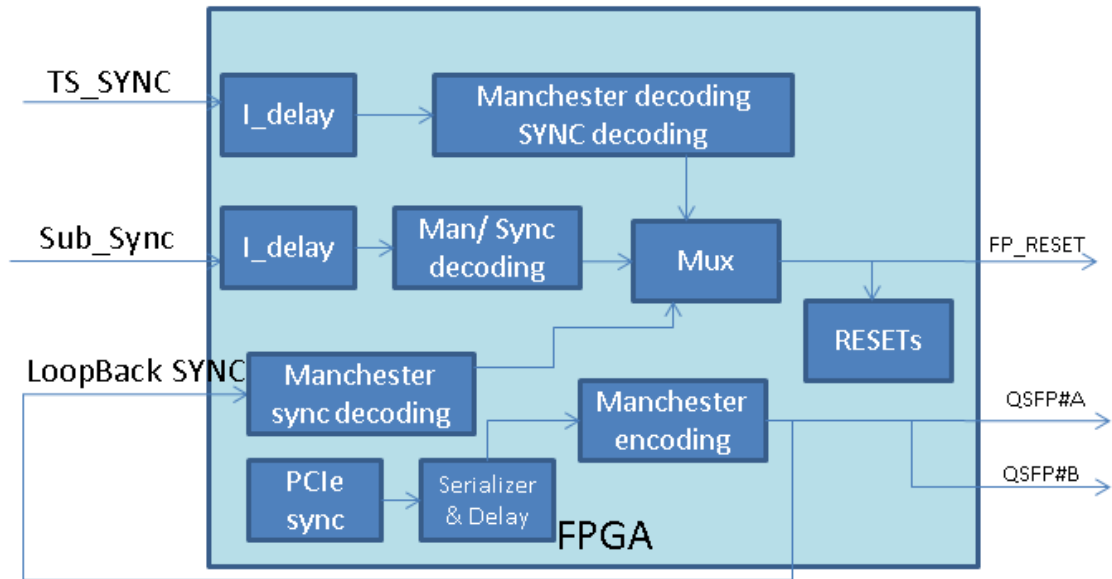


Figure 5 The block diagram of the SYNC distribution on the TIpicieUS board.

### 3.7 PCI express interface

The TIpicieUS board is a PCI express endpoint device. It is capable of PCIe GEN3x8 (8 Gbps, 8 lanes, up to 8 GB/s duplex). It is compatible with PCI express x8 or wider physical connector, though it is backward compatible with PCI GEN1x1 electrically.

The TIpicieUS has two 32-bit Base Address (BA) spaces. The BA0 has 8K bytes (Addr(12:0)), and divided this way:

Addr(12)	Addr(11)	Addr(10)	usage
1	0	x	JTAG path#A
	1	x	JTAG path#B
0	1	1	I2C for QSFPA#B
	1	0	I2C for QSFPA#A
0	0	1	reserved
0	0	0	1K byte 32-bit registers

BA#1 has 64K bytes, used for DMA transfers.

### 3.8: The FPGA flash memory programming.

Two Micron Tech. MT25QU512ABB flash memories, configured as SPIx8, are used to program the Xilinx XCKU3P FPGA. The flash memory can be programmed through the on-board JTAG connector via the FPGA

(indirect program) using Xilinx JTAG cable and Vivado design software. The flash memories can also be programmed by the PCI express directly (remotely) via the FPGA translation (FPGA PCI express to flash memory). The remote (through PCIe) firmware loading is under development. The 50 MHz on-board oscillator sets the FPGA programming speed (SPIx8 master mode).

### **3.9: Readout Synchronization**

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

First, the SyncEvent can be generated periodically by the TS (or TImaster). The last event of 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 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 TIpcieUS if the TIpcieUS 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 setting register offset 0x100, bit#20.

Third, it is also possible that the SyncEvent is generated by a front panel signal input (TS\_code). With certain TS\_code input combination, the SyncEvent can be generated (Event type table lookup).

The TS (or TImaster) will assert a short BUSY (3us) after generating the SyncEvent, and stop further triggers. The TI(pcie) will generate BUSY after receiving the SyncEvent,, 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 and up 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 period, the event number and trigger time will be continued.

## **4. Specification Sheet**

### **4.1 Mechanical**

- Single width, low profile, half-size PCI express x8 module. It can be plugged in any PCI express x8 compatible slot.
- Optionally, double width, low profile, half-size PCI express x8 module, if the user choose to bring the on-board 40-pin connector to the front panel (second single width front panel).

### **4.2 PCI express:**

- PCI express GEN3 x8 compatible, that is 8 lanes with 8 Gbps full duplex;
- Reference Clock at 100 MHz;
- JTAG on PCI express slot can be used by setting the jumpers on the board, but many computers do not support (supply) JTAG;

### **4.3 Onboard connectors:**

- Two 68-pin high density connectors, can be configured as 32-channels of LVDS or 64-channels of LVCMOS18, either input or output depending on the FPGA configuration.
- One 40-pin DIN connector, which can be connected to the second front panel for easier access.

#### **4.4 Front panel inputs and outputs:**

- Two QSFP connectors, which are equivalent to fiber#1 and fiber#5 of VXS/VME TI boards.
- One USB-C connector, for standalone power supply, and USB-C control (to be developed).

#### **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.
- CLOCK
  - 250MHz Clock <3ps RMS Jitter

#### **4.6 LED Indicators: Front Panel:**

- LED#1: RGB, Green: (farther away from the motherboard) FPGA programmed and the clock (DCM locked) is ready;
- LED#2: RGB, Green: PCI express activity
- LED#3: RGB, RED: Trigger\_1 is sent out / received;
- LED#4: RGB, RED: QSFP 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:**

- JTAG to FPGA indirect flash memory loading;
- PCIexpress to flash memory loading to be developed.

#### **4.8 Power requirements:**

- PCIexpress: +3.3V @ 3 Amps; +12V @ 1 Amp;
- Or USB-C: +5V @ 2 Amps.
- Regulators for other required voltages: +0.85V, +0.95V, +1.2V, +1.8V, +2.5V, +3.3V, +5V and -5V.

#### **4.9 Environment:**

- Well ventilated computers, especially in the PCI express expansion region
- Commercial grade components (0-75 Celsius)

### **5 TIpicieUS operation procedures:**

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

The TIpicieUS standalone (no PCI express, not to be confused with the trigger standalone mode) mode (USB-C configuration) will be developed later.

## 5.1 TIpcieUS Power supply:

There are two options for the TIpcieUS power supply. Option#1: The TIpcie uses +3.3V and +12V from the PCI express connector, option#2: it uses the front panel USB type-C connector with +5V supply.

When in option#1, the TIpcieUS generates +0.85V, +0.95V, +1.2V, +1.8V, and +2.5V from the +3.3V input using LTM4615 and TPS74401. It generates the +5V and -5V supply by a DC-DC converter (NMH1205SC) from +12V.

When in option#2, the TIpcieUS generates +3.3V from +5V\_USB using LTM4604. It generates the +0.85V, +0.95V, +1.2V, +1.8V, and +2.5V from the generated +3.3V with the same regulators as option#1. It generates the +5V, and -5V supply by a DC-DC converter (NMH0505SC, which uses the same footprint as the NMH1205SC) from +5V\_USB.

The power supply options are configured by populating the appropriate fuses. There will not be problems if the TIpcieUS is plugged into a PCI express slot, and the USB-C connector is plugged in too. Here is the fuse configuration for the two power options:

Option#1 (PCI express power): F3PCIE and F12PCIE are populated, and F3USB and FUSB5M are not populated. NMH1205SC is populated, and a +12V DC cooling fan can be used.

Option#2 (USB-C power): F3USB and FUSB5M are populated, and F3PCIE and F12PCIE are not populated. NMH0505SC is populated, and a +5V DC cooling fan can be used.

## 5.2 Hardware setting (Switch etc.):

There is one 8-bit switch on the TIpcieUS marked as SC1. This switch is used to set the sources of the four front panel outputs, and the four bit outputs signal sources on the 40-pin connector (pin#5/6, 7/8, 9/10, 11/12).

Bit[3:1]: LVTTTL, open=high. Source selection for the four differential outputs. When “000”, the outputs are controlled by software BR#0, offset 0x4C, bit(3:0).

Bit[6:3]: LVCMOS not used

Bit[8:7]: TIpcieUS clock source selection. The setting is overwritten by the FPGA firmware/software.

## 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 TIpcieUS card is recognized at the computer BIOS boot. During the boot process, the PCI express enumeration will set the proper working parameters to TIpcieUS 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 cycling the computer. If the new firmware keeps the original base address settings, the /sys/bus/pci/devices/0000.22:00.0/config file can be reused without restarting the computer (saving this configure file to a local area, and copying this file back to that directory after firmware reloading, where the 0000.22:00.0 is the host computer enumeration of the TIpcieUS).

### 5.3.2 Driver loading:

The TIpcieUS card is set as “memory controller device”, ‘D0E1’, for DOE lab), and ‘:9038’ (UltraScale+ Kintex FPGA), subsystem device as ‘D0E1’ and ‘:7105’ (TI pci Express with UltraScale Plus Kintex FPGA, and 71E0 for TI PCI Express with Virtex-5 FPGA). The drivers are automatically loaded upon power up. If the driver is loaded successfully, the devices can be used:

/dev/xdma0\_user : for PCIexpress register access, (BA0, 8 KB);

/dev/xdma0\_C2H\_0: DMA readout channel of TI trigger data;

/dev/xdma0\_H2C\_0: DMA write to TIpcieUS. The current firmware just dumps (ignores) the data.

/dev/xdma0\_C2H\_1: DMA readout channel of streaming TDC data (to be implemented).  
/dev/xdma0\_C2H\_2: DMA readout channel of front QSFP data input (to be implemented)

### 5.3.3 TIpcieUS setup:

The TIpcie card needs to be setup properly:

Main Clock selection: BR#0, offset 0x2C: 0x0 for onboard oscillator, .....

SYNC source selection: BR#0, offset 0x24: 0x10 for loopback, .....

Trigger source enables: BR#0, offset 0x20: 0x94 for internal trigger, .....

Busy, Data format etc.: BR#0, offset 0x28, 0x18 etc.

### 5.3.4 Trigger table loading:

There is a 6-bit inputs, 8-bit output look up table built in the TIpcieUS(master). The 6-bit inputs correspond to the six front panel inputs (marked as TS#1, TS#2, ... TS#6). The eight-bit outputs define 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 words (BR#0, offset 0x140 – 0x17C)

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

There are two 32-bit Base Address memory regions defined on the TIpcieUS card.

The BA#0 has 8K byte addressable in 32-bit word only. These memory spaces are further divided into two JTAG spaces, two I2C spaces, one spare 1K bytes space, and one 1K bytes 32-bit registers.

The BA#1 has 64K bytes used for PCI express DMA implementation, which should not be accessed by users directly.

### 6.1 BA#0 memory (Address(12:10) == 000, lower 1K bytes):

These 1K bytes are equivalent to the VME TI A24D32 read/write. They are addressed in 32-bit word only (not byte addressable, to be simple) with Address(1:0) == "00".

➤ Address offset: 0x00000: Board ID:

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

Bit 15-8 (R): Board ID; right now, it is 0x48;

Bit 19-16 (R): PCB related setting, 0x4: for TIpcieUS, 0x1: for TIpcieV5;

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

➤ Address offset: 0x00004: Optic transceiver enable:

Bit 31-0 (R/W): not used

➤ 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: 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 011;

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 trigger code, higher 4-bit of event number, and higher 16-bit of trigger timing;

bit 3: '1' to enable the front panel TS-code pattern readout (bit(5:0). This is before the pre-scaling.

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: control setting; Reset default 0x011:

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

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 31 (R/W): '1' to slow down the "slow clock" further for a factor of 32 in trigger rules logic.

➤ Address offset: 0x00020: Trigger source register:

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

Bit 1: QSFP#A optic transceiver trigger input;

Bit 2: TI-master 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 10: QSFP#B optic transceiver trigger input;

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

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

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

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

Bit 15: SubTS#4 trigger enable, this is valid on fiber 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: QSFP#A (or fiber#1) optic transceiver input;  
 Bit 2: QSFP#B (or fiber#5) optic transceiver input;  
 Bit 3: Front panel trigger inhibit enable;  
 Bit 4: Loopback SYNC enable, when TIpcieUS is in master or standalone mode  
 Bit 6: automatic Sync-Reset 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 QSFP#A;  
 Bit(15:12) (R): last SYNC code from QSFP#B;  
 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 5: '1' enable the front panel (bracket) busy input;  
 Bit 6: '1' BUSY if the trigger is out, but the acknowledge (from slave TI) has not been received;  
 Bit 7: '1' enable TIpcie feed-back BUSY, '0' disable the busy. (used in master or standalone mode)  
 Bit 8: QSFP#A BUSY enable: '1' enable the QSFP#A BUSY input, '0' disable;  
 Bit 12: QSFP#B BUSY enable: '1' enable the QSFP#B 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] = 01: QSFP#B (HFBR#5) clock input;  
 Bit[1:0] = 10: QSFP#A (HFBR#1) clock input;  
 Bit[1:0] = 11: Front panel (40-pin connector) clock input;  
 Bit 19-16 (R/W): TCS daisy chain (fanout) mode if set to 0101: QSFP#A clock is used, and the clock is fanned out to QSFP#B.

➤ Address offset: 0x00030: Trigger\_1 pre-scale:

Bit 15-0 (R/W): pre-scale factor:  $\text{Rate} = \text{Rate}_0 / (\text{bit}(15:0)+1)$ . Factor = 1, 2, 3, 5, 9, 17, ...

➤ 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 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;

If the register 0x1C bit#31 is set 1, the clock step increase from 500ns to 16us. This means that the rule can cover up to 2 ms.

➤ 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 generic trigger input enable:

Bit 5-0 (R/W): Front panel trigger code (TS-code(6:1)) input enable; 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 QSFP#A input;

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

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

Bit 23-16 (R): when TIpcieUS is in slave mode: SYNC phase of QSFP#B input;

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



- 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: 0x00064: Front Panel generic trigger input pre-scale: Reset default 0x00000000
  - Bit 3-0 (R/W): FP Generic trigger input TS-code#1;
  - Bit 7-4 (R/W): FP Generic trigger input TS-code#2;
  - Bit 11-8 (R/W): FP Generic trigger input TS-code#3;
  - Bit 15-12 (R/W): FP Generic trigger input TS-code#4;
  - Bit 19-16 (R/W): FP Generic trigger input TS-code#5;
  - Bit 23-20 (R/W): FP Generic trigger input TS-code#6;
- 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 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;
  - 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 a ~ 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 00,0111
- 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  $(\text{Bit}(6:0) * (4/32 \text{ ns}))$ , 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.
    - Bit(11-0) = 0xC06: (A=0xC) Set the Buffer level (size), and the buffer level is set to 0x06.
  
- Address offset: 0x00088 (R/W): software Random Trigger Command Register:
  - Bit 3-0: Random trigger\_1 rates:  $466 \text{ KHz}/(2^{\text{Bit}(3:0)})$ ;
  - 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:  $466 \text{ KHz}/(2^{\text{Bit}(11:8)})$ ;
  - 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): 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.  $T = (32+8*\text{Bit}(30:16))*1024^{\text{Bit}(31)}$  ns. (Assuming that the PCI express reference clock is 100MHz, and the interface clock is 125 MHz or 8ns 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.  $T = (40+10*\text{Bit}(30:16))*1024^{\text{Bit}(31)}$  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): Tlpcie master generated sync code valid;
  - Bit (18:15): Tlpcie master generated sync code;
  - Bit (14): Loopback sync valid;
  - Bit (13:10): Loopback sync code;
  - Bit (9): QSFP#B (HFBR#5) sync valid;
  - Bit (8:5): QSFP#B (HFBR#5) sync code;
  - Bit (4): QSFP#A fiber sync valid;
  - Bit (3:0): QSFP#A 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 from QSFP#A:
  - 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 QSFP#B
  - 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  $\text{Bit}(31:0)*256*8\text{ns}$ . (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  $\text{Bit}(31:0)*256*8\text{ns}$ . 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 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 Credits available to the remote link partner.
- Address offset: 0x000B4 (R): MGT STATUS\_B registers (not assigned on TIpcie):
  - 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;

Bit 27: TI is in running mode if '1';

Bit 28: QSFP MGT receiver error (the used one);

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 TIpcieUS in master mode:

Bit 7-0: Number of blocks to be readout on QSFP#A (blocks need ROCack);

Bit 15-8: Number of blocks is still missing on QSFP#A (blocks need TRGack);

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: 0x000C8 (R): valid for TIpcieUS in master mode:

Bit 7-0: Number of blocks to be readout on QSFP#B (blocks need ROCack);

Bit 15-8: Number of blocks is still missing on QSFP#B (blocks need TRGack);

➤ Address offset: 0x000D0 (R): valid for TIpcie in master or standalone mode:

Bit 15-0: Extra ROCack received on QSFP#A (that is no block is readout, but got ROCack);

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: 0x000EC (R/W): ROC enable

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

➤ 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 (TIpacieUS 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 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 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 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 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): 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: 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.  $MinimumWidth = bit(30:24) * Tclock$ .  $Tclock = 480ns$  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 the 16 32-bit words;

- 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:

- 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: 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 (QSFP#A) readout busy counter
  - Bit 31-0: 32-bit Busy timer of TI#1 (QSFP#A) (same steps as live/busy timer).
- Address offset: 0x001B0 (R): TI#5 (QSFP#B) readout busy counter
  - Bit 31-0: 32-bit Busy timer of TI#5 (QSFP#B) (same steps as live/busy timer).
- Address offset: 0x001D0 (R): optic transceiver QSFP#A 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.
  - Bit 31-24: Buffer level set on TI connected to fiber#1.
- Address offset: 0x001E0 (R): optic transceiver QSFP#B (HFBR#5) TI ID (in master mode)
  - Bit 7-0: Trigger Source Enable of the TI connected to QSFP#B (fiber#5);
  - Bit 15-8: Crate ID of the TI connected to QSFP#B (fiber#5).
  - Bit 23-16: Block level set on TI connected to QSFP#B (fiber#5).
  - Bit 31-24: Buffer level set on TI connected to QSFP#B (fiber#5).
- Address offset: 0x001F0 (R): The TI master (itself) ID (in master mode)
  - Bit 7-0: Trigger Source Enable of itself, should be the same as bit7-0 of register 0x20;
  - Bit 15-8: Crate ID of 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).

## 6.2 BA#0 memory (Address Bit(12:10) = 001): reserved:

These 1K bytes have not been used yet. It could be 1K bytes registers, or another I2C engine.

### **6.3 BA#0 memory (Address Bit(12:11) = 01): PCI express to I2C (master) engines:**

The middle 2K bytes are used for PCI express to I2C engines. Address Bit(10) is used to select the I2C engines. The lower eight bits of the 32-bit data (Address Bit(1:0)="00") are used for I2C read or write.

Address Bit(10): '0': for optic transceiver QSFP#A I2C; '1': for transceiver QSFP#B I2C.

Address Bit(9:2): eight bits of the I2C byte address.

Address Bit(1:0): to be set as "00".

The device address is generated as "0xA0" automatically as required by the QSFP I2C interface standard. (This means that the engine, firmware design, will not work as generic I2C engine, which require 7 bits of device address, or 32K bytes memory space minimum).

### **6.4 BA#0 memory (Address Bit(12) = 1): PCI express to JTAG engine:**

The upper 4K bytes are used for PCI express to JTAG engines. Address Bit(12) = 1, and Address Bit(11) for the two JTAG engines. There is no specific use of the JTAG on the TlpcieUS yet. For data transfer less than 32 bits, the lower N bits (specified by the address) are used.

Address Bit(10:6): Number of bits to shift (Instruction register or Data register),  $n = \text{Addr}(10:6) + 1$ , 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 Bit(11). Bits(10:0) are DONOT care bits. 32-bit data read only

### **6.5 PCI express data acquisition:**

The Xilinx DMA engines are used for data acquisition. The C2H (Core to Host-computer) slave DMA are implemented for data readout.

Channel#0 (C2H\_0) is used for trigger data, which is the same format as VXS/VME TI.

Channel#1 (C2H\_1) is used for Streaming TDC data readout (to be implemented).

Channel#2 (C2H\_2) is used for QSFP data readout, where the QSFP can be 40 GBE, 100 GBE, or custom protocols (to be implemented).

The DMA controls (bookkeeping) are implemented in BA#1 as 32-bit addressed 64K bytes memory space.



## 7 Connector 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	<a href="#">SMBus</a> and <a href="#">JTAG</a> port pins
8	+3.3 V	TMS	
9	TRST#	+3.3 V	
10	+3.3 V aux	+3.3 V	<a href="#">Standby power</a>
11	WAKE#	PERST#	Link reactivation; fundamental reset
<b>Key notch</b>			
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	

PCI Express ×4 cards end at pin 32

33	HSOp(4)	Reserved	
34	HSON(4)	Ground	Lane 4 transmit data, + and -
35	Ground	HSIp(4)	
36	Ground	HSIn(4)	Lane 4 receive data, + and -
37	HSOp(5)	Ground	
38	HSON(5)	Ground	Lane 5 transmit data, + and -
39	Ground	HSIp(5)	
40	Ground	HSIn(5)	Lane 5 receive data, + and -
41	HSOp(6)	Ground	
42	HSON(6)	Ground	Lane 6 transmit data, + and -
43	Ground	HSIp(6)	
44	Reserved	HSIn(6)	Lane 6 receive data, + and -
45	HSOp(7)	Ground	
46	HSON(7)	Ground	Lane 7 transmit data, + and -
47	Ground	HSIp(7)	
48	PRSNT2#	HSIn(7)	Lane 7 receive data, + and -
49	Ground	Ground	



## 7.2 Front panel attached to the PCB:

From top to bottom (closest to the PCIexpress connector), they are: QSFP#A, USB type C, LED#1, LED#2, LED#3, LED#4, and QSFP#B.

## 7.3 Front panel (the second one, or the onboard 40-pin connector) pin table

Twenty pairs of the 40-pin connector. The pin#1-34 are assigned the same input/output signals as VXS/VME TI boards, but (maybe) different signal levels.

Pin name	Signal Name	Signal Level
Pin 1+/2-	BUSY out	LVDS
Pin 3+/4-	Trigger out	LVDS (can be ECL)
Pin 5+/6-	Output #1	LVDS
Pin 7+/8-	Output #2	LVDS
Pin 9+/10-	Output #3	LVDS
Pin 11+/12-	Output #4	LVDS
Pin 13+/14-	OT#1 out	LVDS (can be ECL)
Pin 15+/16-	OT#2, PromptTrigger out	LVDS (can be ECL)
Pin 17+/18-	Trigger Inhibit/Busy input	Any-level differential
Pin 19+/20-	Trigger input	Any-level differential
Pin 21+/22-	TS-code #1 input	Any-level differential
Pin 23+/24-	TS-code #2 Input	Any-level differential

Pin 25+/26-	TS-code #3 Input	Any-level differential
Pin 27+/28-	Clock Input	Any-level differential
Pin 29+/30-	TS-code #4 Input	Any-level differential
Pin 31+/32-	TS-code #5 Input	Any-level differential
Pin 33+/34-	TS-code #6 Input	Any-level differential
Pin 35+/36-	GenericIn#1 Input	Any-level differential
Pin 37+/38-	GenericIn#2 Input	Any-level differential
Pin 39+/40-	Clock out	LVPCEL33

## 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 TIpicieUS Operation examples:

The TIpicieUS test software are located at `~jgu/PCIeTI/TIpicieUS/` (`~jgu/PCIeTI/TIpicieV5` is for the PCI express TI with Xilinx VIRTEX-5 FPGA):

xdma: Xilinx DMA drivers, which is automatically loaded on power up

tests: Xilinx xdma test programs from Xilinx driver package, not tested yet

tools: modified test program for TIpicieUS. Two programs are particularly useful:

- TIcontrol: “TIcontrol /dev/xdma0\_user”: The TIpicieUS test program, similar functions as the `trigger.c` for VXS/VME TI.
- `dma_from_device`: “`dma_from_device -c 2 -s 1984 -f OutputDataFile.bin`”: Read two (`-c 2`) blocks of data with block size (`-s 1984`) of 1984 bytes, and dump the data to file. The default xdma device is (`-d /dev/xdma0_c2h_0`).

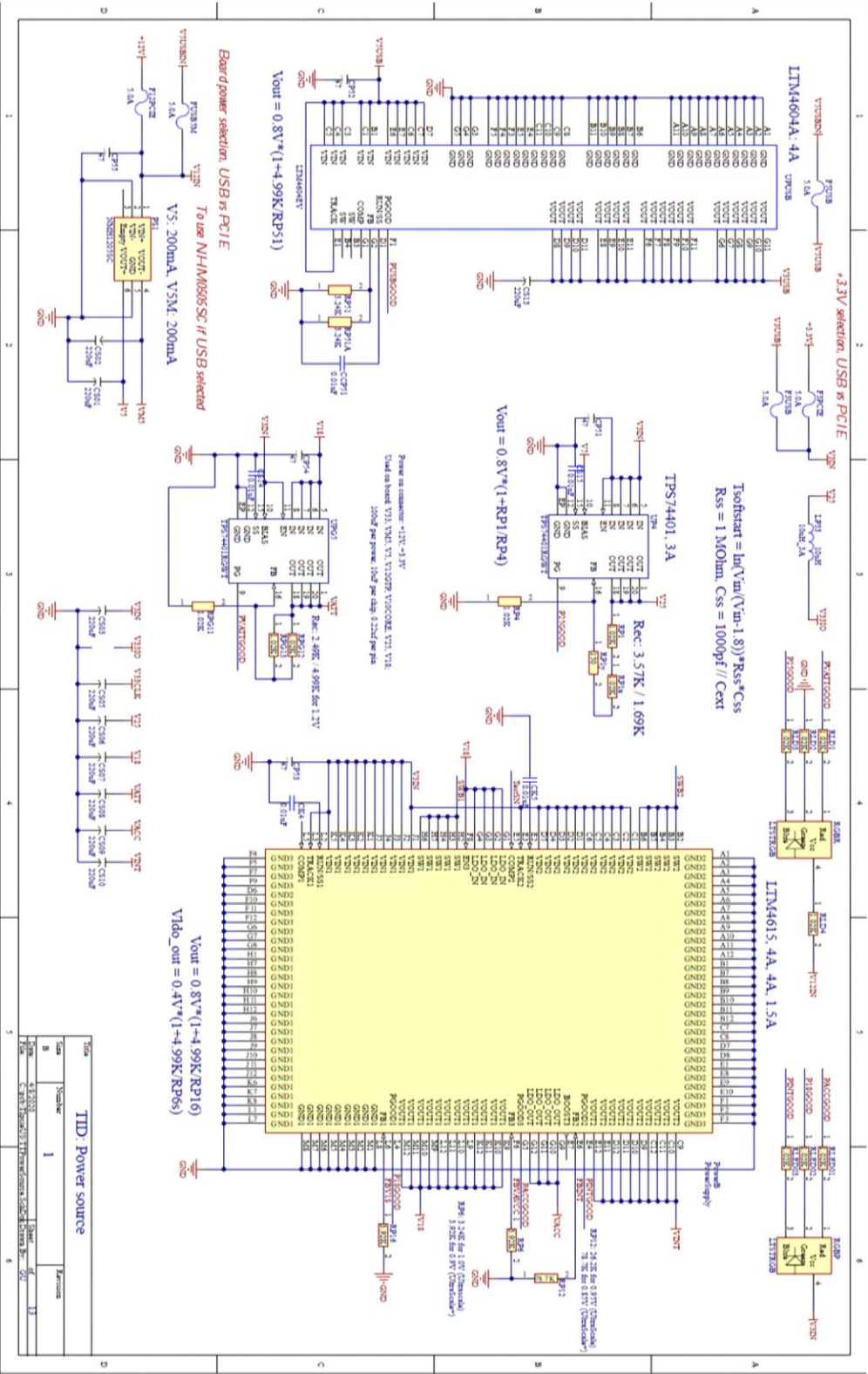
## **9. Citations:**

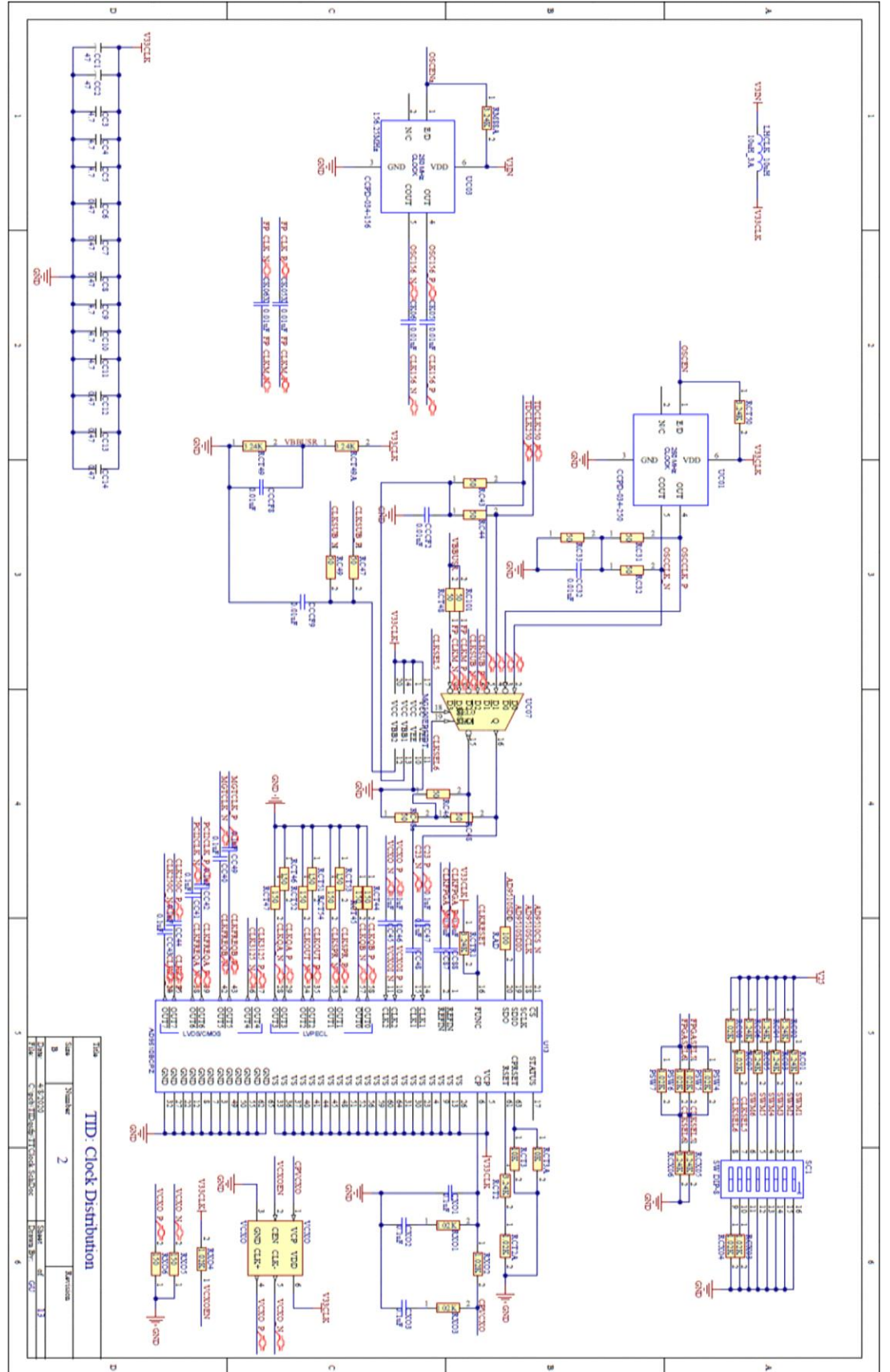
### **Works Cited**

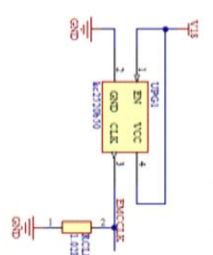
GU, (2010). TI manual

GU, (2014). TIpcie (TIpcieV5) manual

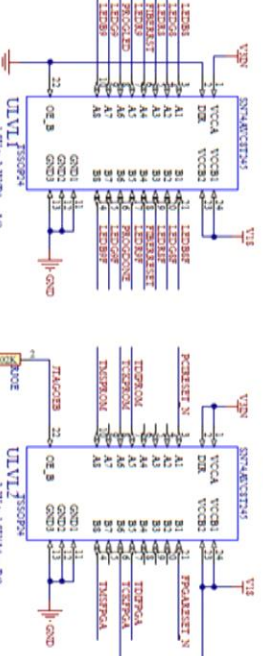
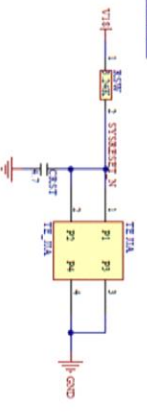
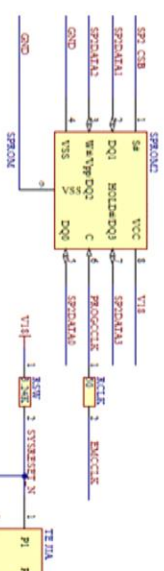
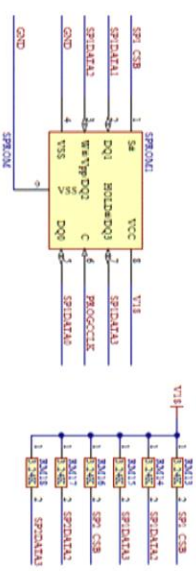
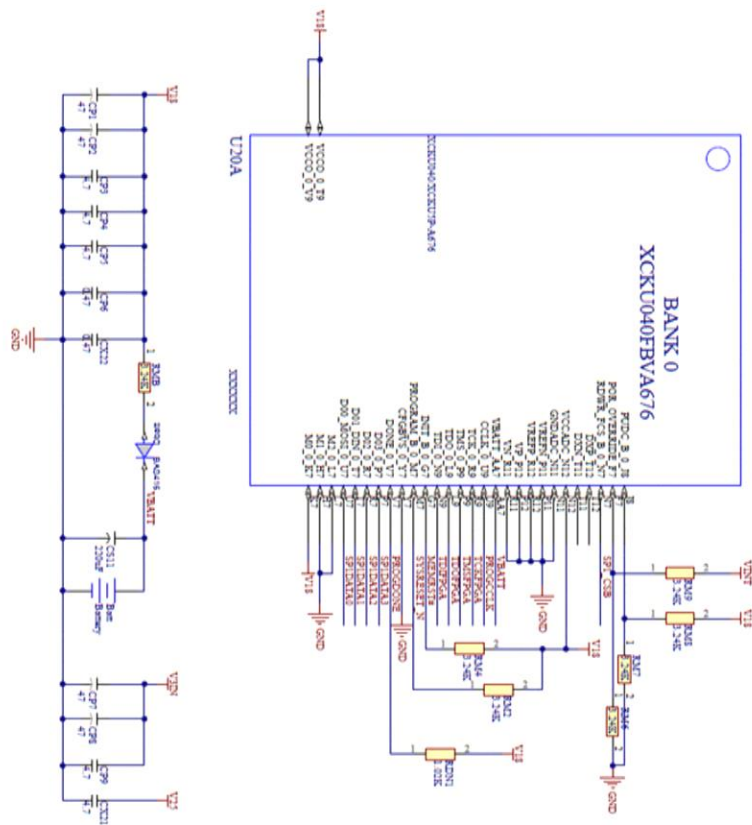
# Appendix A: TlpcieUS PCB schematics:







POR: Power On Reset Delay Override:  
 HIGH: shorter delay (before INIT\_B)  
 LOW: Internal PullUp During Configuration  
 LOW: pullup enabled



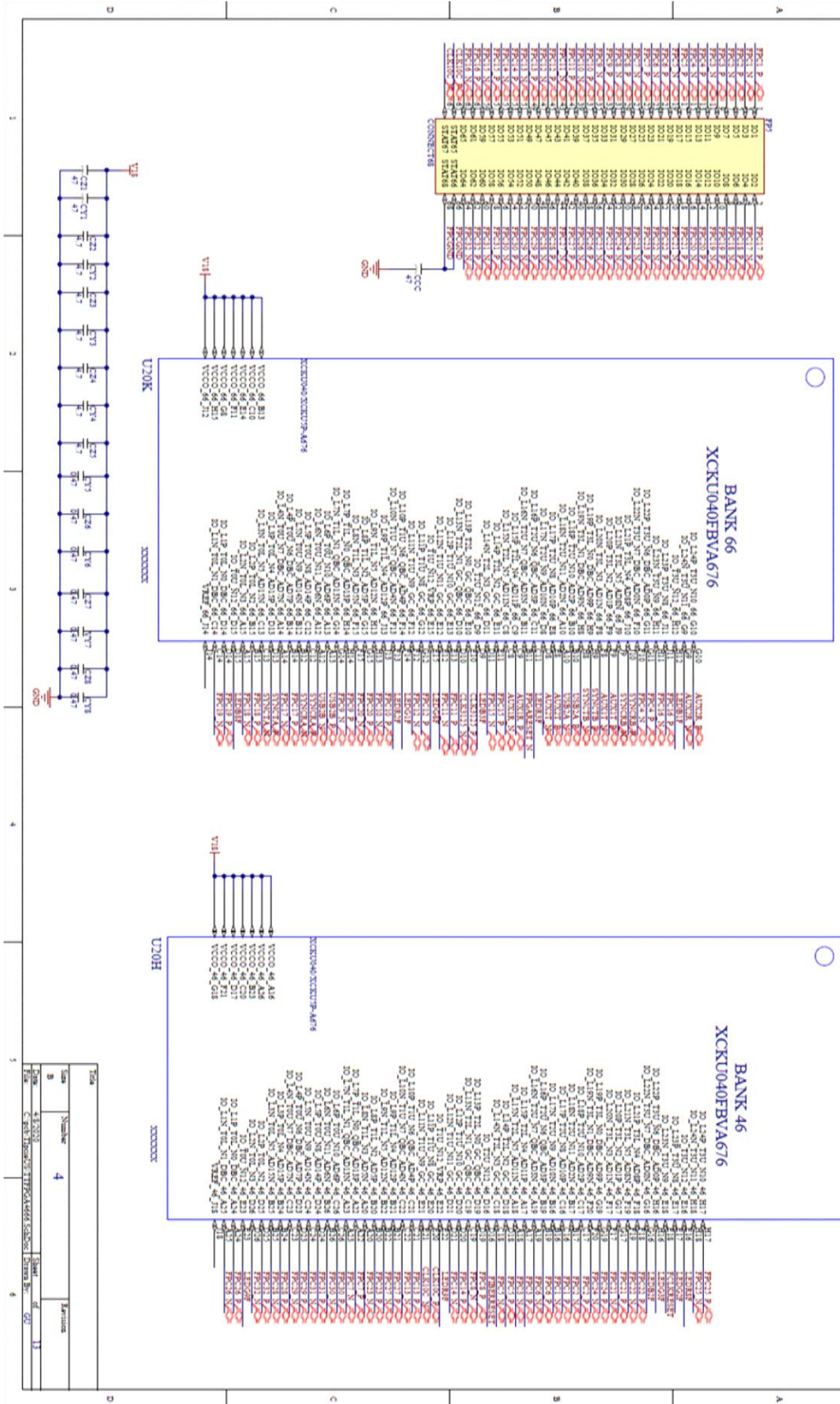
DIR/IOE\_B are supplied by the VCCA  
 OE\_B: HIGH: disable the outputs  
 DIR: HIGH: A->B, LOW: A<-B

Symbol	Value	Reference
R1	10k	
R2	10k	
R3	10k	
R4	10k	
R5	10k	
R6	10k	
R7	10k	
R8	10k	
R9	10k	
R10	10k	
R11	10k	
R12	10k	
R13	10k	
R14	10k	
R15	10k	
R16	10k	
R17	10k	
R18	10k	
R19	10k	
R20	10k	
R21	10k	
R22	10k	
R23	10k	
R24	10k	
R25	10k	
R26	10k	
R27	10k	
R28	10k	
R29	10k	
R30	10k	
R31	10k	
R32	10k	
R33	10k	
R34	10k	
R35	10k	
R36	10k	
R37	10k	
R38	10k	
R39	10k	
R40	10k	
R41	10k	
R42	10k	
R43	10k	
R44	10k	
R45	10k	
R46	10k	
R47	10k	
R48	10k	
R49	10k	
R50	10k	
R51	10k	
R52	10k	
R53	10k	
R54	10k	
R55	10k	
R56	10k	
R57	10k	
R58	10k	
R59	10k	
R60	10k	
R61	10k	
R62	10k	
R63	10k	
R64	10k	
R65	10k	
R66	10k	
R67	10k	
R68	10k	
R69	10k	
R70	10k	
R71	10k	
R72	10k	
R73	10k	
R74	10k	
R75	10k	
R76	10k	
R77	10k	
R78	10k	
R79	10k	
R80	10k	
R81	10k	
R82	10k	
R83	10k	
R84	10k	
R85	10k	
R86	10k	
R87	10k	
R88	10k	
R89	10k	
R90	10k	
R91	10k	
R92	10k	
R93	10k	
R94	10k	
R95	10k	
R96	10k	
R97	10k	
R98	10k	
R99	10k	
R100	10k	



Bank66 HPIO for UltraScale and UltraScale+

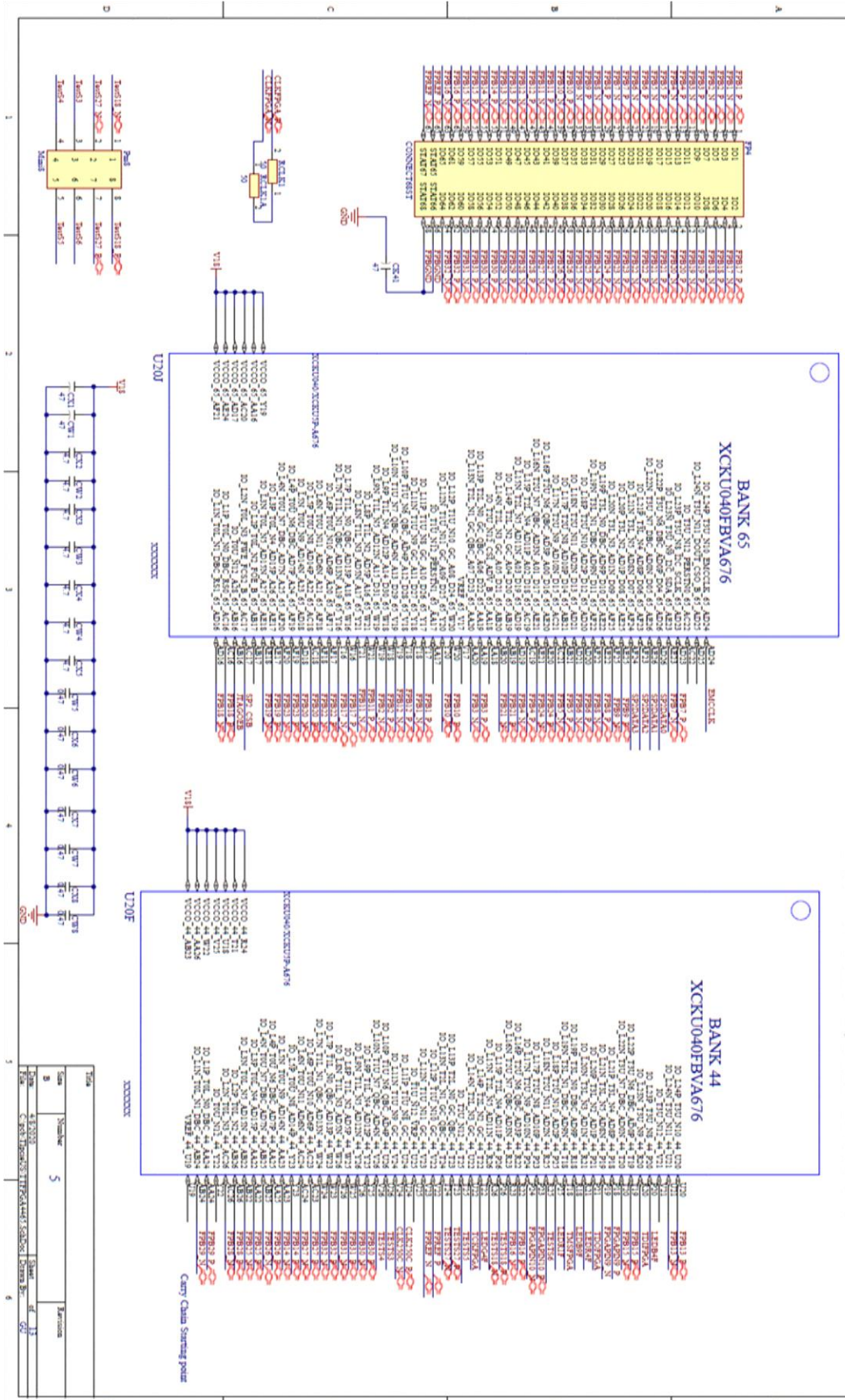
Bank46 for UltraScale, bank 67 for UltraScale+



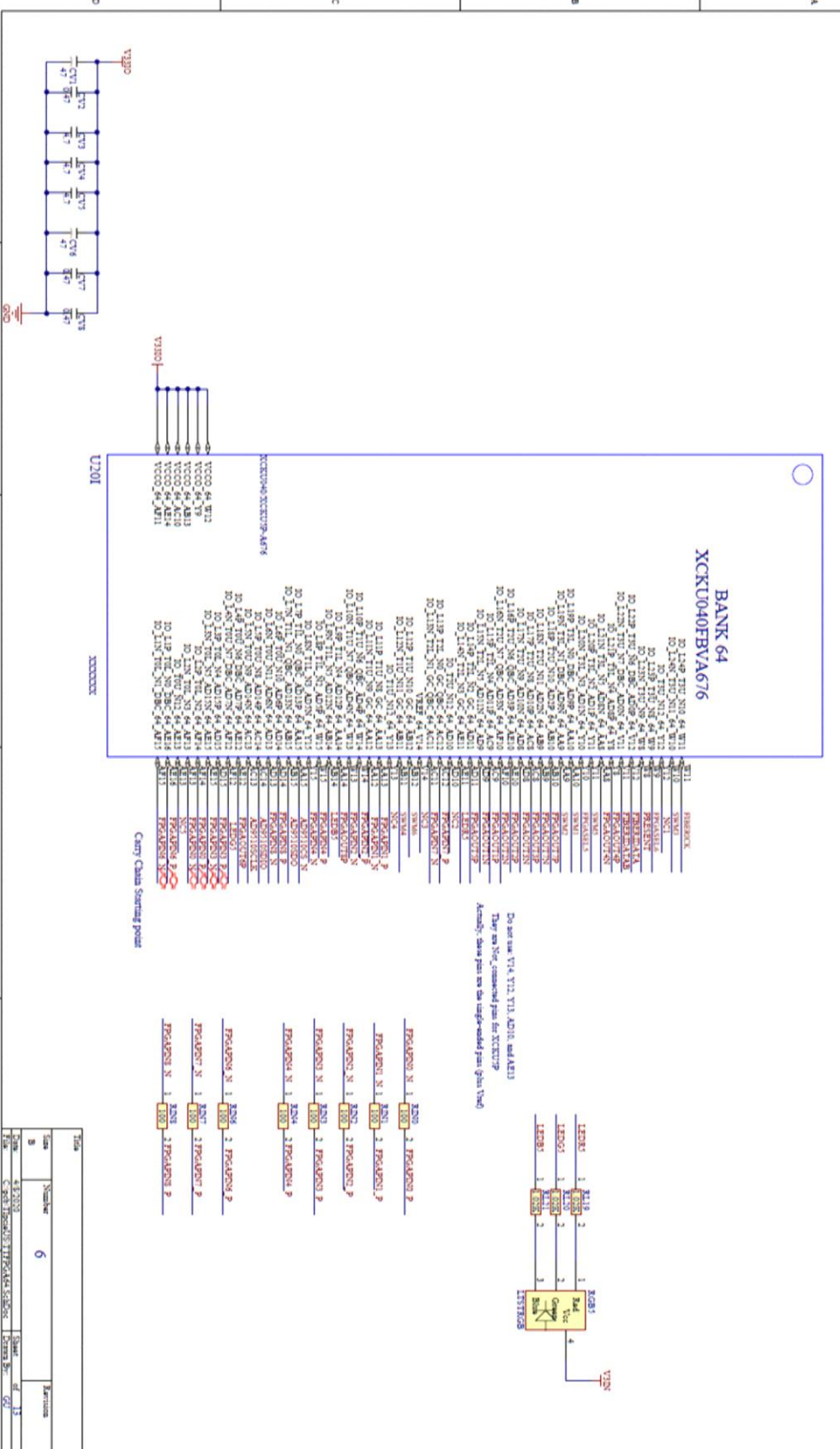
Pin	Signal	Bank	Component
PR0_0	PR0_0	Bank 66	U70K
PR0_1	PR0_1	Bank 66	U70K
PR0_2	PR0_2	Bank 66	U70K
PR0_3	PR0_3	Bank 66	U70K
PR0_4	PR0_4	Bank 66	U70K
PR0_5	PR0_5	Bank 66	U70K
PR0_6	PR0_6	Bank 66	U70K
PR0_7	PR0_7	Bank 66	U70K
PR0_8	PR0_8	Bank 66	U70K
PR0_9	PR0_9	Bank 66	U70K
PR0_10	PR0_10	Bank 66	U70K
PR0_11	PR0_11	Bank 66	U70K
PR0_12	PR0_12	Bank 66	U70K
PR0_13	PR0_13	Bank 66	U70K
PR0_14	PR0_14	Bank 66	U70K
PR0_15	PR0_15	Bank 66	U70K
PR0_16	PR0_16	Bank 66	U70K
PR0_17	PR0_17	Bank 66	U70K
PR0_18	PR0_18	Bank 66	U70K
PR0_19	PR0_19	Bank 66	U70K
PR0_20	PR0_20	Bank 66	U70K
PR0_21	PR0_21	Bank 66	U70K
PR0_22	PR0_22	Bank 66	U70K
PR0_23	PR0_23	Bank 66	U70K
PR0_24	PR0_24	Bank 66	U70K
PR0_25	PR0_25	Bank 66	U70K
PR0_26	PR0_26	Bank 66	U70K
PR0_27	PR0_27	Bank 66	U70K
PR0_28	PR0_28	Bank 66	U70K
PR0_29	PR0_29	Bank 66	U70K
PR0_30	PR0_30	Bank 66	U70K
PR0_31	PR0_31	Bank 66	U70K
VDD	VDD	Bank 66	U70K
VSS	VSS	Bank 66	U70K
VDD	VDD	Bank 46	U70H
VSS	VSS	Bank 46	U70H

Bank65; HRIO for UltraScale; HP10 for UltraScale+

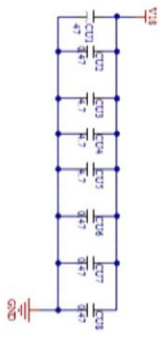
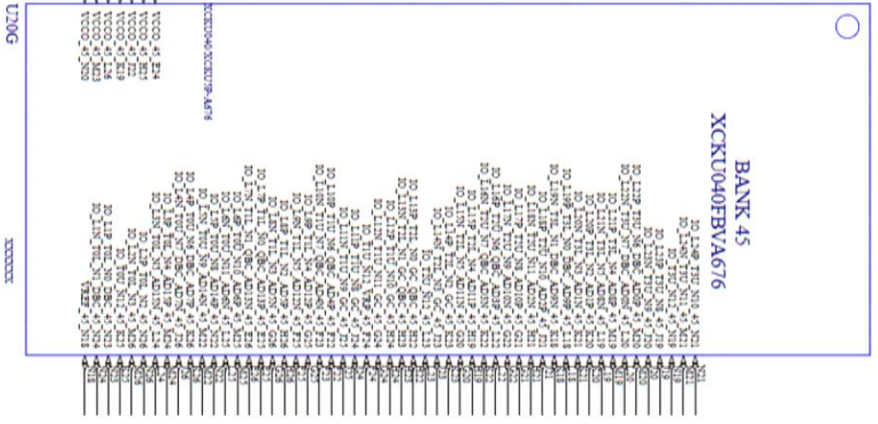
Bank44 for UltraScale; Bank64 for UltraScale+



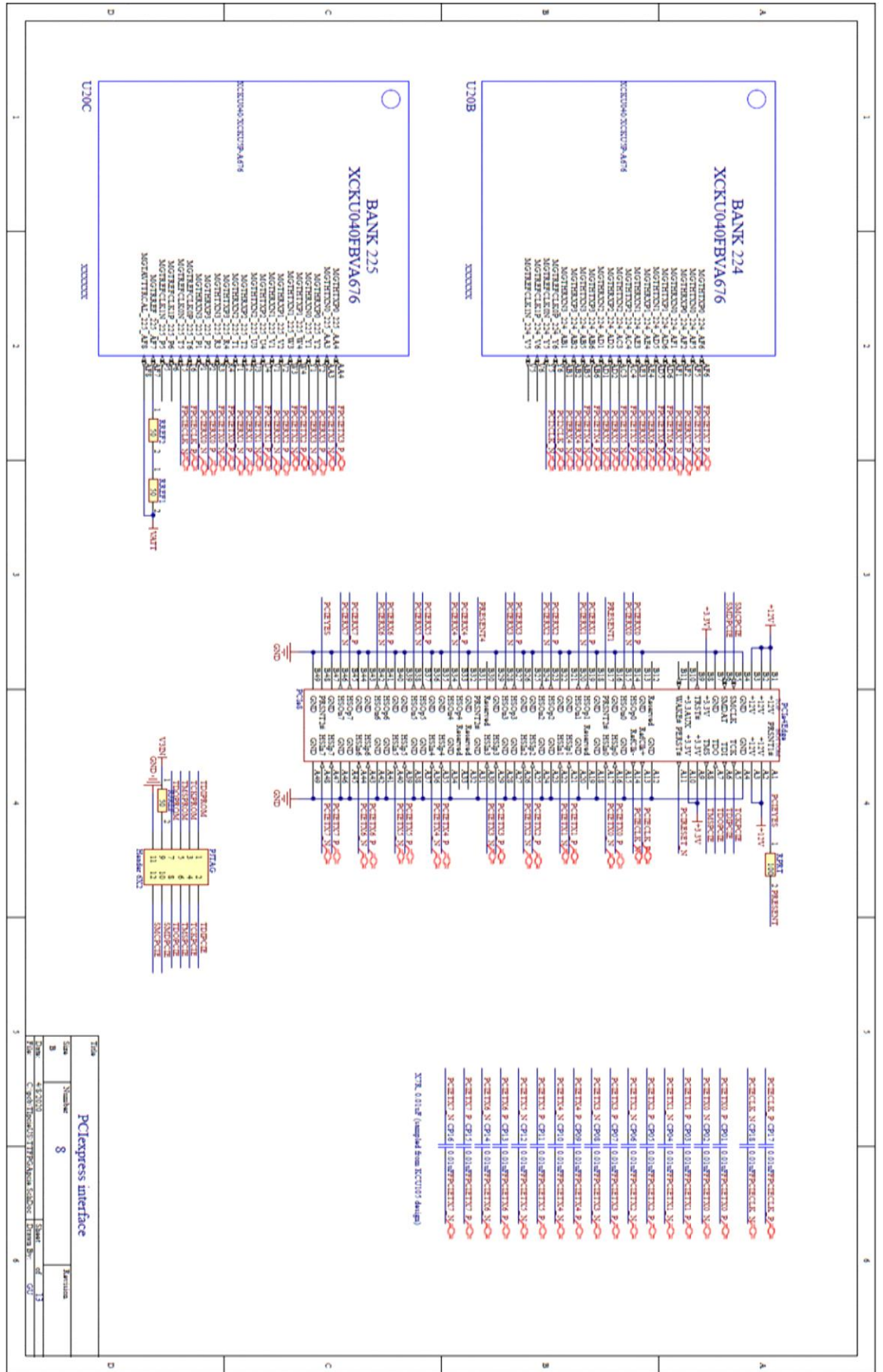
Bank64, HRIO for UltraScale. Bank84/85 HDIO for UltraScale+

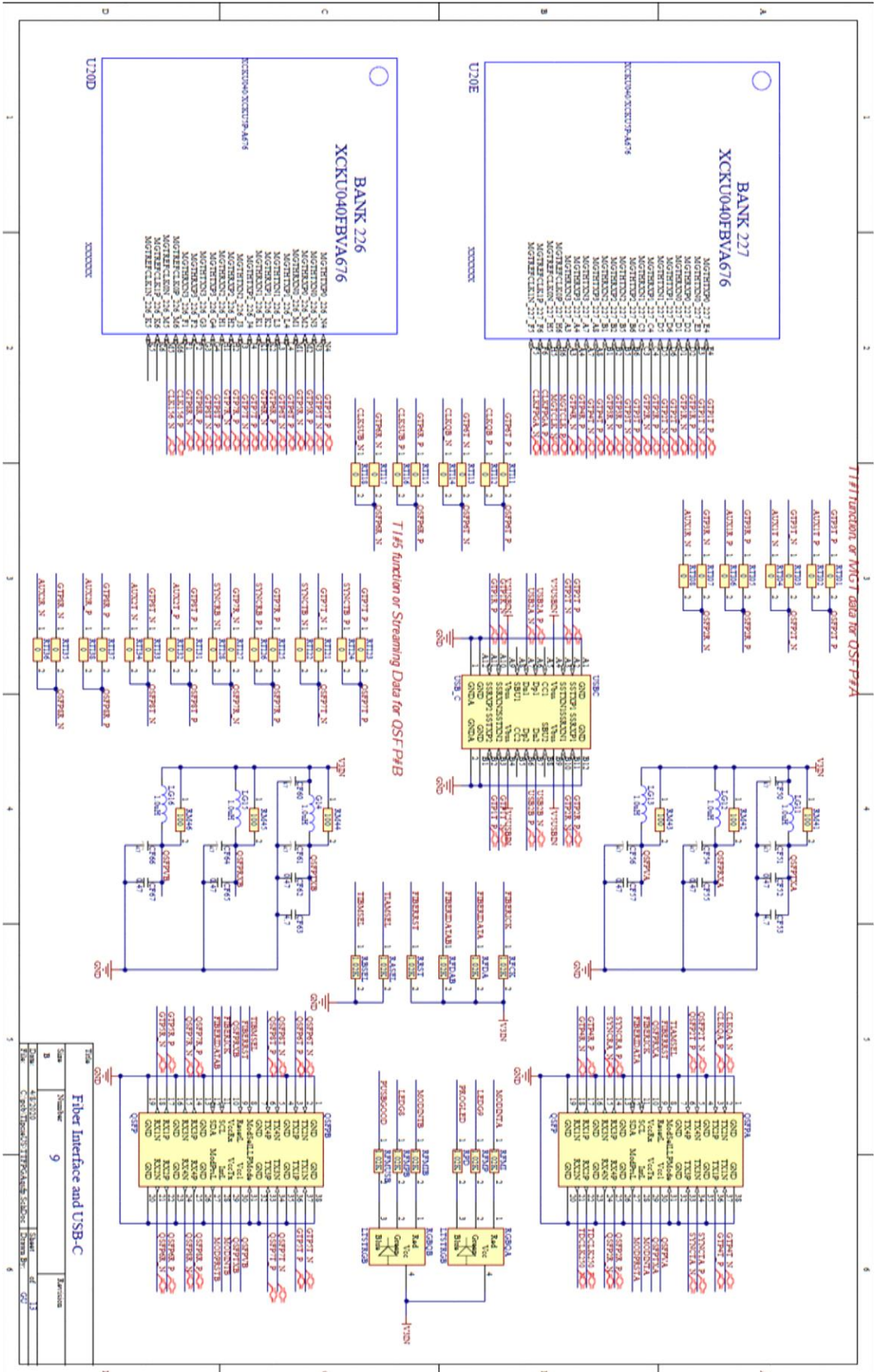


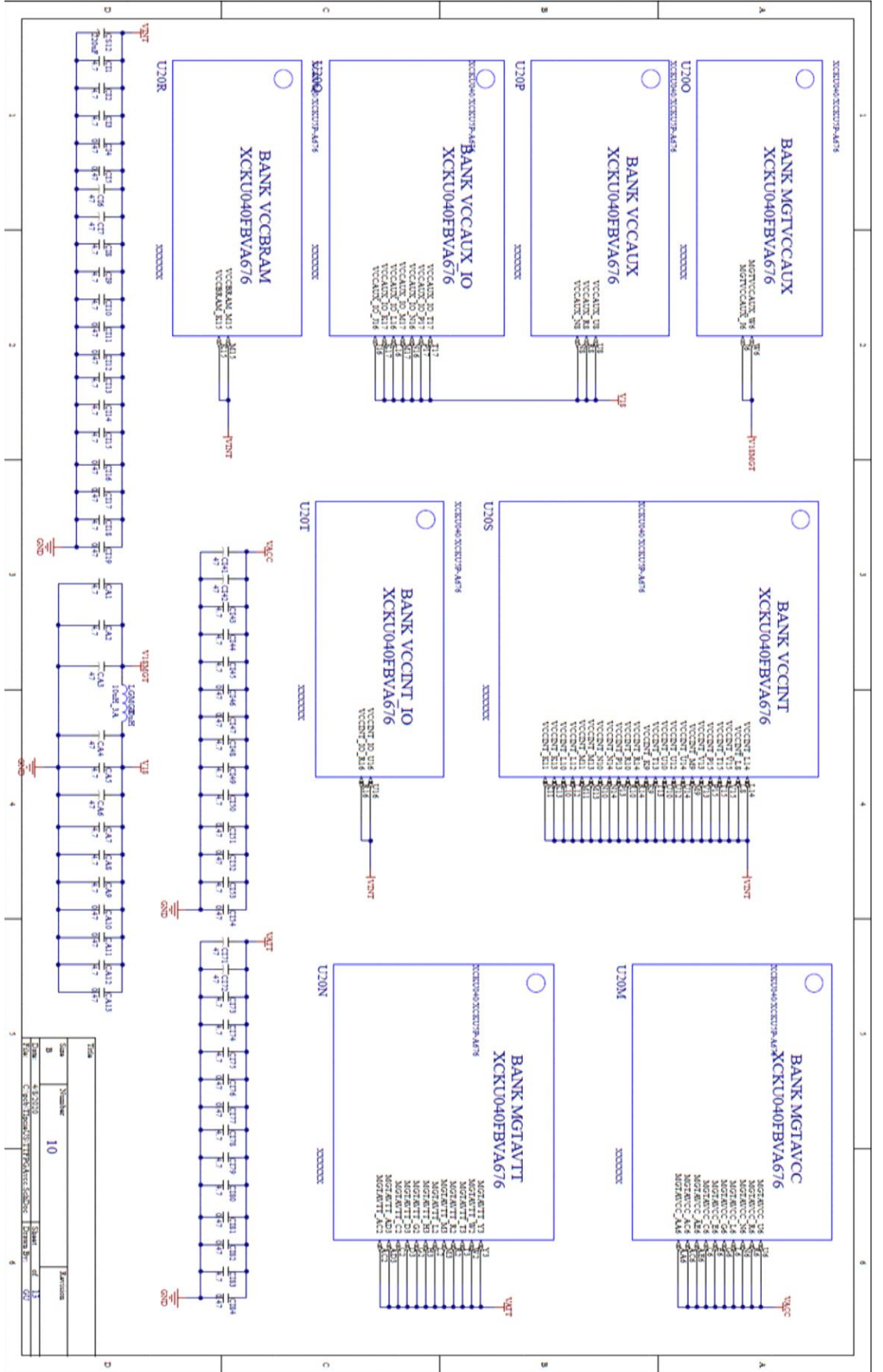
Bank45 is available for UltraScale only, not UltraScale+



Item	Quantity	Part Number	Revision
BANK#45, only available on UltraScale	7		
Part Number	452020		
Part Name	C:\projects\20131126\K45\BANK45		
Sheet No.	02	of	13



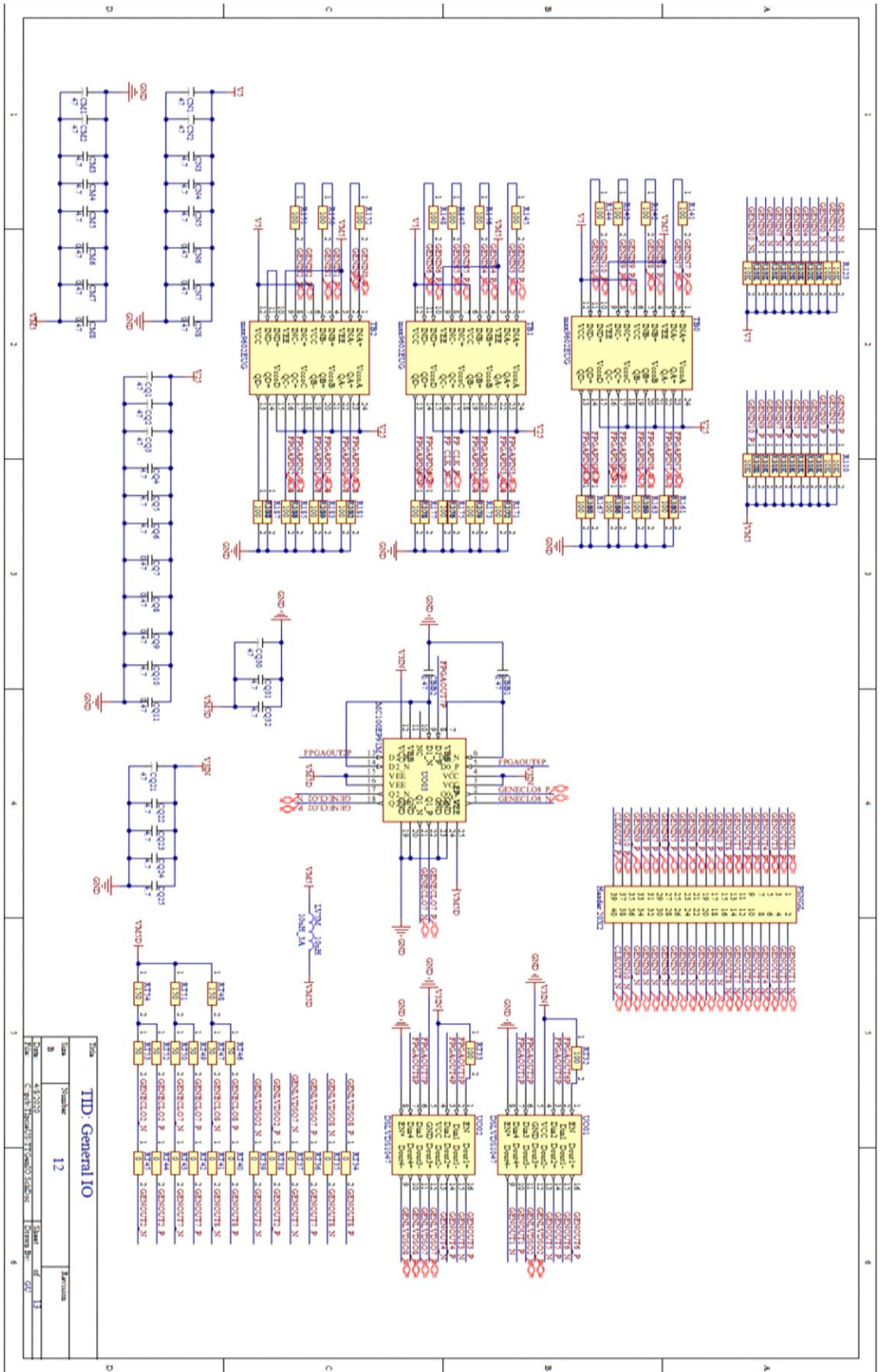




Item	Quantity	Part Number	Manufacturer
1	10	412 2010	0805
2	10	412 2010	0805
3	10	412 2010	0805
4	10	412 2010	0805
5	10	412 2010	0805
6	10	412 2010	0805
7	10	412 2010	0805
8	10	412 2010	0805
9	10	412 2010	0805
10	10	412 2010	0805

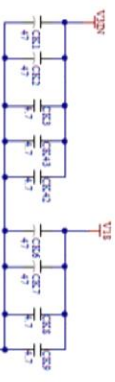
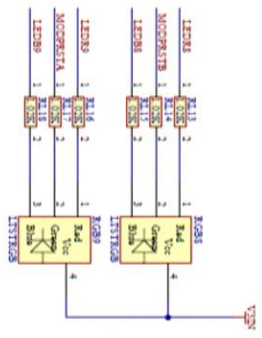
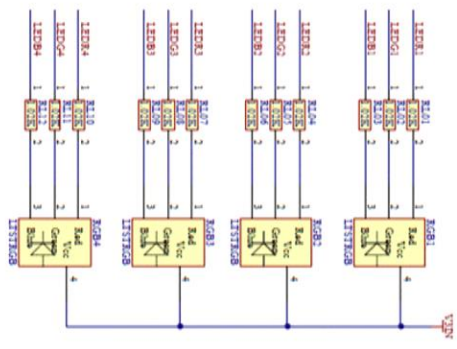
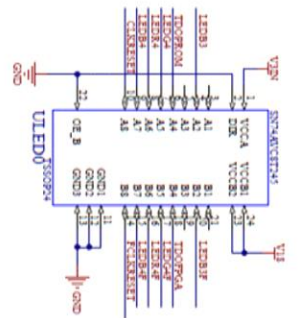
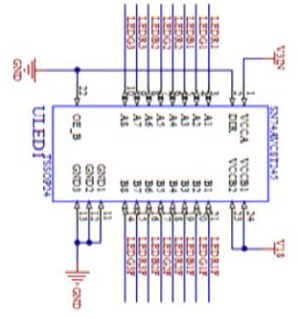






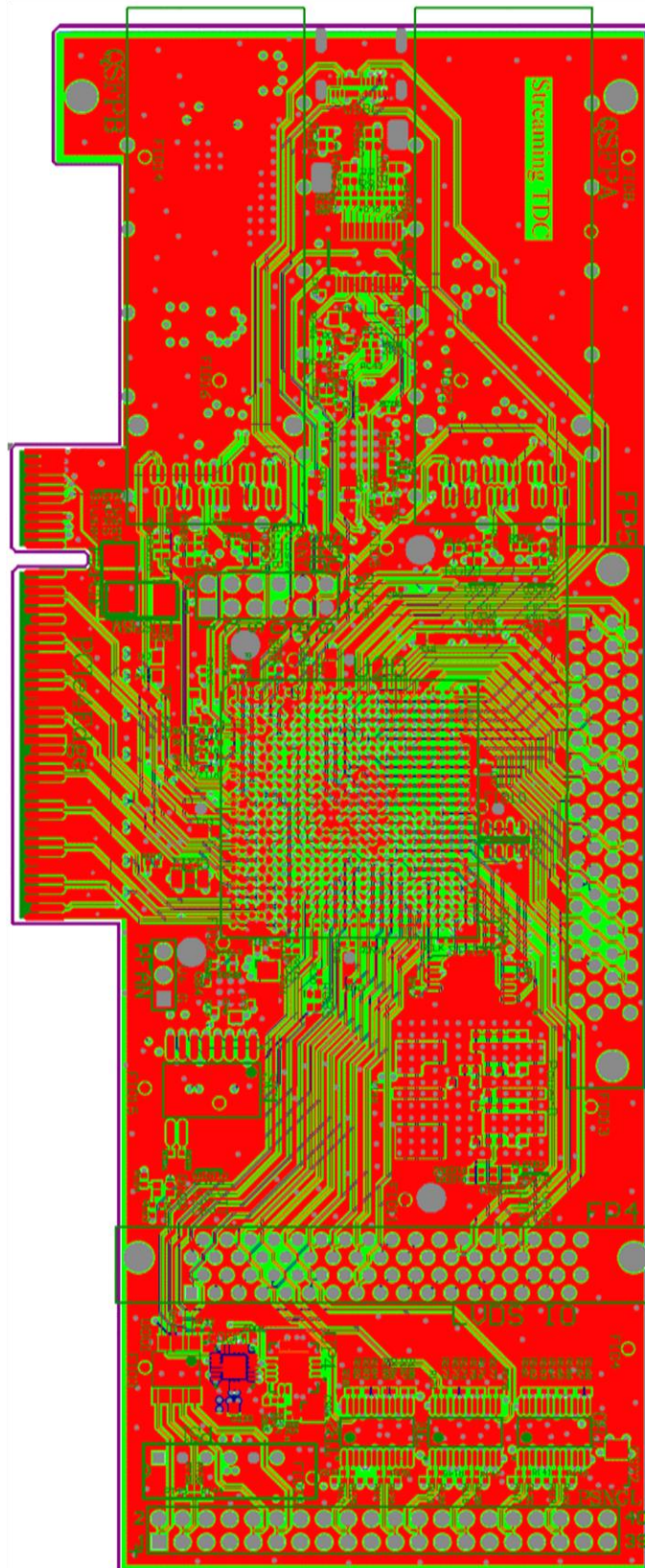
TTD: General IO

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



Size	Number	Series
5	13	
Part No.	C-RED-THROAT-LIGHT-LED-BOX	
Rev.	Rev. No.	Rev. No.
	01	01

## Appendix B: TlpcieUS top layer,



# Appendix C: Bill of materials:

Comment	Description	Designator	Footprint	Quantity	Manufacture PartNumber	alternative allowed
4.7		CA1, CA2, CA5, CA7, CA8, CA9, CA12, CC3, CC4, CC5, CC9, CC10, CC11, CF53, CF63, CI1, CI2, CI3, CI8, CI9, CI10, CI13, CI14, CI15, CI18, CI43, CI44, CI45, CI48, CI49, CI50, CI53, CI73, CI74, CI75, CI78, CI79, CI80, CI83, CK3, CK8, CK9, CK42, CK43, CM3, CM4, CM5, CN3, CN4, CN5, CP3, CP4, CP5, CP9, CQ4, CQ5, CQ6, CQ10, CQ22, CQ23, CQ24, CQ25, CQ31, CQ32, CRST, CU3, CU4, CU5, CV3, CV4, CV5, CW2, CW3, CW4, CX2, CX3, CX4, CX5, CX21, CY2, CY3, CY4, CZ2, CZ3, CZ4, CZ5	CR0402	86	Taiyo Yuden JMK105BBJ475MV-F	YES, 4.7uF, >6V
47		CA3, CA4, CA6, CC1, CC2, CCC, CF50, CF51, CF54, CF56, CF60, CF61, CF64, CF66, CI6, CI7, CI41, CI42, CI71, CI72, CK1, CK2, CK6, CK7, CK41, CM1, CM2, CN1, CN2, CP1, CP2, CP7, CP8, CP51, CP52, CP53, CP54, CP55, CQ1, CQ2, CQ3, CQ21, CQ30, CU1, CV1, CV6, CW1, CX1, CY1, CZ1	CR0805	50	JMK212BC6476MG-T	YES, 47uF, >6V
0.47		CA10, CA11, CA13, CBB1, CBB2, CC6, CC7, CC8, CC12, CC13, CC14, CF52, CF55, CF57, CF62, CF65, CF67, CI4, CI5, CI11, CI12, CI16, CI17, CI19, CI46, CI47, CI51, CI52, CI54, CI76, CI77, CI81, CI82, CI84, CM6, CM7, CM8, CN6, CN7, CN8, CP6, CQ7, CQ8, CQ9, CQ11, CU2, CU6, CU7, CU8, CV2, CV7, CV8, CW5, CW6, CW7, CW8, CX6, CX7, CX8, CX22, CY5, CY6, CY8, CZ6, CZ7, CZ8, VY7	CR0402	67	JMK105B7474KV-F EMK105B7103KV-F	YES, 0.47uF, >6V
0.01uF	Capacitor (Semiconductor SIM Model), Capacitor	CC32, CCCF2, CCCF8, CCCF9, CCP51, CK4, CK05, CK5, CK05X, CK06, CK06X, CP01, CP02, CP03, CP04, CP05, CP06, CP07, CP08, CP09, CP10, CP11, CP12, CP13, CP14, CP15, CP16, CP17, CP18, CS14, CS15	0402	31		YES, 0.01uF
0.1uF	Capacitor (Semiconductor SIM Model)	CC40, CC41, CC42, CC43, CC44, CC45, CC46, CC47, CC48, CC49, CC87, CC88, CX01, CX02, CX03	0402	15	EMK105B7104KV-F	YES, 0.1uF, >6V
220uF		CS01, CS02, CS03, CS05, CS06, CS07, CS08, CS09, CS10, CS11, CS12, CS13	CPB	12	JMK325ABJ227MM-P	YES, >=220uF, >6V
DIODE_SMD_SOD-323	DIODE_SMD_SOD-323	Dbatt	SOD-323_DIO DE_SMD_SOD-323(Primary)	1	Nexperia USA Inc. BAS416	YES, low leakage
5.0A	FUSE 5A SLO BLO NANO 2 SMD	F3PCIE, F5USB, F12PCIE	NANO_FUSE	3	BEL FUSE INC, SST 5	Yes, >=5A
CONNECT68ST	68-pin high density connector	FP4, FP5	P50E068ST	2	3M P50E-068P1-S1-EA	NO
1.0uH	INDUCTOR 1.0UH 300MA 20% 0805	G14, LG11, LG12, LG13, LG15, LG16	0805	6	Abracon LLC, ASMC1-0805-1R0M-T	YES, 1.0uH, >300mA
10uH_3A	Inductor	LGMGT, LHCLK, LP33	INDC3216	3	Würth Electronics, 742792312	YES, >3A
Header 3	Header, 3-Pin	PFAN	HDR1X3	1	Amphenol ICC (FCI), 68001-103HLF	YES
Header 6X2	Header, 6-Pin, Dual row	PJTAG	HDR2X6	1	Amphenol ICC (FCI), 68001-103HLF x4	yes, four Header3

Mini8		Pm8	Mini8	1	Sullins Connector Solutions, GRPB042VWQS-RC	YES
PowerSupply		PowerB	LT-LGA-144-05-08-1816_V	1	Analog Devices, LTM4615EV#PBF	NO
NMH1205SC		PS1	NMH1205SC - duplicate	1	Murata Power Solutions, NMH1205SC	NO
Header 20X2	Header, 20-Pin, Dual row	PSNGL	HDR2X20	1	3M, N2540-6002-RB	YES, 40-pin 100 mil spacing
1.02K		PSW4, PSW5, PSW6, PSW7, RASEL, RBSEL, RC07, RC08, RCLK2, RCT2A, RCX03, RCX04, RDN1, RFCK, RFDA, RFDAB, RFMI, RFMIB, RFMP, RFMPB, RFMUSB, RJCLK, RJOE, RL01, RL02, RL03, RL04, RL05, RL06, RL07, RL08, RL09, RL10, RL11, RL12, RL13, RL14, RL15, RL16, RL17, RL18, RL19, RL20, RL21, RLD1, RLD2, RLD3, RLD4, RLED01, RLED02, RLED03, RP1, RP1x, RP4, RPD, RPG11, RPG12, RPG13, RRST, RXO1, RXO2, RXO3, RXO4	0402	63	ERA-2AEB1021X	YES< 0.5%
QSFP		QSFPA, QSFPB	QSFP38 cage	2	Amphenol ICC, FS1-R38-20A2-00	YES, QSFP+
10K		R110, R111, R112, R113, R114, R115, R116, R117, R118, R119, R120, R123, R124, R125, R126, R127, R128, R129, R130, R131, R132, R133, RCT3, RCT3A	0402	24	ERJ-2RKF1002X	YES, <=1%
100		R141, R142, R143, R144, R145, R146, R147, R148, R150, R151, R152, R161, R162, R163, R164, R165, R166, R167, R168, R171, R172, R173, R174, R175, R176, R177, R178, R181, R182, R183, R184, R185, R186, R187, R188, RAD, RIN0, RIN1, RIN2, RIN3, RIN4, RIN6, RIN7, RIN8, RM41, RM42, RM43, RM44, RM45, RM46, RPRT, RT32, RT33	0402	53	ERJ-2RKF1000X	YES, <1%
3.24K		RC01, RC02, RC03, RC04, RC05, RC06, RCT2, RCT49, RCT49A, RCT50, RCTR1, RCX05, RCX06, RM2, RM4, RM6, RM7, RM8, RM9, RM13, RM14, RM15, RM16, RM17, RM18, RM88A, RMB, RP51, RP51A, RPCIE, RSW	0402	31	ERA-2AEB3241X	YES, <0.5%
50		RC31, RC32, RC33, RC43, RC44, RC46, RC47, RC48, RC48a, RC49, RC101, RCLK, RCLK1, RCLK1A, RCT48, RJREF, RREF1, RREF2, RT46, RT47, RT49, RT50, RT52, RT53	0402	24	ERJ-2RKF51R0X	YES, <1%
150		RCT44, RCT45, RCT46, RCT47, RCT51, RCT52, RCT53, RCT54, RP1y, RT48, RT51, RT54, RXO5, RXO6	0402	14	ERJ-2RKF1500X	YES, <1%
LTSTRGB		RGB1, RGB2, RGB3, RGB4, RGB5, RGB8, RGB9, RGBP, RGBQA, RGBQB, RGBR	LTSTRGB	11	Lite-On Inc, LTST-C19HE1WT	NO
3.92K		RP6, RP16	0402	2	ERA-2AEB3921X	YES, <0.5%
78.7K		RP12	0402	1	Panasonic Electronic Components, ERA-2AEB7872X	YES, <0.5%
0		RT34, RT35, RT36, RT37, RT38, RT39, RTI02, RTI04, RTI06, RTI08, RTI12, RTI14, RTI16, RTI18, RTI22, RTI24, RTI26, RTI28, RTI32, RTI34, RTI36, RTI38	0402	22	ERJ-2GE0R00X	YES, 0 Ohm
SW DIP-8	DIP Switch, 8 Position, SPST	SC1	SW16_M	1	C&K, TDA08H0SB1	YES

					Micron Technology Inc.MT25QU512AB B1EW9-0SIT TR	YES, 1.8V, >128 Mb
SPROM		SPROM1, SPROM2	VPDFN8	2		
max9602E UG		TB0, TB1, TB2	MTC24_N	3	Maxim Integrated, MAX9602EUG	NO
TE_JJA		TE JJA	SOP4_JJA	1	TE Connectivity, JJAV0UJ253NONM RTR	YES,
AD9510B CP	AD9510	U13	CP-64-1_N	1	Analog Devices Inc.AD9510BCPZ	NO
XCKU040/XCKU5P-A676		U20	BGA676_1MM_27x27	1	Xilinx Inc. XCKU3P-1FFVA676E	NO
CCPD-034-250	CCPD-034, 250 MHz	UC01	SO6-5X7	1	EPSON, SG7050EAN 250.000000M-KEGA3	YES, 250 MHz, LVPECL
CCPD-034-156	CCPD-034, 156.25 MHz	UC03	SO6-5X7	1	EPSON, EG-2102CA 156.2500M-LGPAL0	YES, 156.25MHz, LVDS
MC100EP57DT	4:1 Differential Multiplexer	UC07	948E-02_L	1	ON Semiconductor, MC100EP57DTG	NO
SN74AVC8T245		ULED0, ULED1, ULVL1, ULVL2	TSSOP24_4.4MM	4	Texas Instruments, SN74AVC8T245PW	NO
DSLVD51047	TI LVDS driver	UO01, UO02	PW0016A	2	Texas Instruments, DSLVDS1047PWR	NO
MC100EP91M	On Semiconductor, Any level positive to ECL translator	UO03	QFN50P400X400-24W4M	1	ON Semiconductor, MC100EP91MNG	NO
TPS74401RGWT	IC LDO REG 3.0A W/SS 20-VQFN	UP4, UPG5	QFN-20	2	Texas Instruments, TPS74401RGWT	NO
kc2520b50	Miniature Oscillator	UPG1	XTAL_OSC_DFN_1.6X1.2MM	1	Kyocera International, KC2520B50.0000C10E00	YES, 50 MHz, 1.8V
LTM4604EV	LTM4604EV	UPUSB	LGA-66_LTM4604EV(P primary)	1	Analog Devices, LTM4604AEV#PBF	NO
USB_C		USBC	USBC	1	Molex, 1054500101	YES, USB-C connector
VCXO		VCXO	OSC3X5	1	EPSON, VG-4513CB 156.2500M-GGCT3	YES, VCXO

## **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 interface.
- TIpcieV5:** TIpcie using Xilinx Vertex-5 FPGA, manufactured in 2014.
- TIpcieUS:** TIpcie using Xilinx UltraScalePlus Kintex FPGA, manufactured in 2020.
- 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:** TI 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 boards 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 Gigabit per Second Transceiver; A built-in transceiver module in Xilinx FPGA. In XC5VLX30T FPGA, it supports up to 8 MGTs and up to 3.125 Gbps.

## Appendix E: TlpcieUS data format:

The TI 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(31:20): Trigger code received as the trigger (the code is decoded as trigger)  
Bit(19:16): trigger number bit(35:32), to form 36 bit counter with word2;  
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(31:16): fixed as hex 0xDA56;  
Bit(15:0): Latched front panel trigger code input. For TIpcieUS, bit(5:0) are used, which are the same as front panel input TS-code(6:1).

The data is aligned with the DMA interface of 256-bit. This means that the DMA data bits(255:64) are just filler bytes. Here is the DMA super word (256-bits):

Bit(255:192): fixed as hex 0x71E5,DA7A,5948,6921, except the Block trailer, Bit#199 = 1 (0x69A1).

Bit(191:128): TI data(63:0), two TI data words. It is the repeat of Bit(63:0).

Bit(127:64): fixed as hex 0x71E5,DA7A,5948, A521, except the Block trailer, Bit#71 = 1 (0xA5A1).

Bit(63:0): TIdata(63:0), two TI data words.

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(31:16): 64-bit word count of remaining data; bit(15:0): status of the readout

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:**

Nov. 2, 2020: First release, based on TI.docx and TIpacie(V5).docx;