Fermi National Accelerator Laboratory

Particle Physics Division
Beams Division

Tevatron

Ion Profile Monitor (IPM)
TCLK DECODER
Hardware Manual

T. Fitzpatrick

March 8th, 2005

Table of Contents

1. Design Overview

The IPM TCLK decoder card is a modified IPM clock card. The function of the card has been changed to offer a simple TCLK and MDAT decoder card that has interrupt capability on the PCI bus. At this time there is no output function of this card other than the PCI interface. Figure 1-1 is a block diagram of the Ion Profile Monitor (IPM) TCLK Decoder card. The card was designed to reside in a computer's standard PCI slot. The main function of the IPM TCLK Decoder card is to provide Accelerator information derived from TCLK and MDAT to the host computer via the PCI interface.

TCLK and MDAT are received on a LEMO connector at the back panel of the computer. The Altera FPGA contains two state machines that decode TCLK and MDAT using an oversampling 80Mhz clock derived from a 40MHz Oscillator. All decoded TCLK events are stored in a TCLK History RAM that holds a running count of each TCLK event. MDAT decoded data is stored in the MDAT RAM and is updated/overwritten on each MDAT sequence.

A TCLK Action RAM is implemented that allows the user to program a single or multi-level interrupt sequence. One interrupt line is defined in the PCI interface. The TCLK decoder card drives the INTA* line on the PCI bus until the interrupt handling software clears the interrupt by reading the interrupt vector register.

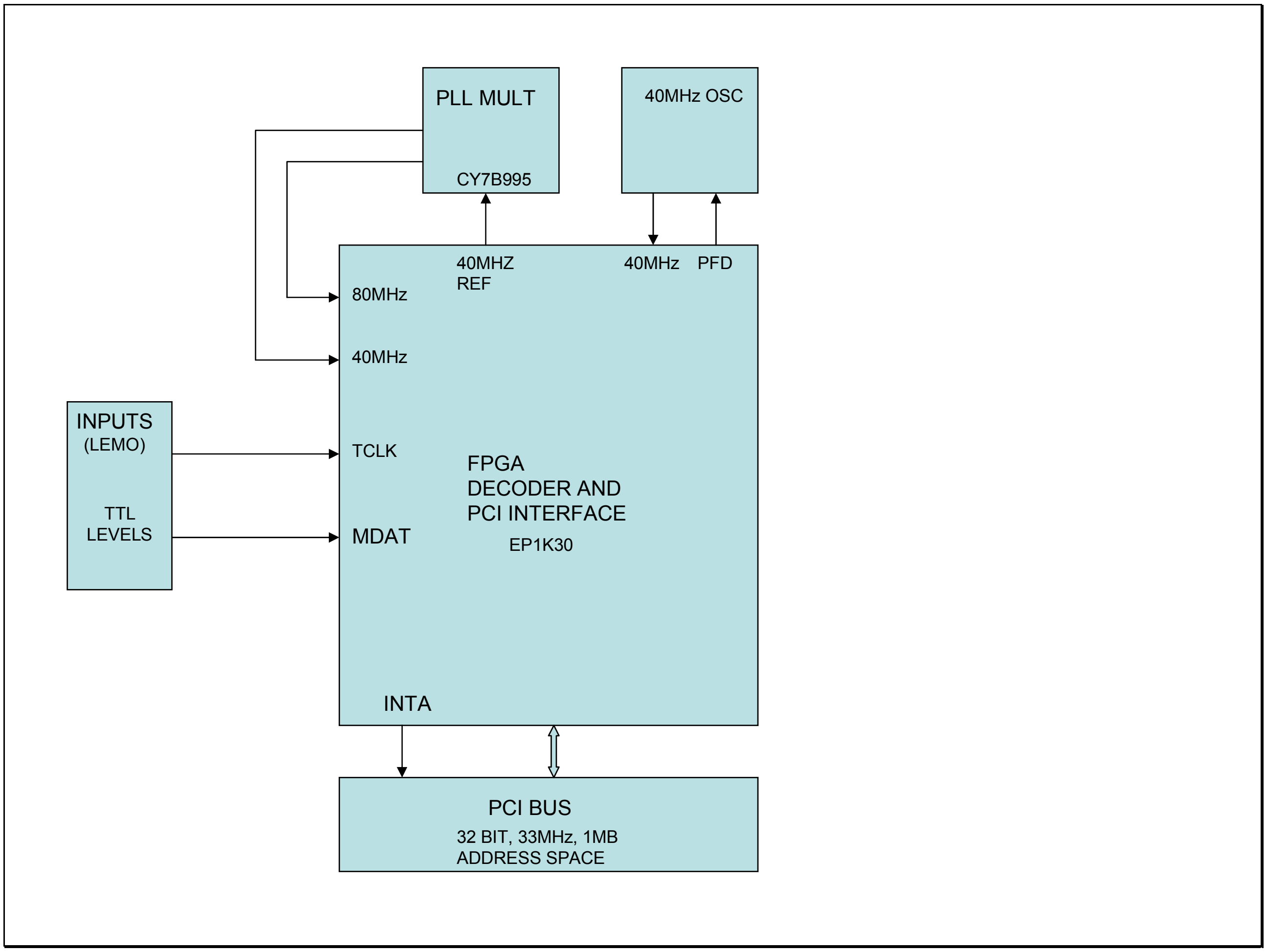

Figure 1-1 – IPM CLOCK CARD BLOCK DIAGRAM

2. PC Board Physical Layout Overview

The IPM TCLK card is designed to reside in a free 32 bit PCI slot of any standard personal computer. The PCI connector was notched to fit in a 5V or 3.3V PCI slot. Figure 1-2 is a picture of the IPM Clock card. The IPM TCLK decoder card has no output function, therefore no parts associated with the IPM Clock card outputs are attached. To the left are the rear panel input/output connectors. TCLK and MDAT are TTL level signals received on standard right angle LEMO connectors. The signals are terminated into 50 ohms and buffered by a 74LVT125. These encoded clocks are passed along to the Atera ACEX1K FPGA in the middle of the card. A 40MHz Oscillator and Cypress CY7C995 multi-phase buffer provide the sampling clock for the TCLK and MDAT decoding state machines. The FPGA code is held in the ALTERA EPC2 configuration device to the lower right of the FPGA. The EPC2 is downloaded via a standard JTAG 10 pin header and an ALTERA Byteblaster cable. A TI 73HD325 regulator to the right of the FPGA provides the 2.5V FPGA core voltage and power on reset.

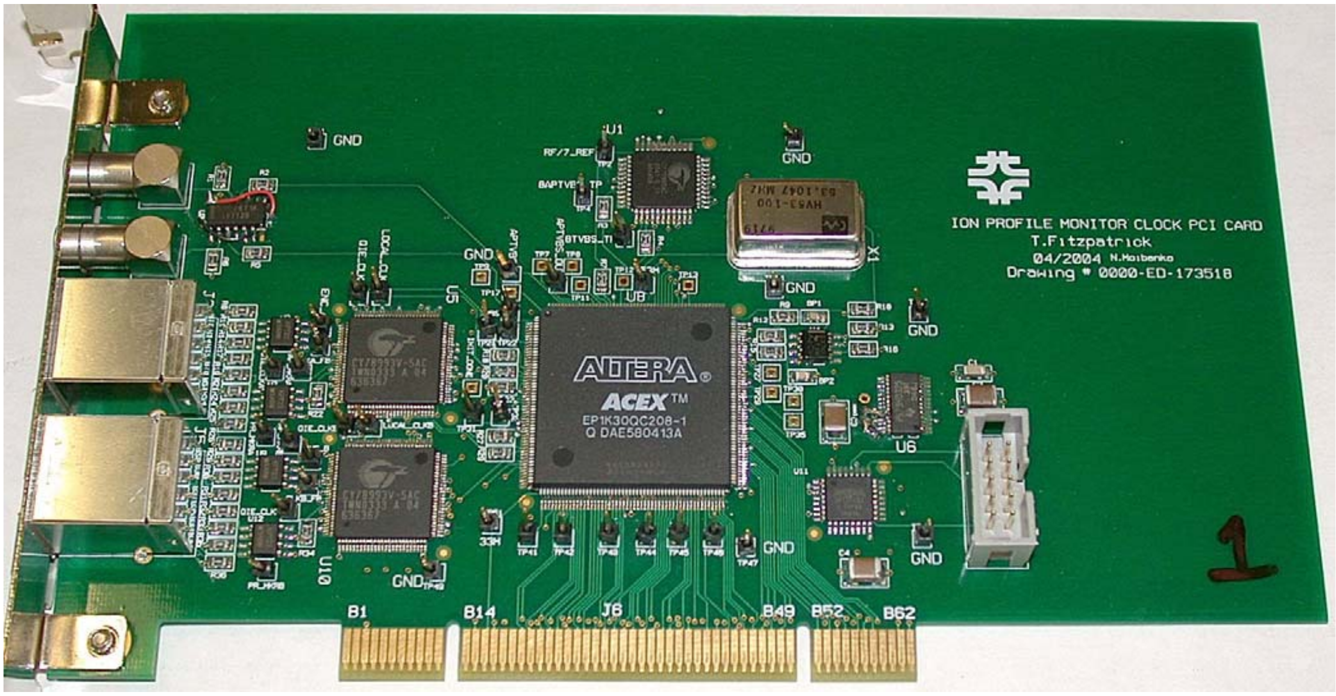

Figure 2-1 – IPM CLOCK CARD PICTURE

3. PCI Interface, Memory Map, and Register Function

The PCI interface for the IPM Clock card is implemented in the ACEX1K FPGA using an ALTERA IP core. The specific core used is the “pci_t32”, a target only, 32-bit, 33MHz device. Figure 1-3 outlines the registers and control functions contained within the 1MB of PCI address space allocated to the clock card. The actual base address is unique to each installation. The address space listed below is the offset from the base address.

Bar0	Function	Function
Offset	**Name**	**Description**
0x00	CSR0	Control Status Register (write 8 bits, read 24 bits)
0x1000-13FC	TCLK HISTORY RAM	TCLK History, records number of each TCLK events received
0x2000-23FC	TCLK ACTION RAM	TCLK Action, defines outputs and interrupts for TCLK events
0x3000	INTERRUPT	Force interrupt (Write), read interrupt vector, release interrupt
0x4000-43FC	MDAT MEMORY	Read MDAT event data
0x8000	SOFTWARE RESET	Board reset initiated from software

Figure 3-1 – PCI Memory Map

3.1. CSR0 Description

Control/Status Register (CSR0, 0x00) is a general purpose register used to enable/disable functions and read pertinent status from the card. CSR0 functions are outlined in figure 3-2 below.

CSR0 BIT	Function	CSR0 bit function description
0	TCLK DECODE ENABLE	1 = TCLK DECODE ENABLE, 0 = DISABLE
1	TCLK HISTORY RAM ENABLE	1 = TCLK HISTORY RAM ENABLE, 0 = DISABLE
2	TCLK ACTION ENABLE	1 = TCLK DECODED ACTION ENABLE, 0 = DISABLE
3	MDAT DECODE ENABLE	1 = MDAT DECODE ENABLE, 0 = DISABLE
7..4	RESERVED	RESERVED CSR BITS
8	ROBOCLOCK PLL LOCK	1 = PLL LOCKED
9	TCLK PARITY ERROR	1 = LATCHED TCLK PARITY ERROR
10	MDAT PARITY ERROR	1 = LATCHED MDAT PARITY ERROR
23..20	FPGA VERSION NUMBER	HIGHER NUMBER IS MORE RECENT VERSION

Figure 3-2 – CSR0 Function Bit Map

3.1.1. CSR0 Control Bit Functions (0x00)

There are 4 control functions associated with CSR0.

Bit 0 = 1 enables the TCLK decoding state machine. No interrupt or TCLK history function can occur when TCLK decoding is disabled.

Bit 1 = 1 enables the TCLK history RAM write process. No TCLK history will be recorded if the TCLK History RAM is disabled.

Bit 2 =1 enables the TCLK Action Ram interrupt processing. No interrupts will be generated if the TCLK Action process is disabled.

Bit 3 = 1 enables the MDAT Memory. MDAT events will not be updated if the MDAT memory write process is disabled.

Bits 4-7 are reserved for possible future applications.

3.1.2. CSR0 Status Bit definitions (0x00)

Status bits 8-23 are read only bits. Bit 8 is the lock status of the multi-phase PLL buffer on the board. This bit should always be 1. A constant zero on this bit indicates a clock problem.

Bits 9-10 monitor the serial parity of the TCLK and MDAT encoded data. Each TCLK and MDAT encoded event is ended with a parity bit. This simple parity is checked against the running decoding state machine parity. An error is latched if the parity bit comparison fails. A software reset must be issued to clear these error flags. Recurring latching of parity error indicates a problem with the card, incoming signal, or cable.

Bits 20-23 are a fixed four bit number corresponding to the FPGA firmware version number.

3.2. TCLK History RAM (0x1000-0x13FC)

The TCLK History RAM is a 32 bit, 256 location RAM that records the occurrence of each TCLK event. The address of the RAM corresponds to the TCLK event number. From the PCI perspective, each consecutive address is incremented by four to follow the PCI specification. As an example, the most frequent TCLK event, TCLK $07 (720Hz), is accessed at PCI address location 0x101C ($07 shift left 2 plus the base 0x1000 offset). Each time a TCLK $07 event is decoded, the TCLK History state machine looks up the current value at location 0x101C, increments the value by one, and writes this sum back to location 0x101C. With a 32 bit number, TCLK $07 can be recorded for 69 days without the History Ram overflowing. The value of TCLK $07 gives a relative time stamp of about 1.4mS. The entire History Ram should be zeroed at the beginning of a data taking run. CSR0 bit 1 should be set to zero (disabled) while the memory is zeroed.

A complete listing of all the currently defined TCLK events can be viewed at:

http://www-bd.fnal.gov/controls/hardware_vogel/tclk.htm

3.3. TCLK Action RAM (0x2000-0x23FC)

The TCLK Action RAM is a 16 bit, 256 location RAM that is currently configured to initiate a PCI interrupt based on one or more TCLK events. Multiple single TCLK event interrupts can be defined, but currently only one multi-level interrupt sequence can be defined. Only 9 out of the 16 bits of this RAM are used to define interrupts. Other bits are reserved for future use, possibly defining output sequences.

The address of the RAM corresponds to the TCLK event number. From the PCI perspective, each consecutive address is incremented by four to follow the PCI specification. As an example TCLK $29 is accessed at PCI address location 0x20A4 ($29 shift left 2 plus the base 0x2000 offset). Each time a TCLK $29 event is decoded, the TCLK Action state machine looks up the value at location 0x20A4 and uses this data to determine if any action is requested. Most TCLK events have zero in the Action RAM and the state machine does nothing.

The following example in figure 3-3 shows a simple TCLK $29 interrupt definition. Data bit 8 = 1 is the key bit that signals the state machine to immediately initiate an interrupt or start a multiple TCLK event interrupt sequence. Data bits 7..0 = $AA is the code that tells the state machine to immediately issue an interrupt. $AA was chosen since this is an undefined TCLK event.

TCLK EVENT	TCLK ACTION RAM ADDRESS	DATA BIT 8	DATA BITS 7..0
$29	$20A4	1	$AA

Figure 3-3 – Single TCLK event interrupt

Figure 3-4 shows an example of a multiple TCLK event interrupt sequence. In this example TCLK $21 is the key to the beginning of an interrupt sequence, as defined by bit 8 = 1. Unlike the above single TCLK event interrupt, $AA is NOT written to bits 7..0. Instead, $22 is written, and is interpreted as a pointer by the state machine to the next TCLK event in the interrupt sequence. In this case, TCLK $22 is the next event. When TCLK $22 is decoded, the Action state machine already is aware that it is in the middle of an interrupt sequence and looks for the code $AB on bits 7..0 to signify the end of a multiple TCLK event interrupt sequence. $AB was chosen since this is an undefined TCLK event. An interrupt is issued and the state machine returns to idle, looking for a TCLK event with bit 8 = 1. The length of the interrupt sequence is not expressly limited. One can define any sequence by following the simple formula:

First TCLK event in a sequence has bit 8 = 1 and data bits 7..0 pointing to the next TCLK event.

Any middle TCLK events have bit 8 = 0 and data bits 7..0 pointing to the next TCLK event.

Last TCLK event in a sequence has bit 8 = 0 and data bits 7..0 = $AB.

TCLK EVENT	TCLK ACTION RAM ADDRESS	DATA BIT 8	DATA BITS 7..0
$21	$2084	1	$22
$22	$2088	0	$AB

Figure 3-4 – Sequenced TCLK event interrupt

3.4. Force Interrupt, Read Interrupt Vector (0x3000)

The Interrupt register has two distinct functions. A write to this address location will force an immediate interrupt, regardless of the data written. The value of the data written is not stored. This function is used for software testing only. A read from the register during a normal interrupt will yield the 8 bit TCLK event number that caused the interrupt and will also clear the interrupt line. In the example above in figure 3-4, reading the interrupt vector would yield $22, the last TCLK event of the interrupt sequence.

3.5. MDAT Memory (0x4000-0x43FC)

The MDAT memory is a 16 bit, 256 location RAM used to store decoded MDAT event data. The address of the RAM corresponds to the MDAT event number. From the PCI perspective, each consecutive address is incremented by four to follow the PCI specification. Each time an MDAT sequence is received and decoded, the MDAT results are written to their respective memory locations. A history is not kept. Only the most recent MDAT data is available to read from PCI. The MDAT decoder state machine writes to the MDAT memory while the PCI interface only reads from the memory.

A complete listing of all the currently defined MDAT events can be viewed at:

http://www-bd.fnal.gov/controls/hardware_vogel/mdat.htm

3.6. Software Reset (0x8000)

The software reset command is issued when a write to address 0x8000 is executed. The data sent is ignored. There is no read function at this address. Software Reset returns state machines to their idle state and clears any latched errors. It also clears interrupt sequences that may be in progress or hung.

4. Altera FPGA code

The Altera ACEX1K30 FPGA is the main logic device on the IPM Clock card. It handles the PCI interface, clock and timing decoding state machines, clock synchronization, encoded bit state machine, and all coarse delay functions. The code is written in the Altera Hardware Description Language (AHDL). The bulk of the code is defined and commented in the file http://www-ppd.fnal.gov/EEDOffice-w/Projects/Beams_Div/Beams_IPM_Clock/pci_local_target_tclk.tdf. This file is best viewed within the Altera design environment but can be viewed using a standard text editor.