Nuclear Physics Division
Fast Electronics Group

Firmware for
VXS Crate Trigger Processor Module

10 March 2009

Hai Dong

Table of Contents

Section	Title	Page
1.0	Introduction	3
2.0	LV50 (U1, U3) VHDL Code	
3.0	LV110 (U24) VHDL Code	
4.0	I-SQUARE-C Code	
5.0	Total Delay	
6.0	Register File	

1.0 Introduction

The firmware for the Crate Trigger Module (CTP) is written in VHDL. The CTP module has two XC5VLV50 (U1, U3) that has same VHDL code and one XC5VLV110 (U24) that has its own VHDL code. LV50 VHDL code receives data from five FADC, aligns the data, and sends the data in sync with system clock to the LV110. The LV110 receives data from six FADC and both LV50, aligns the data, processes the data, and sends the result to Fiber located at the front panel.

2.0VLV50 VHDL Code

2.1 Receive data from FADC

The VHDL code receives data from five FADC using Xilinx Aurora operating at 2.5Gbits, aligns the data, and sends the data 80 bits synchronized with 250MHz clock. The Aurora protocol provides control and interface signals: Lane_up, Channel_up, Rx_src_rdy_n, Tx_src_rdy_n, Tx_dst_rdy_n. The functions for these signals are as follow:

- Lane_up : indicates FADC MGT and CTP GTP are able to send and receiving data from each other.
- Channel_up: indicates the data from the lanes if more than one are in alignment.
- Tx_dst_rdy_n: indicates that the transmitter is ready to send data
- Tx_src_rdy_n: tell the transmitter to send user data.
- Rx_src_rdy_n: indicates that the receiver has user data

2.2 Aligning Data

In the 12 Gev Trigger System, Sync is distributed in time to all sixteen FADC250V2 and CTP modules in a crate. When Sync goes high, FADC250 stops sending data and CTP reset all its circuitries and buffers. Sync has to be high for at least 500 nS to allow the MGT to completely flush its buffers. When Sync goes low, the FADC250 send value 2 for one 16 bits and value 1 for the other 16 bits on the 2 MGT lanes 0 and 1. Values 2 and 1 are sent 3 times. Which value is sent on lane 0 and 1 is depend on when the Sync arrived at the FADC250. The CTP uses these values to align data from all 16 FADC250 in time and words order. After aligning the data from 5 FADC250, U1 sends data to U24 via 80 LVDS lines. U3 does the same as U1. U24 aligns the data from 6 FADC250, wait for data from U1 and U3, processes, and sends data to SSP through fiber. First Word Fall Through FIFO is used to used to indicate when first values 2 and 1 are received.

2.3 Demultiplexed Data to VLX110 (U24)

The 32-bits data from each FADC is de-multiplexed into two 16-bits words with bits 31-16 send to U24 first. The 16-bits word is registered (F1_F3_D) and sends to U24 along with DatRdy. After Sync goes low, DatRdy goes high along with all values 2 output. DatRdy goes low when Sync goes high. The 250 MHz input clock is routed inside the FPGA to output pins (CLK_FPGA1_3). The LV110 uses CLK_FPGA1_3 to capture F1_F3_D and Datrdy.

2.4 Disable FADC

Data from FADC can be disabled by writing to Configuration register via I-square-C interface. When disabled, the code does not wait for data from that FADC250. The bits corresponded to that FADC are zero going.

2.5 Processing Delay

2.6 Xilinx System Monitor.

Xilinx System Monitor feature that indicates the die temperature and VCCINT and VCCAUX Supply is included. The data is mapped into register file that can be read back through I-Square-C. The host has to convert binary data to temperature and voltage (See Xilinx UG192).

3.0 VLX110 VHDL code

3.1 Receive data from FADC

The procedure for receiving data from six FADC is identical to VLX50.

3.2 Receive data from VLX50

When Datrdy is high, 80 bits data is clocked into 512_80 FIFO. Five-hundred words deep can handle 2uS skewing between the two VLX50. The number of clock from Datrdy going high to data FIFO empty going low is 8 (32 ns).

3.3 Aligning Data.

Aligning data from six FADC250 is the same as U1 and U3. When Datrdy from U1 goes high, U24 allows words from U1 to be written into First-Word-Fall-Through FIFO. It does the same with U3. When words values of 2 are at the output of these FIFO, the FIFO are read at the same time.

3.4 Summing Data.

The data is de-multiplexed and add in stages as shown below. The stages are used to make 250 MHz clocking using Xilinx FPGA lowest cost (speed grade 1).

3.4.1 First stage

- 1^{st}_A0 = VLX50 #1 bit (15..0), (31..16), (47..32)
- 1^{st}_A1 = VLX50 #1 bit (63..48), (79..64)
- 2^{nd}_A0 = VLX50 #2 bit (15..0), (31..16), (47..32)
- 2^{nd}_A1 = VLX50 #2 bit (63..48), (79..64)

2.1.2 Second stage

- 1^{st}_B = 1^{st}_A0 + 1^{st}_A1
- 2^{nd}_B = 2^{nd}_A0 + 2^{nd}_A1
- SUM_A = FADC_0 + FADC_1 + FADC_2
- SUM_B = FADC_3 + FADC_4 + FADC_5

2.1.3 Third stage

- 1^{st}_2^{nd} = 1^{st}_B + 2^{nd}_B
- SUMA_B = SUMA + SUMB

2.1.4 Fourth stage

- Final_SUM = 1^{st}_2^{nd} + SUMA_B

3.5 Multiplexing Final SUM

In the CTP the FIBER data consisted of 4 Aurora Lanes operating at 2.5Gibits. Each lane provides 16bits for a total of 64-bits every 8 NS. Since the 20-bits Final Sum is occurring at 4 NS, two Final Sum is packed (multiplexed) into one transmission. Final Sum # even is packed into bits 39 to 30 while Final Sum # odd is packed into bits 19 to 0. As of this date 03/10/09 bits 63..40 are set to zeroes.

3.6 Disable FADC

An FADC can be Data from FADC can be disabled by writing to Configuration register via I-square-C interface. When disabled, the code does not wait for data from that FADC. The bits corresponded to that FADC are zero in the summing process.

3.7 Processing Delay

The number of clock from Datrdy (from VLX50) going high to FIFO empty going low is 8 clock (32 nS). Adder stages take 11 clock (44 ns).

3.8 Threshold

A register accessed through I-Square-C provides a threshold for the final sum . When the final sum is above the threshold, Trigger Output at the front panel is active high. If the History Buffer described below is armed, 256 additional sums are stored and ready to be read out.

3.9 History Buffer

The History Buffer stores 512 Final Sum values. Halves of the values are before Final Sum cross the threshold and halves are after. It is controlled and read out via I-Square-C. The steps to use the history buffer are as follow:

- Host sets ARM to one.
- Host sets ARM to zero.
- After 256 Final Sum are stored in Memory, waiting for Final Sum to go above threshold. Continues storing Final Sum
- When Final Sum reach threshold. 256 more values are stored.
- Stops storing.
- Host polled for DataReady for high.
- Read as much values as host wanted. Data read back is auto increment.
- Host sets ARM to clear DataReady.

3.10 Automate Testing.

The integrity of the data transfer from FADC250 to CTP can be verified. When bit 1 of MGT_CTRL register in FADC250 is 0, counting sequences (0,1,2,3,4, etc,). The CTP adds the data from all enable FADC250 and compare to expected sum. For example, if 8 FADC250 are enable, the expected Sum are 0, 8, 16, 24, 32, etc. If at any time the actual Sum and the expected Sum are not equaled, ERROR_LATCH_FS variable goes high and stay high until Sync goes high. To use this feature:

- Make bit 1 of MGT_CTRL register in FADC250 zero.
- Bring Sync high for at least 500 nS
- Bring Sync low.
- Wait for however long the testing is desire.
- Read Is2 status register bit ERROR_LATCH_FS. If it is zero the data from the FADC250 are OK.
- Make bit 1 of of MGT_CTRL register in FADC250 one. ERROR_LATCH_FS should be one because data from FADC250 is not counting sequence.

3.11 Xilinx System Monitor

Xilinx System Monitor feature that indicates the die temperature and VCCINT and VCCAUX Supply is included. The data is mapped into register file that can be read

back through I-Square-C. The host has to convert binary data to temperature and voltage (See Xilinx UG192).

4.0 I-Square-C interface

Data is written to and read from the CTP serially from VXS P1 connector. The serial interface consists of an input clock bit and a bidirectional data bit that connected to all three FPGA. The FPGA de-serializes the data bits and listens for its address. The FPGA have non-overlapping address and responses when it is addressed. Each Register is 16 bits wide.

5.0 Total Processing Delay

It takes a total of 27 clocks (8 from VLX50, 8 to receive data from VLX50, 11 from adder stages) from the time data arrived from the slowest FADC to produce the first Final Sum.

6.0 LED

LED(0) <= HEART_BEAT_250 ;
LED(1) <= HEART_BEAT_200;
LED(2) <= ABOVE_THREDSHOLD_BUF_Q(0); -- ; Heartbeat 200
LED(3) <= indicate all MGT lanes and all MGT Channels of U24 are OK.
LED(4) <= HEART_BEAT_INIT;
LED(5) <= Indicate there is not Fiber Fault and Fiber is ready---
LED(6) <= '1';

7.0 SD Test Code

7.1 Count Rising Edge of Sync, Trig1, Trig2 from SD.

7.2 Measure Clock from SD (should be 250 MHz)

7.3 Connect SD_IN_SPARE 0 to SD_OUT_SPARE 0

7.4 Connect SD_IN_SPARE 1 to SD_OUT_SPARE 1

8.0 Register File

I-Square-C Address Mapping

I-Square-C Board Address	I-Square-C Sub Address	R/W	Function
0	0	R	1st VLX50 Status 0
	1	R	1st VLX50 Status 1
	2	R/W	1st VLX50 Config 0
	3	R/W	1st VLX50 Config 1
	4	R	1st VLX50 Die Temperature
	5	R	1st VLX50 Vint
	6	R/W	Undefine
	7	R/W	Undefine
1	0	R	2nd VLX50 Status 0
	1	R	2nd VLX50 Status 1
	2	R/W	2nd VLX50 Config 0
	3	R/W	2nd VLX50 Config 1
	4	R	2nd VLX50 Die Temperature
	5	R	2nd VLX50 Vint
	6	R/W	Undefine
	7	R/W	Undefine
2	0	R	VLX110 Status 0
	1	R	VLX110 Status 1
	2	R/W	VLX110 Config 0
	3	R/W	VLX110 Config 1
	4	R	VLX110 Die Temperature
	5	R	VLX110 Vint
	6	R/W	Undefine
	7	R/W	Undefine
	8	R/W	Final Sum Threshold LSB
	9	R/W	Final Sum Threshold MSB
	10	R	History Buffer Data LSB
	11	R	History Buffer Data MSB
	12	R/W	SD Test Control Register Bits: 0 : Reset SYNC Count Reg 14 1 : Reset TRIG 1 Count Reg 15 2 : Reset TRIG 2 Count Reg 16
	13	R	SD Clock Frequency (MHz) Bits (7.. 0) : Clock 250 Count (Should be 250 +/- 2)
	14	R	Count Rising Edge of Sync From

			SD **Bits (15:0) Count Rising Edge of SYNC from SD. Sync must be high for at least 30 nS**
	15	**R**	**Count Rising Edge of Trig1 from SD** **Bits (15:0) Count Rising Edge of TRIG1 from SD. Sync must be high for at least 30 nS**
	16	**R**	**Count Rising Edge of Trig2 from SD** **Bits (15:0) Count Rising Edge of TRIG2 from SD. Sync must be high for at least 30 nS**

Register Definition:

- Status 0
  - Bit 0 : FADC 0 LANE 0 up
  - Bit 1 : FADC 0 LANE 1 up
  - Bit 2 : FADC 1 LANE 0 up
  - Bit 3 : FADC 1 LANE 1 up
  - Bit 4 : FADC 2 LANE 0 up
  - Bit 5 : FADC 2 LANE 1 up
  - Bit 6 : FADC 3 LANE 0 up
  - Bit 7 : FADC 3 LANE 1 up
  - Bit 8 : FADC 4 LANE 0 up
  - Bit 9 : FADC 4 LANE 1 up
  - Bit 10: FADC 5LANE 0 up VLX110 only
  - Bit 11: FADC 5LANE 1 up VLX110 only
  - Bit 12: FADC 0 Channel Up
  - Bit 13: FADC 1 Channel Up
  - Bit 14: FADC 2 Channel Up
  - Bit 15: FADC 3 Channel Up
- Status 1
  - Bit 0: FADC 4 Channel Up
  - Bit 1: AllChanUp_BUF_Q
  - Bit 2: FADC 5 Channel Up, VLX110 only
  - Bit 3: History Buffer Data Ready, VLX110 only
  - Bit 4: FIBER_LANE_CH_ALIGN, VLX110 only
  - Bit 5: FIBER_LANE_BYTE_ALIGN, VLX110 only
  - Bit 6: FIBER_LANE_REMOTE_UP, VLX110 only
  - Bit 7: FIBER_GTX_CHANNEL_RDY, VLX110 only
  - Bit 8: ERROR_LATCH_FS U24 only
  - Bit 15-9 : Firmware Version Number
- Configuration 0

    - Bit 0: Unable FADC in Payload 1 in sum
    - Bit 1: Unable FADC in Payload 2 in sum
    - Bit 2: Unable FADC in Payload 3 in sum
    - Bit 3: Unable FADC in Payload 4 in sum
    - Bit 4: Unable FADC in Payload 5 in sum
    - Bit 5: Unable FADC in Payload 6 in sum
    - Bit 6: Unable FADC in Payload 7 in sum
    - Bit 7: Unable FADC in Payload 8 in sum
    - Bit 8: Unable FADC in Payload 9 in sum
    - Bit 9: Unable FADC in Payload 10 in sum
    - Bit 10: Unable FADC in Payload 11 in sum
    - Bit 11: Unable FADC in Payload 12 in sum
    - Bit 12: Unable FADC in Payload 13 in sum
    - Bit 13: Unable FADC in Payload 14 in sum
    - Bit 14: Unable FADC in Payload 15 in sum
    - Bit 15: Unable FADC in Payload 16 in sum
    -
- Configuration 1 (VLX110)
    - Bit 0 : Arm History Buffer
    - Bit 1 : Reset All GTP connected to PayLoad Port
    - Bit 2 : Reset GTP connected to SSP
    - Bit 15-3 : Fixed Latency Delay

The history data advance after Data MSB is read. Hence read LSB then MSB.

Channel Number to PayLoad Mapping:

FPGA	**FADC # Channel**	**Payload**
U24	0	3
	1	1
	2	5
	3	2
	4	4
	5	6
U1	0	7
	1	9
	2	11
	3	13
	4	15
U3	0	8
	1	10
	2	12
	3	14
	4	16


```
Temperature_C = ((float)TempRegValue * 503.975/1024) - 273.15;
```