Target 7 Evaluation board interface write-up

4/11/2014

Very initial, copy of Target 5 with some development for Target 7

1 PURPOSE

This document describes the user interface for the Target 5 Evaluation board (T5EV) with communication over Ethernet UDP protocol

2 SCOPE

This document gives an overview of the user interface of the Target 5 Evaluation board (T5EV)

3 INTRODUCTION

T5EV is designed to communicate and control acquisitions in current version using standard 1Gbit Ethernet connection. It uses UDP protocol to send and receive packets. Also, for image data it can send packets (one packet per image pixel) with maximum size smaller then Jumbo packet.(No support for Jumbo packets required)

4 Requirements:

1. Support Processing Command/Image data from 1 Target 5 ASIC.
2. TBD.

5 Memory Map Demo Baord:

5.1 Master Board FPGA Memory Map

Registers:

1. FpgaVersion
2. Scratch pad
3. Serial ID LSW
4. Serial ID MSW
5. Status register
6. Latched Status register
7. FIFO Status register
8. Latched FIFO Status register
9. Trigger statistics
10. FIFO statistics
11. Packet statistics
12. Ramp count statistics
13. Monitor results register
14. Monitor control register
15. Time register
16. Control register
17. Trigger control register
18. Row/Column control/status
19. ROVDD feedback control

20. Special feature register
21. VDEL/ROVDD calculated vlaues
22. VPED DAC control
23. Enable sample readout 31-0
24. Enable sample readout 63-32
25. Enable channel readout
26. Packet count enable register
27. Unused
28. Unused
29. Unused
30. Unused
31. Mode register
32. Feedback register
33. Target configuration register
34. SST feedback control
35. SST feedback DCM control register.

FPGA version: Address 0x0

Bits	Function	R/W	Default
31-0	Assigned value of 0xFED000003, assigned in Firmware	R	0

Figure 1: FPGA Version: Address 0x0

Scratch pad: Address 0x1

Bits	Function	R/W	Default
31-0	Any value for control software, does not have effect on any FPGA logic	RW	0

Figure 2: Scratch pad: Address 0x1

Serial number LSW: Address 0x2

Bits	Function	R/W	Default
31-0	Serial number the least significant word	R	0

Figure 3: Serial number LSW: Address 0x2

Serial number MSW: Address 0x3

Bits	Function	R/W	Default
31-0	Serial number the most significant word	R	0

Figure 4 : Serial number MSW: Address 0x3

Status register: Address 0x4

Bits	Function	R/W	Default
31-16	Fifo packet errors (one per channel), bit 16 – channel 0, bit 17 – channel 1, … Indicate that ether packet count underflow or overflow	R	0
15	State of event Done bit	R	0
14	unused	R	0
13	Packet size error, one for all channel due to simultaneous processing	R	0
12	mgt_AVTT_OK is OK, 1- OK, 0 – is not	R	0
11	mgt_AVCCPLL_OK is OK, 1- OK, 0 – is not	R	0
10	mgt_AVCC_OK is OK, 1- OK, 0 – is not	R	0
9	+1_8V is OK, 1- OK, 0 – is not	R	0
8	+2_5V VIO to Target 1 is OK, 1- OK, 0 – is not	R	0
7	OT from FPGA monitoring, see FPGA monitoring section	R	0
6	EOS from FPGA monitoring, see FPGA monitoring section	R	0
5	EOC from FPGA monitoring, see FPGA monitoring section	R	0
4	DRDYfrom FPGA monitoring, see FPGA monitoring section	R	0
3	BUSY from FPGA monitoring, see FPGA monitoring section	R	0
2	USER_TEMP_ALARM_OUTfrom FPGA monitoring, see FPGA monitoring section	R	0
1	VCCINT_ALARM_OUT from FPGA monitoring, see FPGA monitoring section	R	0
0	VCCAUX_ALARM_OUT from FPGA monitoring, see FPGA monitoring section	R	0

Figure 5: Status register: Address 0x4

Latched Status register: Addressw 0x5

Bits	Function	R/W	Default
31-16	Fifo packet errors (one per channel), bit 16 – channel 0, bit 17 – channel 1, … Indicate that ether packet count underflow or overflow	RW*	0
15	State of event Done bit	RW*	0
14	unused	RW*	0
13	Packet size error, one for all channel due to simultaneous processing	RW*	0
12	mgt_AVTT_OK is OK, 1- OK, 0 – is not	RW*	0
11	mgt_AVCCPLL_OK is OK, 1- OK, 0 – is not	RW*	0
10	mgt_AVCC_OK is OK, 1- OK, 0 – is not	RW*	0

9	+1_8V is OK, 1- OK, 0 – is not	RW*	0
8	+2_5V VIO to Target 1 is OK, 1- OK, 0 – is not	RW*	0
7	OT from FPGA monitoring, see FPGA monitoring section	RW*	0
6	EOS from FPGA monitoring, see FPGA monitoring section	RW*	0
5	EOC from FPGA monitoring, see FPGA monitoring section	RW*	0
4	DRDYfrom FPGA monitoring, see FPGA monitoring section	RW*	0
3	BUSY from FPGA monitoring, see FPGA monitoring section	RW*	0
2	USER_TEMP_ALARM_OUTfrom FPGA monitoring, see FPGA monitoring section	RW*	0
1	VCCINT_ALARM_OUT from FPGA monitoring, see FPGA monitoring section	RW*	0
0	VCCAUX_ALARM_OUT from FPGA monitoring, see FPGA monitoring section	RW*	0

Figure 6: Latched Status register: Address 0x5

* - Writing one will reset corresponding bit

FIFO Status register: Address 0x6

Bits	Function	R/W	Default
31-0	Data Storage FIFO underflow(bits 0,2,4,.) and overflow (1,3,5,7,..) errors. Two bits per acquisition channel. Channel 0 errors reported in bits 0(underflow) and 1(overflow), channel 1 in bits 2 and 3,….	R	0

Figure 7: FIFO Status register: Address 0x6

Latched FIFO Status register: Address 0x7

Bits	Function	R/W	Default
31-0	Data Storage FIFO underflow(bits 0,2,4,.) and overflow (1,3,5,7,..) errors. Two bits per acquisition channel. Channel 0 errors reported in bits 0(underflow) and 1(overflow), channel 1 in bits 2 and 3,….	RW*	0

Figure 8: Latched FIFO Status register: Address 0x7

* - Writing one will reset corresponding bit

Trigger statistics: Address 0x8

Bits	Function	R/W	Default
31-16	Count trigger low to high transitions on all enabled for counting (bits 11-6 of register 0x10) incoming triggers (4 from Target 2, 1 software trigger(bit 31 of register 0x10), and 1 external hardware trigger). Trigger transition observed on same clock counted as one trigger. Counter is reset by writing value of 1	R	0

	into bit 15 of register 0x10		
15-0	Count trigger low to high transitions on all incoming triggers (4 from Target 2, 1 software trigger(bit 31 of register 0x10), and 1 external hardware trigger). Trigger transition observed on same clock counted as one trigger. Counter is reset by writing value of 1 into bit 15 of register 0x10	R	0

Figure 9: Trigger statistics: Address 0x8

FIFO statistics: Address 0x9

Bits	Function	R/W	Default
31-16	Count all enabled for counting (bits 15-0 of register 0x19) built packets on all incoming channels. Counter is reset by writing any value into register 0xA	R	0
15-0	Count all built packets on all incoming channels. Counter is reset by writing any value into register 0xA	R	0

Figure 10: FIFO statistics: Address 0x9

Packet statistics: Address 0xA

Bits	Function	R/W	Default
31-16	Internal Arbiter Counter, count all packet forwarded from internal storage to MAC. Counter is reset by writing any value into register 0xA	RW*	0
15-0	MAC Counter, count all transmitted packet by MAC. Counter is reset by writing any value into register 0xA	RW*	0

Figure 11: Packet statistics: Address 0xA

* Writing any value into this register will reset all counters in registers 0x9,0xA, and some on 0xB

Ramp count statistics: Address 0xB

Bits	Function	R/W	Default
31-16	Count command issued to Evaluation board. Counter is reset by writing any value into register 0xA	R	0
15-0	Count number of ramp transition. Counter is reset by writing value of 1 into bit 15 of register 0x10	R	0

Figure 12: Ramp count statistics: Address 0xB

Monitor results register: Address 0xC

Bits	Function	R/W	D

			efault
31	Always 1	R	0
30-28	Unused	R	0
27-23	Unused	R	0
22-16	Data address for FPGA monitoring interface, DADDR on page 13 of http://www.xilinx.com/support/documentation/user_guides/ug192.pdf	R	0
15-0	Output data for FPGA monitoring interface, DO on page 13 of http://www.xilinx.com/support/documentation/user_guides/ug192.pdf	R	0

Figure 13: Monitor results register: Address 0xC

Monitor control register: Address 0xD

Bits	Function	R/W	Default
31	Data write for FPGA monitoring interface, DWE on page 13 of http://www.xilinx.com/support/documentation/user_guides/ug192.pdf	RW	0
30-28	Unused	R	0
27-23	Unused	RW	0
22-16	Data address for FPGA monitoring interface, DADDR on page 13 of http://www.xilinx.com/support/documentation/user_guides/ug192.pdf	RW	0
15-0	Input data for FPGA monitoring interface, DI on page 13 of http://www.xilinx.com/support/documentation/user_guides/ug192.pdf	RW	0

Figure 14: Monitor control register: Address 0xD

Time register: Address 0xE

Bits	Function	R/W	Default
31-16	Free running time counter value starting to run from value in bits 15-0 at the load	R	0
15-0	Time counter start value	RW	0

Figure 15: Time register: Address 0xE

Control register: Address 0xF

Bits	Function	R/W	Default
31-30	Few function depending on bit combination: 00 - normal operation 01 - Special mode to test trigger pattern 10 normal mode but clr always 0	RW	0

	11 normal mode but clr always 1		
29-28	Specify ramp signal duration addition to Target 5 in addition to bits26-24 and 15-8 to study AC noise	RW	0
27	Enable to simulate test input to Target ASIC. 0- disabled, 1- enabled. Output waveform at half system clock frequency, or 62.5MHz.	RW	0
26-24	Specify ramp signal duration addition to Target 2 in addition to bits 15-8	RW	0
23	Special test sampling mode enable. Wr_strb = 1, SSP=1, SST=1, wr_en=1	RW	0
22	Control assertion of SR_sel (0 – only during event, 1 – all the time)	RW	0
21	Invert sense of select_any to be able to debug event with known pattern (0-normal, 1 – inverted)	RW	0
20	Directly control clr_reg input to Target 2	RW	0
19-16	Specify number of buffer for readout, with 0 corresponding to 1 buffers With 1- 2 buffers With 15- 16 buffers (maximum) If value set to more than 1 bit 2 of same register (offset usage) is ignored	RW	0
15-8	Specify ramp signal duration to Target 2 , high order bits 26-24	RW	0
7-3	Ramp start delay after trigger, in system clocks (8 ns)	RW	0
2	Offset usage enable (1). In order to optimize 48 samples readout relatively to trigger position, logic implement option to sync readout to 8 sample bins (4 bins per buffer). Setting bit to one will enable this feature. Otherwise buffer readout always start from beginning. Should be noted that that setting number of readout buffer to more then 2(Bits 19-16) will disable offset usage. And for software buffer this feature disabled as well (bit 9 of 0x11)	RW	0
1	Disable stoppage of analog sampling during ADC conversion and readout, 1 –disable(analog sampling run continuously)	RW	0
0	Enable bit, 1- enable analog sampling, 0 -disable	RW	0

Figure 16: Control register: Address 0xF

Trigger control register: Address 0x10

Bits	Function	R/W	Default
31	Software trigger	RW	0
30	SST acquisition trigger. Trigger event like response which will deliver SST calculated data. Used for diagnostic of SST feedback logic	RW	0
29	Special trigger mode. When set to 1 enable J5 output generate pulse on every sampling buffer turn for 10 event and stop. To reactivate need to bring it back to 0, if 0 - trigger sync command generated ~ at 1kHz	RW	0
28-16	Trigger delay, to compensate for time between trigger arrival and trigger decision making, need to be found experimentally. Count in 8ns steps	RW	0
15	Reset bit, reset trigger(register 0x8) and ramp(register 0xb bits 15-0) counters	RW	0
14-12	Unused	RW	0

11-6	Enable trigger input contribution to trigger counter(register 0x8 bits 31-16). Bits assigned as followed: 0 – Target 2 trigger bit 0, 1 – Target 2 trigger bit 1, 2 – Target 2 trigger bit 2, 3 – Target 2 trigger bit 3,4 – external hardware trigger, 5 – software trigger	RW	0
5-0	Enable trigger input. Bits assigned as followed: 0 – Target 2 trigger bit 0, 1 – Target 2 trigger bit 1, 2 – Target 2 trigger bit 2, 3 – Target 2 trigger bit 3,4 – external hardware trigger, 5 – software trigger	RW	0

Figure 17: Trigger control register: Address 0x10

Row/Column control/status: Address 0x11

Bits	Function	R/W	Default
31-30	Unused	RW	0
29-27	Latched by latest trigger value of row counter	R	0
26-21	Latched by latest trigger value of column counter	R	0
20-19	Latched by latest trigger value of sample counter (4 samples per buffer for finer data extraction)	R	0
20-19	Latched by trigger value of sample counter (4 samples per buffer for finer data extraction)	R	0
18-16	Free running current row value to simulate sampling buffer write row pointer	R	0
15-10	Free running current column value to simulate sampling buffer write column pointer	R	0
9	Specify if row and column for readout computed by FPGA internal logic(0) or bits 8-0 of this register are used	RW	0
8-3	Specify column for readout when trigger received and bit 9 of this register set to 1.	RW	0
2-0	Specify row for readout when trigger received and bit 9 of this register set to 1.	RW	0

Figure 18: Row/Column control/status: Address 0x11

ROVDD feedback control: Address 0x12

Bits	Function	R/W	Default
31-20	Set VDD value for ROVDD feedback loop, this value should be as close as possible to final desired sampling frequency. Since RCO is product of both VdlyP and VdlyN settings on Target ASIC, technically we can try to control both. But from practical point only VdlyN control sampling frequency, so we should use only VdlyN to control RCO frequency to adjust sampling. Default value for 1GHz sampling is 0x910, therefore to control VdlyN we need to invert this value to 0x6ef as starting RCO feedback value. Similarly for 400MHz sampling we have 0xBC0, so we have to set feedback initial value to 0x43f. VdlyN Feedback enabled by bit 1 of the register 0x1E.	RW	0

	VdlyP Feedback enabled by bit 2 of the register 0x1E, should not be activated.		
19-0	Compare value for ROVDD feedback loop. This is actual value which effect RCO final frequency. To find rough value we need to multiply desired RCO period in ns by 256. It is a little tricky task, because RCO runs faster than desired sampling frequency, so we need to target slower value of RCO frequency to get desired frequency. To have 1GHz sampling we need to set RCO roughly to 1GHz/128 = `7.8MHz(or period of 128ns), however in practice real 1GHz sampling achieved at 5.95MHZ (or 168ns) given that VdlyP set to value of 0x680 (to make RCO closer to 50% duty cycle). In this case value of this register need to be set 168*256=43008 (or 0xa800). Similarly for 400MHz sampling we need to set this value to about 1.88MHz(531ns) or value of 256*531= ~0x212f0 (need to be verified for 400MHz sampling)	RW	0

Figure 19: ROVDD feedback control: Address 0x12

Special feature register: Address 0x13

Bits	Function	R/W	Default
31-9	Unused	R/W	0
8	Select buffer enable logic. With current sampling timing this bit need to be set to 1. All odd buffers will get incremented earlier and as result special counting as increment by 3 and decrement by 1 need to be used (Set to 1)	R/W	0
7-5	Unused	R/W	0
4	Select if done signal is used to speed up conversion (1 – done is enabled)	R/W	0
3	Unused	R/W	0
2-0	Select Wilkinson clock frequency divided by 2: 000 – 250MHz 001 – 104MHz 010 – 250MHz 011 – 62.5MHZ 100 – 250MHz 101 – 250MHz 110 – 250MHz 111 – 250MHz	R/W	0

Figure 20: Special feature register: Address 0x13

VDEL/ROVDD calculated values: Address 0x14

Bits	Function	R/W	Default
31-28	Unused	R	0
27-16	Computed ROVDD (Vdelp) value, Vdeln = NOT(Vdelp)	R	0
15-12	Unused	R	0
11-0	Computed VDEL DAC value	R	0

Figure 21: VDEL/ROVDD calculated values: Address 0x14

VPED DAC control: Address 0x15

Bits	Function	R/W	Default
31-12	Unused	RW	0
11-0	VPED value from 0 to 2.5V	RW	0

Figure 22: VPED DAC control: Address 0x15

Enable sample readout 31-0: Address 0x16

Bits	Function	R/W	Default
31-5	Unused	RW	0
4-0	Partial buffer, from 0 to 31 samples. Going in combination with number of buffers in register 0xf(bits19-16)	RW	0

Figure 23: Enable sample readout 31-0: Address 0x16

Enable sample readout 63-32: Address 0x17

Bits	Function	R/W	Default
31	Enable special mode (1) to have two set of timed responses on single trigger request. Useful only in software readout mode. All other bit in this register have meaning only if this bit set to 1. Otherwise (0) ir is normal operation	RW	0
30-25	Unused	RW	0
24-9	Delay between two internal triggers plus 36 clocks (in clock cycles). This delay and selection of alternative buffers in bit8-0 enable user to study effect of different intervals of Wilkinson digitization and Serial output on quality of sampled data.	RW	0
8-3	Alternative read column for special mode of operation to test effect of digitization on sampling. Data read from this location and effect on other buffers during sampling will be studied	RW	0
2-0	Alternative read row for special mode of operation to test effect of digitization on sampling. Data read from this location and effect on other buffers during sampling will be studied	RW	0

Figure 24: Enable sample readout 63-32: Address 0x17

Enable channel readout: Address 0x18

Bits	Function	R/W	Default
31-16	Unused	RW	0
15-0	Enable data inputs, one per data line (1 –enable)	RW	0

Figure 25: Enable channel readout: Address 0x18

Packet count enable register: Address 0x19

Bits	Function	R/W	Default

31-16	Unused	RW	0
15-0	Packet count enable for data inputs, one per input. Effect calculations of packet count in 0x9 bits 31-16 (enable 1)	RW	0

Figure 26: Packet count enable register: Address 0x19

VDEL DAC control: Address 0x1A -- UNUSED

Bits	Function	R/W	Default
31-12	Unused	RW	0
11-0	VDEL value from 0 to 2.5V , UNUSED , related to Target 2/3 development	RW	0

Figure 27: VDEL DAC control: Address 0x1A

VDISCHARGE DAC control: Address 0x1B -- UNUSED

Bits	Function	R/W	Default
31-12	Unused	RW	0
11-0	VDISCHARGE value from 0 to 2.5V , UNUSED , related to Target 2/3 development	RW	0

Figure 28: VDISCHARGE DAC control: Address 0x1B

ISEL DAC control: Address 0x1C -- UNUSED

Bits	Function	R/W	Default
31-12	Unused	RW	0
11-0	ISEL value from 0 to 2.5V , UNUSED , related to Target 2/3 development	RW	0

Figure 29: ISEL DAC control: Address 0x1C

Mode register: Address 0x1D

Bits	Function	R/W	Default
31	Force special mode to force readout from random buffer when sampling performed from desired buffer -- Unused	RW	0
30-22	Specify offset from desired buffer in buffer counts. On trigger request Wait until sampling reach this special buffer and start to read after sampling completed. This operation force readout during desired buffer sampling to study effect of readout on sampling -- Unused	RW	0
21-20	Sampling mode selection: -- There are 3 options for 1G sampling option tracking hold wr_strb 00 8 16 8 -- 01 8 8 16 -- 10 16 8 8 -- There are 1 options for 0.4G sampling	RW	0

	-- option tracking hold wr_strb -- 11 16 48 16		
19-16	Unused	RW	0
15	To study drooping effect of sample 31(32nd) we can read samples in increasing (0) or decreasing (1) order.	RW	0
14-8	Adjust event serial data input delay to make sure all channels properly clock input data	RW	0
7-3	Unused	RW	0
2	Select options when buffer never incremented, stay same as start one	RW	0
1	Select if Sel signal asserted all times (1) or only when actual data transferred	RW	0
0	To select data tacking buffer from fixed (1) or computed in relation to trigger timing and related delays -- Unused	RW	0

Figure 30: Mode register: Address 0x1d

Feedback register: Address 0x1E

Bits	Function	R/W	Default
31-23	Unused	R	0
22	Captured bit of Target 5 configuration, Select output between SST/SSP (0) and RCO(1)	R	0
21	Captured bit of Target 5 configuration, Select output between SST and SSP , 0 – SST, 1 – SSP | 0 – SSP_RCO, 1 – SSt_RCO	R	0
20	Captured bit of Target 5 configuration, Select additional load capacitor for sampling logic, 0 – 1GHz sampling, 1 – 400MHz sampling	R	0
19-5	Unused	R	0
4	Unused	RW	0
3	Unused	RW	0
2	Unused	RW	0
1	VdlyN feedback on (1)/off(0) control	RW	0
0	Vdly feedback on (1)/off(0) control	RW	0

Figure 31: Feedback register: Address 0x1E

Configuration waveform register: Address 0x1f

Bits	Function	R/W	Default
31-24	PCLK width plus 1 when SIN high, need to be set to 0x3	RW	7
23-16	SIN settling time after PCLK plus 1, need to set to 0	RW	20
15-8	SIN settling time before PCLK plus 1, need to set to 0	RW	20

7-0	PCLK width plus 1 when SIN low, need to be set to 0x7	RW	7

Figure 32: Configuration waveform register: Address 0x1F

SST feedback control register: Address 0x21

Bits	Function	R/W	Default
31-20	Computed DAC value for VdlyP/N control as result of SST control	R	0
19-16	Select computational compare value for feedback control, nominally should be 0x8	RW	0x8
15-12	Select computational compare hysteresis value for feedback control, nominally should be 0x1	RW	0x1
11-0	SST setup value. Starting value for SST feedback control (should be same as for RCO control)	RW	0

Figure 33: SST feedback control register: Address 0x21

SST feedback DCM control register: Address 0x22

Bits	Function	R/W	Default
31-20	Value recorded into PLL delay	R	0
19-15	Unused	R	0
14	RCO sampled with PLL clock and re-sampled with system clock to find proper edge (low to high), ignore of high to low.	R	0
13	Captured bit of Target 5 configuration, Select output between SST/SSP (0) and RCO(1)	R	0
12	Captured bit of Target 5 configuration, Select output between SST and SSP , 0 – SST, 1 – SSP | 0 – SSP_RCO, 1 – SSt_RCO	R	0
11	Captured bit of Target 5 configuration, Select additional load capacitor for sampling logic, 0 – 1GHz sampling, 1 – 400MHz sampling	R	0
10-9	Unused	R	0
8	PLL setup value to shift direction. Every update to this register will reset PLL to 0 shift position and will shift in proper direction for proper amount specified in bits 7-0. Reset done to avoid keeping bookkeeping of direction and amount for every reload.	RW	0
7-0	PLL setup value to shift sampling clock timing to align with SST input (from Target) for proper SST type feedback control. Actual value range is from 0 to 0xFF. Each step is about 125ps (estimate)	RW	0

Figure 34: SST feedback DCM control register: Address 0x22

SST feedback PLL control register: Address 0x23 -- UNUSED

Bits	Function	R/W	Default
31	Specify read (0) or write (1) operation for PLL register	RW	0

30-27	Encoded Address of PLL registers -- My address[3:0] DADDR[4:0] (BIN) Usage DI[15:0] Value -- 000 01100 CLKFBOUT 000X000100000100 -- 000 01101 CLKFBOUT XXXXXXXX00000000 -- 001 11011 CLKOUT0 111X101000101000 -- 001 11100 CLKOUT0 XXXXXXXX00111111 -- 010 11001 CLKOUT1 000X000100000100 -- 010 11010 CLKOUT1 XXXXXXXX00000000 -- 011 10110 CLKOUT2 000X001000001000 -- 011 10111 CLKOUT2 XXXXXXXX00000000 -- 100 10100 CLKOUT3 000X010000010000 -- 100 10101 CLKOUT3 XXXXXXXX00000000 -- 101 10010 CLKOUT4 000X100000100000 -- 101 10011 CLKOUT4 XXXXXXXX00000000 -- 110 01110 CLKOUT5 000X000000000000 -- 110 01111 CLKOUT5 XXXXXXXX00000000 -- 111 00000 Digital filter control XXXXXX0010XXXXXX -- 111 00001 Digital filter control XXXXXXXXXXXX0001 -- 1000 00110 CLKIN XXX1XXXXXXXXXXX	R	0
26-0	Value of PLL register to be written on write operation	RW	0

Figure 35: SST feedback PLL control register: Address 0x23

- Write to register start PLL register change
- In practice only clock 0 register need to be changed. Only value of phase adjustment needs to be changed. It is bits 20:13 for given setup.
- Procedure: - read register,
  - Modify bits 13-20 accordingly
  - Write back modified value
  - One bit change corresponds to 125ps delay
  - Write_command 0x23 1 0x88VVVRRR will update PLL

SST feedback PLL status register: Address 0x24

Bits	Function	R/W	Default
31-24	Value recorded into Idelay delay	R	0
23-15	Unused	R	0
14	RCO sampled with PLL clock and re-sampled with system clock to find proper edge (low to high), ignore of high to low.	R	0
13	Captured bit of Target 5 configuration, Select output between SST/SSP (0) and RCO(1)	R	0
12	Captured bit of Target 5 configuration, Select output between SST and SSP , 0 – SST, 1 – SSP | 0 – SSP_RCO, 1 – SSt_RCO	R	0

11	Captured bit of Target 5 configuration, Select additional load capacitor for sampling logic, 0 – 1GHz sampling, 1 – 400MHz sampling	R	0
10-9	Unused	R	0
8	Idel setup value to shift direction. Every update to this register will shift in proper direction for proper amount specified in bits 7-0.	RW	0
7-0	Idelay setup value to shift sampling clock timing to align with SST input (from Target) for proper setup. Actual value range is from 0 to 0x1f. Each step is about 70ps (estimate). Set to midrange at power up to make correction in both directions. Generally it is not used. Need for secondary feedback loop if implemented. For now just register control.	RW	0

Figure 36: SST feedback PLL status register: Address 0x24

Trigger efficiency control register: Address 0x25

Bits	Function	R/W	Default
31	Indicate completion of counting	RW	0
30-0	Specify duration of trigger statistic collection in 8ns. So, full value is ~17 seconds	RW	0

Figure 37: Trigger efficiency control register: Address 0x25

- Write to this register will start time interval counting in system clock cycles (8ns)
- Value of this register indicate time interval when trigger statistic collected after start (write operation)

Trigger input counter register: Address 0x26

Bits	Function	R/W	Default
31	Indicate completion of counting	RW	0
30	Number of trigger low to high transitions counted	RW	0

Figure 38: Trigger input counter register: Address 0x26

Trigger efficiency counter 0 register: Address 0x27

Bits	Function	R/W	Default
31	Indicate completion of counting	RW	0
30	Number of trigger low to high transitions counted	RW	0

Figure 39: Trigger efficiency counter 0 register: Address 0x27

Trigger efficiency counter 1 register: Address 0x28

Bits	Function	R/W	Default
31	Indicate completion of counting	RW	0
30	Number of trigger low to high transitions counted	RW	0

Figure 40: Trigger efficiency counter 1 register: Address 0x28

Trigger efficiency counter 2 register: Address 0x29

Bits	Function	R/W	Default
31	Indicate completion of counting	RW	0
30	Number of trigger low to high transitions counted	RW	0

Figure 41: Trigger efficiency counter 2 register: Address 0x29

Trigger efficiency counter 3 register: Address 0x2a

Bits	Function	R/W	Default
31	Indicate completion of counting	RW	0
30	Number of trigger low to high transitions counted	RW	0

Figure 42: Trigger efficiency counter 3 register: Address 0x2a

Trigger 16 efficiency counter register: Address 0x2b

Bits	Function	R/W	Default
31	Indicate completion of counting	RW	0
30	Number of trigger low to high transitions counted	RW	0

Figure 43: Trigger 16 efficiency counter register: Address 0x2b

Sampling PLL config register: Address 0x2c

Bits	Function	R/W	Default
31	Writing 1 to this bit will start PLL update operation	RW	0
30	Unused	RW	0
29-25	Pulse high duration of wr_strb signal (1 bit = 32ns/32 = 1ns)	RW	0
24-17	Phase delay of wr_strb signal (1 bit = 32ns/256 = 0.125ns)	RW	0
16-9	Phase delay of wr_advclk signal (1 bit = 32ns/256 = 0.125ns)	RW	0
8-0	Phase delay of SSP signal (1 bit = 64ns/512 = 0.125ns)	RW	0

Figure 44: Sampling PLL config register: Address 0x2c

Sampling PLL status register: Address 0x2d

Bits	Function	R/W	Default
31-30	Unused	R	0
29-25	Pulse high duration of wr_strb signal (1 bit = 32ns/32 = 1ns)	R	0
24-17	Phase delay of wr_strb signal (1 bit = 32ns/256 = 0.125ns)	R	0
16-9	Phase delay of wr_advclk signal (1 bit = 32ns/256 = 0.125ns)	R	0
8-0	Phase delay of SSP signal (1 bit = 64ns/512 = 0.125ns)	R	0

Figure 45: Sampling PLL status register: Address 0x2d

HV control: Address 0x2e

Bits	Function	R/W	Default
31-12	HV low side 0 to XX V. Direct Write to configuration space of MAX5715.	RW	0
11-10	Load Low voltage side DAC, need to be set to 11	RW	0
9	Unused	RW	0
8	Load High voltage side DAC, need to be set to 1	RW	0
7-0	HV high side control value between 5 and 90 V, each step is 1/256 of the range	RW	0

Figure 46: HV control: Address 0x2e

To set MAX5715 need to do the following steps:

1. Write value of 0x71000c00 enable internal 2.5V reference (or 0x73000c00 for 4.096 ref)
2. Write value of 0x82800c00 load value of 1.25V(or 2.0V with 4.096 ref)
3. Optional to adjust termination 0x430f0c00

Peltier uC control: Address 0x2f

Bits	Function	R/W	Default
31	Done bit. If mode bit (bit 1) set to 0 indicate completion of individual write operation, in programming mode (bit 1 set to 1), this bit indicate that programming to uC is enabled and can be followed with uC programming commands	RW	0
30-16	Unused	RW	0
15-8	Define clock period. Which will be equal Value * 256 (ns). For programming the blank device this value need to be set to 0x20 For normal operation in is function of many configuration parameters of actual code running on uC	RW	0
7-3	Unused	RW	0
2	Complement reset bit (invert) . In normal mode reset needs to be at high level (5v), so asserting this bit will assert reset (0V). And in programming mode reset needs to 0V (asserted), and this bit set to 1 will deassert reset (5v)	RW	0
1	Mode bit (0) normal operation, (1) programming mode. By setting this bit to 1 before bit 0 is set to 1, enable execution or programming enable instruction to uC. See http://www.atmel.com/Images/Atmel-8271-8-bit-AVR-Microcontroller-ATmega48A-48PA-88A-88PA-168A-168PA-328-328P_datasheet.pdf Page 300-301 for details	RW	0
0	Enable start of communication over SPI, 1 – start of operation. Need to be 1 for both programming and normal mode for SPI to work	RW	0

Figure 47: Peltier uC control: Address 0x2f

Peltier uC data to uC: Address 0x30

Bits	Function	R/W	Default
31-0	32 bit value which need to be written to uC, Function of write to this registaer depend on normal or programming mode. Programming mode defined on page 301 of http://www.atmel.com/Images/Atmel-8271-8-bit-AVR-Microcontroller-ATmega48A-48PA-88A-88PA-168A-168PA-328-328P_datasheet.pdf For normal mode details of protocol will be specified by uC firmware implementer.	RW	0

Figure 48: Peltier uC data to uC: Address 0x30

Peltier uC data from uC: Address 0x31

Bits	Function	R/W	Default
31-0	Data captured from uC as response to communication to uC. Again, programming mode defined on page 301 of http://www.atmel.com/Images/Atmel-8271-8-bit-AVR-Microcontroller-ATmega48A-48PA-88A-88PA-168A-168PA-328-328P_datasheet.pdf Normal is up to implementation	RW	0

Figure 49: Peltier uC data from uC: Address 0x31

Instruction on how to set programming mode:

1. Write value of 0x2002 into register 0x2f
2. Write value of 0x2003 into register 0x2f
3. Read both registers 0x2f, it should have value of 0x80002003 and register 0x31 with value of 0x5300. If both is true programing got enable, and other programming command can be issued
4. Now it is up to user to deal with programming
5. Example I did just to understand interface
    - Wrote 0x30 = 0x500000ab (0xab is just random selection, can be anything)
    - Read 0x31 -> 0xab500062
    - Wrote 0x30 = 0xaca00022 (to enable internal clock output on pin 12)
    - Read 0x31 -> 0xabac0000
    - Wrote 0x30 = 0x500000ab (0xab is just random selection, can be anything)
    - Read 0x31 -> 0xab500022
    - Write 0x2f = 0x2007 (to remove reset signal) as recommended by uC manual
    - Power cycled and observed clock on pin 12

To write Bootloader to flash memory the following steps need to take:

1. Write value of 0x2002 into register 0x2f
2. Write value of 0x2003 into register 0x2f
3. Read both registers 0x2f, it should have value of 0x80002003 and register 0x31 with value of 0x5300. If both is true programing got enable, and other programming command can be issued
4. Flash Memory divided into pages (64 16bits words, 256 pages), Flash section start depend on Bootsz1/0 bits. With current selection “11” it start at location of 0x3f00 (Register Extended Fuse,

bits 2-1). Also in hex file it is byte addressing so we need to convert from byte addressing to word addressing by dividing by 2 and round down.

5. There are command for lower byte and higher byte in the word. Order load lower byte and then higher byte
6. Wrote 0x30 = 0x40000085 Load lower byte, and 0x30 = 0x480000E0
7. …. Write full page (64 16-bit words)
8. Store page by write page command 0x30 = 0x4c3f0000
9. Write 0x2f = 0x2007 (to remove reset signal) as recommended by uC manual
10. Power cycle and verify writing by reading locations
    0x203f0000
    0x283f0000

Flash memory write: Address 0x32

Bits	Function	R/W	Default
31	Write to this bit (1) start operation of writing to flash command intermediate storage	RW	0
30-26	Unused	R	0
25-16	Address of flash intermediate command data storage for write.	RW	0
15-0	Value to be written to flash intermediate storage at location specified in bits 25-16.	R	0

Figure 50: Flash memory write: Address 0x32

Flash memory control: Address 0x33

Bits	Function	R/W	Default
31	Write to this bit (1) start flash command pre-stored in flash intermediate storage programmable in register 0x32. If command involve read operation data will be stored in intermediate storage accessible from register 0x34	RW	0
30-25	Unused	R	0
24-16	Size of read operation in bytes to be to be stored into intermediate read store, 0x0 – no read required	RW	0
15-10	Unused	R	0
9	If set to 1, flash chip select will not be asserted16	R	0
8-0	Size of command in bytes to be played from intermediate command store	RW	0

Figure 51: Flash memory control: Address 0x33

Flash memory read: Address 0x34

Bits	Function	R/W	Default
31	Write to this bit (1) start operation of reading which can be completed on follow up read	RW	0
30	Done bit, indicate that flash command is completed	RW	0
30-26	Unused	R	0
25-16	Address of flash intermediate data storage to read.	RW	0
15-0	Value read from flash intermediate storge at location specified in bits 25-16.	R	0

Figure 52: Flash memory read: Address 0x34

5.1.1 Target 2 related registers

Target 5 configuration register: Address 0x20

Bits	Function	R/W	Default
31-20	Read back value. If bit 19 is set (1) read back followed immediately after write. Otherwise, read of previously written register	R	0
20	If bit 20 is set (1) read back followed immediately after write. Otherwise, read of previously written register	RW	0
19	Define if data latching on Target is required. 0 – latching is required, 1 – no latching.	RW	0
18-12	Target register address to write. See Target 5 documentation for specific register address map	RW	0
11-0	Target register value to write. See Target 5 documentation for specific registers and bit allocation	RW	0

Figure 53: Target 5 configuration: Address 0x20

5.2 Packet data word

Packet for Buffer 0, channel 0:

F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
Size. Real size = size * 8 in bytes.								Sequence number per ID							
Time Stamp[15:0] at time of TACK arrival, 8ns resolution															
Read Sft (1b)	Use offset (1b)	*Current column of buffer at time of sampling (6 bits)*						*Current row of buffer at time of sampling(3 bits)*					Current sample offset (5 bits)		
Sample offset (2bits)		Contributing triggers(6 bits)						Number of buffers(4bits) as specified in Reg0xF(bits19-16)				ID of serial data connection (4 bits)			
Indicate Stale Data(1) – if data read from same buffer more than once with sampling update		MBZ(5)						Adjusted sample delay due to trigger time and selected trigger delay(5)				Partial number of samples(5) as specified in Reg0x16(bits4-0)			
MBZ(1)		Total number of samples in event (10 bits)						Total number of buffers digitized(5 bits)							
MBZ(2)		Time Stamp[23:16] at time of TACK arrival, 8ns resolution													
MBZ(16)															

MBZ(3)	Trigger delay (8 bits)
Sample ID (only 4 bits) Samples 0 and 16 look same. First enabled sample of buffer 0	12 bit ADC value for sample ID indicated. Even so, sample ID is limited to 4 bits, information can be easily restored from list of enabled samples. Sample ID start from 0 to 31(63). Buffers are going in order as well after all samples collected.
Sample ID (only 4 bits) Samples 0 and 16 look same,. Second enabled sample of buffer 0	12 bit ADC value for sample ID indicated. Even so, sample ID is limited to 4 bits, information can be easily restored from list of enabled samples. Sample ID start from 0 to 31(63). Buffers are going in order as well after all samples collected.
…	…
Sample ID (only 4 bits) Samples 0 and 16 look same. Last enabled sample of buffer 0	12 bit ADC value for sample ID indicated. Even so, sample ID is limited to 4 bits, information can be easily restored from list of enabled samples. Sample ID start from 0 to 31(63). Buffers are going in order as well after all samples collected.
Sample ID (only 4 bits) Samples 0 and 16 look same. First enabled sample of buffer 1	12 bit ADC value for sample ID indicated. Even so, sample ID is limited to 4 bits, information can be easily restored from list of enabled samples. Sample ID start from 0 to 31(63). Buffers are going in order as well after all samples collected.
Sample ID (only 4 bits) Samples 0 and 16 look same,. Second enabled sample of buffer 1	12 bit ADC value for sample ID indicated. Even so, sample ID is limited to 4 bits, information can be easily restored from list of enabled samples. Sample ID start from 0 to 31(63). Buffers are going in order as well after all samples collected.
…	…
Sample ID (only 4 bits) Samples 0 and 16 look same. Last enabled sample of buffer 1	12 bit ADC value for sample ID indicated. Even so, sample ID is limited to 4 bits, information can be easily restored from list of enabled samples. Sample ID start from 0 to 31(63). Buffers are going in order as well after all samples collected.
…	
Sample ID (only 4 bits) Samples 0 and 16 look same. First enabled sample of Last buffer.	12 bit ADC value for sample ID indicated. Even so, sample ID is limited to 4 bits, information can be easily restored from list of enabled samples. Sample ID start from 0 to 31(63). Buffers are going in order as well after all samples collected.
Sample ID (only 4 bits) Samples 0 and 16 look same,. Second enabled sample of Last buffer	12 bit ADC value for sample ID indicated. Even so, sample ID is limited to 4 bits, information can be easily restored from list of enabled samples. Sample ID start from 0 to 31(63). Buffers are going in order as well after all samples collected.
…	…
Sample ID (only 4 bits) Samples 0 and 16 look same. Last enabled sample of Last buffer	12 bit ADC value for sample ID indicated. Even so, sample ID is limited to 4 bits, information can be easily restored from list of enabled samples. Sample ID start from 0 to 31(63). Buffers are going in order as well after all samples collected.
CRC on above words	
0xbeef - This is filler words to keep size at multiple of 8 bytes and is function of actual enabled samples	

(see above). Up to 3 words possible		
MBZ(14 bits)	Timeout	Error
MBZ		

6 Command Input data format:

Word #	Byte 1								Byte 0								Type
	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	
0	DNC																PGP
1	DNC														MBZ		PGP
2	OC 00read 01write		DNC						Address MSB								PGP
3	Address LSW																PGP
4	Data MSW, write – actual word, read - 0																PGP
5	DATA LSW, write – actual word, read - 0																PGP
6	DNC																
7	DNC																

Table 1: Command input packet format

Formatted according to PGP requirements to deal with double board usage (UDP and PGP)

7 Response output data format:

Word #	Byte 1								Byte 0								Type
	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	
0	Feedback command header word																PGP
1	Feedback command header word														MBZ		PGP
2	OC 00read 01write		0x0						Address MSB								PGP
3	Address LSW																PGP
4	Data MSW, write – feedback written word, read – actual read word																PGP
5	DATA LSW, – feedback written word, read – actual read word																PGP
6	MBZ														Timeout error	Other Error	PGP
7	MBZ																PGP

Table 2: Response output packet format

8 Virtex-5 FPGA System Monitoring

User guide for Virtex-5 FPGA system monitoring can be found under

http://www.xilinx.com/support/documentation/user_guides/ug192.pdf

T5EV implement readout of 14 VAUX channels. The first 7 monitor interface to Evaluation board

VAUX channel definition table

Signal	Function	Conversion formula, V=,V
VAUX0	+2.5V analog supply to Target 2	Value/1024/0.200
VAUX1	HV measured current, A	Value/1024/0.200/200
VAUX2	VPED voltage to Target 2	Value/1024/0.200
VAUX3	VDLY voltage from Target 2	Value/1024/0.200
VAUX4	VDISCHARGE voltage from Target 2	Value/1024/0.020
VAUX5	ISEL voltage from Target 2	Value/1024/0.200
VAUX6	Unused, can be connected to measure any desired voltage under 2.5V	Value/1024/0.200

Figure 54: VAUX channel definition table

To read any register used in FPGA monitoring interface the following sequence need to be executed:

1. Write value of 0xAA0000, where AA is FPGA monitoring register address into T5EV register number 0xD
2. Read T5EV register 0xC, the 16 LS bits will have register value

To write any register used in FPGA monitoring interface the following sequence need to be executed:

1. Write value of 0x80AA0000, where AA is FPGA monitoring register address into T5EV register number 0xD
2. Read T5EV register 0xC, the 16 LS bits will have updated (written) register value

Example of reading temperature in register 0:

1. Write value of 0 into T5EV register 0xD
2. Read register 0xD and extract ADC value = (Reg0xC & 0xffff) >> 6
3. Calculate by T,C = (ADCValue x 503.975)/1024 – 273.15

Example of reading VCCINT in register 1:

4. Write value of 0x10000 into T5EV register 0xD
5. Read register 0xD and extract ADC value = (Reg0xC & 0xffff) >> 6
6. Calculate by Voltage, V = (ADCValue x 3)/1024

Example of reading VAUX0 in register 1:

7. Write value of 0x100000 into T5EV register 0xD
8. Read register 0xC and extract ADC value = (Reg0xC & 0xffff) >> 6
9. Calculate by Voltage, V = (ADCValue)/1024/0.20 – formula from table 47

Virtex-5 FPGA monitoring interface include many options for operation, for basic operation all registers is configured at power up and will not require additional configuration. For expert usage need to consult

http://www.xilinx.com/support/documentation/user_guides/ug192.pdf

9 UDP requirements and setups:

T5EV support UDP communication through external network connections. There are set of limitations that need to be considered when interface to T5EV is designed:

1. T5EV board has Fiber-optic Ethernet interface. As result in order to talk to any regular PC there are two possible solutions:
   - Use NIC card supporting Fiber-optic interface. Possible NIC card which was tested with this interface is Hewlett-Packard-HP-NC373F-Mltfunc-PCI-E-1000SX-GbEth-ADP-NC373F
   - Use Fiber Transceiver to convert from FO Ethernet to 1Gbit copper connection. Adapter used is FIB1-1000TS
2. Any 1Gbit NIC card will support communication with evaluation board
3. T5EV hardwired to IP address 192.168.0.173
4. T5EV hardwired to port 8105
5. T5EV responds to only ARP and UDP packet with IP192.168.0173 and port 8105, does not responds to ping
6. On Linux core in order to keep up with flood of packets command

   sudo /sbin//sysctl -w net/core/rmem_max=2000000

   need to be executed to increase kernel buffering for incoming data. Value of 2000000 need to be verified. Window version of this setup is not known at this point.
7. The following Python command used to set and talk to T5EV
   - from socket import *
   - UDPSock = socket(AF_INET,SOCK_DGRAM)
   - UDPSock.bind((HOST0,PORT0))
   - UDPSock.settimeout(1)
   - UDPSock.sendto(buffer,(HOST,PORT))
   - data,addr = UDPSock.recvfrom(buffer)
   - UDPSock.close()
8. For each command sent to T5EV there are response (each command is acknowledged)