PRELIMINARY

Time Tagging Module Specification SDE Upgrade

Version 2.0

Robert Sobin, Case Western Reserve

January 21st, 2015

1 Background

The time tagging module for the SDE Upgrade development originally began by using the SDE Upgrade Firmware Specification Version 1.0 and the time tagging module (ttag.bdf schematic April 11, 2014) for Auger North (AN) as references. The module is written in Verilog and developed using Vivado 2014.2 tools or the latest version of Vivado exceeding 2014.2. As a result of the Beyond 2015 SDE Software Meeting in October 2014, the module design has been reorganized per the input of the meeting participants.

Prior to the October 2014 software meeting, we were using AXI GPIO status and control registers to communicate with the time tagging module. As a result of the meeting we are now using an AXI Slave Peripheral interface to communicate with the time tagging module.

2 AXI Slave Peripheral Interface

2.1 General Description

The AXI Slave Peripheral Interface consists of sixteen registers of which thirteen are status registers, one is a control register, and two are spare registers. The AXI Slave Peripheral Interface is created using the standard “create and package IP” tool in Vivado. As a result of using this tool, the time tagging module will now reside in the Vivado IP catalog as time tagging IP. The time tagging IP consists of two modules and a top level wrapper. The wrapper and modules are named as follows:

a.) time_tagging_v1_0 (note – this is the top level wrapper)
b.) time_tagging (note – this is the user time tagging hardware module written in structural Verilog)
c.) time_tagging_v1_0_S00_AXI (note – this is the AXI Slave Peripheral Interface module generated in behavioral Verilog)

The time tagging hardware module consists of the time tagging hardware specific to the application and runs on the 120MHz external oscillator. The AXI Slave Peripheral Interface module runs on the PL fabric clock and consists of two parts: the AXI Slave Peripheral Interface

to the AXI Master, and the interface connecting the time tagging hardware module to the AXI Slave Peripheral Interface. The top level wrapper connects the related ports of the two modules together, and provides the ports external to the IP, needed to interface to the Z_system AXI Master and to the other SDE upgrade hardware at the project level.
All time tagging registers are 32 bits wide and are memory mapped into the Zynq PS address space. The thirteen status registers are read only registers. The single control register and two spare registers are read write registers. The control register is still named “ttagctrl” and is now required mainly to provide a soft reset to the time tagging hardware. The original control register address signals needed to access individual memory locations within the shower and muon buffers are now input ports to the time tagging module. The original control register address signals needed to drive the time tagging register output multiplexer are no longer needed because of memory mapping.
It is assumed that the time tagging IP will be instantiated and added to the overall project block diagram, at which time the register address mapping will be created automatically. The block diagram is part of the Z_system.bd file which can be found in the project hierarchy.

2.2 Hardware Changes Resulting From October 2014 Software Meeting

The following hardware changes have been incorporated as a result of the October 2014 Software meeting:

a.) The fast trigger is no longer distinguished by rising and falling edges.
b.) A new module input port named “dead” and a new dead counter have been created.
c.) The dead counter will begin counting when “dead” is HI and reset when “dead” is LO
d.) The value of the dead counter is stored in a register for reading before the counter is reset
e.) The fast counter value will be captured in registers at the following times:
    i. Captured on the occurrence of the fast (or shower) trigger
    ii. Captured on the occurrence of the slow (or muon) trigger
    iii. Captured on the occurrence of pps, then re-captured on the occurrence of fast trigger
    iv. Captured on the occurrence of pps, then re-captured on the occurrence of slow trigger
f.) The calibration counter value will be captured in registers at the following times:
    i. Captured on the occurrence of pps, then re-captured on the occurrence of fast trigger
    ii. Captured on the occurrence of pps, then re-captured on the occurrence of slow trigger
g.) The Auger North register formerly named “timems” has been eliminated

h.) The original control register address signals needed to access individual memory locations within the shower and muon buffers are now input ports to the time tagging module.
i.) The original control register address signals needed to drive the time tagging register output multiplexer are no longer needed because of memory mapping.

In addition the following additions/changes prior to October 2014 still apply:

a.) A new register named “teststatus” has been created to allow for a stand alone “C” test program in SDK that will exercise the time tag module hardware
    i. In the test program, artificial events including pps, fast trigger, and slow trigger are created by toggling individual control bits in Virtual I/O IP.
    ii. The register values of the time tag module are printed to the terminal after each occurrence of an artificial event
    iii. In the test set-up, the external 120MHz clock is being generated by the clock generator IP using the ZED external 100MHz oscillator as a source.
b.) The register named ”tt_constant” on the Auger North schematic has been eliminated. This register apparently was intended to keep track of the compilation version of the firmware code.
c.) A calibration counter has been created that will start counting over on pps occurrences

2.3 Control Register Signals

The control register function has been reduced in scope. The control signals, associated with the time tagging control register (ttagctrl), are as follows:

a.) ttagres – time tag reset hi true
b.) clr_pps – clear pps occurrence status bit, used for testing only
c.) clr_clkf – clear fast trigger occurrence status bit, used for testing only
d.) clr_clks – clear slow trigger occurrence status bit, used for testing only
e.) clr_dead – clear dead time occurrence status bit, used for test only

The above control signals add up to 5 total signals being used in the 32 bit control register.

3 Time Tagging Module User I/O Ports

3.1 Description

The following is a list of the user I/O ports in the time tagging module. The list is a subset of the I/O ports as it does not include I/O ports of the AXI Slave Bus Interface. It is assumed that all relevant logic signals are HI true:

USER TIME TAGGING PORTS DESCRIPTION

clk_120m	120MHz clock, formerly 100MHz clock in AN
pps	gps 1 pulse per second
evtcnt [3:0]	tag from trigger memory
evtclkf	fast trigger
evtclks	slow trigger
dead	dead time
address_wsb [2:0]	shower buffer write address
address_rsb [2:0]	shower buffer read address
address_wmb [2:0]	muon buffer write address
address_rmb [2:0]	muon buffer read address

It is worthy of note that, although the shower and muon buffers are double buffers in this implementation, assigning 3 bits of write address and 3 bits of read address allows for up to 8 buffer addresses in the future for each type of buffer. It also means that the processor will decide which buffer address is being used for capturing data at any given time and which buffer address is being used for reading data at any given time.
One topic which bears discussion is the freeing up of shower and muon buffers for the writing of new events. It is assumed here that the processor logically will not issue a new buffer write address (shower or muon) to the time tag module unless it (the processor) has cleared that buffer for the next event. Therefore it is assumed that an explicit buffer “done” input port to the time tag module is not needed.

Another topic of discussion concerns the clocks used by the time tag module. The former AN “adcclk “(alias clk_100m} is now replaced by “clk_120m” (the 120 MHz clock). The former AN bus clock “mck” (alias pck0), is now replaced by the Zynq PL fabric clock “fclk_clk0”. The “clk_120m” clock is sourced by an external PLL VCXO. The “fclk_clk0” clock is sourced by the I/O PLL of the Zynq. The former clock is used by the counter modules in the time tagging hardware module. The latter clock is used by the AXI Slave Peripheral Interface module. The two clocks are asynchronous. Therefore I/O port control signals and register data connecting the two modules together, will cross clock domains and will have to be double synchronized between the two clocks, or else the implemented time tagging module will not pass the timing constraints test.

3.2 AXI Slave Peripheral Handshaking

The generic Verilog code generated by Vivado in the creation of the AXI Slave Peripheral Interface module uses fixed handshaking that assumes static read register access. Therefore the read valid response (axi_rvalid) of the slave peripheral interface occurs once a valid read address is placed on the AXI Slave interface by the Master. This quick read valid response of the Slave does not allow for the transfer of valid read data to the Master when the register being read is not static. In the time tagging hardware, the register with the name of "timeseconds" is not static. The register's data value is made available in the time tagging hardware only when the PS accesses the register. As a result latency is created by the register address selection crossing clock domains when entering the time tagging hardware module, and when the register's returned data crosses clock domains when entering the AXI Slave Peripheral Interface module. To accommodate this latency the generic Verilog code generated by Vivado in the creation of the AXI Slave Peripheral Interface module, has been modified to allow for delayed valid read responses (axi_rvalid) by the Slave. Currently there is a long delay for the specific register discussed here, and a shorter default delay for all other time tagging registers that are read by the Master. The long read delay allows for clock synchronization in two directions, and the short read delay allows for clock synchronization in one direction, as with the case of semi-static registers.

Similarly, in the generic Verilog code, the write valid response (axi_bvalid) of the Slave Peripheral Interface module occurs once a valid write address and write data are placed on the AXI Slave Interface by the Master. The design of this generic Verilog code by Vivado expects no outstanding transactions. This short write handshake does not engage the PS while the write data synchronization occurs in the time tagging hardware module. Therefore the generic Verilog code has been modified to allow for a delayed valid write response (axi_bvalid) by the Slave that will keep the PS engaged during this synchronization period before releasing it.

3.3 AXI Slave Peripheral Address Cycle Stretching

The access of a time tagging register that is not semi-static requires generating address decoding for the specific register. One such register is the read only register named "timeseconds". An individual address decode signal is generated by the AXI Slave Interface module and passed over to the time tagging hardware module to be used for accessing the register. The address decode signal must be double synchronized in the clock domain of the time tagging hardware module before it can be used there. Because of the a-stable nature of cross domain clock synchronization, the address decode signal can either be shortened or

lengthened by one clock cycle once it has been synchronized to the domain of the time tagging hardware clock. As a result if the address decode signal is only one clock cycle in length when it is generated at the AXI Slave Peripheral Interface module, there is a chance that it will either be two clock cycles in length or zero clock cycles in length once it has been synchronized in the clock domain of the time tagging hardware module. If the length of the signal ends up being zero clock cycles in length, the intended transaction will never occur and lack of valid data handshaking will hang the AXI transaction.
To avoid this problem the generic Verilog code in the Slave Peripheral Interface module has been modified. The modification involves stretching the amount of time that the AXI address is available to the slave peripheral. To accomplish this, the acceptance of the address handshake by the slave is delayed two clock cycles for both read and write cycles (axi_aready & axi_awready).

3.4 ASYNC_REG Property

Vivado Design Suite Properties Reference Guide UG912 recommends setting the ASYNC_REG property to “TRUE” with synchronizing registers in order to place these registers in the same slice during implementation. The ASYNC_REG property has been set to “TRUE” in the AXI Slave Peripheral Interface module for the synchronizing 32 bit register pairs associated with the thirteen time tagging read only registers, as well as two single bit register pairs associated with two address decode signals returning from the time tagging hardware module. The same property has also been set “TRUE” in the time tagging hardware module for the three single bit register pairs associated with the “pps” input and two address decode signals coming from the AXI Slave Peripheral Interface module.
At the time of this writing it is not known what clock domain the remaining user input ports, identified in section 3.1 above and excluding the input clock, reside in. If these input ports reside in a clock domain other than that of the 120MHz external oscillator, then the ASYNC_REG property will also have to be set “TRUE” for their synchronizing register pairs.

4 Time Tagging Module Registers

4.1 Description

The sixteen time tagging registers are implemented as 32 bit register memory types. As the shower and muon registers will be implemented as double buffered registers in this version of the firmware, the plan is to implement each buffer with two 32-bit registers. The output data from each register within a buffer will be multiplexed according to the value of the respective

buffers read address, which is set by the processor. The value of each respective buffer's write address will be decoded to determine which register in that buffer is enabled for writing. Again, this buffer write address is set by the processor.

4.2 Register Bit Assignment Changes from Auger North

By replacing the shower fifo memory in Auger North (AN) with double buffered register memory, some changes need to be made in the definition and assignment of the data bits within the respective 32-bit data words for the shower buffers. In AN, the high order bits in the buffer which captures the 27-bits of the seconds counter value, were assigned to the "wrful" and "rdempty" outputs of the shower nanoseconds FIFO memory. Since the FIFO's are being replaced in this implementation, the AN definition and assignment of these two bits is no longer needed. Also in AN, one of the high order bits in the buffer which captures the 27-bits of the nanosecond counter value, was assigned to "fast trigger". This stored value, then, gave the state (hi or lo) of fast trigger for that particular word. Since the fast trigger is no longer being distinguished by rising and falling edges, the bit assignment is no longer necessary.

Also, regarding the shower buffer which captures the 27-bits of the nanosecond counter value, the value of the fast counter now represents the number of 120 MHz clock periods (8.33ns), rather than the number of 100 MHz clock periods (10ns). Because of this frequency change in the clock, the fast counter will now roll-over in 1.11 seconds instead of 1.34 seconds.

4.3 Register Memory Mapping and Description

Below is a list of the time tagging module register names that are being used with the AXI Slave Peripheral Interface, as well as their relative address offset relative to the base address assigned by Vivado:

REGISTER NAME	ADDRESS OFFSET	DESCRIPTION
Onanosec	0	value of nanosecond (fast) counter at time of fast trigger
Oseconds	1	value of seconds counter at time of fast trigger
c120mout_sb	2	value of nanosec. counter at last pps occurrence and fast trigger
c120calout_sb	3	value of calibration counter at last pps occurrence and fast trigger
slowtriggerns	4	value of nanosecond counter at time of slow trigger
slowtriggersec	5	value of seconds counter at time of slow trigger
c120mout_mb	6	value of nanosec. counter at last pps occurrence and slow trigger

c120calout_mb	7	value of cal. counter at last pps occurrence and slow trigger
timeseconds	8	value of seconds counter at last pps occurrence when read by ps
c120mout_ps	9	value of nanosec counter at last pps occurrence when read by ps
c120calout_ps	10	value of cal counter at last pps occurrence when read by ps
c120deadout	11	value of dead counter at last pps occurrence when read by ps
teststatus	12	value of event status bits for test purposes only
ttagctrl	13	control register contains time tag soft reset control bit
spare	14	
spare	15	

Reading of the “timeseconds” register by the PS at address offset 8 results in the latching of three registers at the same time: "timeseconds", "c120mout_ps", and "c120calout_ps". Therefore, of the three registers, “timeseconds” must be read first.

4.4 Individual Register Bit Assignments

4.4.1 onanosec Register

REGISTER BIT LOCATION	BIT DESCRIPTION – READ ONLY
onanosec [26:0]	value of nanosecond (fast) counter at time of fast trigger
onanosec [27]	LO
onanosec [31:28]	value of evtcnt tags from trigger memory

4.4.2 oseconds Register

REGISTER BIT LOCATION	BIT DESCRIPTION – READ ONLY
oseconds [27:0]	value of seconds counter at time of fast trigger
oseconds [31:28]	all LO

4.4.3 c120mout_sb Register

REGISTER BIT LOCATION	BIT DESCRIPTION – READ ONLY

c120mout_sb [26:0] value of nanosecond counter at last pps occurrence and fast trigger
c120mout_sb [31:27] all LO

4.4.4 c120calout_sb Register

REGISTER BIT LOCATION BIT DESCRIPTION – READ ONLY

c120calout_sb [26:0] value of 120MHz cal. counter at last pps occurrence and fast trigger
c120calout_sb [31:27] all LO

4.4.5 slowtriggerns Register

REGISTER BIT LOCATION BIT DESCRIPTION – READ ONLY

slowtriggerns [26:0] value of nanosecond (fast) counter at time of slow trigger
slowtriggerns [31:27] all LO

4.4.6 slowtriggersec Register

REGISTER BIT LOCATION BIT DESCRIPTION – READ ONLY

slowtriggersec [27:0] value of seconds counter at time of slow trigger
slowtriggersec [31:28] all LO

4.4.7 c120mout_mb Register

REGISTER BIT LOCATION BIT DESCRIPTION – READ ONLY

c120mout_mb [26:0] value of nanosecond counter at last pps occurrence and slow trigger
c120mout_mb [31:27] all LO

4.4.8 c120calout_mb Register

REGISTER BIT LOCATION BIT DESCRIPTION – READ ONLY

c120calout_mb [26:0]	value of 120MHz cal. counter at last pps occurrence and slow trigger
c120calout_mb [31:27]	all LO

4.4.9 timeseconds Register

REGISTER BIT LOCATION	BIT DESCRIPTION – READ ONLY
timeseconds [27:0]	value of seconds counter at last pps occurrence when read by ps
timeseconds [31:28]	all LO

4.4.10 c120mout_ps Register

REGISTER BIT LOCATION	BIT DESCRIPTION – READ ONLY
c120mout_ps [26:0]	value of nanosecond counter at last pps occurrence when read by ps
c120mout_ps [31:27]	all LO

4.4.11 c120calout_ps Register

REGISTER BIT LOCATION	BIT DESCRIPTION – READ ONLY
c120calout_sb [26:0]	value of 120MHz cal. counter at last pps occurrence when read by ps
c120calout_sb [31:27]	all LO

4.4.12 c120deadout Register

REGISTER BIT LOCATION	BIT DESCRIPTION – READ ONLY
c120deadout [26:0]	value of 120MHz dead counter at last pps occurrence when read by ps
c120deadout [31:27]	all LO

4.4.13 teststatus Register

The teststatus register is only intended to be used when testing the time tagging hardware. The teststatus register has the following bit assignments:

REGISTER BIT LOCATION	BIT DESCRIPTION – READ ONLY
teststatus [0]	slow trigger occurrence when HI, clear with “clr_clks”
teststatus [1]	pps occurrence when HI, clear with “clr_pps”
teststatus [2]	fast trigger occurrence when HI, clear with “clr_clkf”
teststatus [3]	fast trigger occurrence when HI, clear with “clr_dead”
teststatus [31:4]	all LO

4.4.14 ttagctrl Register

REGISTER BIT LOCATION	BIT DESCRIPTION – READ/WRITE
ttagctrl [0]	ttagres – time tag reset hi true
ttagctrl [1]	clr_pps – clear pps occurrence status bit, used for testing only
ttagctrl [2]	clr_clkf – clear fast trigger occurrence status bit, used for testing only
ttagctrl [3]	clr_clks – clear slow trigger occurrence status bit, used for testing only
ttagctrl [4]	clr_dead – clear dead time occurrence status bit, used for test only
ttagctrl[31:5]	spare – read/write

4.4.15 spare1 Register

REGISTER BIT LOCATION	BIT DESCRIPTION – READ/WRITE
spare1 [31:0]	read/write

4.4.16 spare2 Register

REGISTER BIT LOCATION	BIT DESCRIPTION – READ WRITE
spare2 [31:0]	read/write

5 120 MHz PLL VCXO

5.1 Background

To date we have been calibrating ILotus GPS receivers using a Verilog counter module instantiated in a Xilinx CPLD. The counter clock is driven by an external fundamental and 3rd overtone 150 MHz crystal (CTS –CB3LV) which has a frequency stability of +/- 50 ppm and peak to peak jitter of 50 ps. We have also run one GPS calibration by instantiating a counter module into the Zynq PL on the ZED board. The counter clock is driven by the I/O PLL of the Zynq running at about 244 MHz. The latter single test result indicated that, although the standard deviation appeared to improve at the higher frequency, the mean real time difference at the higher frequency Zynq test did not track the mean of the lower frequency CPLD tests as closely. It is suspected that the timing path for the 1PPS GPS input will have to be constrained in order to better understand this result.

5.1 120 MHz VCXO

The PLL VCXO planned as the source for the “clk_120m” clock to the Zynq is the Texas Instrument CDCE913. The CDCE913 has a peak to peak jitter of 100 ps. It is not known which crystal will be used as the clock input to the PLL.

6 Attachments

6.1 Z_System Block Diagram for Test