

The following are various tidbits of information in no particular order. They are organized as an answer to the various questions we faced during the progress of the project. They have been loosely organized into some categories.

QUESTIONS NOTICED IN THE SCHEMATICS:

- The schematics in the binder are not up to date. A number of minor changes had been noticed and the vhdl code was based on the new schematic printouts
- The sysclk input goes to a non-clock pin on the fpga and explicit manual routing in the code through GCLK and IBUF (just the gclk does not work) was required to route the clk on pin22.
- Similar manipulation as above was required to route rd_event because it is used as the clock to a flip-flop.
- As in the schematics multiple copies of enable were used to generate enable signal to decrease fan out and decrease delay (with just one enable timing failed for 60 MHz routing).
- The fifowrite pin was made a fast pin to improve the rise time of the signal in an attempt to solve some problems with the FIFO chips. However its impact on noise has not been considered.
- The synthesis did not like counters greater than 4 bits and the two counters in the schematics had to be broken into smaller cascaded counters to solve that problem.
- The sync signal as implemented in the schematics was observed to have a problem of generating a wider enable signal (2 sysclk ticks wide) against the required signal and was corrected by making the clear of the shift register implementing the enable signal as a synchronous clear.
- The component MYRS used in the schematics seems to be a modification of the standard RS flip-flop component but with a modified set signal to save on 2 gates in the implementation.
- The cmd_ph1 signal generated seems to be a test signal which is not used in the implementation.
- The ele_datb signals connected to the latches of buf_rd and buf_wr seems to be dummy signals serving no direct purpose.
- There are pull up resistors on the eledatb and locksum input lines to avoid the input lines going to an impedance state other than high or low to avoid problems of bad input to the CMOS logic.

ALGORITHM FOR HALF SAMPLNG IMPLEMENTED

Assuming samples are numbered 0,1...31

If (sampling even sample) then

If (sample is a TDC hit) then

*(Keep the sample;
 Set firstsamplewasTDC = true ;)*

Else

*(Throw away the sample;
 Set firstsamplewasTDC = false ;)*

If (sampling odd sample) then

If (firstsamplewasTDC = true) then

Throw away the sample;

Else

Retain the sample;

BRIEF SUMMARY OF THE STATE MACHINE CYCLING THROUGH THE PHASES

The phases with numbering in alphanumeric were added new, compared to the old firmware to accommodate the readout at 15MHz from the elephant chip.

Phase 1 -> initializes the state machine on receipt of the rd_event signal.

Phase 2 -> load the hitmap. *It was noted that when the chip select is asserted, or when hitmap is requested there appears to be a spurious data serve from the elephant chip (actually the previous hitmap) which is not the real hitmap. The real hitmap loads a little later on the clk15 rise. It is important to note this and latch the hitmap on the correct rising edge of clk 15.*

Phase 2a and Phase 2b -> It was noted that the circuit implementing the test hit map, generating the more_channel signal was a long delay path and to give enough time from when the next signal comes to when the more_channel signal is valid, two extra phases were added as time delay phases. It was noted that when the timing was on the edge for the above path the wrong hit map was loaded or the more_channel signal was not valid when required.

Phase 3 -> write the elephant address

Phase 4 -> write tag bits

Phase 5 -> write the sysclk count

Phase 6 -> write the trigger tag bits

Phase 3 through Phase 6 accesses SRAM2 in the elephant chip.

Phase 7 -> point to the first data in the channel in the RO_RAM in the elephant chip to start waveform readout.

Phase8_datainc -> time delay phase to account for the fact that there is a 2-cycle latency to access data from the RO_RAM in response to the data_start signal ; and to send out the first data_inc signal.

Phase 8 -> cycle through reading the first 31 samples and writing the appropriate half sampled data.

Phase 9a -> read the last sample and point back to the first data of a channel.

The readout requires 1 data_start and 31 data_inc to read all the 32 samples of the waveform and 1 extra data_inc to point back to the first data of channel (like resetting a counter).

Phase 9 -> test to cycle to next_channel or next_elefant.

Phase 10 -> test to check if all the elephants have been read out.

Phase 11 -> signals end of readout and re initializes the state machine to wait for next rd_event

Also a significant change in the new state machine is the clocking of the fifo_dat before it goes to the FIFO chip, effectively it keep the ele_data valid for an extra cycle before it changes, which helps to send the correct the fifo_write signal over the period which fifo_dat is valid. This was added primarily to satisfy the constraint of the FIFO chip that the fifo_write signal has to be at least 25ns wide.

NOTED IN THE WORKING OF THE ELEFANT CHIPS

- The elephant chip works of a clk15 signal of 15MHz derived from the sysclk. However they are not in phase and the sysclk of the slave and the clk15 signals were noted to vary as either as +4ns or -8ns in phase difference.
- The control signals data_start, data_inc, ch_sel, buf_rd, buf_wr were latched by the elephant chip on the rising edge of the clk15.
- The elephant chip puts out data with a 1-cycle latency (of clk15) in response to the data_start, on the rising edge of clk15 and in response to the data_inc, on the falling edge of the clk15.
- Glitch noted in the software simulations on data_inc during phase9a due to combinatorial logic, which did not seem to be a problem since the rising edge of clk15, which latches the signal, is not present during the glitch.
- During the readout of the SRAM2 the control signal ch_num is latched on the rising edge of clk15 and the data appears nearly instantaneously with no latency.
- **As Noted Before when the chip select is asserted, or when hitmap is requested there appears to be a spurious data serve from the elephant chip (actually the previous hitmap) which is not the real hitmap. The real hitmap loads a little later on the clk15 rise. It is important to note this and latch the hitmap on the correct rising edge of clk 15.**

MISCELLANEOUS INFORMATION

- Sometimes during power on or during a reset of all the FEA's glitches were notices either in rd_event or in the ll_accept line. This either leads to a spurious startup of the state machine or an unintentional increment of the buf_rd counter. To get around this problem a clrout to clear the buf counters and a FIFO reset in the software is required every time on power on or a reset of all FEA's.
- To set the correct polarity of reset for the prom's configuring the fpga's 4 bytes of zeroes starting from the location 0x2000(typically the address after the maximum capacity of the prom) has to be written while programming the proms.
- While running the synthesis tool the option of 'FPGA reentrant routing' to do a time delay based cleanup is required to meet the timing requirements of 60MHz design.
- A number of kludge wires from the slave pins to the master are soldered on the board. But no purposes for them have been found yet; they connect pins, which are not used by the slaves. They seem like the vestiges of a previous test process.
- The readout had to be sped up to 15Mhz from the previous 7.5Mhz. Since with half sampling and a readout of 7.5Mhz the effective write to the FIFO's goes down to 3.75Mhz(roughly) from the previous 7.5Mhz. This creates cracks in the readout when the master reading the FIFO can catch up with the slave writing to the FIFO.

LIST OF TIMING CONSTRAINTS GIVEN TO THE SYNTHESIS TOOL

- **TIG (the timing analyzer neglects any path through the net with a TIG) on dis_ele:**

The critical path here is from ele_num changing on phase10, which causes dis_ele signal to change and this controls the loading of the hitmap in phase2a. This path propagates over a number of circuits and since it has time from phase10 to phase2a it is a multi cycle path. But a lot other paths also merge with this, which are not important for the timing analysis like the path from eleoff_in to dis_ele. Therefore a TIG was placed on the dis_ele net.

- **TIG on hitmap(0:7):**

The critical path here is the loading of the hitmap after the next signal is valid and the setting of more_channel, and to ensure that an extra time delay phase phase2b was added to do just that. Hence to avoid the synthesis considering all the other unnecessary paths through the nets a TIG was placed on the hitmap(0:7) signal.

- **TIG ON NET HIT_SEL_N_44, HIT_SEL_N_46:**

There was a software bug in the timing analyzer when the output of a flip-flop in a CLB was looped back as input to the other flip-flop in the same CLB whose output was under consideration. Such paths were manually checked in the FPGA editor and hence to avoid the bug a TIG was placed on these two nets.

- **CHIP_SEL (0:5) - clk -> pad delay of 48ns after sixty mhz sysclk:**

Ele_num changes on phase 10 and the decoder outputs the chip_sel number, which is not used until phase2. This gives a time delay of 5 sysclk cycles before when the chip_sel signal is used. So conservatively a delay of 48ns (nearly 3 sysclk cycles) was constrained on the chip_sel signal.

- **CH_SEL_NUM (0:2) – clk -> pad delay of 24ns after 60 mhz sysclk:**

The ch_sel num, which is used during phase 8 readout and during phase 3,4,5,6, is latched in by the elephant chip on the rising edge of the clk15 signal. After reading out one full channel there is time delay phase 9 to give enough time for the changing of more_channel and ch_sel_num after next signal changes, an effective 4 sysclk cycles. In phase 3,4,5,6 phase 3 has time during phase 2a, phase2b to get the correct ch_sel_num and phase 6 has time till phase 7 before which ch_sel_num has to be valid. Phase 4, Phase5 has the tightest constraint and assuming the elephant chip latches the signal on the rising edge of clk15 there is a time allowance of 1.5 sysclk cycles (from software simulations) and another allowance of 6ns from sixty mhz -> clk on the flip-flop. Thus conservatively the constraint on the ch_sel num signal was 24ns.

- **DATA_START, DATA_INC, SRAMENB – clk -> pad of 24ns after 60 mhz sysclk:**

The above control signals are latched in by the elephant chip on the rising edge of the clk15 and from software simulations and measurement of clk15 wrt. sysclk there is an allowance of 1.5 sysclk cycles and another 6ns from sixty mhz -> clk on the flipflop. Thus the constraints on the signals are 24ns. It should also be noted here that there is some variation on the phase difference of clk15 to sysclk and it has been minimally accounted.

The following constraints are a vestige from the time before, when the `fifo_dat` was not clocked on the output. In some cases they are still pertinent and in some not necessary.

- **FIFO_WR – clk -> pad of 33ns after sixty mhz sysclk:**

Even after the change, under the worst case the `fifo_wr` signal can drift for the time `fifo_dat` is valid on the flip-flop edge, which is 4 `sysclk` cycles. The `fifo_wr` signal is 2 `sysclk` cycles wide starting from the same time, as `fifo_dat` is valid. Hence it has an allowance of 2 `sysclk` cycles and that is the constraint on the `fifo_wr` signal.

- **PAD TO PAD DELAY – 24ns:**

This constraint is no longer necessary since all the outputs are clocked and there is no direct path from the input to the output.

- **FIFO_DAT – clk -> pad of 32ns(for `fifo_dat(0:6)`) and 24ns(for `fifo_dat(7)`):**

In the new case, the `fifo_wr` is 2 `sysclk` cycles wide and the `fifo_dat` is valid for 4 `sysclk` cycles and both signals go valid at the same time. The setup time requirement of the FIFO is 9ns and hence the `fifo_dat` has a leeway of $2\text{sysclk cycles} - 9\text{ns} = 24\text{ns}$ and hence should be the constraint on the `fifo_dat`.

- **ELE_DAT – pad -> setup of 24ns before 60 mhz sysclk:**

The `ele_dat` has to be valid before the decision to latch the data to write is made. The time from `ele_dat` changing on the falling edge of the `clk15` to the `data_latch_en` signal is 1.5 `sysclk` cycles and hence the constraint of 24ns on the signal.