Project: WATCHMAN
University of Hawaii at Manoa

Meeting : Tuesday, 15th of January 2018, Week 18

Participants:

Jonathan Hendriks	jhendrik@hawaii.edu
Anthony Schluchin	schluchi@hawaii.edu
Jose Duron	jduron@hawaii.edu
Kurtis Nishimura	kurtisn@phys.hawaii.edu
Gary Varner	varner@phys.hawaii.edu

Jose :
Data Analysis of Tyler's Raw data on PMT signal

23% percent of the data is actual PMT signal, the rest is noise. The data analysis shows multiple amplitude and pulse wide, for instance the pulse of 8mV is shown on the slides.

From the readout of all waveforms of interest a fourier analysis shows that there are some peaks at :

- 376MHz
- 437MHz
- 967MHz

The frequency of 376MHz corresponds to UHF Ultra high frequency range, which is composed of TV broadcast, microwave ovens, etc

source :https://fccid.io/frequency-explorer.php?lower=345&upper=376

The other frequencies are probably similar, this mean the PMT can catch the noise, like discussed in the WATCHMAN collaboration meeting held from the 9th to 13th of January 2019.

Baseline subtraction should be done dynamically before the PMT pulse, this is to say calculate the average baseline before the pulse is noticed (over threshold) and after the filtering of the noise.

The waveforms have to be filtered to remove the undesired frequencies.

The sampling frequency is specified in the Tyler's email (20GS/s) with all other measurement parameters.

Anthony

During this last week, Anthony worked on finding the best way to correct the data. First, there is the pedestal subtraction, which is an offset correction. Every analog memory location (512*16*32) has its own pedestal value.

Once the pedestal subtraction is done, there is still a correction to do, due to the comparator (inside the ASIC). After testing and analysing, the conclusion is that every memory location has more less the same behavior. Therefore, the idea is to perform the fitting on the average of all samples, with that done all value can be correct using a LUT (look up table) of 2048 possibilities, the advantage of using this method is that it is much faster than computing all for each point.

Jonathan

For the past week, the goal was to drop down the number of LUTs used for the HDL. The utilization of resources is now okay after optimizing the system. The picture below shows the utilization report from the 17th of January after timing optimization.

Name 1	Slice LUTs (17600)	Slice Registers (35200)	F7 Muxes (8800)	F8 Muxes (4400)	Block RAM Tile (60)	Bonded IOB (100)	Bonded IOPADs (130)	IBUFDS (96)	BUFGCTRL (32)	MMCME2_ADV (2)
N base_zynq_wrapper	12281	14185	660	219	1.5	63	130	1	4	1
base_zynq_i (base_...	12281	14185	660	219	1.5	0	0	1	4	1
axi_dma_0 (base...	1042	1355	0	0	1.5	0	0	0	0	0
axi_iic_0 (base_zy...	414	368	9	4	0	0	0	0	0	0
axi_interconnect...	222	255	0	0	0	0	0	0	0	0
axistream_0 (bas...	119	104	0	0	0	0	0	0	0	0
FifoManagerV4...	675	736	24	12	0	0	0	0	0	0
iobuf_0 (base_zy...	0	0	0	0	0	0	0	0	0	0
iobuf_1 (base_zy...	0	0	0	0	0	0	0	0	0	0
processing_syst...	112	0	0	0	0	0	0	0	1	0
ps7_0_axi_peri...	561	731	0	0	0	0	0	0	0	0
rst_ps7_0_50M (...	19	40	0	0	0	0	0	0	0	0
TARGETC_IP_Prot...	9117	10596	627	203	0	0	0	1	3	1
xlconcat_0 (base...	0	0	0	0	0	0	0	0	0	0
xlconcat_1 (base...	0	0	0	0	0	0	0	0	0	0
xlconstant_0 (ba...	0	0	0	0	0	0	0	0	0	0

We see 12281 / 17600 LUT = 70% of LUTs. For the new design it will be best to invest 40 dollars more for a Zynq 7020 (Z-7020) rather than 7010 (Z-7010), which is currently used. The Zynq 7020 has 3 times more resources than the Zynq we are using right now.

		Cost-Optimized Devices						Mid-Range Devices			
	Device Name	Z-7007S	Z-7012S	Z-7014S	Z-7010	Z-7015	Z-7020	Z-7030	Z-7035	Z-7045	Z-7100
	Part Number	XC7Z007S	XC7Z012S	XC7Z014S	XC7Z010	XC7Z015	XC7Z020	XC7Z030	XC7Z035	XC7Z045	XC7Z100
Processing System (PS)	Processor Core	Single-Core ARM® Cortex™-A9 MPCore™ Up to 766MHz			Dual-Core ARM Cortex-A9 MPCore Up to 866MHz			Dual-Core ARM Cortex-A9 MPCore Up to 1GHz[1]			
	Processor Extensions	NEON™ SIMD Engine and Single/Double Precision Floating Point Unit per processor									
	L1 Cache	32KB Instruction, 32KB Data per processor									
	L2 Cache	512KB									
	On-Chip Memory	256KB									
	External Memory Support[2]	DDR3, DDR3L, DDR2, LPDDR2									
	External Static Memory Support[2]	2x Quad-SPI, NAND, NOR									
	DMA Channels	8 (4 dedicated to PL)									
	Peripherals	2x UART, 2x CAN 2.0B, 2x I2C, 2x SPI, 4x 32b GPIO									
	Peripherals w/ built-in DMA[2]	2x USB 2.0 (OTG), 2x Tri-mode Gigabit Ethernet, 2x SD/SDIO									
	Security[3]	RSA Authentication of First Stage Boot Loader, AES and SHA 256b Decryption and Authentication for Secure Boot									
	Processing System to Programmable Logic Interface Ports (Primary Interfaces & Interrupts Only)	2x AXI 32b Master, 2x AXI 32b Slave 4x AXI 64b/32b Memory AXI 64b ACP 16 Interrupts									
Programmable Logic (PL)	7 Series PL Equivalent	Artix®-7	Artix-7	Artix-7	Artix-7	Artix-7	Artix-7	Kintex®-7	Kintex-7	Kintex-7	Kintex-7
	Logic Cells	23K	55K	65K	28K	74K	85K	125K	275K	350K	444K
	Look-Up Tables (LUTs)	14,400	34,400	40,600	17,600	46,200	53,200	78,600	171,900	218,600	277,400
	Flip-Flops	28,800	68,800	81,200	35,200	92,400	106,400	157,200	343,800	437,200	554,800
	Total Block RAM (# 36Kb Blocks)	1.8Mb (50)	2.5Mb (72)	3.8Mb (107)	2.1Mb (60)	3.3Mb (95)	4.9Mb (140)	9.3Mb (265)	17.6Mb (500)	19.2Mb (545)	26.5Mb (755)
	DSP Slices	66	120	170	80	160	220	400	900	900	2,020
	PCI Express®	—	Gen2 x4	—	—	Gen2 x4	—	Gen2 x4	Gen2 x8	Gen2 x8	Gen2 x8
	Analog Mixed Signal (AMS) / XADC[2]	2x 12 bit, MSPS ADCs with up to 17 Differential Inputs									
	Security[3]	AES & SHA 256b Decryption & Authentication for Secure Programmable Logic Config									
	Speed Grades: Commercial	-1			-1			-1			-1
	Speed Grades: Extended	-2			-2,-3			-2,-3			-2
	Speed Grades: Industrial	-1, -2			-1, -2, -1L			-1, -2, -2L			-1, -2, -2L

However timing issues are now the problem indeed the rearranging/re-engineering of the round buffer cost some delays.

These delays are often not a problem like the parameter NBRWINDOW which specifies the number of windows we would like to digitize. This parameter is set and can take how much time to stabilize, because the command to start the digitization process comes after 2 AXI-Lite transaction = many clocks cycles later.

Kurtis recommends adding constraints to the HDL sources codes to remove all these timing errors.

With the timing errors removed the testbench simulation are working, as well as real test on hardware. The worst case is maybe for this test not happening but we can be sure that it will come along once so better to deal with it now.