

TARGET Readout Status

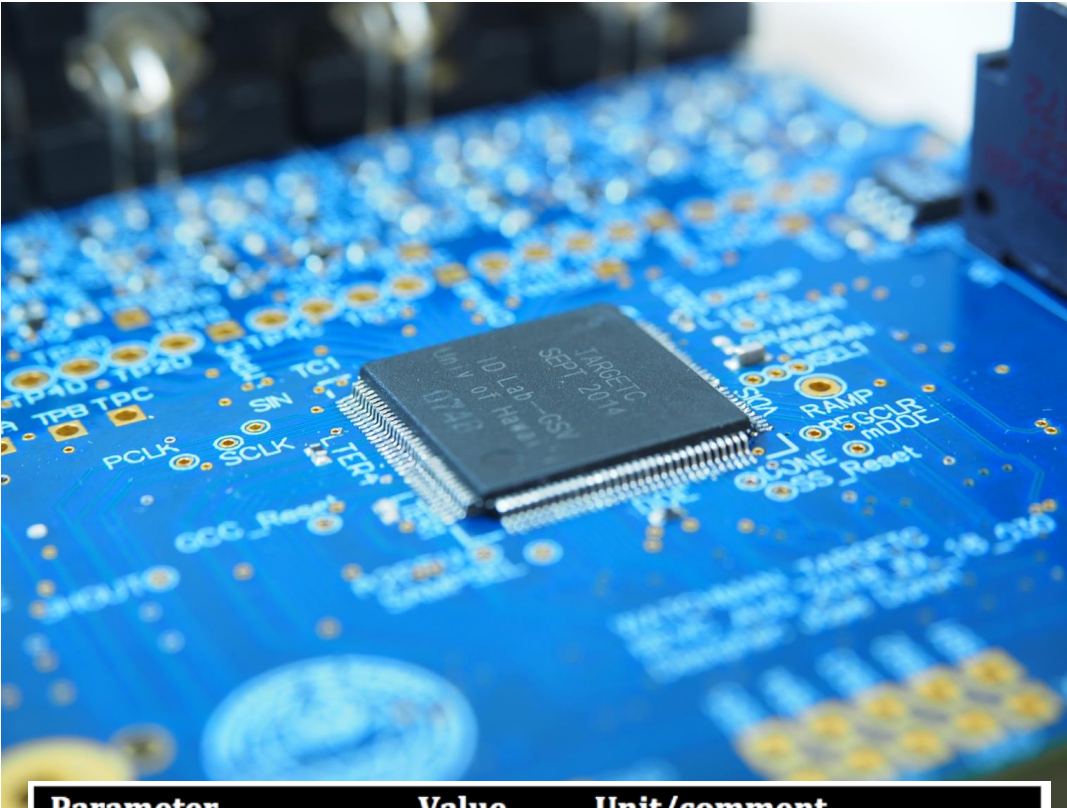
Jose Duron, Jonathan Hendriks,
Kurtis Nishimura,
Anthony Schluchin, Gary Varner

University of Hawaii

WATCHMAN DAQ Pre-Meeting
January 10, 2019



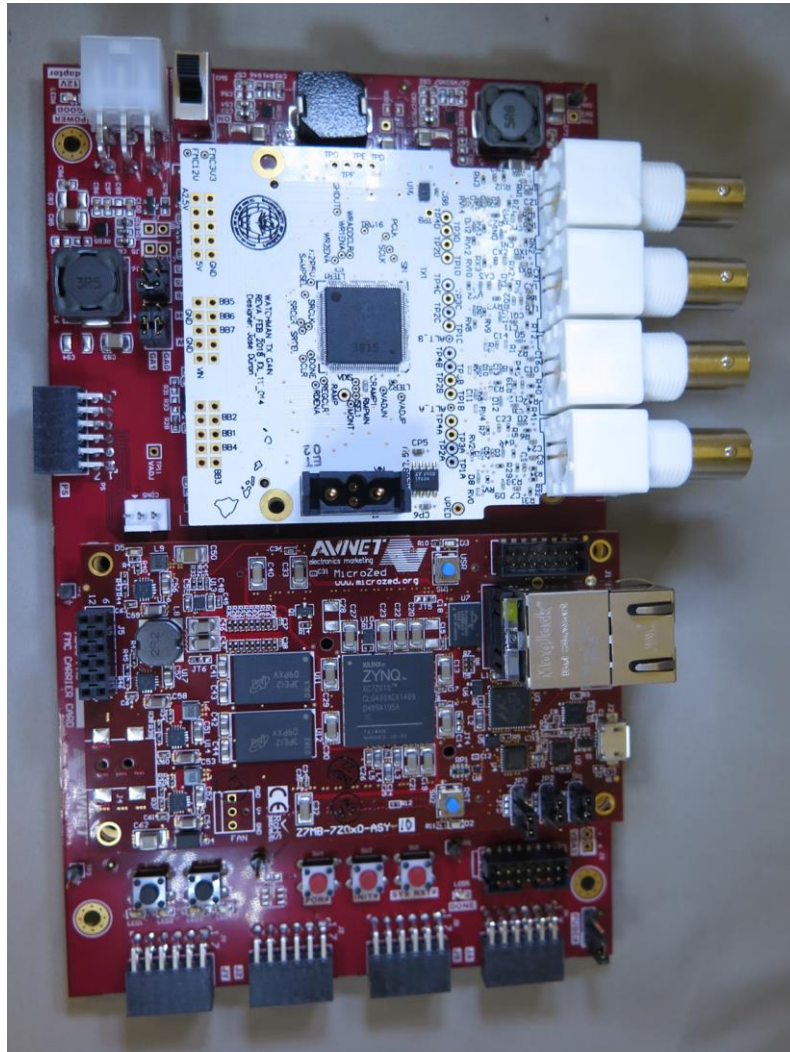
TARGET Introduction/Reminder



- TARGET series of ASICs are fast (~ 1 GSPS) waveform samplers developed by G. Varner.
- Switched-capacitor based sampling with deep analog storage.
- Main variants of interest for this talk:
 - TARGETX: >20k channels installed in Belle II.
 - TARGETC: used in Cherenkov Telescope Array.
 - (More on technical differences on next slide.)
- FMC prototype cards for WATCHMAN developed for both of these ASICs...

Parameter	Value	Unit/comment
Number of channels	16	
Sampling rate	0.4 – 1.2	GSPS
Storage depth	16,384	512 windows of 32 samples
Digitizer	On-chip	Wilkinson
Dynamic range	9-10	bits ENOB
Digitization time	2-8	μ s
Trigger output		Varies by ASIC

FMC Prototype Cards – Common



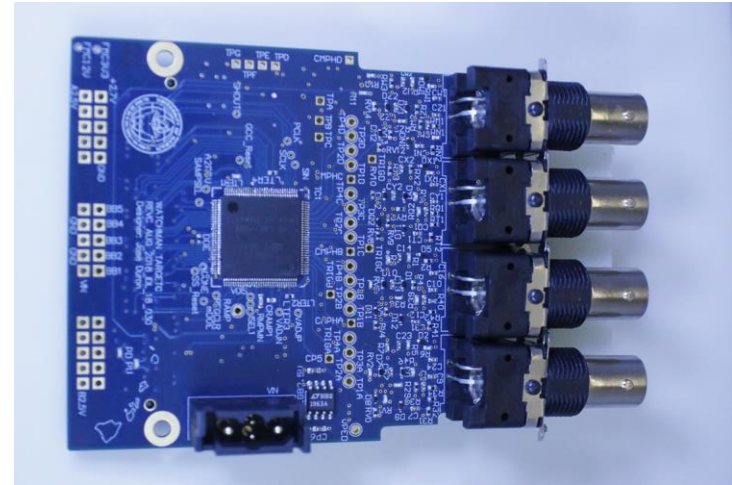
- FMC connector to FPGA card.
 - Tested/developed on μ Zed & Xilinx AC701.
- 4 PMT inputs (BNC).
- 4 gain stages per input:
 - X10
 - X1
 - x1/10
 - x1/100
- TARGET ASIC records waveforms from 16 total channels, on-chip Wilkinson ADC for digitization.
- What's the difference between TARGETX/C?

Two FMC TARGET Cards



TARGETX Card (first):

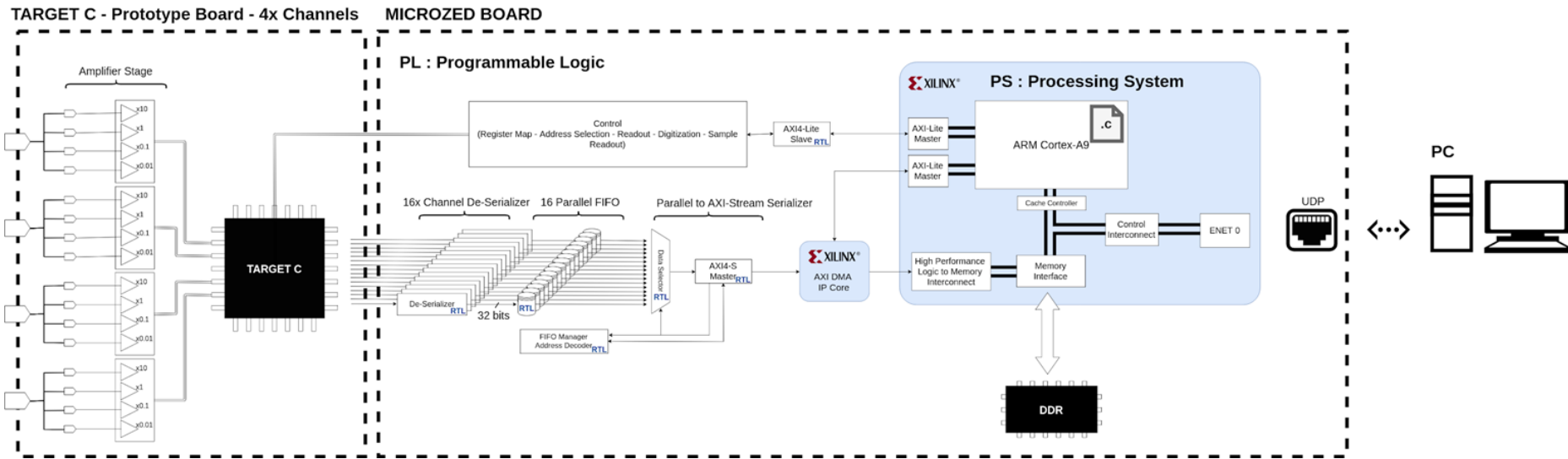
- First prototype, based on another similar design.
- TARGETX has internal trigger comparators, so does not require external components for identifying hits...
- ...but potentially can miss pulses (only has a “write enable” control to avoid overwriting).



TARGETC Card (present):

- TARGETC has no on-chip comparators...
 - Card uses external comparators (ADCMP601) on 10x gain channels to get maximum sensitivity.
- ...but TARGETC has full control over analog memory write addresses, can truly run dead time-less.
- ***Firmware and software development is currently focused on this card.***

TARGETC Firmware & Readout – Data Flow

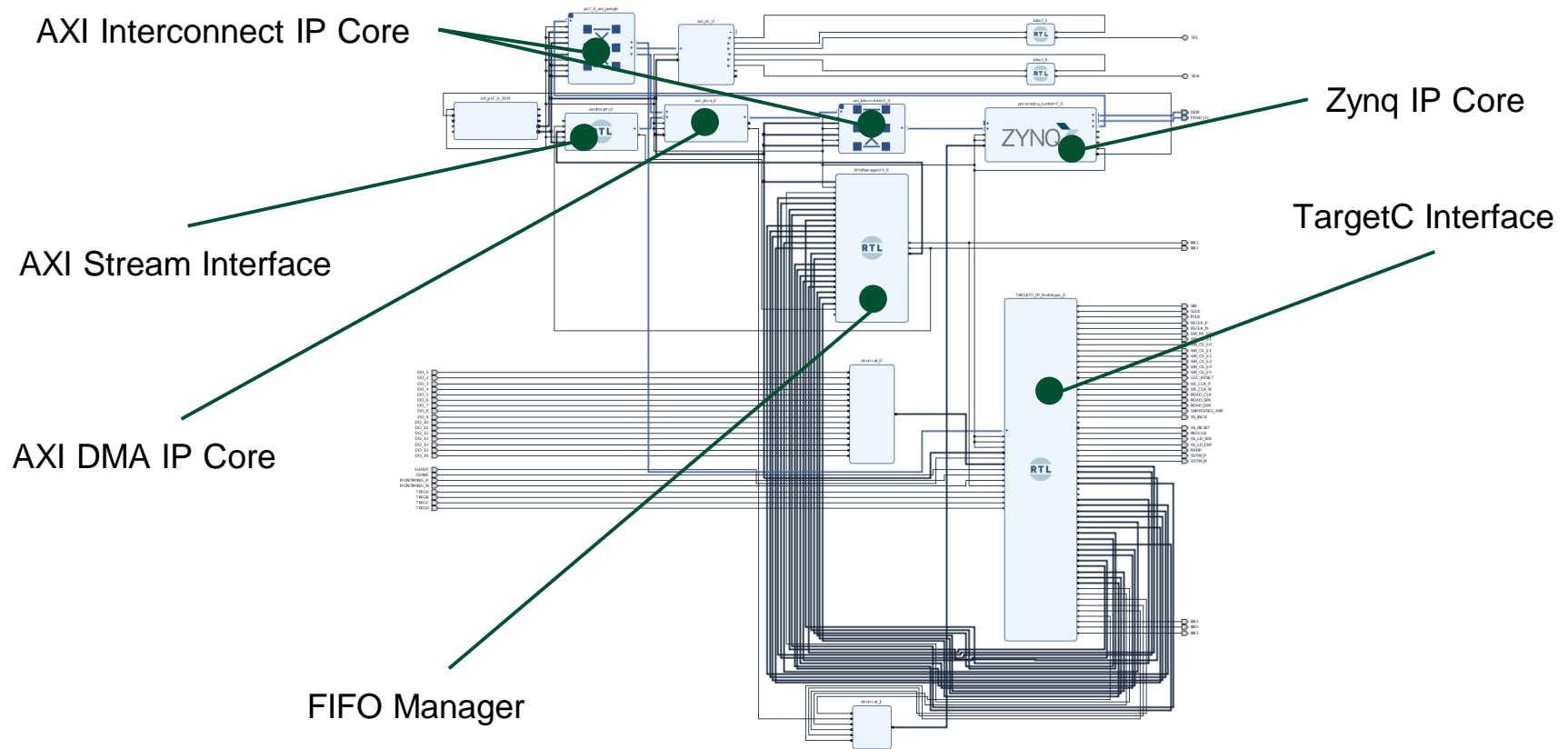


- Note that Zynq devices are FPGA fabric (PL) coupled with ARM microprocessor (PS).
- Present scheme uses PL for ASIC interfaces & readout.
- PL sends data to PS for processing (pedestal subtraction, feature extraction) and readout.
- PS interfaces to PC (GbE), data path for waveforms, control path for register interfaces.
- ➔ Development is for FMC prototype, but straightforward scalable to full system.

Status of UH TARGET Design, FW & SW

- **Prototype Version - (4x Channels)**
- Prototype design is well understood and is capable to meet the expectations.
- Validated :
 - Target C design.
 - VHDL system description.
 - Ethernet design readout.
 - Python Gui application.
- Ongoing:
 - Optimization of System (Fine tune of parameters of the ASIC).
 - Trigger system Hardware issue (VHDL simulated and validated).
 - Signal processing on full wave after signal compensation.
 - Real test with PMT signals.

Firmware Structure



- Developed using Vivado IP integrator, but important TARGET control blocks are VHDL.
- Project lives in github for version control.

Validation and Verification - Timing/Data Rate (Non-Optimized)

1

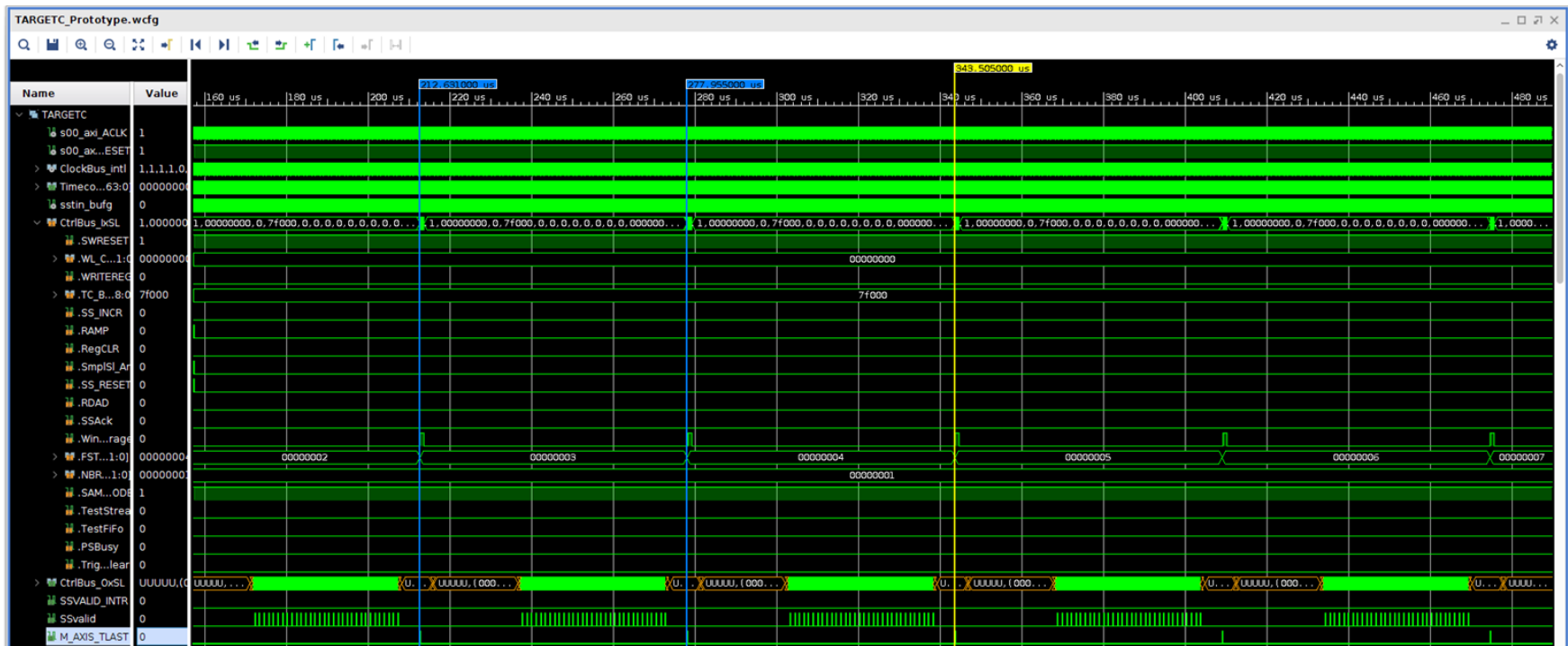
Sampling frequency 1GS/s

If the parameters of TARGET C ASIC are well optimized and calibrated the sampling frequency is up to 1GS/s in accordance with the TARGET Architecture.

Validation and Verification - Timing/Data Rate (Non-Optimized)

2

Readout latency WCS : AVG 65.53 us(sampling till readout 1st window : 32 samples)
Readout frequency : 15kHz



Validation and Verification - Timing/Data Rate (Non-Optimized)

2

Readout latency WCS :	AVG 65.53 us(sampling till readout 1st window : 32 samples)
Readout frequency :	15kHz

Values using non-optimized firmware. Optimization to be done :

- Ramp Charge (~20us overall charge => divide by 4 so 5 us)
Estimated reduction : - **15 us**
- Increment Sample (wait ~500 ns for 1 sample increment => 16 us for 32 samples readout, could be reduced probably to 125 ns)
Estimated reduction : - **12 us**

Overall reduction : 65us - 15 us - 12 us = 38 us

Frequency : 26 kHz

Validation and Verification - Timing/Data Rate (Non-Optimized)

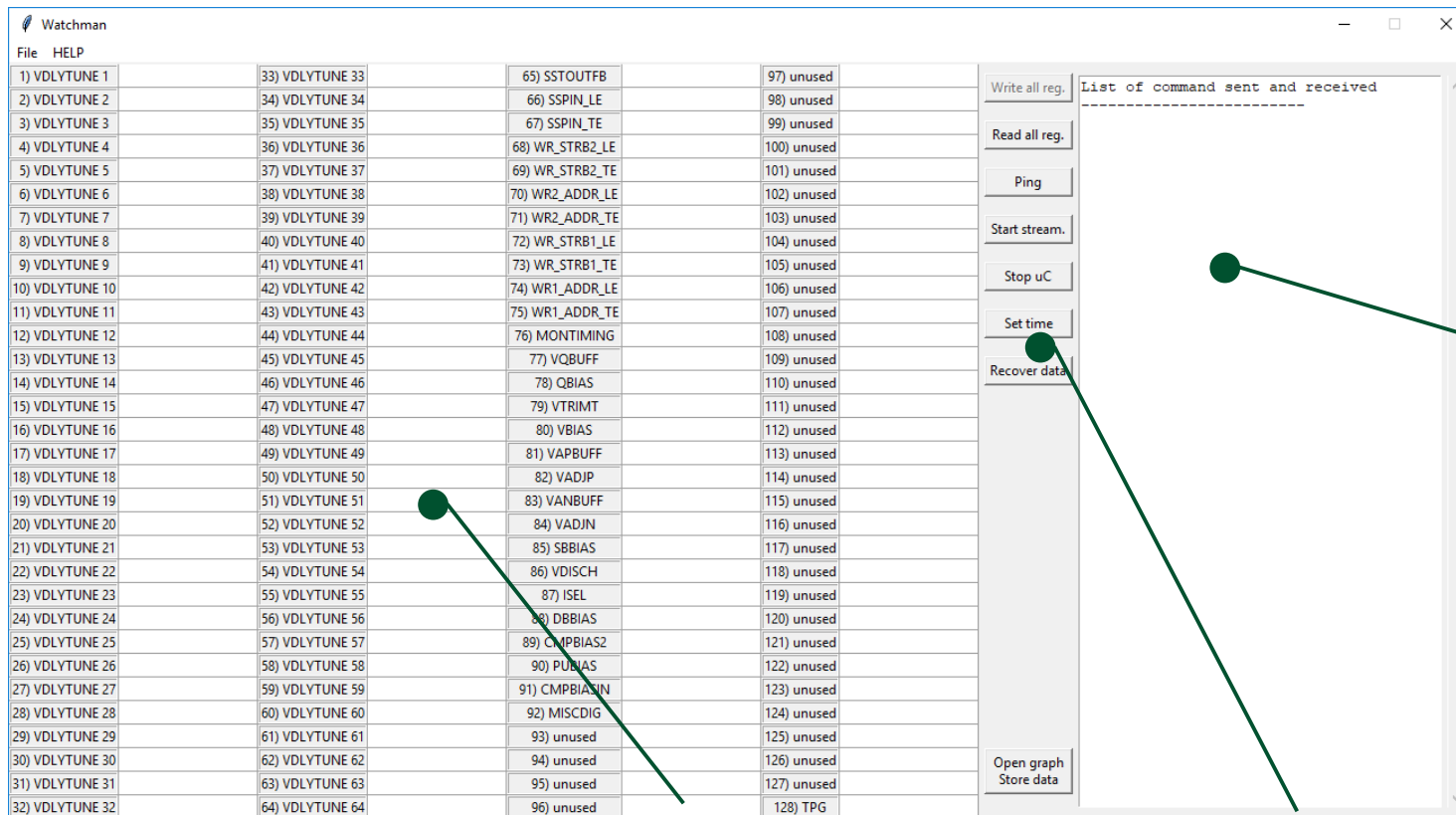
2

Readout latency WCS :	AVG 65.53 us(sampling till readout 1st window : 32 samples)
Readout frequency :	15kHz

Additional speed-up :

- Frequency increase on the different interfaces of TARGET C
- Data Minimizer send 2x 12bits on a single data of 32bits on AXI stream

Software Development – Python GUI



The screenshot displays the Watchman Python GUI. The main window is titled "Watchman" and contains a menu bar with "File" and "HELP". Below the menu bar is a large table with 4 columns, listing register addresses and names. The table is divided into four sections: 1) VDLYTUNE 1 to 32, 2) VDLYTUNE 33 to 64, 3) VDLYTUNE 65 to 96, and 4) 97) unused to 128) TPG. A green dot is placed on the table at row 51, column 2, with a line pointing to the label "TargetC Register Access". To the right of the table is a panel titled "List of command sent and received". This panel contains a list of commands: "Write all reg.", "Read all reg.", "Ping", "Start stream.", "Stop uC", "Set time", "Recover data", "Open graph", and "Store data". A green dot is placed on the "Set time" command, with a line pointing to the label "UDP Commands".

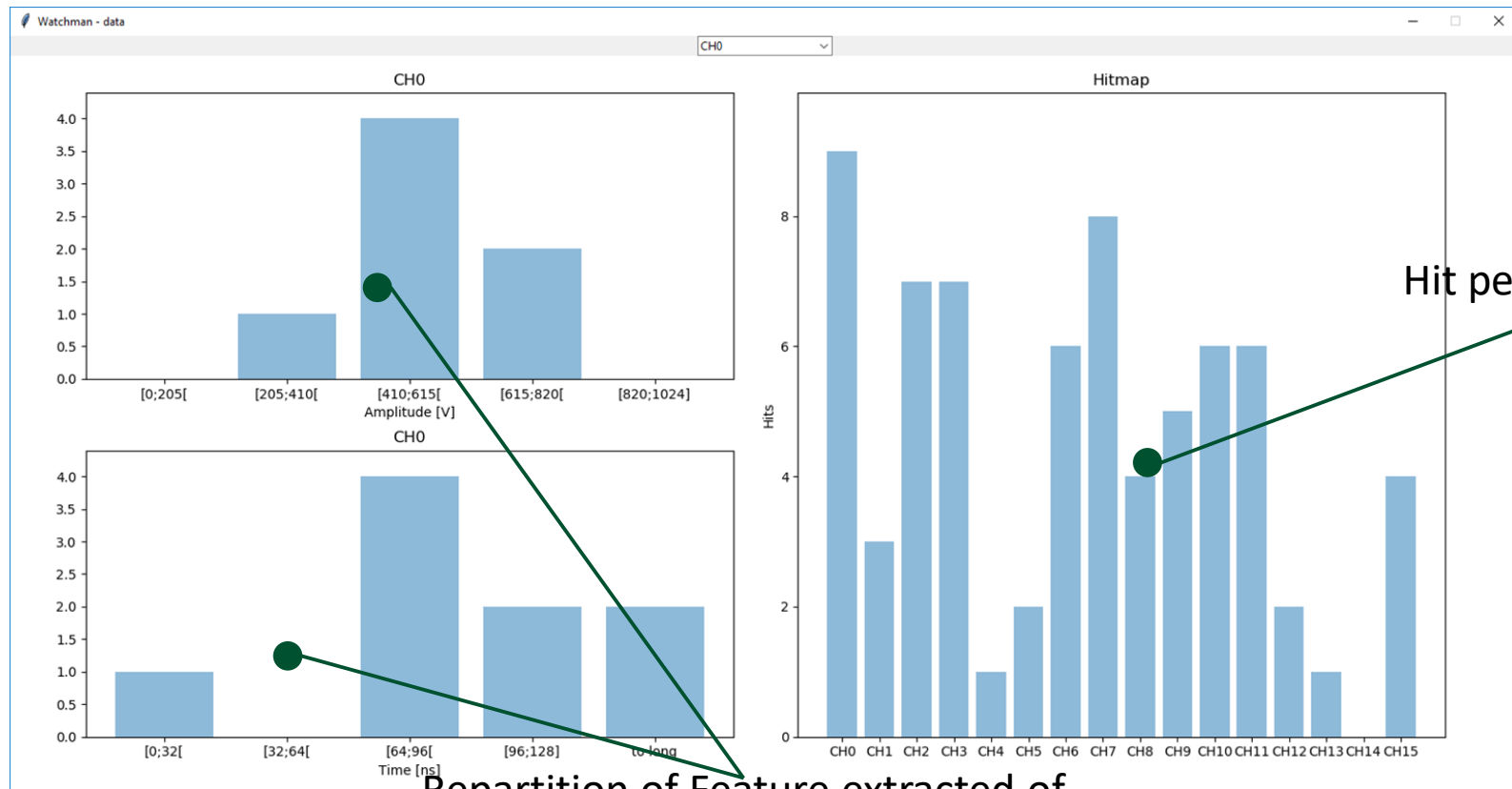
1) VDLYTUNE 1	33) VDLYTUNE 33	65) SSTOUTFB	97) unused
2) VDLYTUNE 2	34) VDLYTUNE 34	66) SSPIN_LE	98) unused
3) VDLYTUNE 3	35) VDLYTUNE 35	67) SSPIN_TE	99) unused
4) VDLYTUNE 4	36) VDLYTUNE 36	68) WR_STRB2_LE	100) unused
5) VDLYTUNE 5	37) VDLYTUNE 37	69) WR_STRB2_TE	101) unused
6) VDLYTUNE 6	38) VDLYTUNE 38	70) WR2_ADDR_LE	102) unused
7) VDLYTUNE 7	39) VDLYTUNE 39	71) WR2_ADDR_TE	103) unused
8) VDLYTUNE 8	40) VDLYTUNE 40	72) WR_STRB1_LE	104) unused
9) VDLYTUNE 9	41) VDLYTUNE 41	73) WR_STRB1_TE	105) unused
10) VDLYTUNE 10	42) VDLYTUNE 42	74) WR1_ADDR_LE	106) unused
11) VDLYTUNE 11	43) VDLYTUNE 43	75) WR1_ADDR_TE	107) unused
12) VDLYTUNE 12	44) VDLYTUNE 44	76) MONTIMING	108) unused
13) VDLYTUNE 13	45) VDLYTUNE 45	77) VQBUFF	109) unused
14) VDLYTUNE 14	46) VDLYTUNE 46	78) QBIAS	110) unused
15) VDLYTUNE 15	47) VDLYTUNE 47	79) VTRIMT	111) unused
16) VDLYTUNE 16	48) VDLYTUNE 48	80) VBIAS	112) unused
17) VDLYTUNE 17	49) VDLYTUNE 49	81) VAPBUFF	113) unused
18) VDLYTUNE 18	50) VDLYTUNE 50	82) VADJP	114) unused
19) VDLYTUNE 19	51) VDLYTUNE 51	83) VANBUFF	115) unused
20) VDLYTUNE 20	52) VDLYTUNE 52	84) VADJIN	116) unused
21) VDLYTUNE 21	53) VDLYTUNE 53	85) SBBIAS	117) unused
22) VDLYTUNE 22	54) VDLYTUNE 54	86) VDISCH	118) unused
23) VDLYTUNE 23	55) VDLYTUNE 55	87) ISEL	119) unused
24) VDLYTUNE 24	56) VDLYTUNE 56	88) DBBIAS	120) unused
25) VDLYTUNE 25	57) VDLYTUNE 57	89) CMPBIAS2	121) unused
26) VDLYTUNE 26	58) VDLYTUNE 58	90) PUBIAS	122) unused
27) VDLYTUNE 27	59) VDLYTUNE 59	91) CMPBIASIN	123) unused
28) VDLYTUNE 28	60) VDLYTUNE 60	92) MISCDIG	124) unused
29) VDLYTUNE 29	61) VDLYTUNE 61	93) unused	125) unused
30) VDLYTUNE 30	62) VDLYTUNE 62	94) unused	126) unused
31) VDLYTUNE 31	63) VDLYTUNE 63	95) unused	127) unused
32) VDLYTUNE 32	64) VDLYTUNE 64	96) unused	128) TPG

TargetC Register Access

UDP Commands

UDP Terminal

Software Development – Python GUI

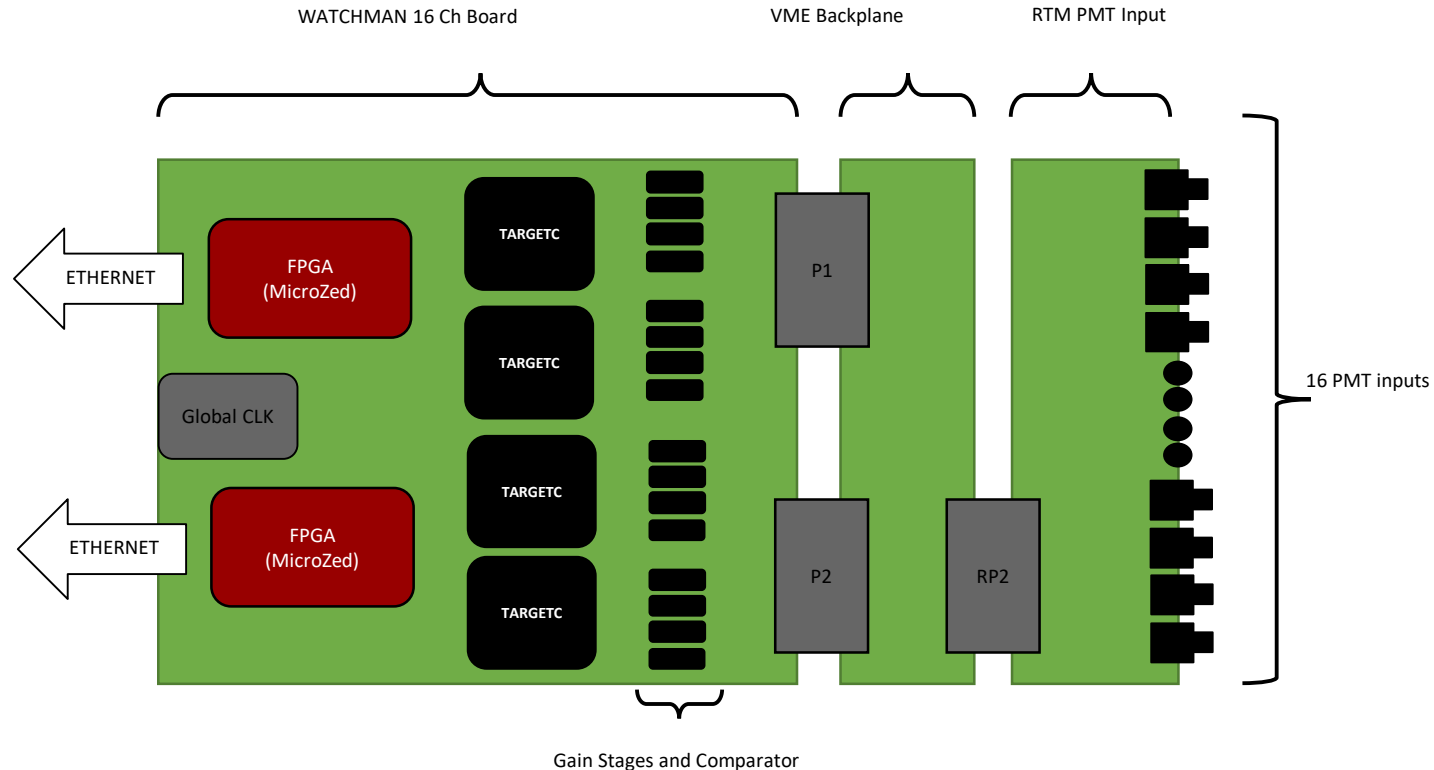


Repartition of Feature extracted of
Channel 0

Hit per Channel

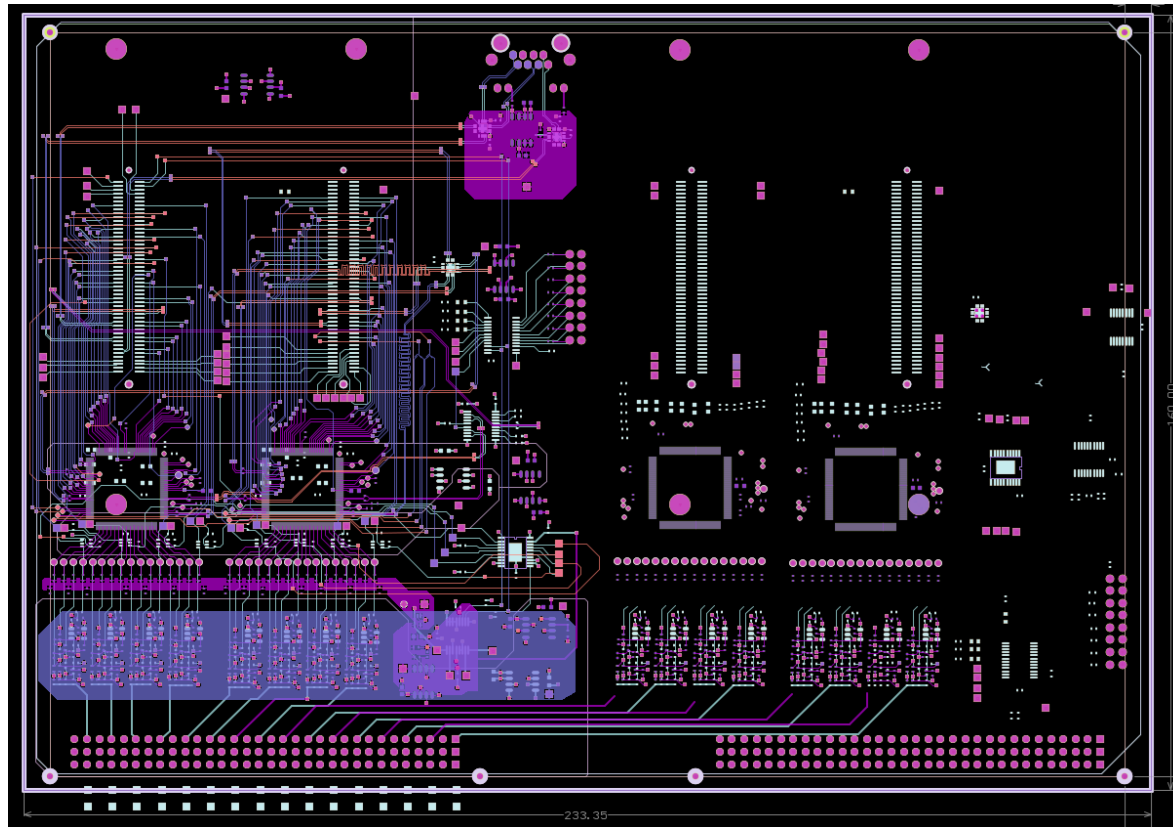
HW Development – Toward 16 Channel VME

16x Channel Version - Conceptual Design



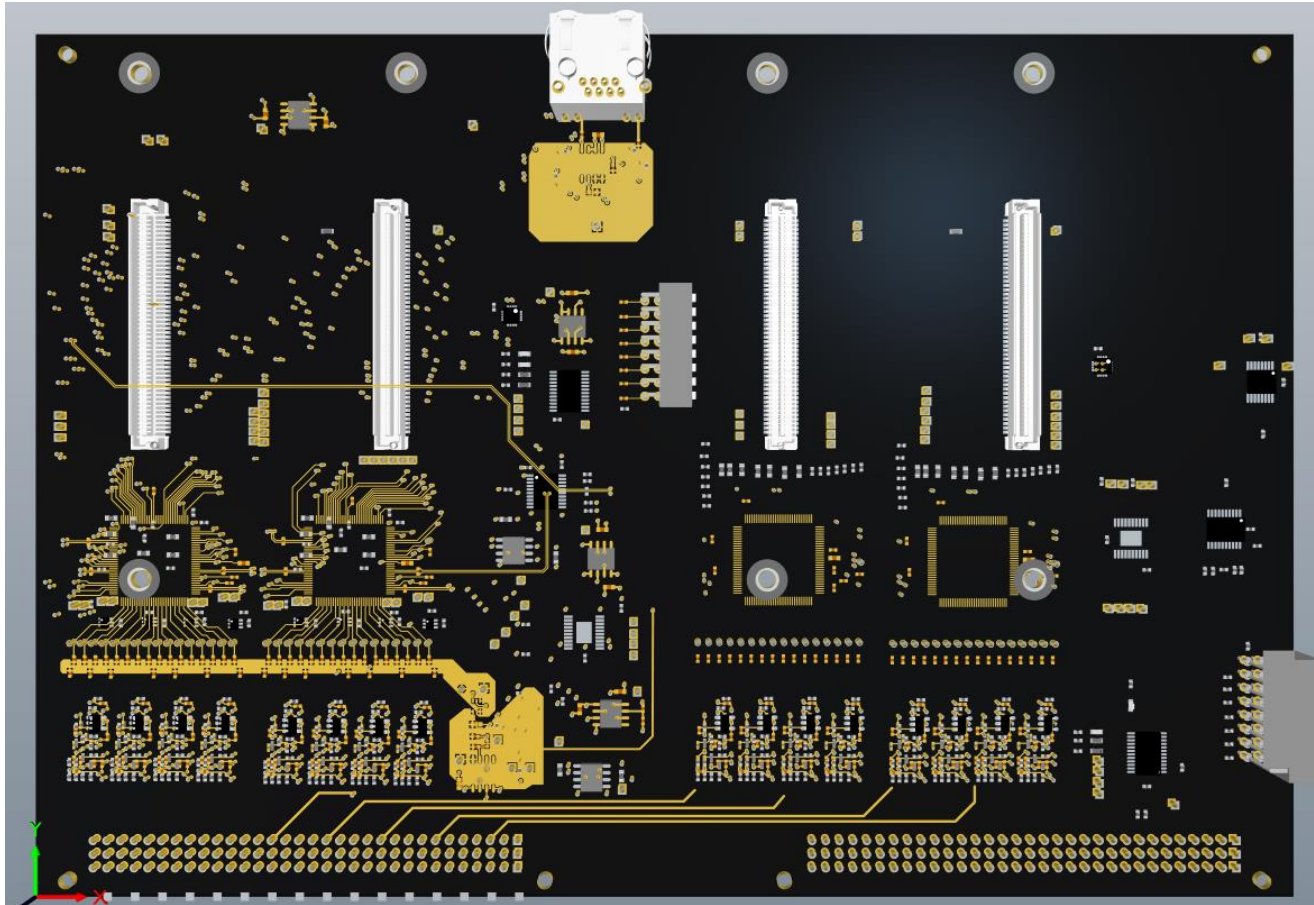
- First VME version uses dual MicroZed boards for rapid development.
 - This version is presently in layout.
- FPGA can be pulled onto board in next revision.

Initial VME Board – Layout in Progress



- Final layout pending some final work on TARGETC FMC prototype:
 - Debugging of trigger comparators, verification of gain stages through full flow.

Initial VME Board – Layout in Progress



- Final layout pending some final work on TARGETC FMC prototype:
 - Debugging of trigger comparators, verification of gain stages through full flow.

Hardware, Firmware Status

Hardware		Status
	TARGET C Prototype Board 4x Channel	ACHIEVED
	- Final analog verification (comparators, amplifiers).	1-2 weeks
	- Investigate power supply noise.	1 week
	Use PMT as test input with HV splitter.	1 week
	Complete initial 16 channel TARGET C VME board.	3-4 weeks



PMT HV/signal splitter based on UCD schematics

Firmware		Status
	PL → PS DMA Interface (AxiStream).	ACHIEVED
	TARGET C interface & control: - Registers, sample readout, & digitization.	ACHIEVED
	Circular buffer – design, simulation with trigger system).	ACHIEVED
	Circular buffer – optimization for resource utilization.	1 week
	Project cleanup.	1-3 day
	System test of trigger & circular buffers.	1-2 week
	Adapt firmware to 16 channel VME board.	1-2 week

Software & Operations		Status
	Initial python GUI.	ACHIEVED
	Interrupt systems - timers, watchdogs, data, GbE.	ACHIEVED
	UDP communication - commands, data, registers.	ACHIEVED
	Timestamping.	ACHIEVED
	Log files saved into the SD card.	ACHIEVED
	Data processing - pedestals.	ACHIEVED
	Data processing – transfer functions.	3 days
	Data processing - feature extraction.	ACHIEVED
	Optimize TARGET C parameters.	1-2 weeks
	Integrated test with PMT.	2 weeks

Current Manpower

Hardware development:

- Jose Duron [Master Student] – graduating spring/summer

Firmware development:

- Jonathan Hendriks [visiting Master's student] – departing February.

Software development:

- Anthony Schluchin [visiting Master's student] – departing February.

Advising/supervising:

- Gary Varner
- Kurtis Nishimura

- ➔ Searching for $\geq 50\%$ postdoc to handoff development.
- ➔ Vasily Shebalin [postdoc] ramping up in a few months.
- ➔ Jeff Kleyner [EE Ph.D. student] joining later this semester.

Summary and Outlook

- 4-channel TARGETC FMC prototype:
 - Firmware and software development is nearly complete for performance evaluations.
 - Should wrap up this month, cards can then be fully characterized.
- 16-channel VME prototype:
 - Layout underway on a first version, fabrication pending some final debugging and performance checks on FMC card.
 - Firmware for this version should be straightforward adaptation of FMC prototypes.
 - A second streamlined version is envisioned shortly after.
- Personnel changes:
 - A number of these coming in the next months.
 - Working to ensure smooth transition, sufficient coverage for work through downselect – V. Shebalin [Postdoc], J. Kleyner [EE Grad].

BACKUP

Feature Extraction

- **Example from another Zynq system (Belle II TOP):**
- Rough benchmarking: 3.6 μsec to process each pair of hits using 1 core of ARM PS.
 - 1.5 μsec for pedestal subtraction,
 - 2.1 μsec for feature extraction.
 - Corresponds to ~ 550 kHz hits processed.
 - Not very optimized, may be able to speed up FE code in the PS, or can move some of it to PL.
- Roughly consistent with initial results seen on TARGETX w/ μZed .

