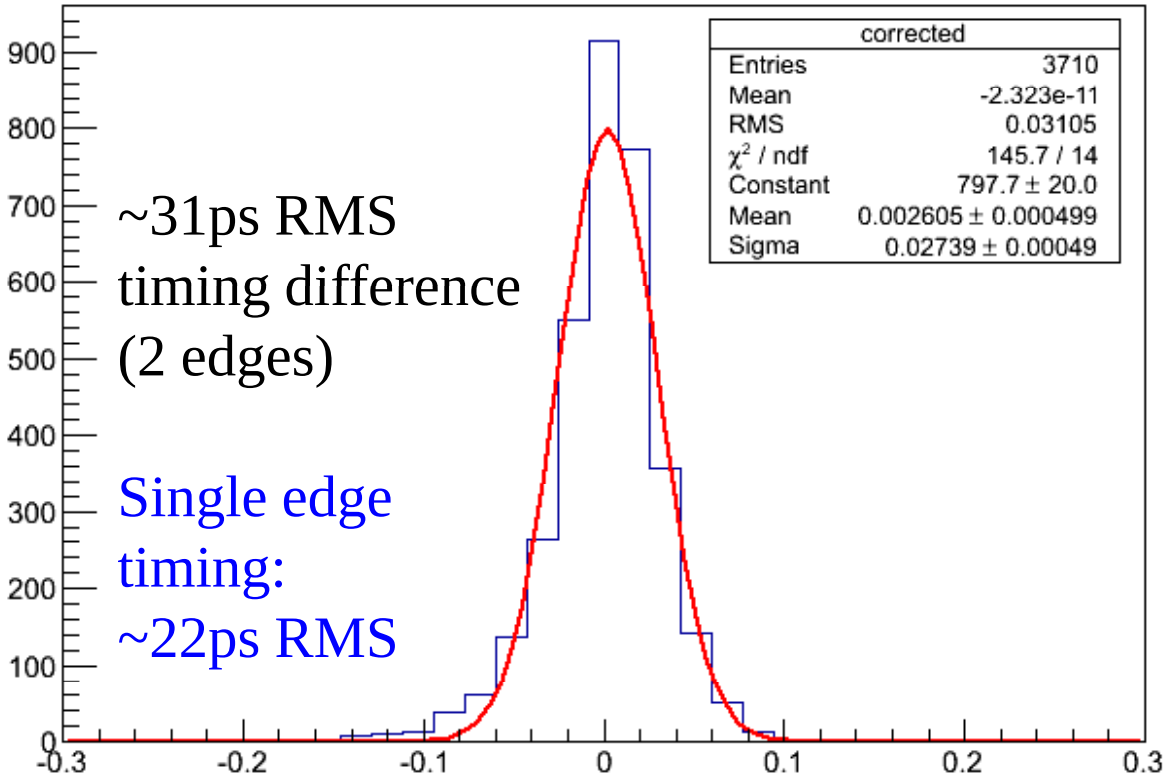
Understanding IRS3B, board-stack and mTC operation/calibration

Calibration in progress



Gary S. Varner

2 JUL 2014

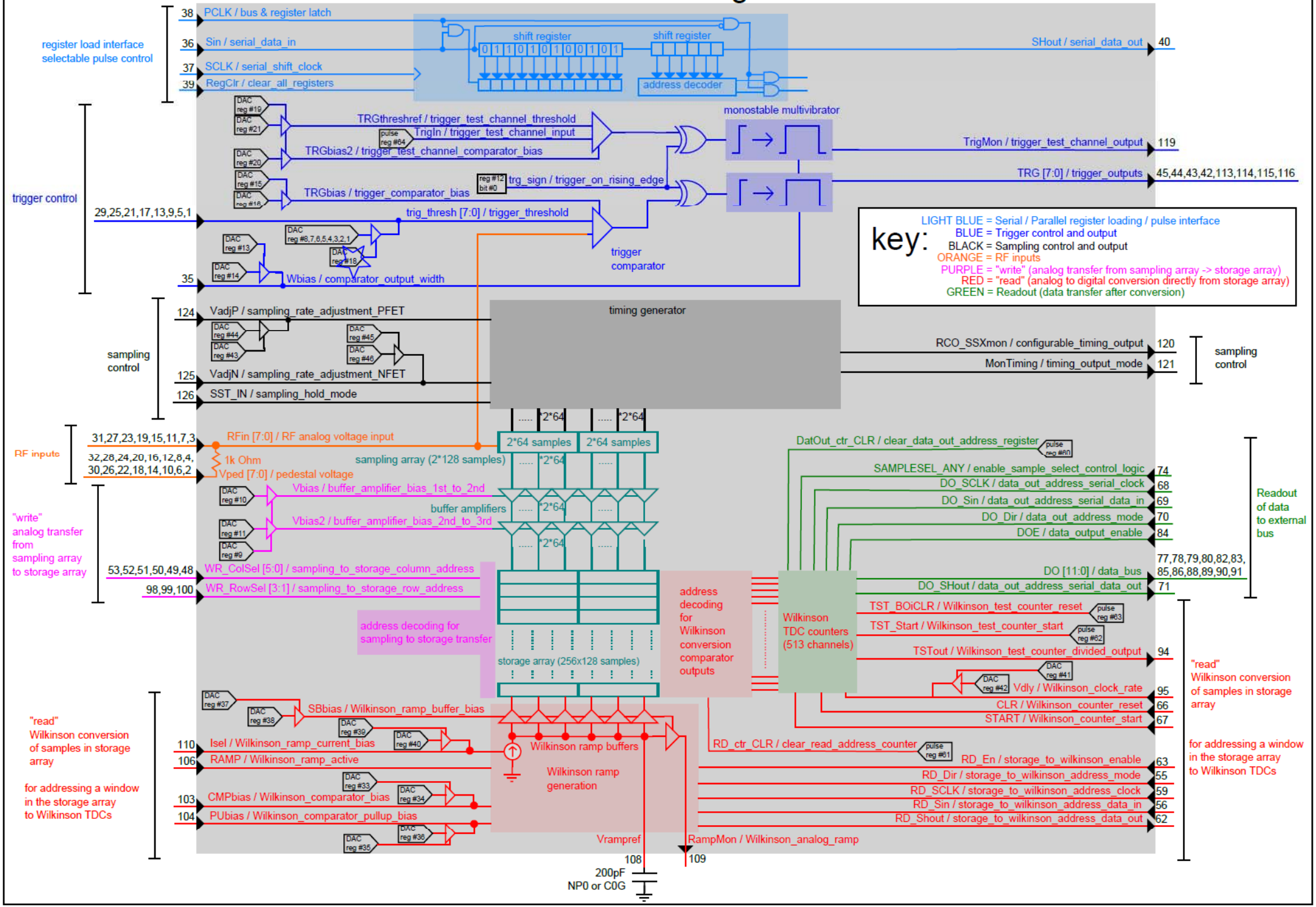
mTC Training Session #3

Roadmap

- mTC Readout
  - Currently at about Phase 1.5
  - You can get us to Phase 2.5
- Specifically
  - Operators need to understand Hardware/Firmware/Software
  - Develop real-time Data Quality Monitoring
- What I hope to convey:
  1. Details of the hardware: ASIC + boardstack
  2. Firmware and Configuration/Operating parameters
  3. Understand how to read and comprehend documentation and ask meaningful questions (“it doesn’t work” notably not amongst them)

IRS3B block diagram

2012-10-23



http://www.phys.hawaii.edu/~mza/ASIC/IRS2-BLAB3A/index.html

"PCLK #"	Register/Value	Default value	# bits		
1	Threshold Ch. 1	TBD (0x000)	12		
2	Threshold Ch. 2	TBD (0x000)	12		
3	Threshold Ch. 3	TBD (0x000)	12		
4	Threshold Ch. 4	TBD (0x000)	12		
5	Threshold Ch. 5	TBD (0x000)	12		
6	Threshold Ch. 6	TBD (0x000)	12		
7	Threshold Ch. 7	TBD (0x000)	12		
8	Threshold Ch. 8	TBD (0x000)	12		
9	VBDbias	0x400	12	DAC buff bias for Vbias, Vbias2	
10	Vbias	0x380	12	Buff amp bias -- initial sample	
11	Vbias2	0x370	12	Buff amp bias -- transfer samples	
12	Misc Reg (incl. SGN)	0x000	8		
13	WBDbias	0x400	12	DAC buff bias for Wbias	
14	Wbias	0x3D7	12	Trigger Width adjust (Wbias)	
15	TCBbias	0x400	12	DAC buff bias for Trigger Comparator	
16	TRGbias	0x350	12	Trigger Comparator bias	
17	THDbias	0x400	12	DAC buff bias for Trigger Thresholds	
18	Tbbias	0x400	12	Internal Trigger Threshold buffer bias	(disable for external drive)
19	TRGDbias	0x400	12	DAC buff bias for TRGbias2, TRGthref	
20	TRGbias2	0x350	12	TRGbias for reference channel	
21	TRGthref	0x800	12	Trigger threshold for reference channel	
22	Leading Edge SSPin	0x060	8	Timing Gen: SSPin	
23	Trailing Edge SSPin	0x010	8		
24	Leading Edge S1	0x028	8	Timing Gen: S1	
25	Trailing Edge S1	0x058	8		
26	Leading Edge S2	0x068	8	Timing Gen: S2	
27	Trailing Edge S2	0x018	8		
28	Leading Edge PHASE	0x018	8	Timing Gen: PHASE	
29	Trailing Edge PHASE	0x030	8		
30	Leading Edge WR_STRB	0x040	8	Timing Gen: WR_STRB	NOTE: WR_ADDR phase CRITICAL
31	Trailing Edge WR_STRB	0x070	8		
32	Timing Generator Reg	0x***	8	Select Timing signals viewed, Phase clear, RCO running	decode for
33	PDDbias	0x400	12	DAC buff bias for CMPbias	
34	CMPbias	0x500	12	Storage Cell Comparators (Pull-Down) bias	
35	PUDbias	0x400	12	DAC buff bias for CMPbias	
36	PUbias	0xBF0	12	Storage Column Comparators (Pull-Up) bias	
37	SBDbias	0x400	12	DAC buff bias for Super Buffer bias	
38	Sbbias	0x400	12	Super Buffer bias	
39	ISDbias	0x400	12	DAC buff bias for ISEL	
40	ISEL	0x900	12	Voltage Ramp Current	Initial config required
41	VDDbias	0x400	12	DAC buff bias for Vdly	
42	Vdly	0xB54	12	Wilkinson counter adj voltage	Servo-lock target value
43	VAPDbias	0x600	12	DAC buff bias for VadjP	
44	VadjP	0x5E8	12	Timing Generator delay adjust PMOS	Initial config required
45	VANDbias	0x600	12	DAC buff bias for VadjN	
46	VadjN	0xAD0	12	Timing Generator delay adjust NMOS	Servo-lock target value
60	DatOut_ctr_CLR	AddrMode	ADDR_M	Clear Data Output Address Registers	
61	RD_ctr_CLR	AddrMode	ADDR_M	Clear Read Address Registers	
62	Start_WilkMon	AddrMode	ADDR_M	Start Wilkinson Reference counter	
63	Boin_CLR	AddrMode	ADDR_M	Stop/Clear Wilkinson Reference counter	
64	Trig_In	AddrMode	ADDR_M	Pulse Trigger test circuit	

scan required

Hopefully, these will start to make sense in terms of how these settings map onto operational parameters of the IRS3B ASIC

irs3dControl_writeDefaultDacRegs.cpp

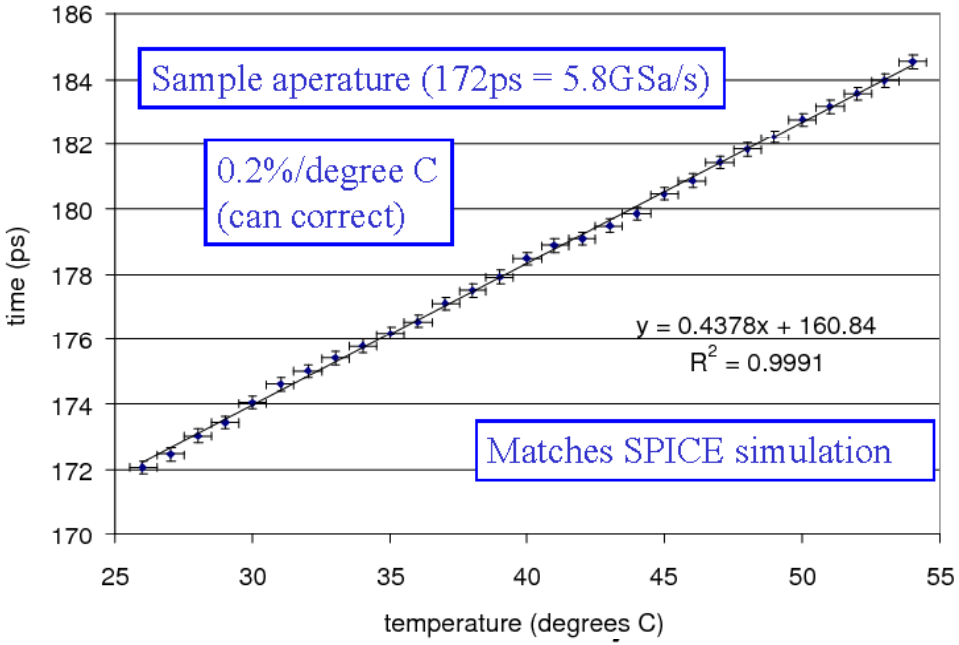
```
// Wilkinson comparator biases
control->writeDACReg(board_id,  169, 737);  // CMPbias2
control->writeDACReg(board_id,  170, 3112);  // PUbias
control->writeDACReg(board_id,  171, 1000);  // CMPbias
//
// Vramp Related Controls  PCLK 173-176
control->writeDACReg(board_id,  172, 1300);  // SBbias
control->writeDACReg(board_id,  173, 0);  // Vdischarge
control->writeDACReg(board_id,  174, 2900);  // ISEL
control->writeDACReg(board_id,  175, 1300);  // DBbias
//
// PLL-related DACs
control->writeDACReg(board_id,  176, 4095);  // VtrimT
control->writeDACReg(board_id,  177, 1500);  // Qbias
control->writeDACReg(board_id,  178, 1300);  // Vqbuff
//
//Misc. Timing Select Register (8-bit) -- PCLK 180
// 1 = Sel0
// 2 = Sel1
// 4 = Sel2
// 32 = SSTSEL
// 64 = Time1Time2
// 128 = CLR_PHASE
//
// value currently = 0 + 0 + 0 + 0 + 0 + 0
control->writeDACReg(board_id,  179, 0);
//
// Timebase Control -- analog   PCLK 181 - 184
control->writeDACReg(board_id,  180, 2800);  // VadjP
control->writeDACReg(board_id,  181, 1300);  // VAPbuff
control->writeDACReg(board_id,  182, 120);  // VadjN
control->writeDACReg(board_id,  183, 0);  // VANbuff -- disable for PLL operation
//
// Timebase Control -- Digital  PCLK 185 - 197
control->writeDACReg(board_id,  184, 0);  // WR_SYNC_LE  -- IRSX only?
control->writeDACReg(board_id,  185, 0);  // WR_SYNC_TE  -- IRSX only?
control->writeDACReg(board_id,  186, 96);  // SSPin_LE  0x60
control->writeDACReg(board_id,  187, 16);  // SSPin_TE  0x10
control->writeDACReg(board_id,  188, 40);  // S1_LE  0x28
control->writeDACReg(board_id,  189, 88);  // S1_TE  0x58
control->writeDACReg(board_id,  190, 104);  // S2_LE  0x68
control->writeDACReg(board_id,  191, 24);  // S2_TE  0x18
```

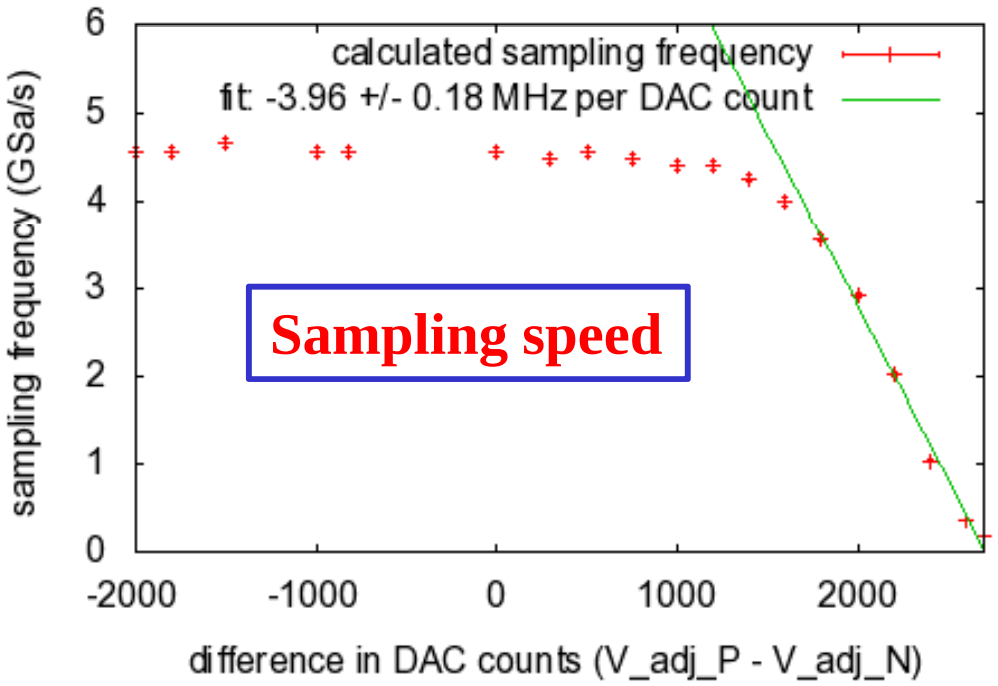
Simple software

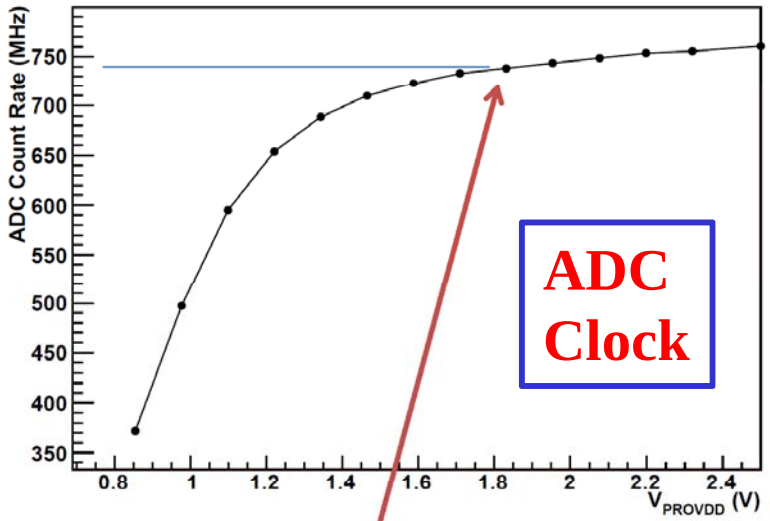
Once have writing template that talks to the picoblaze, tuning operational parameters of the IRS3B ASIC really easy

Servo-controls (3 parameters)

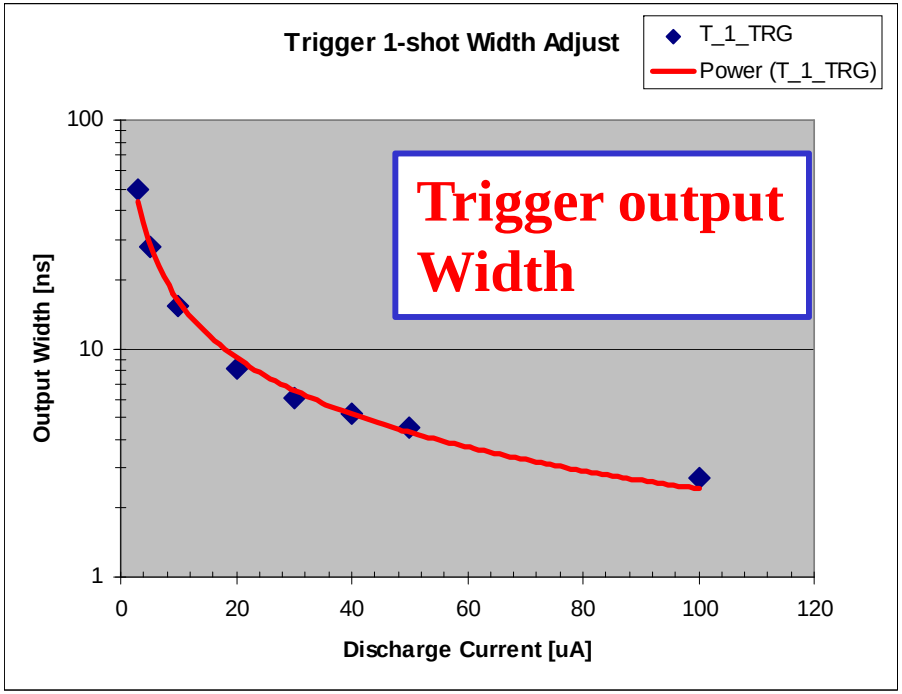
Temperature Dependence





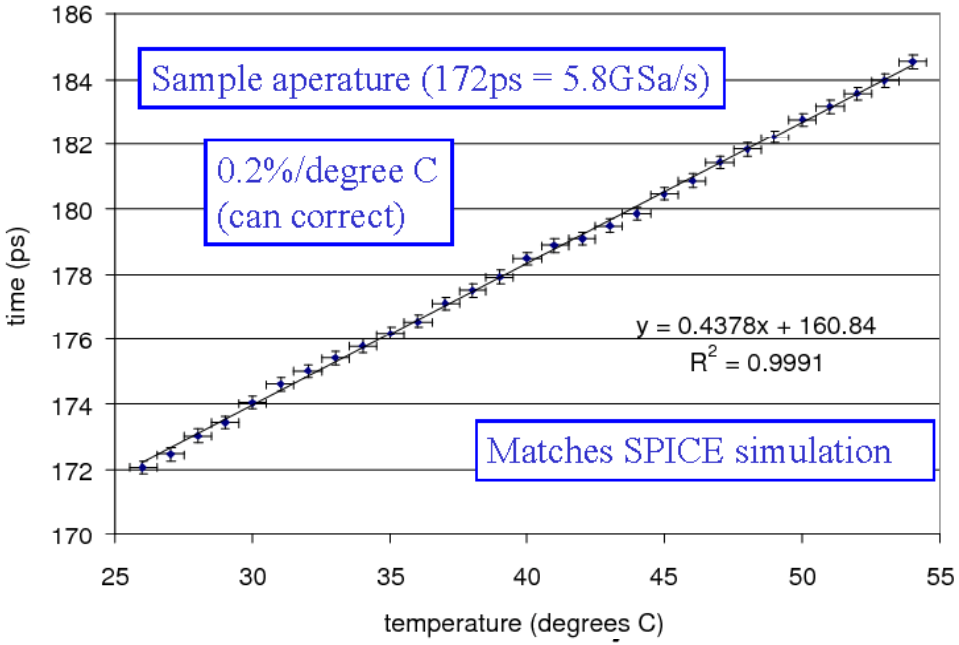


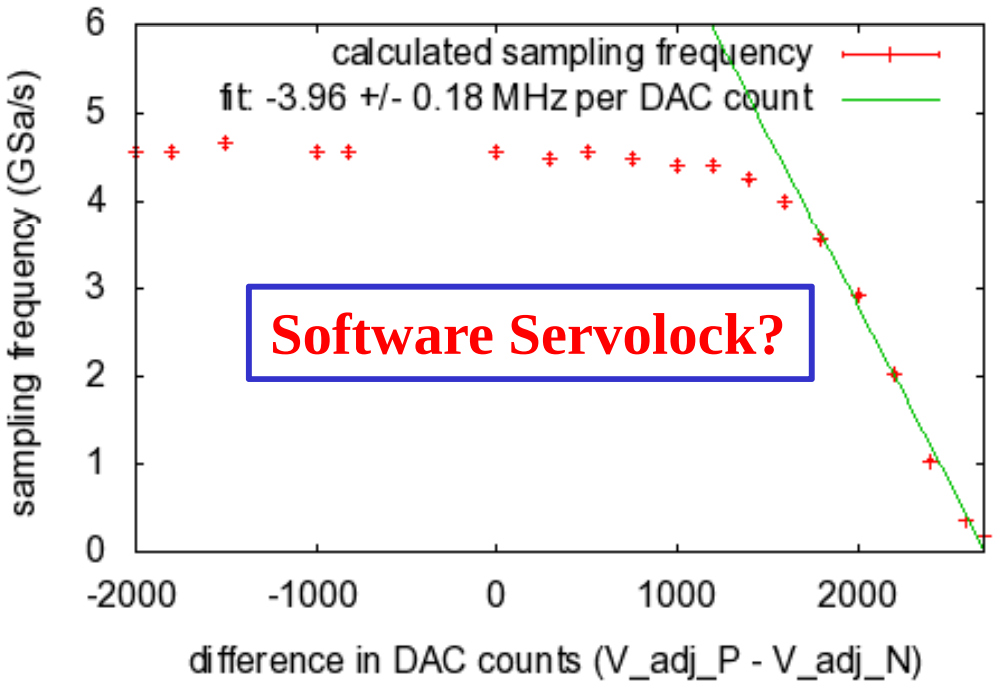
This is the Wilkinson clock adjustment

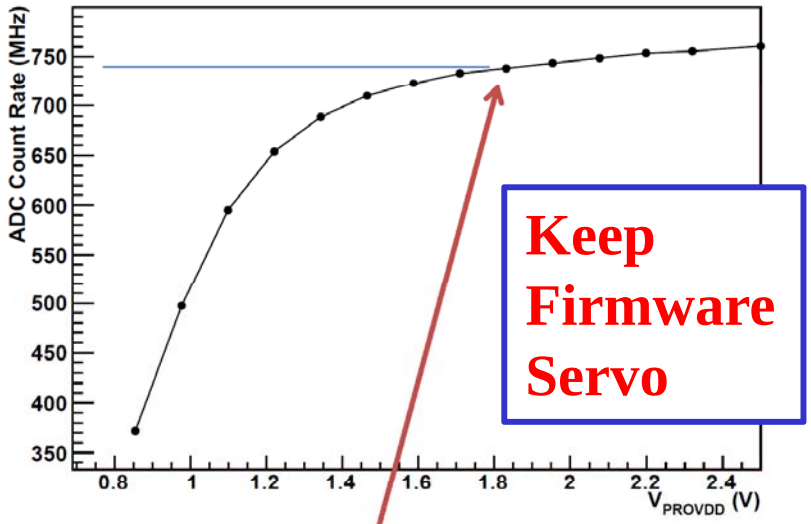


Specifically, for mTC

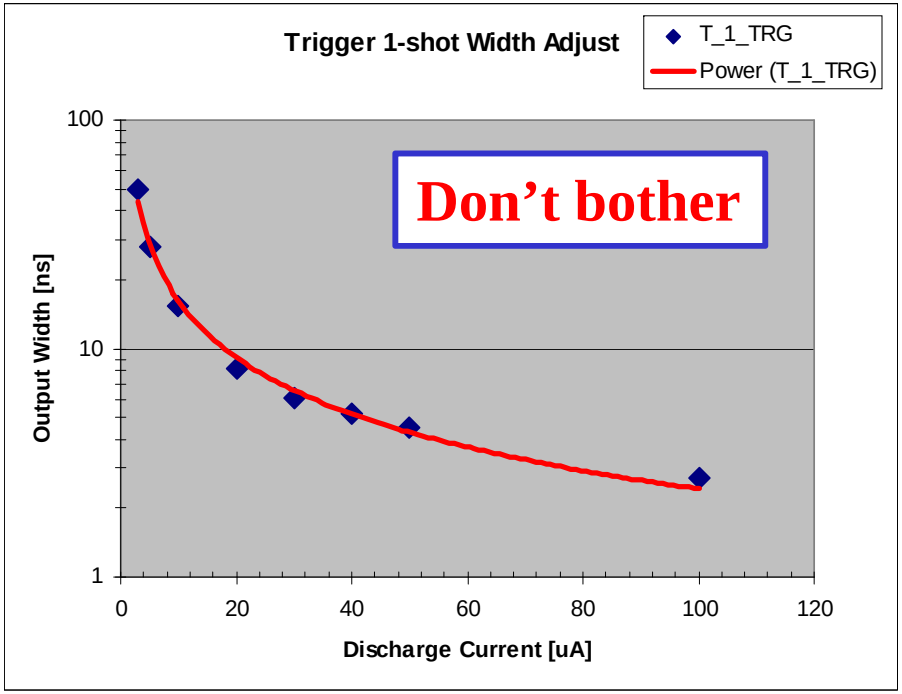
Temperature Dependence







This is the Wilkinson clock adjustment



https://code.google.com/p/idlab-scrod/source/browse/SCROD-boardstack/iTOP/IRS3B_CRT/src/asic_interfaces/wilkinson_monitoring

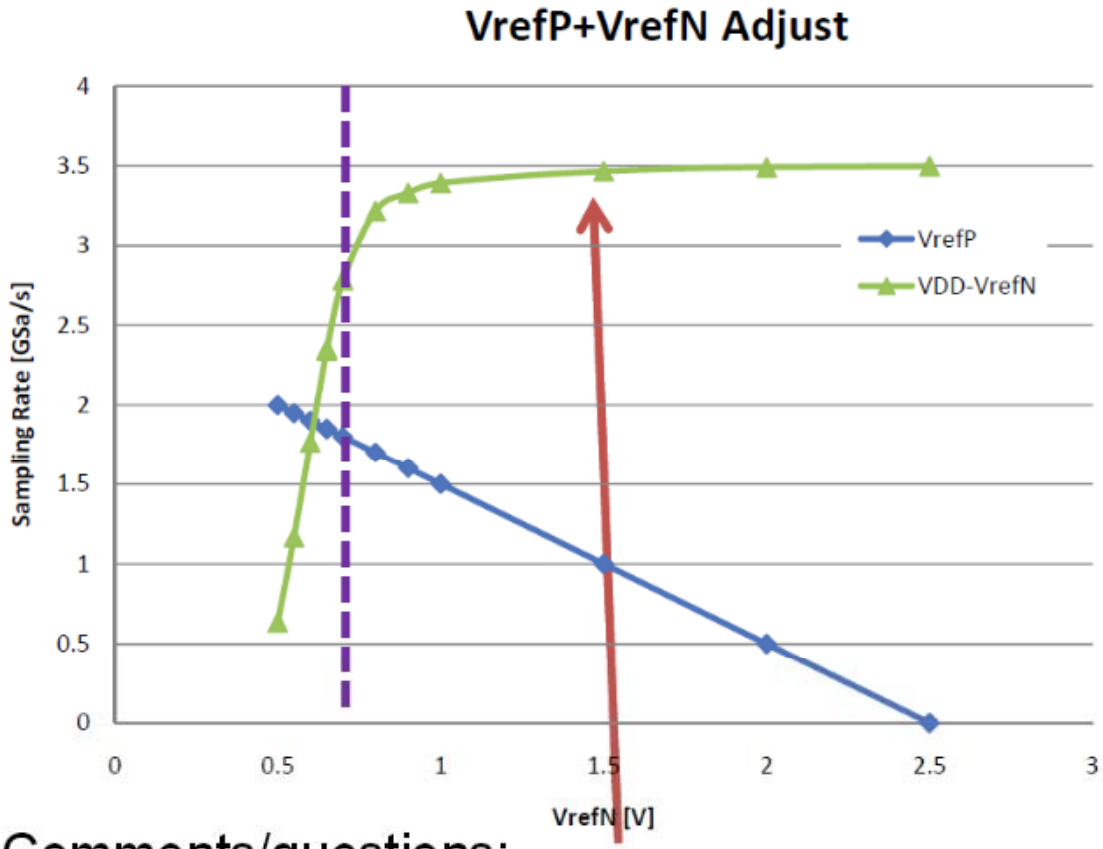
```
--------------------------------------------------------------------------------
-- Single feedback loop (1 ASIC) - more complicated version that moves proportionally
--------------------------------------------------------------------------------
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.NUMERIC_STD.ALL;
use work.asic_definitions_irs3b_carrier_revB.all;
use work.IRS3B_CarrierRevB_DAC_definitions.all;

entity wilkinson_feedback is
	port (
		CLOCK           : in  std_logic;
		CLOCK_ENABLE    : in  std_logic;
		FEEDBACK_ENABLE : in  std_logic;
		CURRENT_VALUE   : in  Counter;
		TARGET_VALUE    : in  Counter;
		DAC_VALUE       : out DAC_Setting;
		STARTING_VALUE  : in  DAC_Setting
	);
end wilkinson_feedback;

architecture Behavioral of wilkinson_feedback is
	signal	internal_DAC_VALUE		: DAC_Setting;
	signal	internal_NEXT_DAC_VALUE		: signed(13 downto 0);
	signal	internal_CURRENT_DIFFERENCE	: signed(16 downto 0);
	signal	internal_STEP			: signed(11 downto 0);
begin
	--Map the signals to the output
	DAC_VALUE <= internal_DAC_VALUE;
	--Calculate the difference between what we have and what we want
	process(CLOCK) begin
		if (rising_edge(CLOCK)) then
			if (CLOCK_ENABLE = '1') then
				internal_CURRENT_DIFFERENCE <= signed('0' & TARGET_VALUE) - signed ('0' & CURRENT_VALUE);
			end if;
		end if;
	end process;
	--The effective proportionality constant is chosen here by truncating the LSBs.
	internal_STEP <= internal_CURRENT_DIFFERENCE(16 downto 5);
	internal_NEXT_DAC_VALUE <= signed(resize(unsigned(internal_DAC_VALUE),internal_NEXT_DAC_VALUE'length)) +
signed(resize(internal_STEP,internal_NEXT_DAC_VALUE'length));
	--Apply the correction
	process(CLOCK) begin
		if (rising_edge(CLOCK)) then
			if (CLOCK_ENABLE = '1') then
				if (FEEDBACK_ENABLE = '0') then
					internal_DAC_VALUE <= STARTING_VALUE;
				else
					if (internal_NEXT_DAC_VALUE < 0) then
						internal_DAC_VALUE <= (others => '0');
					elsif (internal_NEXT_DAC_VALUE > 4095) then
						internal_DAC_VALUE <= (others => '1');
					else
						internal_DAC_VALUE <= std_logic_vector(internal_NEXT_DAC_VALUE(11 downto 0));
					end if;
				end if;
			end if;
		end if;
	end process;
end Behavioral;
```

Firmware Servo-lock

mTC software pointer

A word of caution

Sampling Rate Feedback



We typically operate ~2.7 GSa/s

Slope in this region: ~10GSa/s / V

12-bit DAC can adjust by 0.6 mV per step:
0.6 mV * 10 GSa/s / V = 6 MSa/count

@ 2.7 GSa/s, this is 0.2% per count:
0.2% * 1/2.7 GSa/s = 0.74 ps

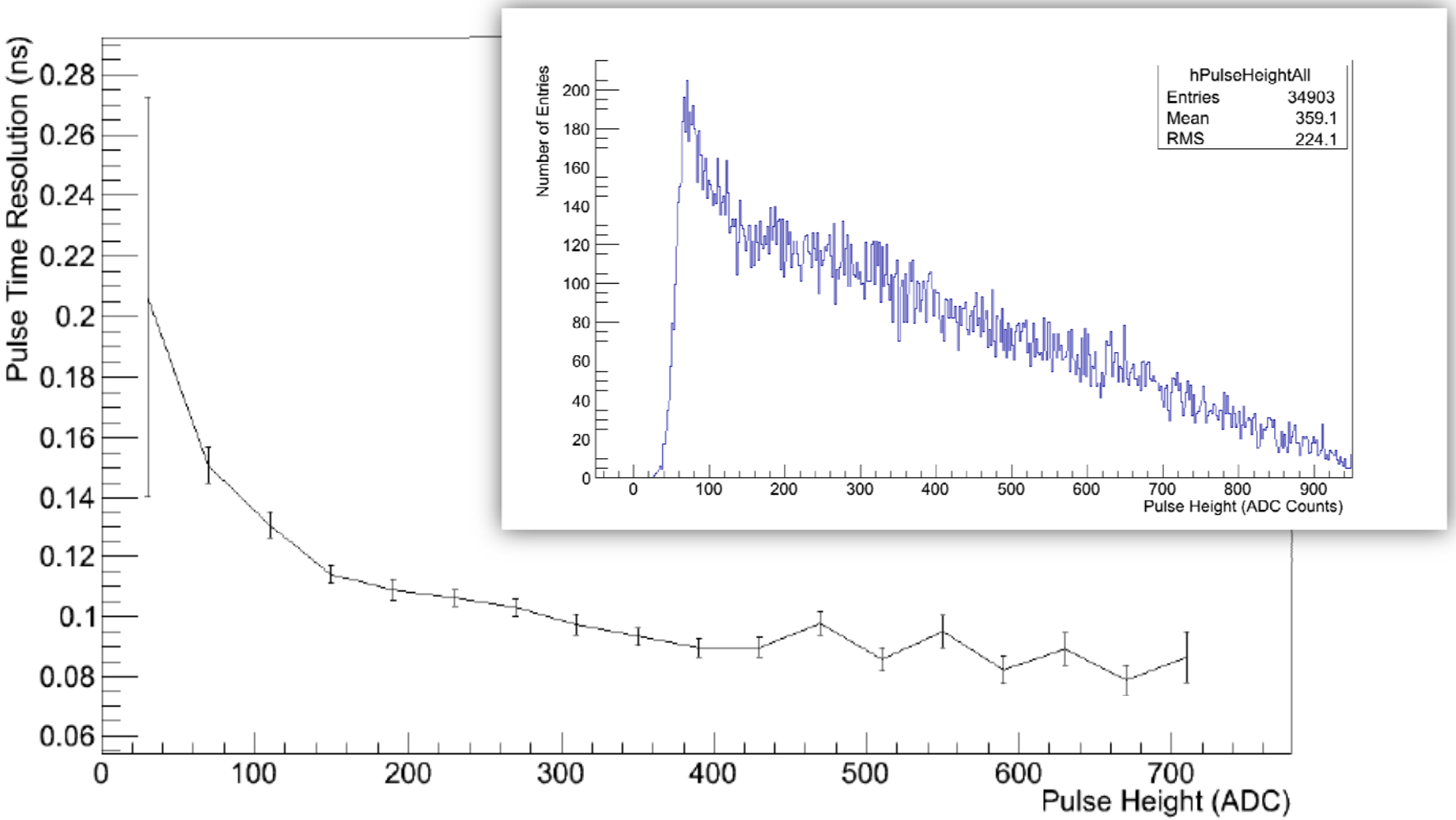
➔Over the full 128 samples, this is ~95 ps of INL.

Comments/questions:

1) 12-bit DAC resolution does not seem sufficient to ever control this effect to < ~50 ps.
   - Higher resolution DAC could help, but only if noise is reduced to the level where the LSBs actually matter.
2) Noise on the two control lines contributes to sampling rate uncertainties. If so, 1 mV of noise would roughly correspond to 150 ps over the full range.

What limits timing?

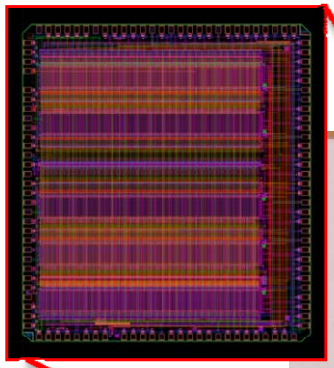
Increased Amplification:

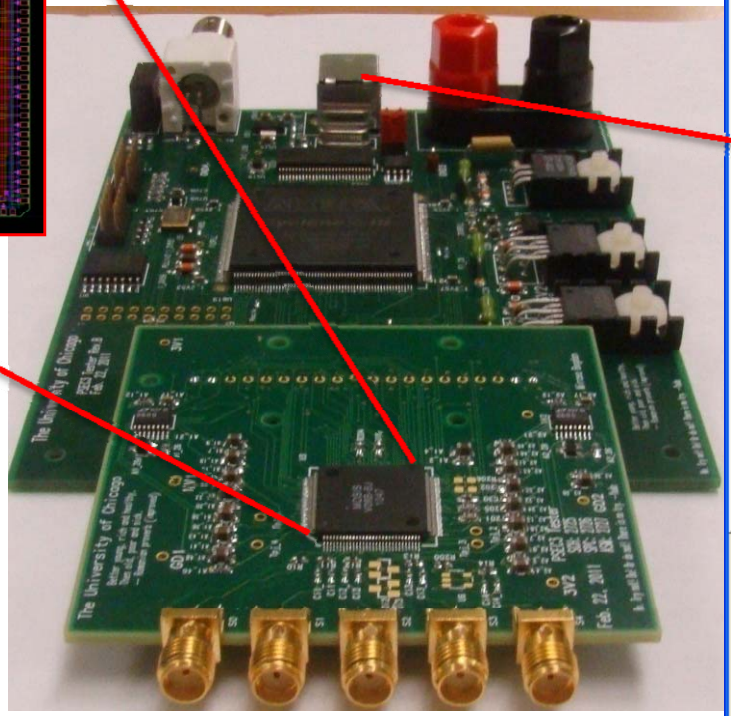


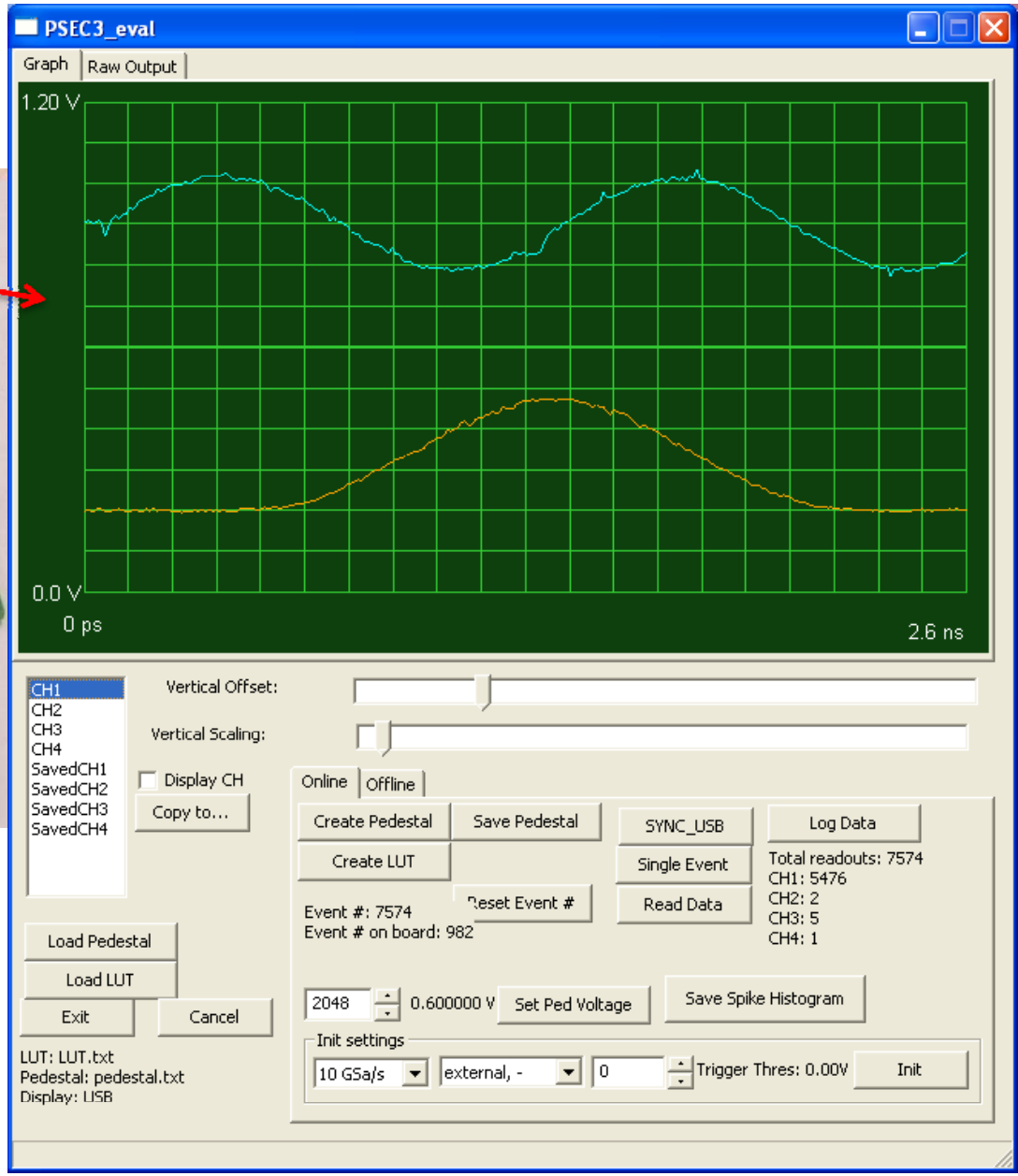
Want >100 ADC counts (>~ 60mV) for smallest pulses:

Carrier Rev C 90-100ps ➔ 60-70ps

Oscilloscope on a chip? ➔ Calibration



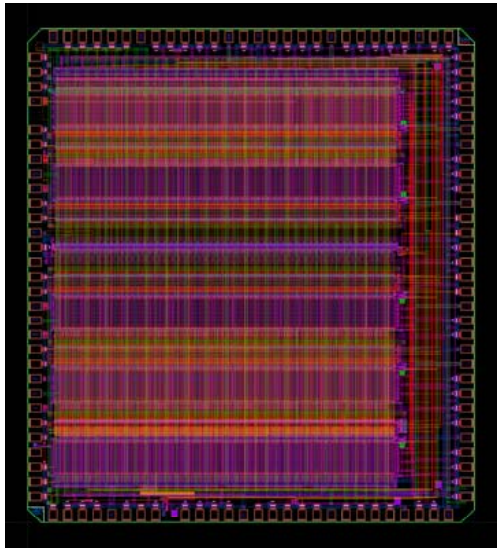




~= ??

Oscilloscope on a chip? ➔ Calibration

- Modified approximation:



+ 

+ 

+ 

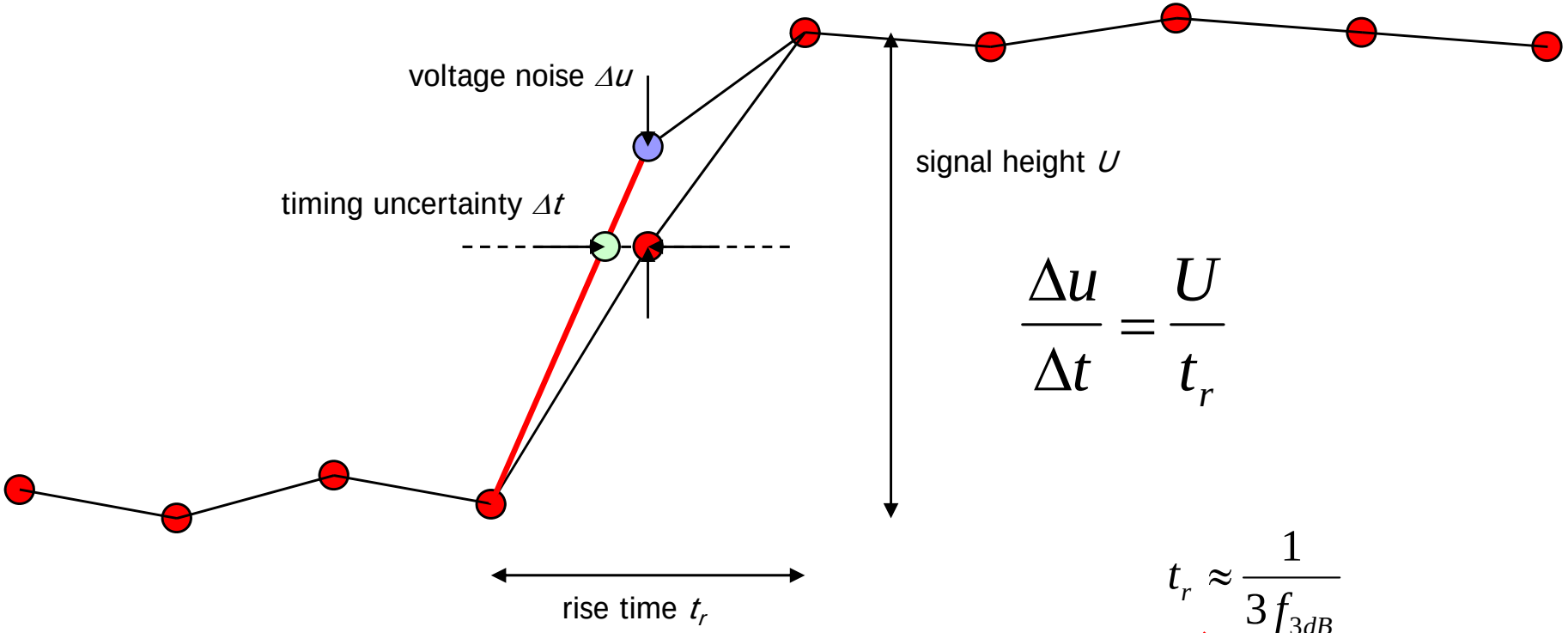
+ 

+ 

+ 

+ ... ~ =

Calibration and Sources of Timing Error



$$\frac{\Delta u}{\Delta t} = \frac{U}{t_r}$$

$$t_r \approx \frac{1}{3 f_{3dB}}$$

$$\Delta t = \frac{\Delta u}{U} \cdot t_r = \frac{\Delta u}{U\sqrt{n}} \cdot t_r = \frac{\Delta u}{U} \cdot \frac{t_r}{\sqrt{t_r \cdot f_s}} = \frac{\Delta u}{U} \cdot \frac{\sqrt{t_r}}{\sqrt{f_s}} = \frac{\Delta u}{U} \cdot \frac{1}{\sqrt{3 f_s \cdot f_{3dB}}}$$

*Diagram, formulas from Stefan Ritt

Calibration and Sources of Timing Error

Contributions to timing resolution:

- Voltage uncertainties
- Timing uncertainties

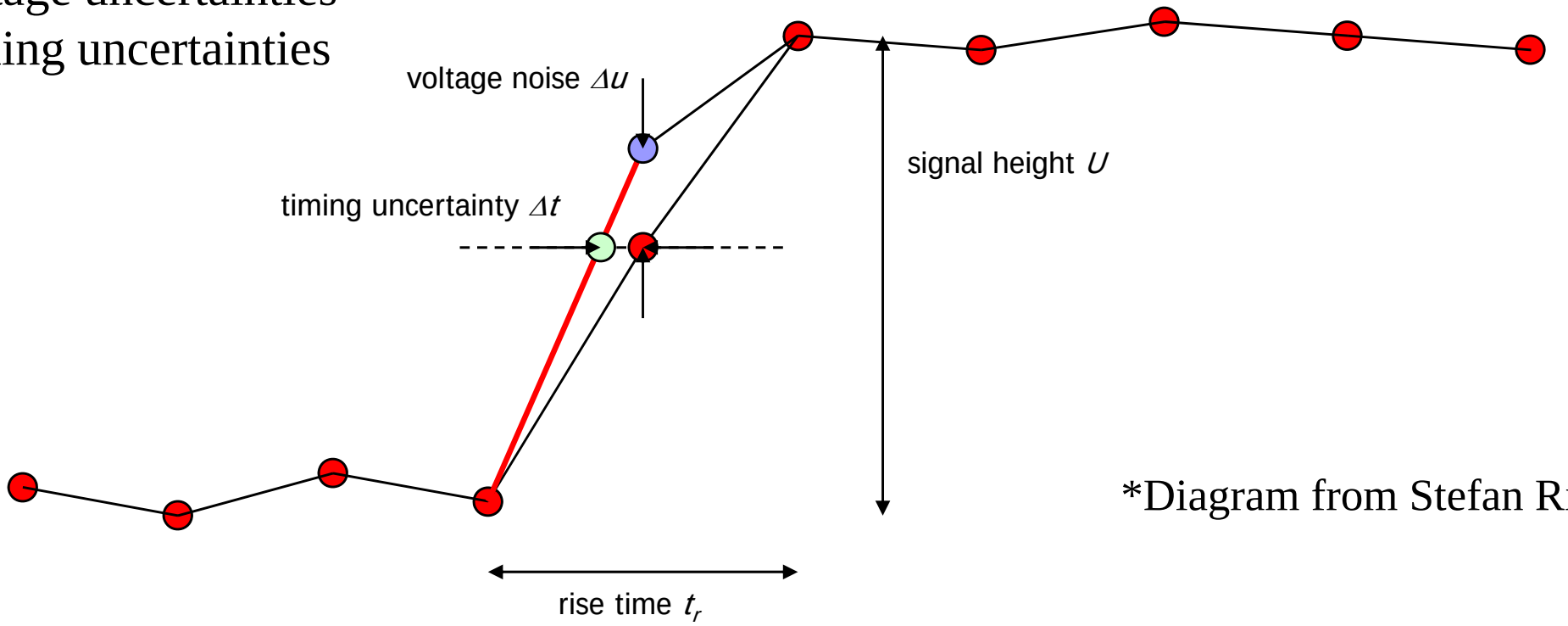


*Diagram from Stefan Ritt

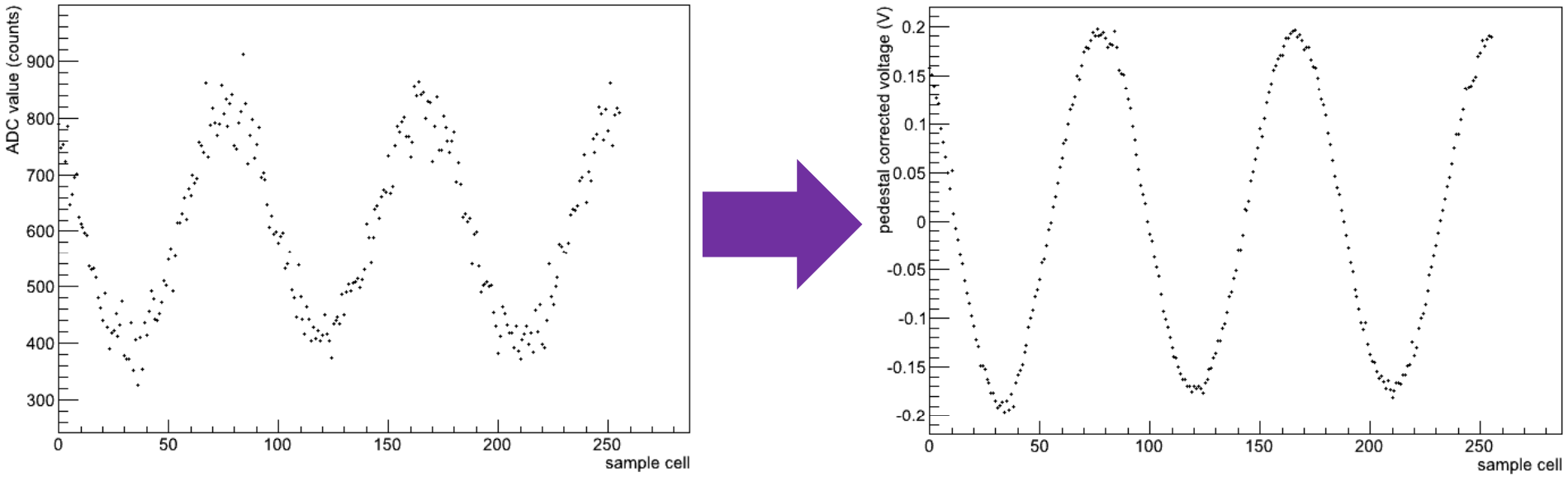
Of these contributions:

- Random – irreducible (without hardware redesign)
- Deterministic – **in principle** can be calibrated away.

Let's talk about where the deterministic pieces come from and what has been done about them.

First ASIC Calibration – Pedestals

- Each storage cell has its own “offset” value.
  - Measure with no signal input.
  - These offsets must be removed in order to see a clean(er) signal.
  - We call this “pedestal subtraction.”
  - **32k pedestals per channel (quarter million per ASIC!)**
- Example:

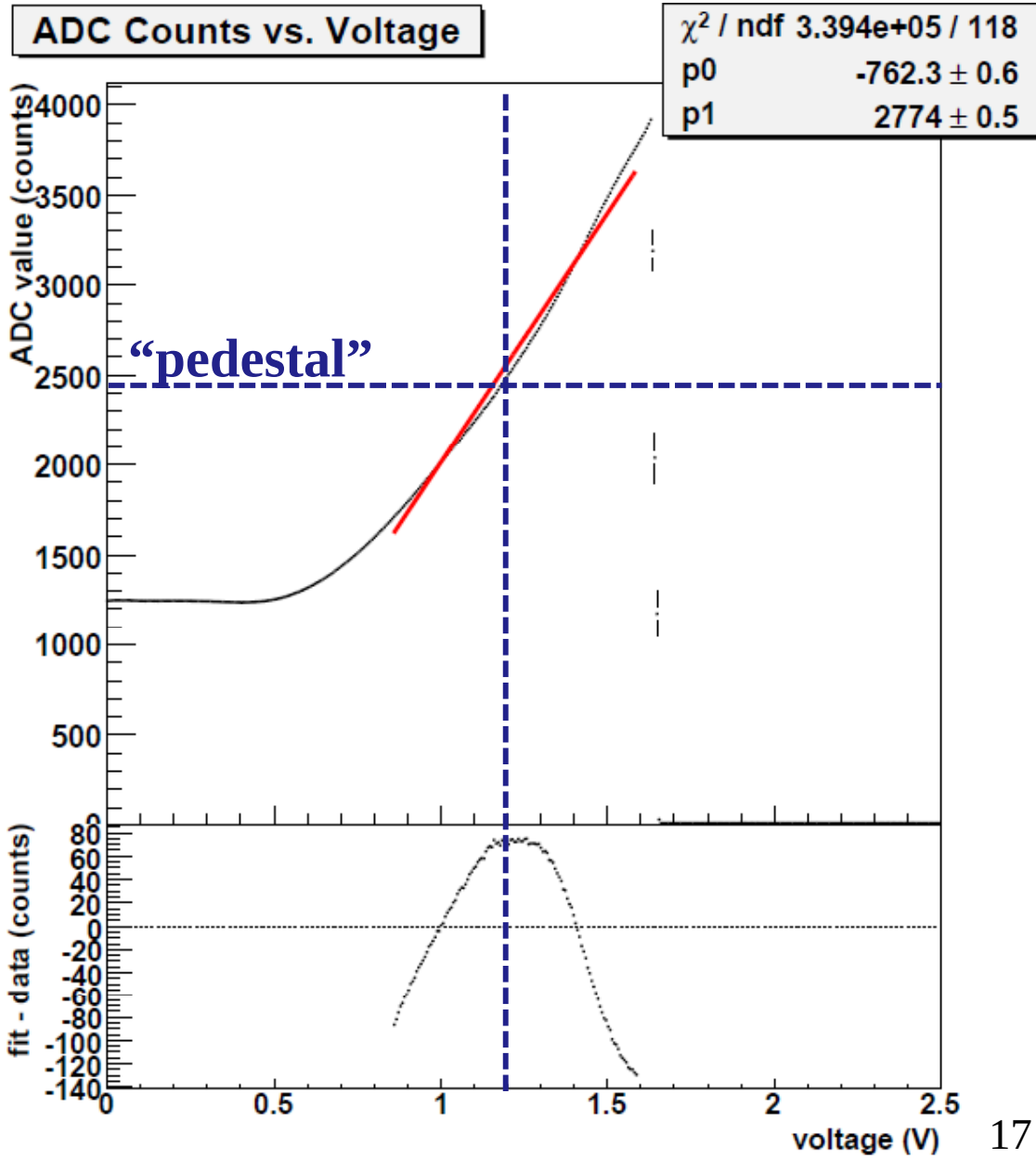


Before pedestal subtraction

After pedestal subtraction

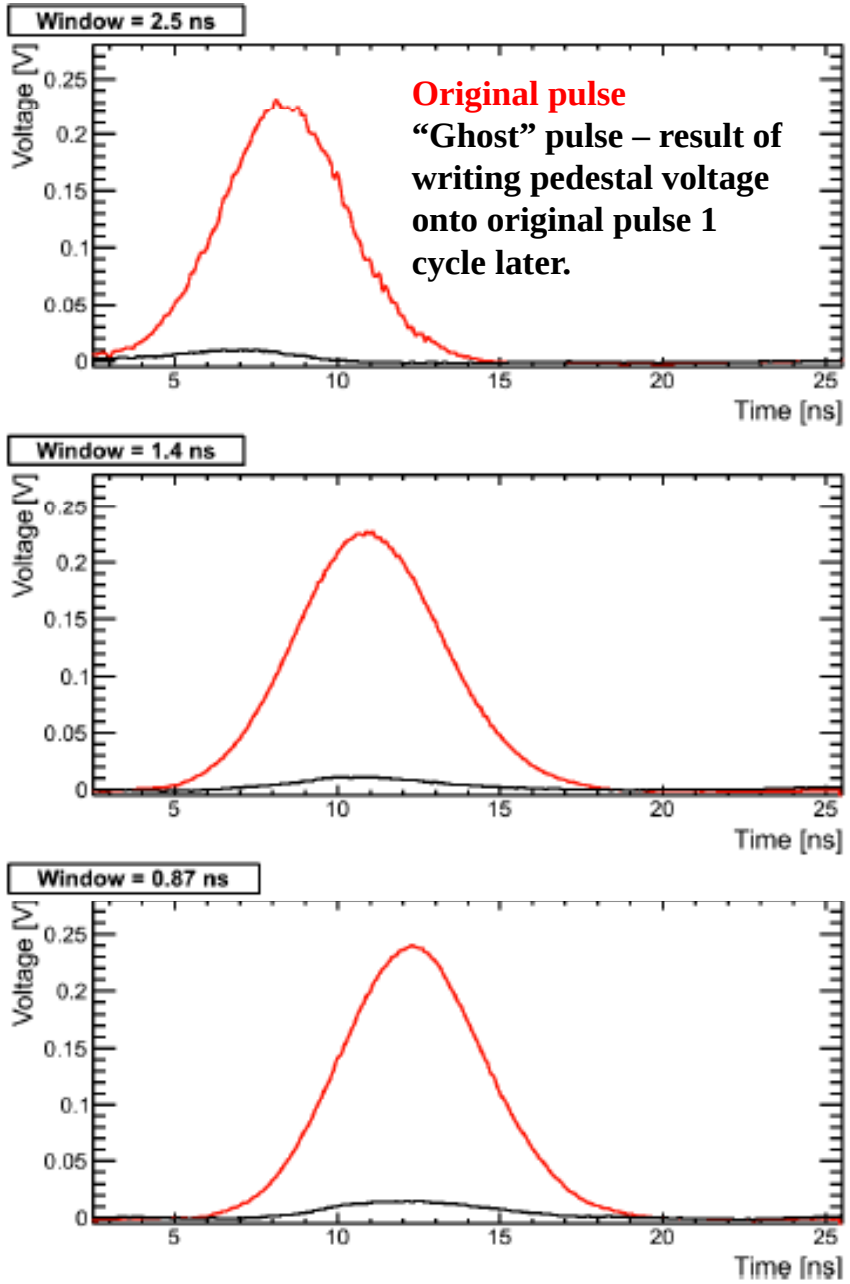
Comparator Transfer Functions

- Wilkinson comparator.
- Ramp is supplied to all storage cells.
- Comparator output fires when ramp exceeds stored voltage.
- Signals are stored with DC offset to fit into the comparator's dynamic range.
  - Offset varies somewhat for each storage cell. This is what we try to remove with pedestal correction.
- Comparator response is nonlinear.
- Example shown for IRS3B comparator



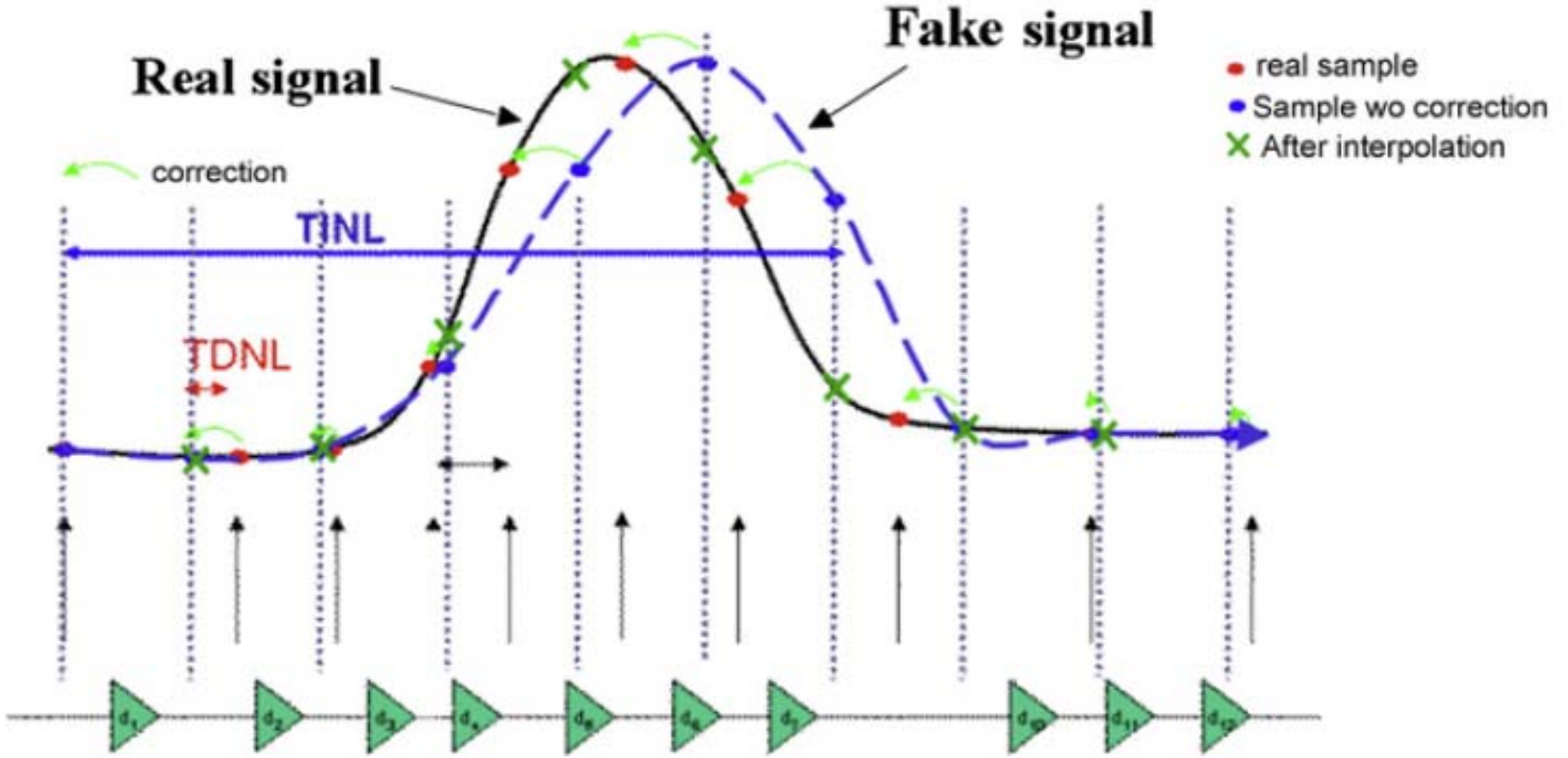
AC vs. DC Response, Pulse Persistence

- Previous slide transfer functions measured with DC inputs.
- AC response may not be the same!
- Why not? One example… persistence.
  - Voltage has some dependence on previously stored voltage.
  - Example from Eric Oberla, PSEC3 ASIC.
    - This shows a pulse whose "ghost" persists for one or more cycles after the pulse.
    - The inverse is almost certainly true: a pulse does not reach its full height due to



Timing Uncertainties and Timing Calibration

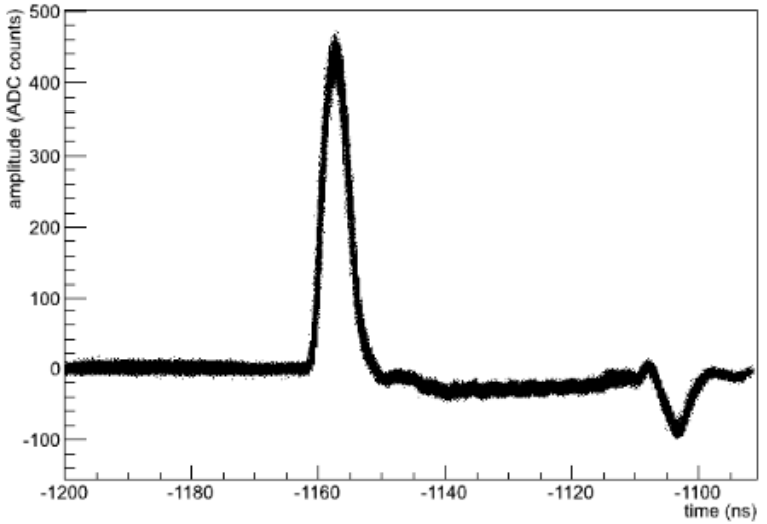
- Time interval between delay line stages has intrinsic variation.
- Not accounting for this properly causes significant timing errors
- Differential (DNL) and Integral (INL) [run-out] Non-Linearity

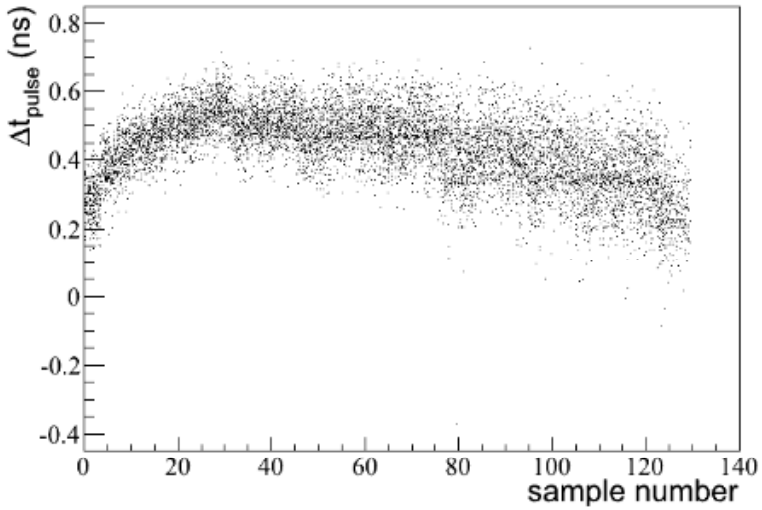


Nuclear Instruments and Methods in Physics Research A 629 (2011) 123–132

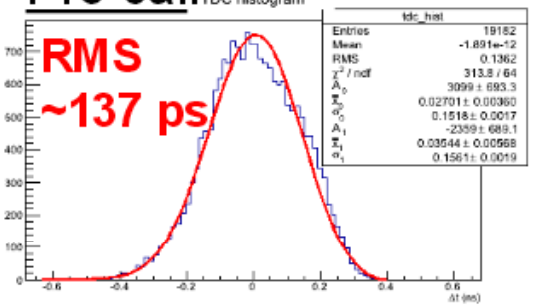
One calibration scheme

- Inject fixed amplitude pulses at a constant time relative to system trigger.
- Measure times using a simple fixed threshold analysis.
  - Simple gain correction applied sample-by-sample.
  - Two point linear fit at threshold crossing.
  - Trying to add information from other samples complicates the procedure, since it includes contributions from many
- Measured time should be a constant value regardless of where the pulse was captured in the sampling array, but we see significant structure.

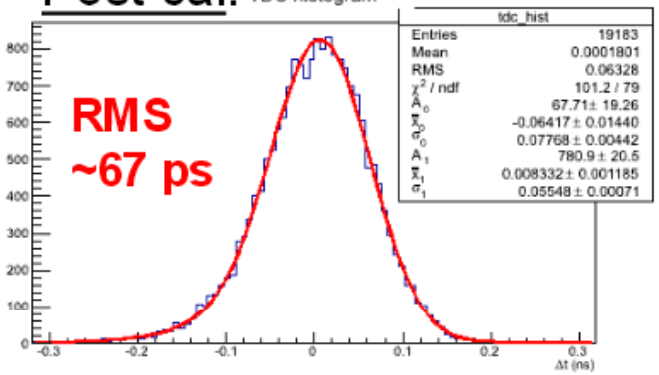




Pre-cal: TDC histogram



Post-cal: TDC histogram

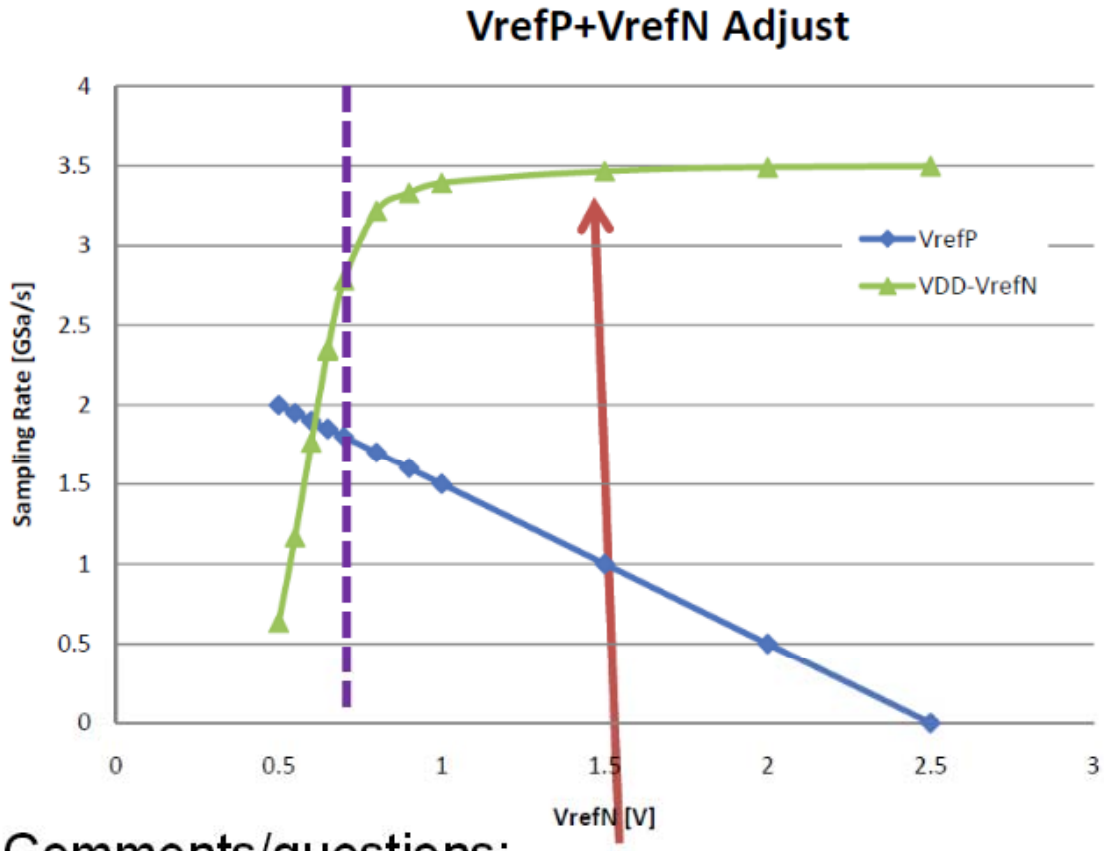


- Have tried many, and they each have their merits and drawbacks

A word of caution (anon)

Sampling Rate Feedback

SLAC



We typically operate ~2.7 GSa/s

Slope in this region: ~10GSa/s / V

12-bit DAC can adjust by 0.6 mV per step:
0.6 mV * 10 GSa/s / V = 6 MSa/count

@ 2.7 GSa/s, this is 0.2% per count:
0.2% * 1/2.7 GSa/s = 0.74 ps

➔Over the full 128 samples, this is ~95 ps of INL.

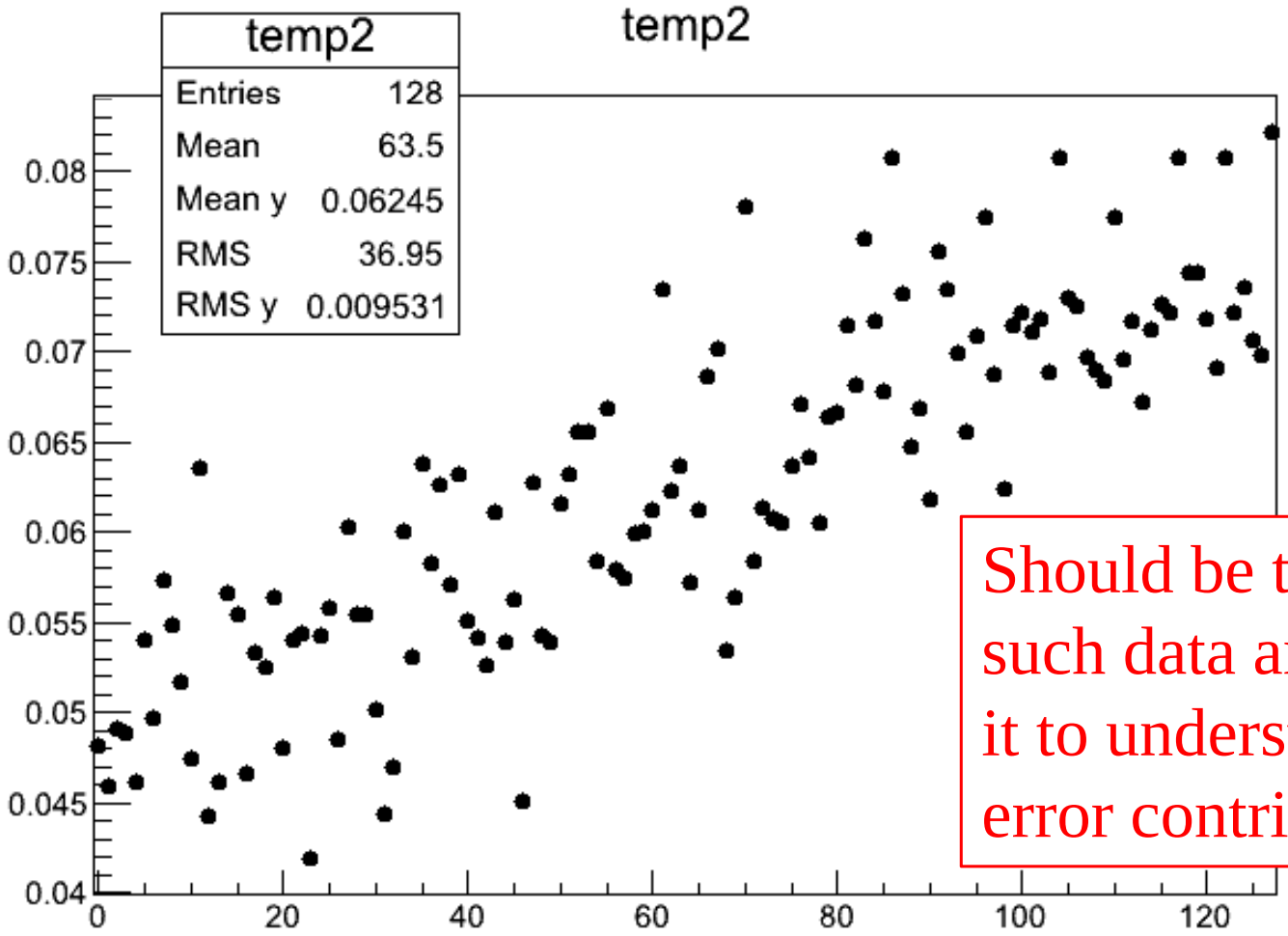
Comments/questions:

1) 12-bit DAC resolution does not seem sufficient to ever control this effect to < ~50 ps.
   - Higher resolution DAC could help, but only if noise is reduced to the level where the LSBs actually matter.
2) Noise on the two control lines contributes to sampling rate uncertainties. If so, 1 mV of noise would roughly correspond to 150 ps over the full range.

Timing Resolution as a Function of Sample Number

75ps

50ps



Should be taking a lot of such data and analyzing it to understand timing error contributions

- **Timing resolution as a function of sample number of the threshold crossing (pulser data).**
- **Indicative of noise contributions.**

IRS3D Improvements over IRS3B

1. Improved Trigger Sensitivity
2. Timebase Servo-locking
3. dT hardware adjust
4. Improved linearity/dynamic range
5. Improved Wilkinson ADC
6. No high current at power-on

= originally reported for TARGET7/X

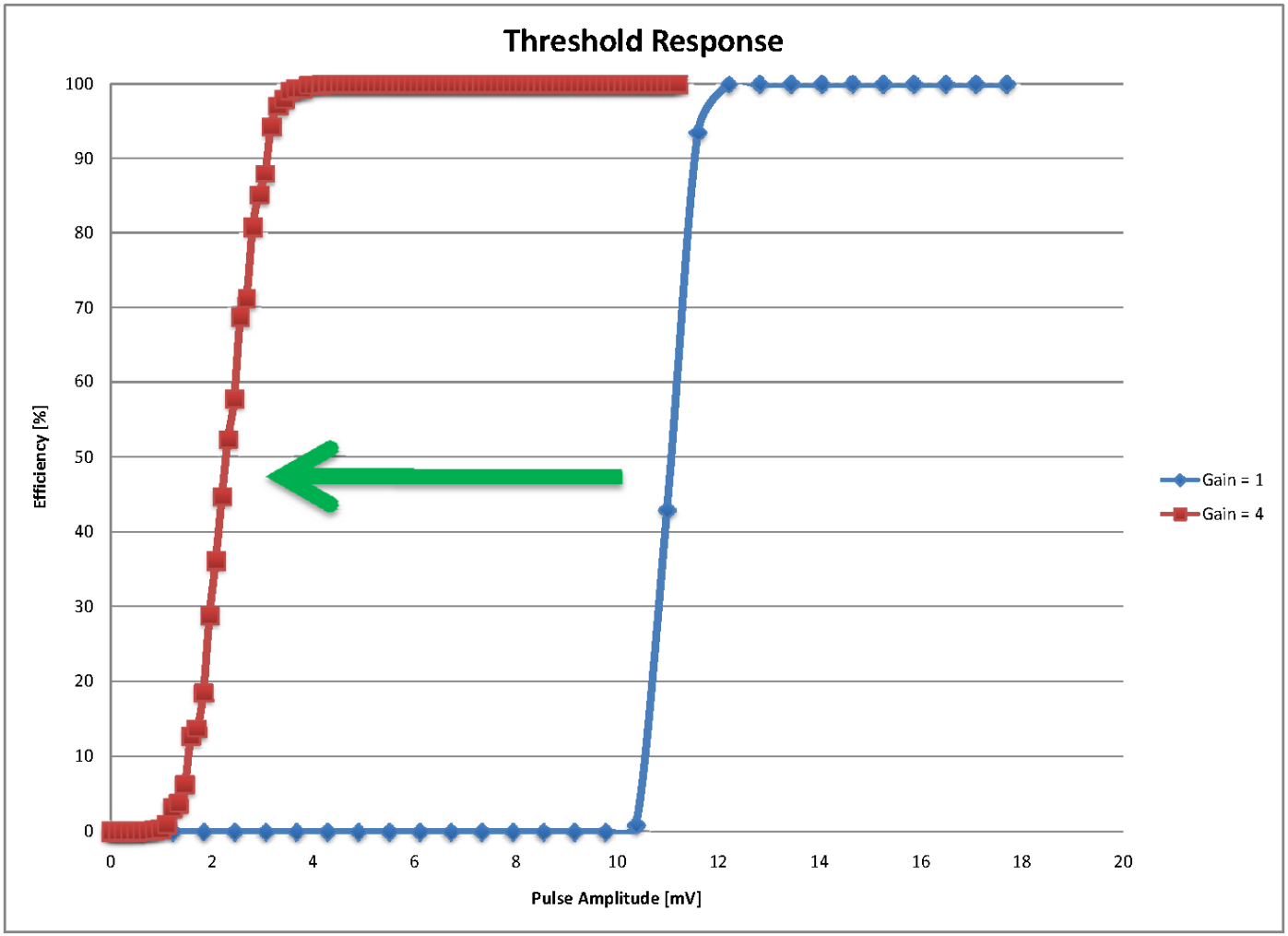
= demonstrated initially (TARGET7/X), detailed timing confirmed

= LABRADOR4 independent confirmation

= All ASICs since IRS3C

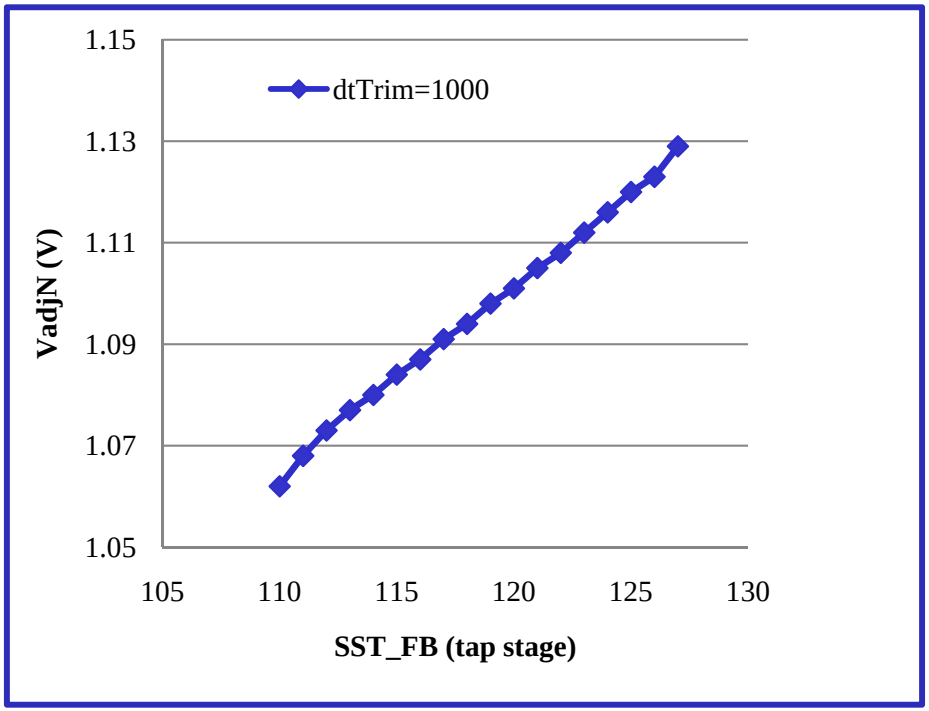
Trigger Threshold Improvement

- A significant improvement for smaller pulses where “first strike” initiation of the MCP charge development is retarded

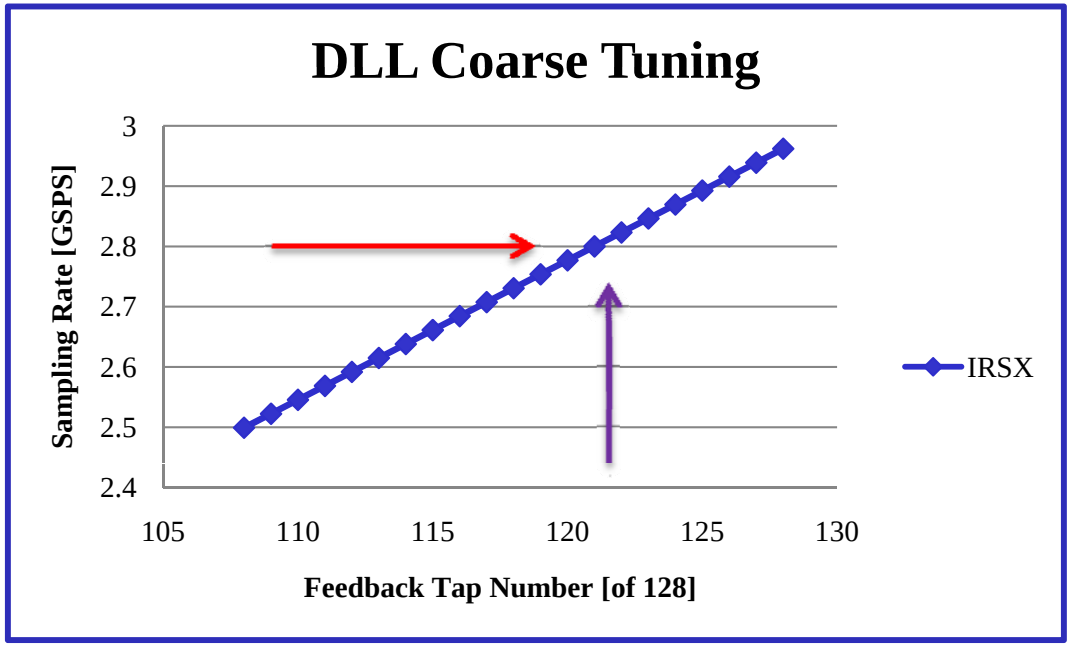


- 16x doesn’t improve further, as already at the signal-to-noise limit

Timebase servo-locking (DLL)



Sampling tracks target delay



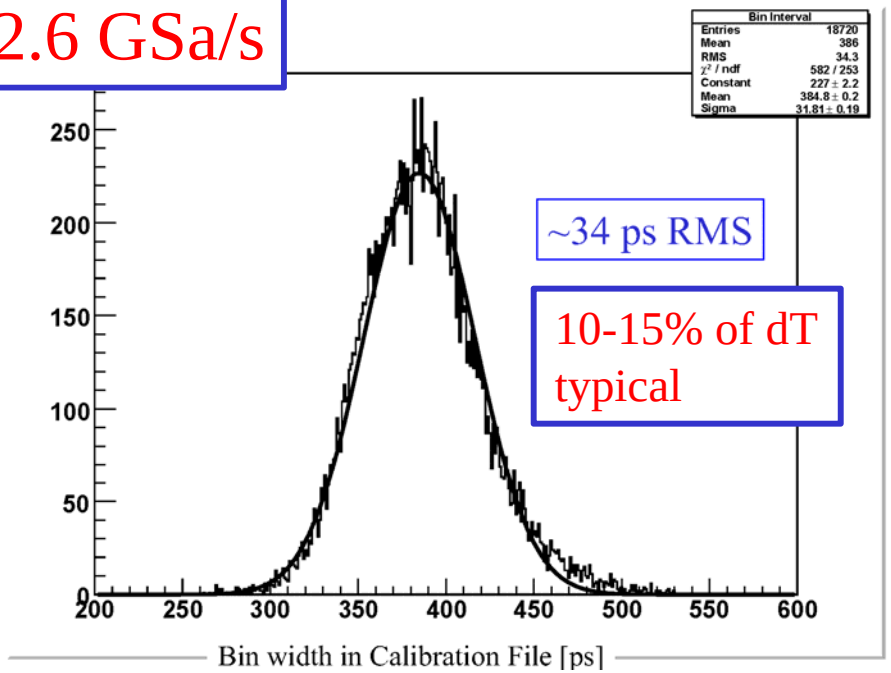
Target sampling rate: **2.8 GSa/s**

Feedback tap = 121

(indirect "RCO" feedback mechanism injects asynchronous noise into timebase generator, degrading timing performance – so this is a significant improvement)

Time base non-uniformity…

2.6 GSa/s

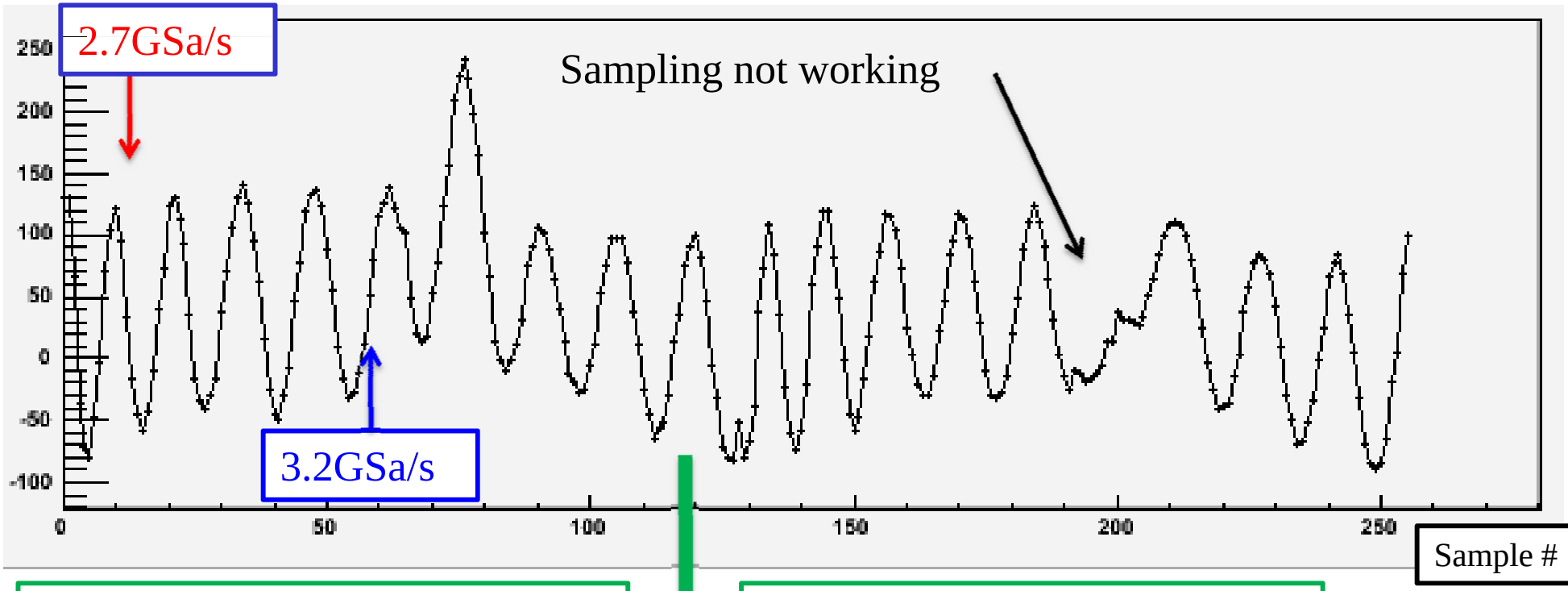


~34 ps RMS

10-15% of dT typical

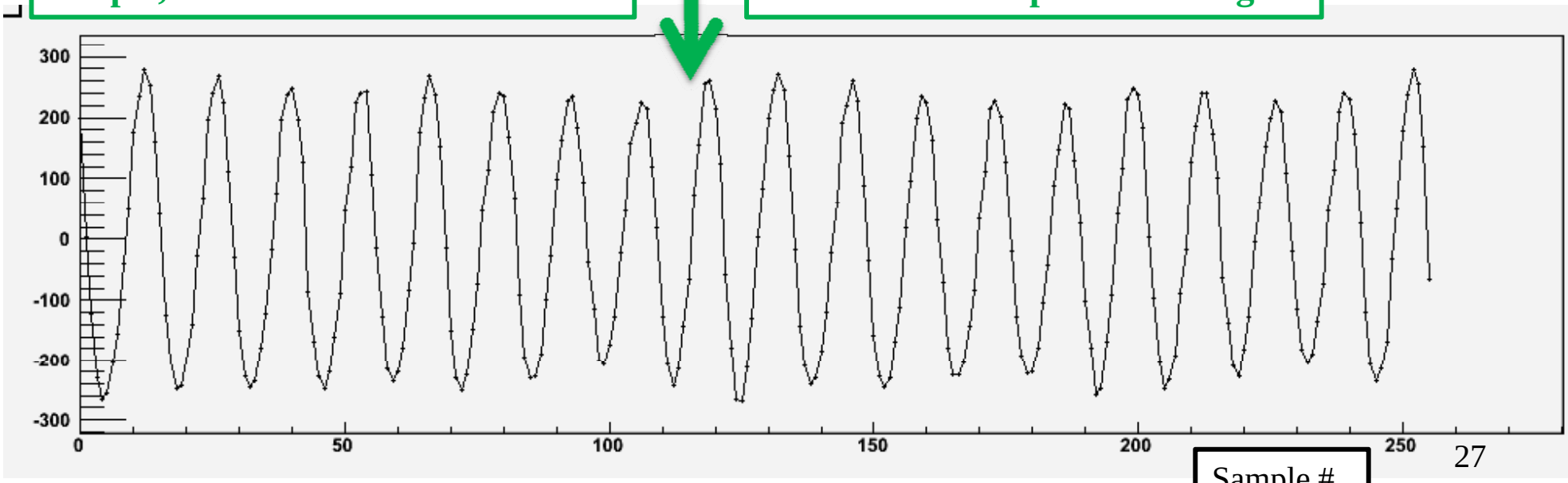
If can correct, reduces processing time dramatically, as this is the most computationally-intensive aspect of “fast feature extraction”

Roughly Adjusted dT Sampling

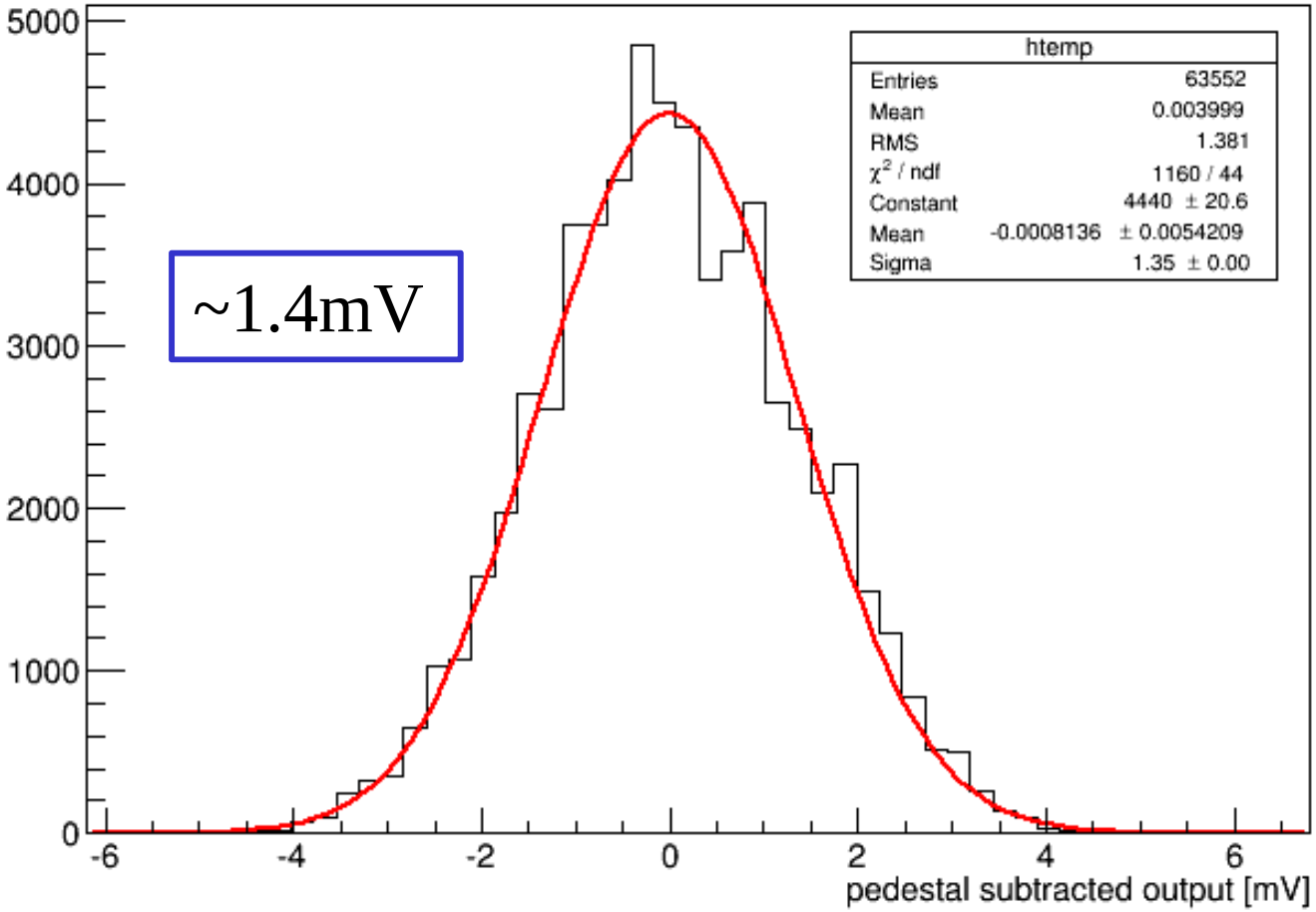


Simple, linear dT slew correction

Still room for improved tuning



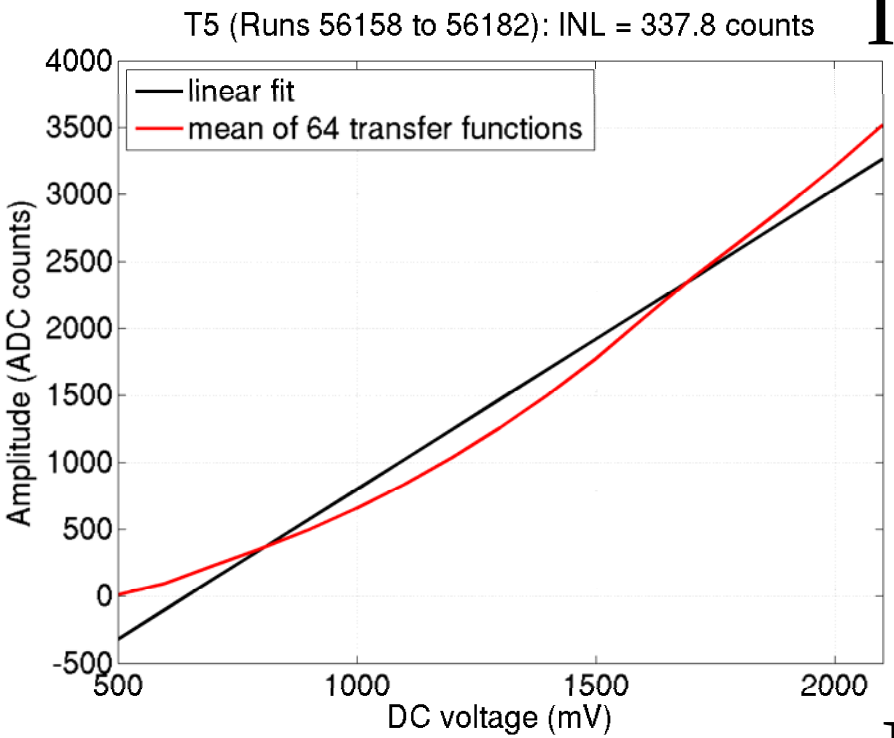
Observed IRSX noise

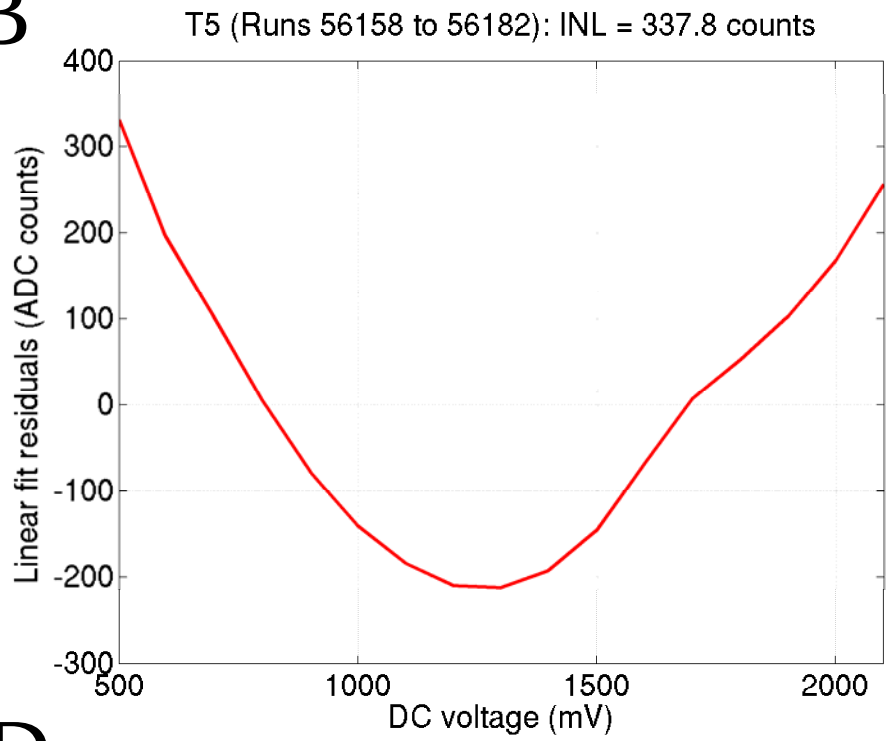


Non-gaussian distributions expected for small noise amplitude due to non-linearity in Gray-code least count

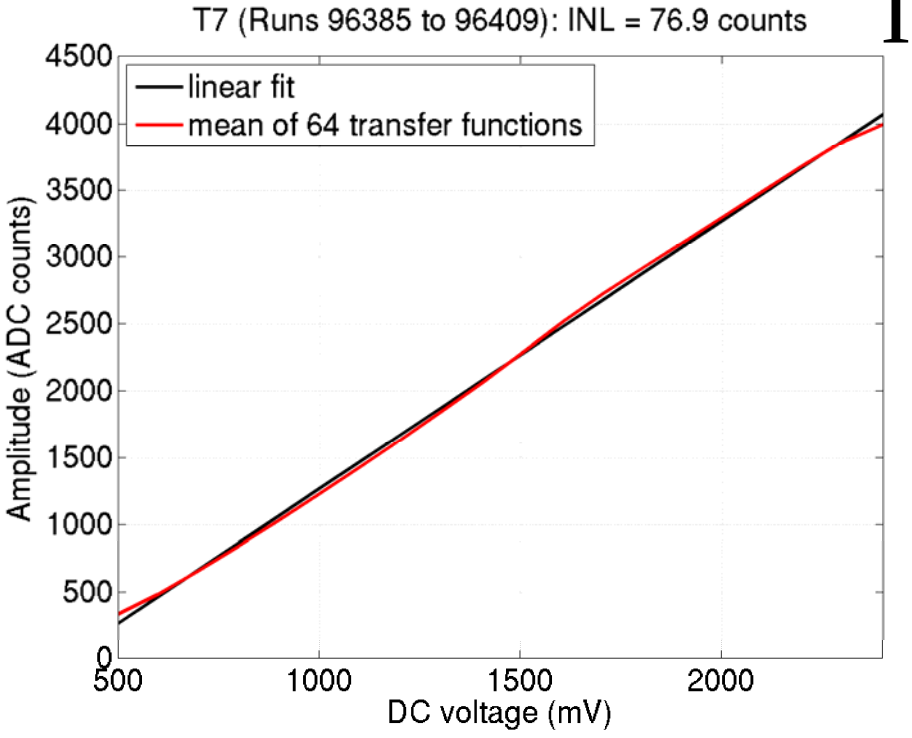
Take away message: noise is comparable, or better than IRS3B, and acquired while sampling continues to run

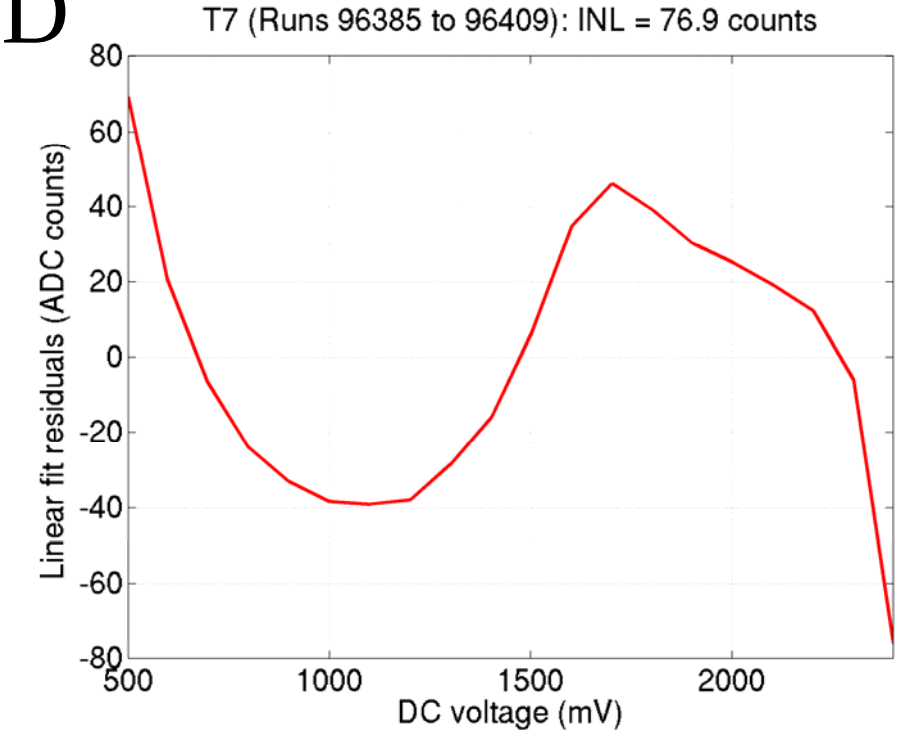
IRS3B


T5 (Runs 56158 to 56182): INL = 337.8 counts
linear fit
mean of 64 transfer functions
Amplitude (ADC counts)
DC voltage (mV)


T5 (Runs 56158 to 56182): INL = 337.8 counts
Linear fit residuals (ADC counts)
DC voltage (mV)

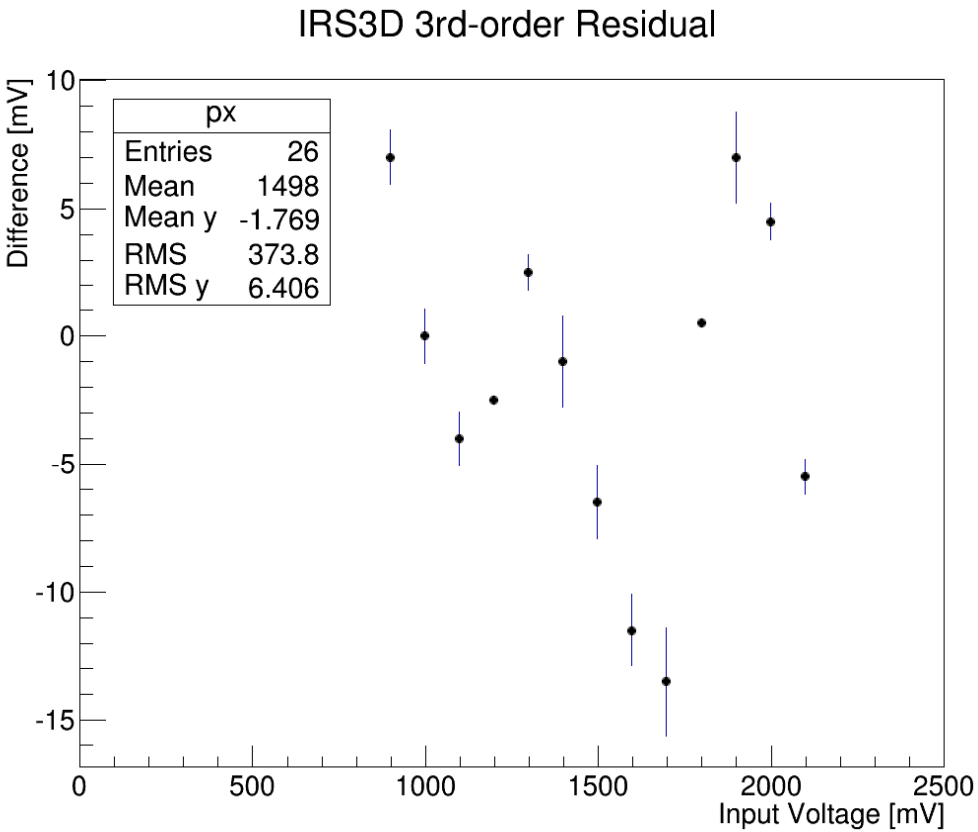
IRS3D

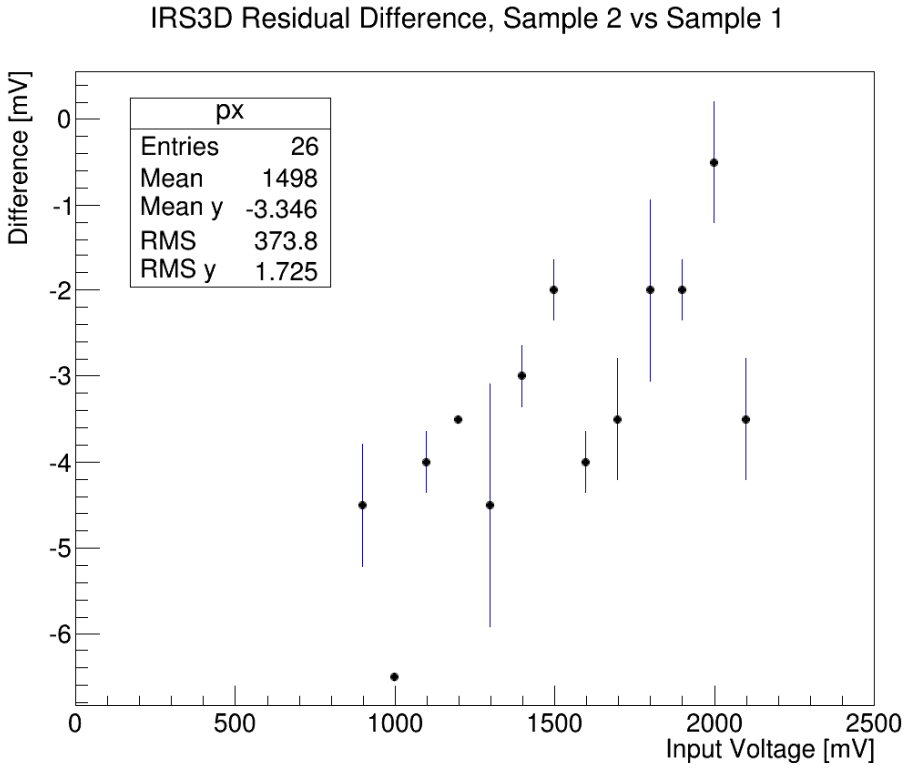

T7 (Runs 96385 to 96409): INL = 76.9 counts
linear fit
mean of 64 transfer functions
Amplitude (ADC counts)
DC voltage (mV)


T7 (Runs 96385 to 96409): INL = 76.9 counts
Linear fit residuals (ADC counts)
DC voltage (mV)

Improved Residuals, repeatability

Note: IRS3D -- no comparator bias tuning yet done



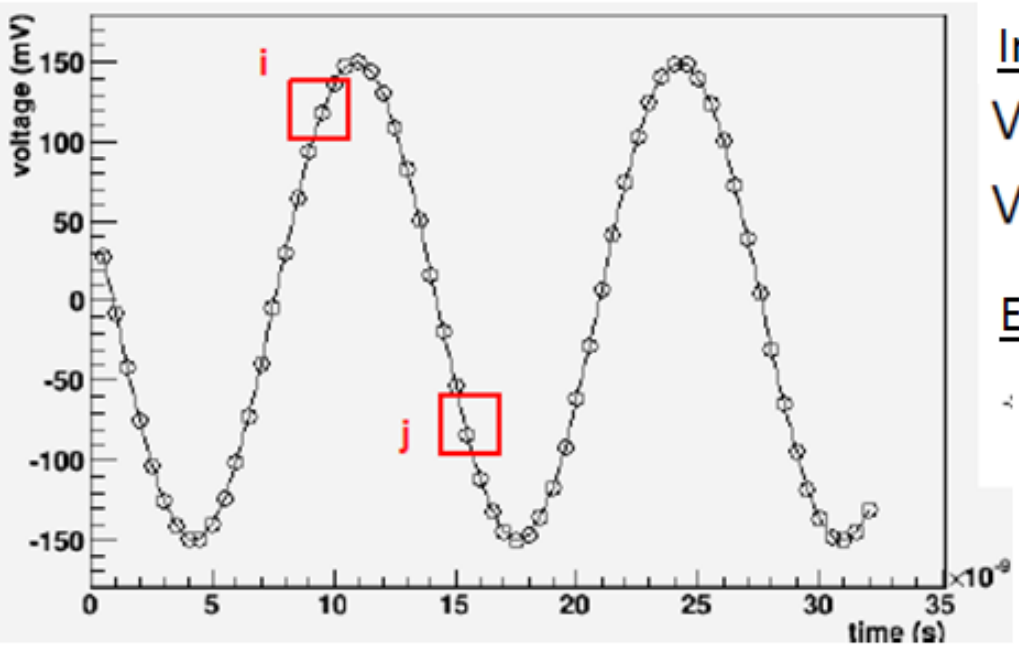


~1% Integral deviation from 3rd-order over key sensitivity range

Shape repeatable sample-sample (common lookup table, with only pedestal offset)

Useful Diagnostic tool

- Plot correlations between pairs of samples:
  - To determine Δt_{ij}, plot $V_i - V_j$ versus $V_i + V_j$



Input signals given by:

$V_i = A \sin(\omega t_i + \phi)$

$V_j = A \sin(\omega t_j + \phi)$

Effectively rotate by 45±:

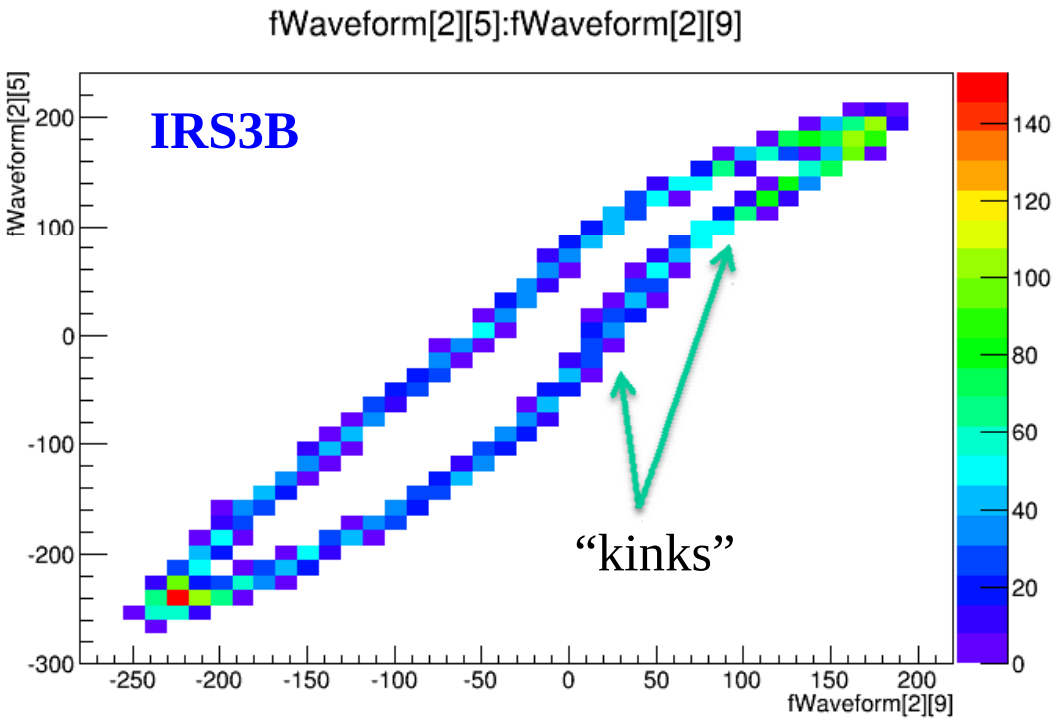
- $x := V_i + V_j$
- $y := V_i - V_j$

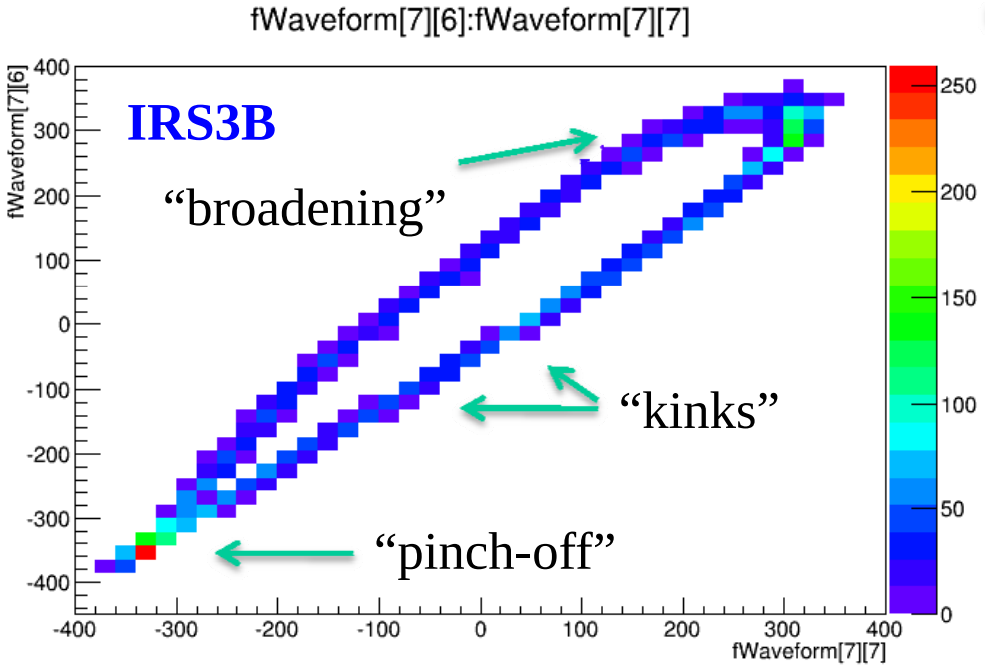
➔ $$\frac{x^2}{4A^2\cos^2(\omega\delta t/2)} + \frac{y^2}{4A^2\sin^2(\omega\delta t/2)} = 1$$

- **i and j can be adjacent (or not), but cycle ambiguities exist if > 1 period apart.**

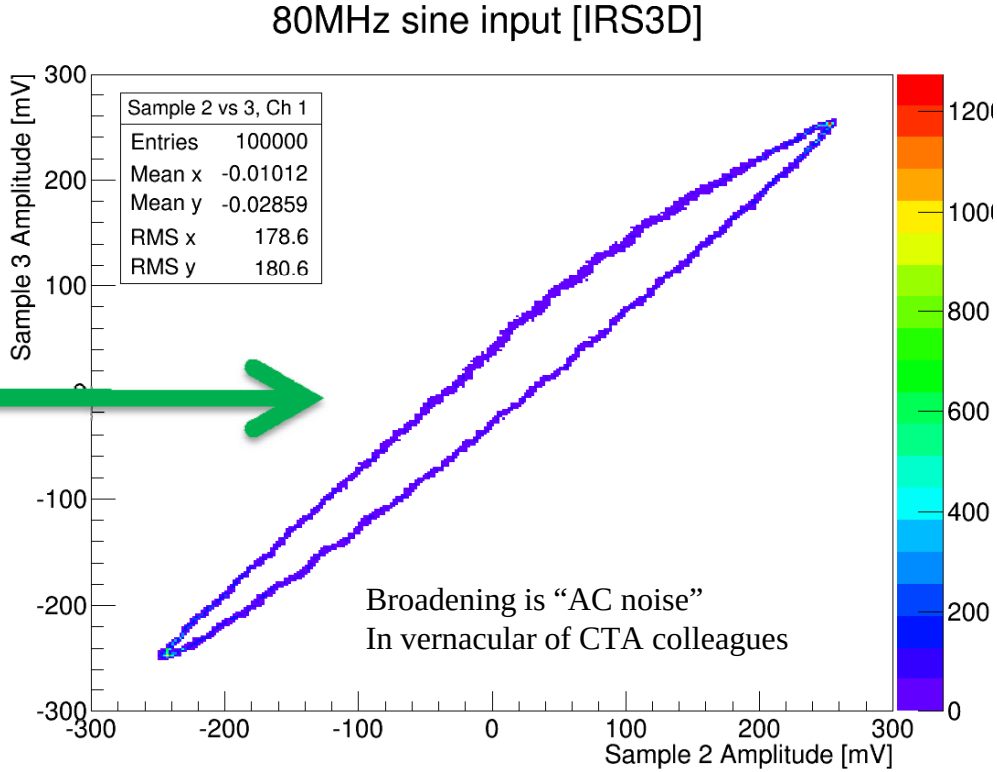
Quite sensitive to yuckiness in the data

Improved Wilkinson “cross-feed”



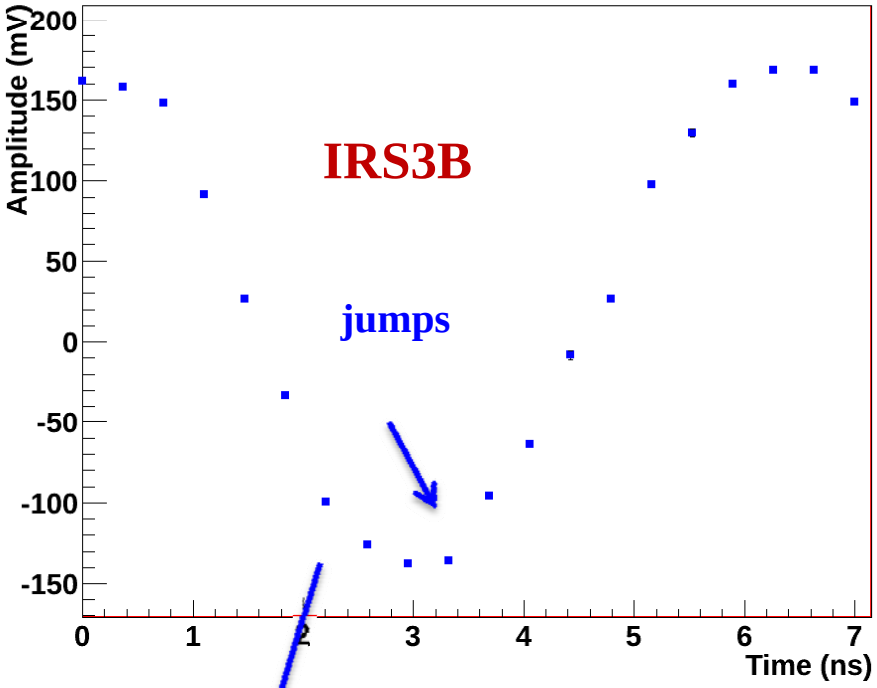


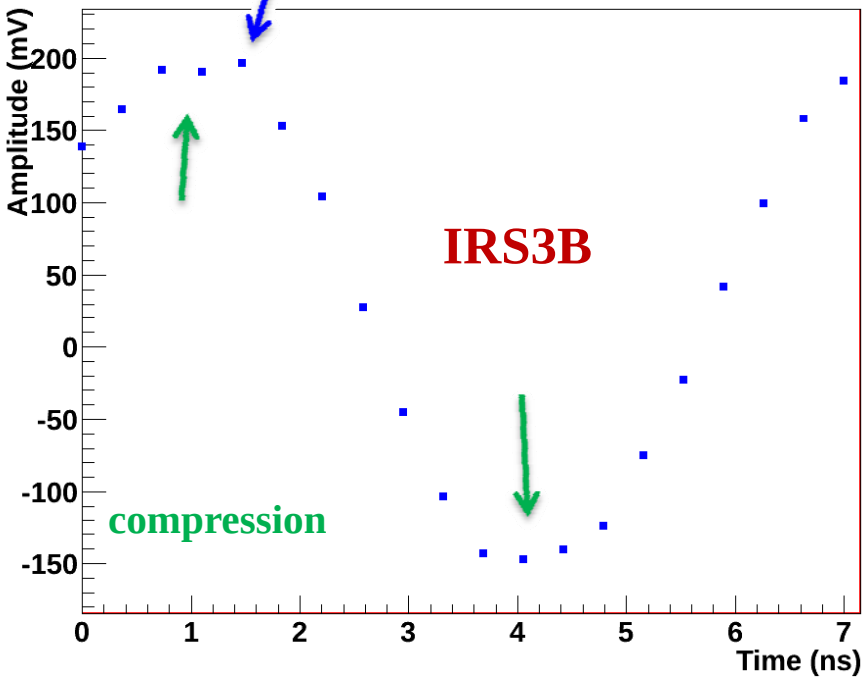
- Broadening at the extrema
- Breakdown of simple ellipses expect otherwise
- Kinks/inflections hard to manufacture without some type of digital interference



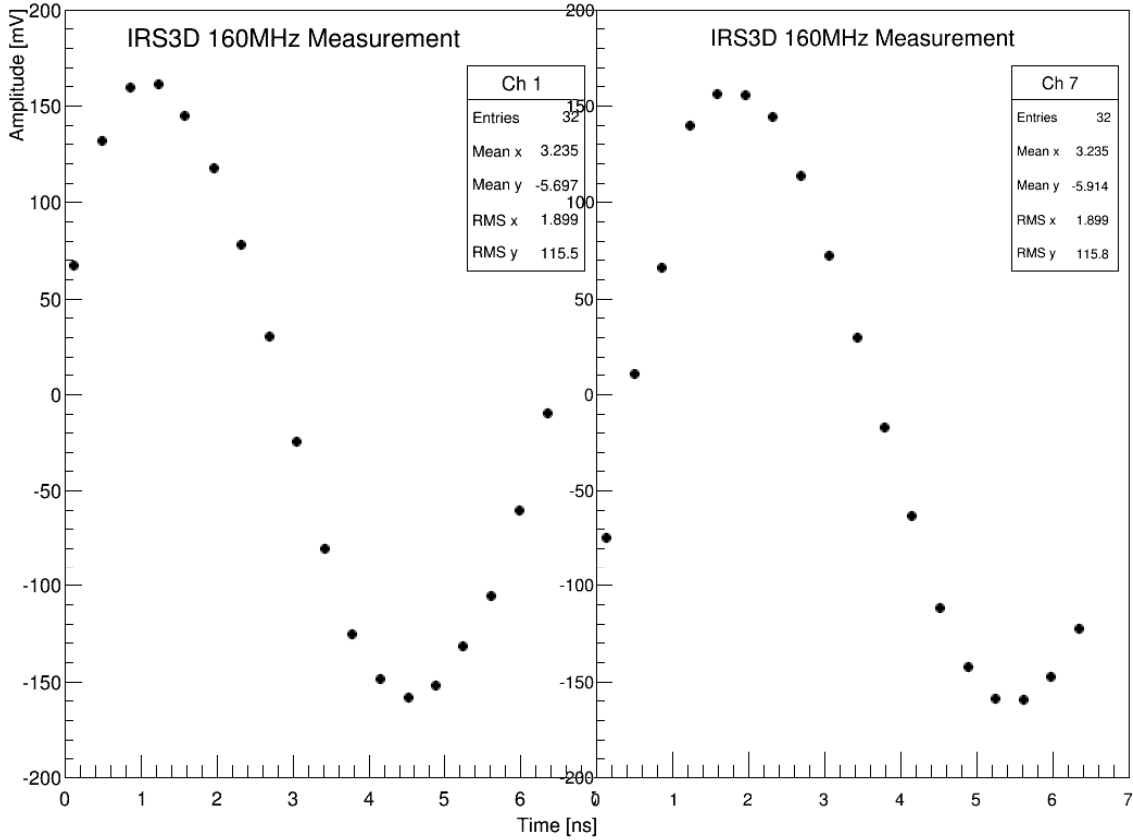
Much improved – some additional improvement expected with linearity correction

Result: visually nicer waveforms



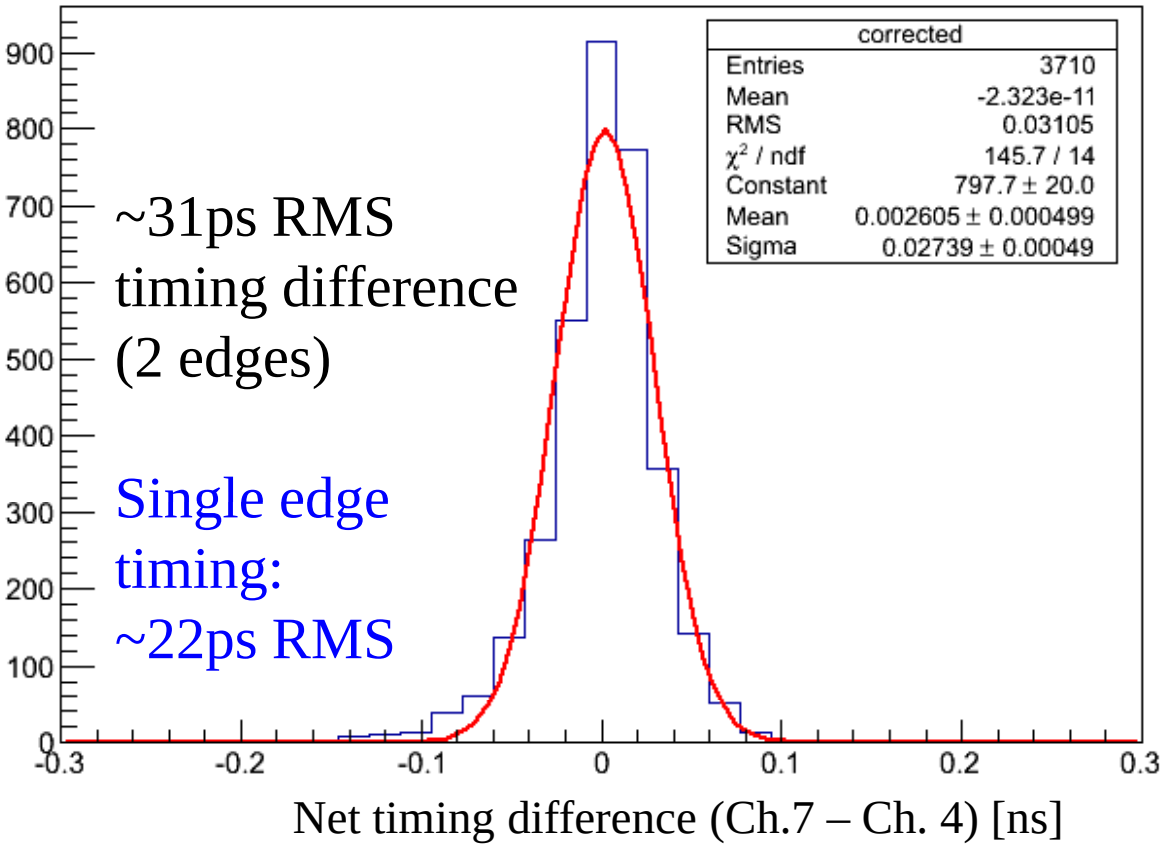


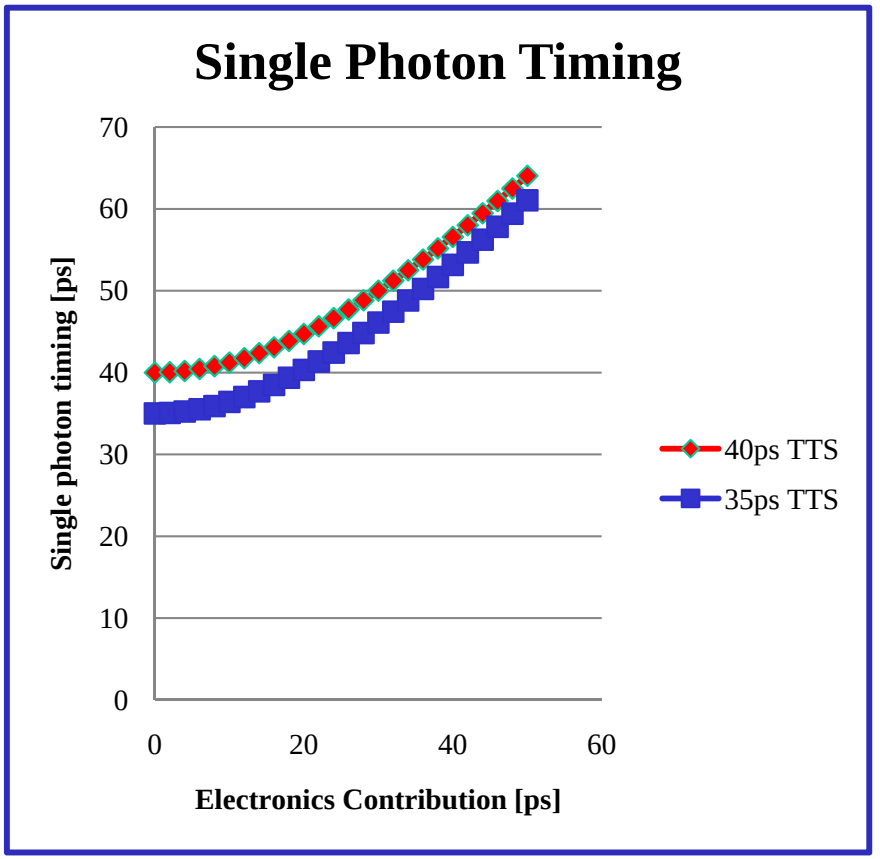
Difference most evident at the extrema of the waveforms



IRS3D timing

(no detailed timebase calibration)





For stretch goal of <= 50ps single p.e. timing, the electronics contribution should be <= 36ps for 35ps MCP-PMT TTS (best case) [<= 30ps for 40ps MCP-PMT TTS (worst case)]

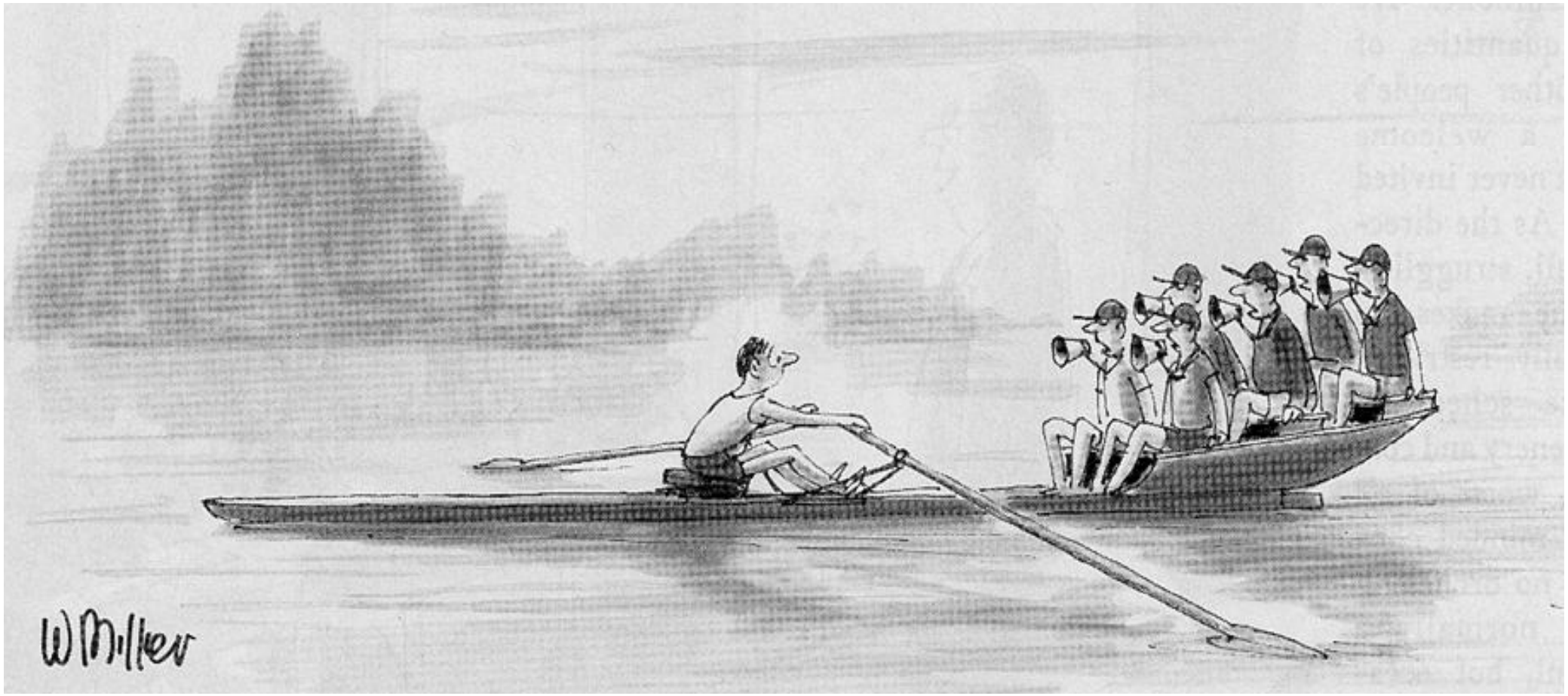
IRS3D looks capable of achieving this goal.

Summary – Day 3

Final batch of detailed information presented

- **Hopefully useful**
- **All of these concepts are straightforward, though much, much, much to be assimilated all at once**
- **Typically 3+ times through needed to 'get it'**
- **Essential issues to be addressed for quality mTC data-taking:**
  - **Are register configurations/feedbacks being set properly ?**
  - **Can we tell ? (meaningful DQM tools ?)**
  - **Understanding what is being done ?**
  - **<u>Calibration!</u> (and diagnostics)**

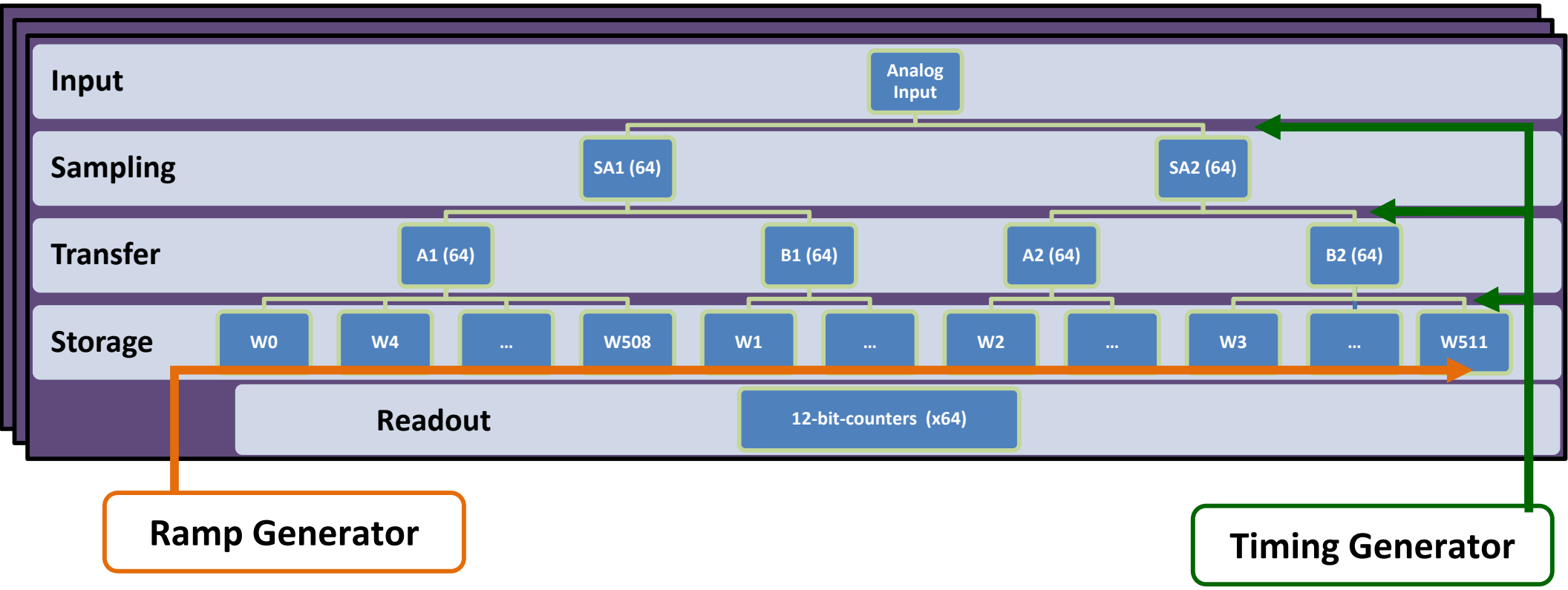
Back-up slides



Resources (where to find more)

- Hardware:
  - IRS3B webpage:
    http://www.phys.hawaii.edu/~idlab/taskAndSchedule/ASIC/IRS3B/IRS3B_homepage.html
  - Board stack schematics:
    http://www.phys.hawaii.edu/~mza/PCB/iTOP/carriers/index.html
    http://www.phys.hawaii.edu/~mza/PCB/iTOP/index.html (Interconnect)
    http://www.phys.hawaii.edu/~mza/PCB/SCROD/index.html
- Firmware:
  - References link:
    http://www.phys.hawaii.edu/~idlab/taskAndSchedule/ASIC/Firmware/Firmware_homepage.html
  - Repositories: http://idlab-scrod.googlecode.com/svn/SCROD-boardstack/iTOP/IRS3B_CRT/
    https://code.google.com/p/idlab-general/source/browse/#svn%2Funiversal_eval%2FIRS3B_DC-stand-alone-firmware%2Fsrc
- Software:
  - Will talk about next time: https://www.phys.hawaii.edu/elog/
    https://www.phys.hawaii.edu/elog/mtc/152

Simplified IRS3B Block Diagram

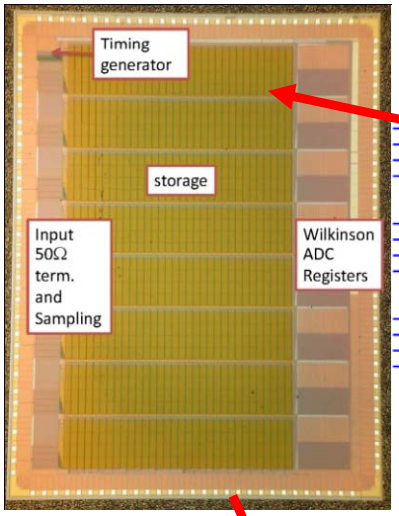


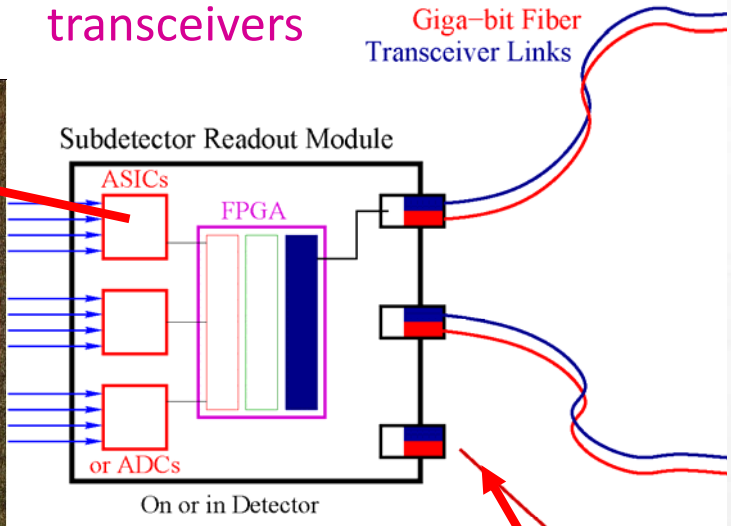
- Per channel:
  - Single input line
  - 128 sampling cells/capacitors
  - 256 transfer cells/capacitors
  - 32,768 storage cells/capacitors
  - 64 counters used to digitize 64-samples in parallel
- Common to all channels:
  - Timing generator
  - Ramp generator

mTC Readout

IRS3B sampling ASIC

12 DAQ fiber transceivers





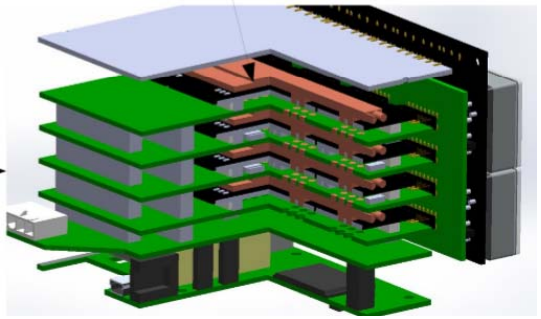
FPGA firmware consists of 3 parts:

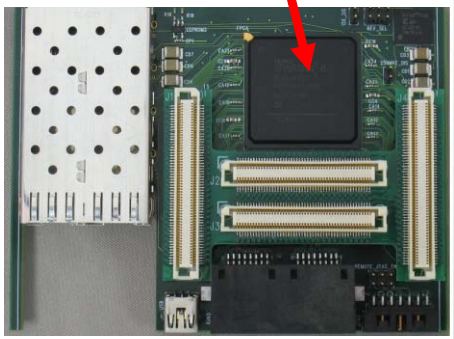
1) ASIC/ADC driver (common)
2) Trigger/feature extract (subdet. specific)
3) Unified DAQ transport protocol

1,536 channels

192 8-ch. ASICs

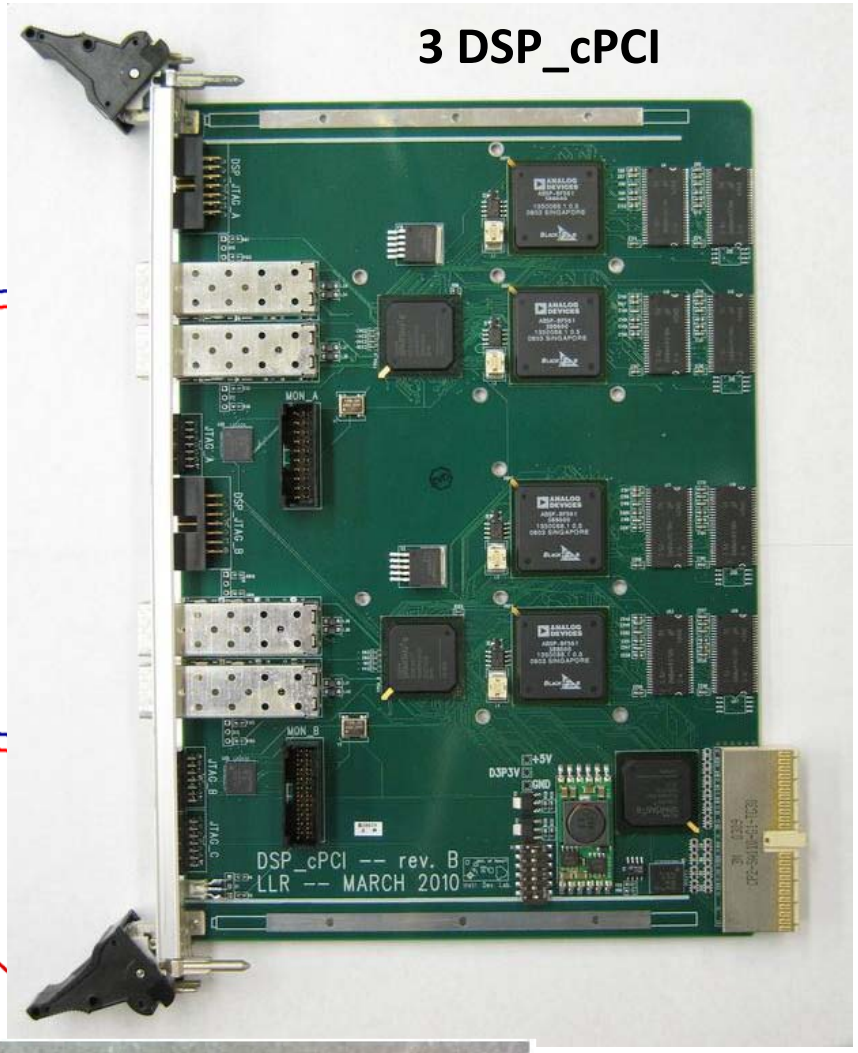
12 SRM "board stacks"

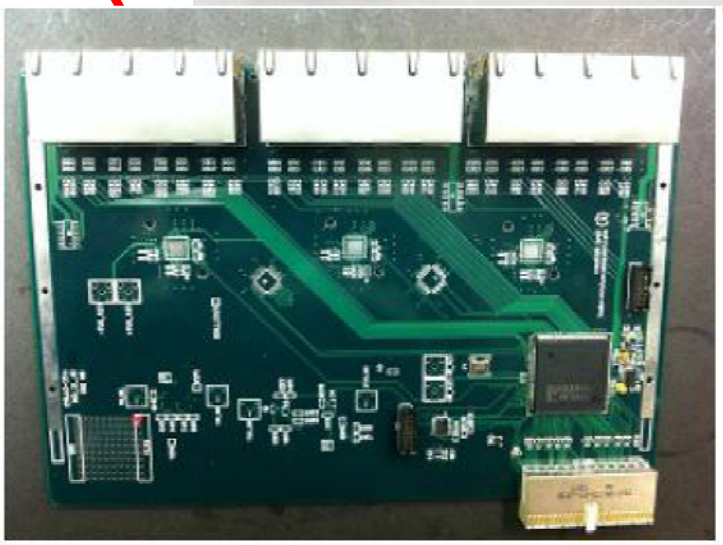




12 SCROD

3 DSP_cPCI

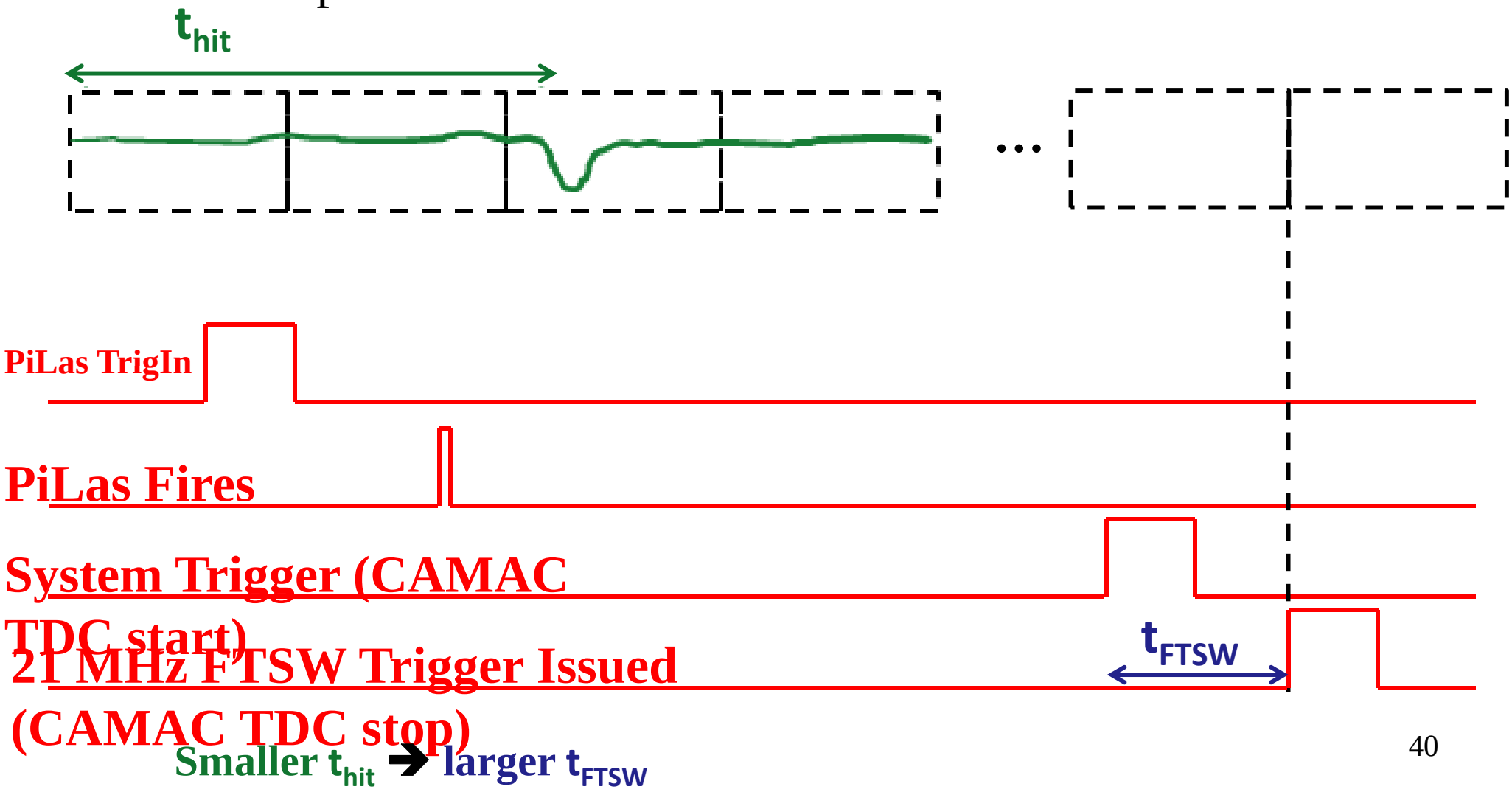




CAJIPCI clock, trigger, programming

Example window buffer mgmt

- Laser fired randomly with respect to FTSW clock…
  - …but at a fixed time relative to the global trigger.
  - Example 1:

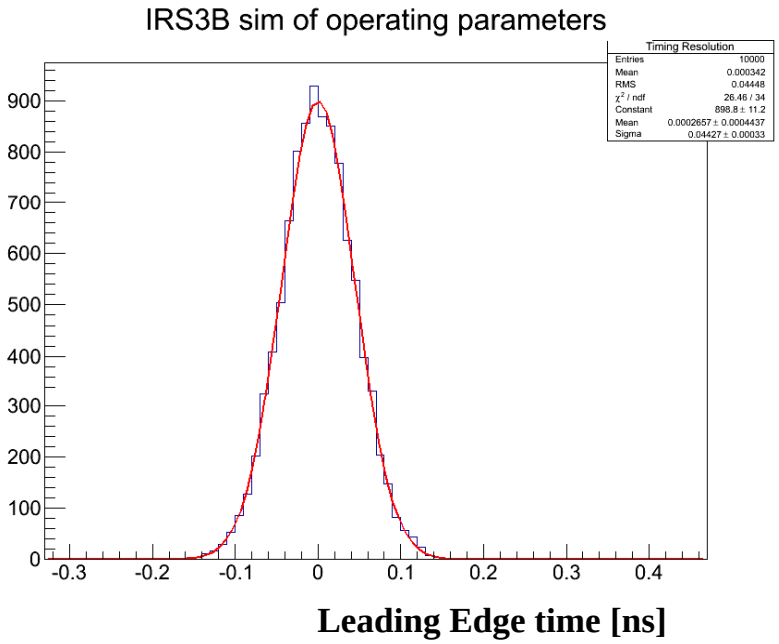


Understanding Expectations: IRS3B "toy" Monte Carlo

Vpeak	**100 ADC**
Risetime	**2.7 ns**
Sampling rate	**2.72 Gsa/s**
nom dT	**0.368 ns**
nom dV	**13.617 ADC/sample**

40% CFD ratio:

Applied between 2 points on leading edge that bracket this transition



~44ps for 100mV peak, 2mV noise