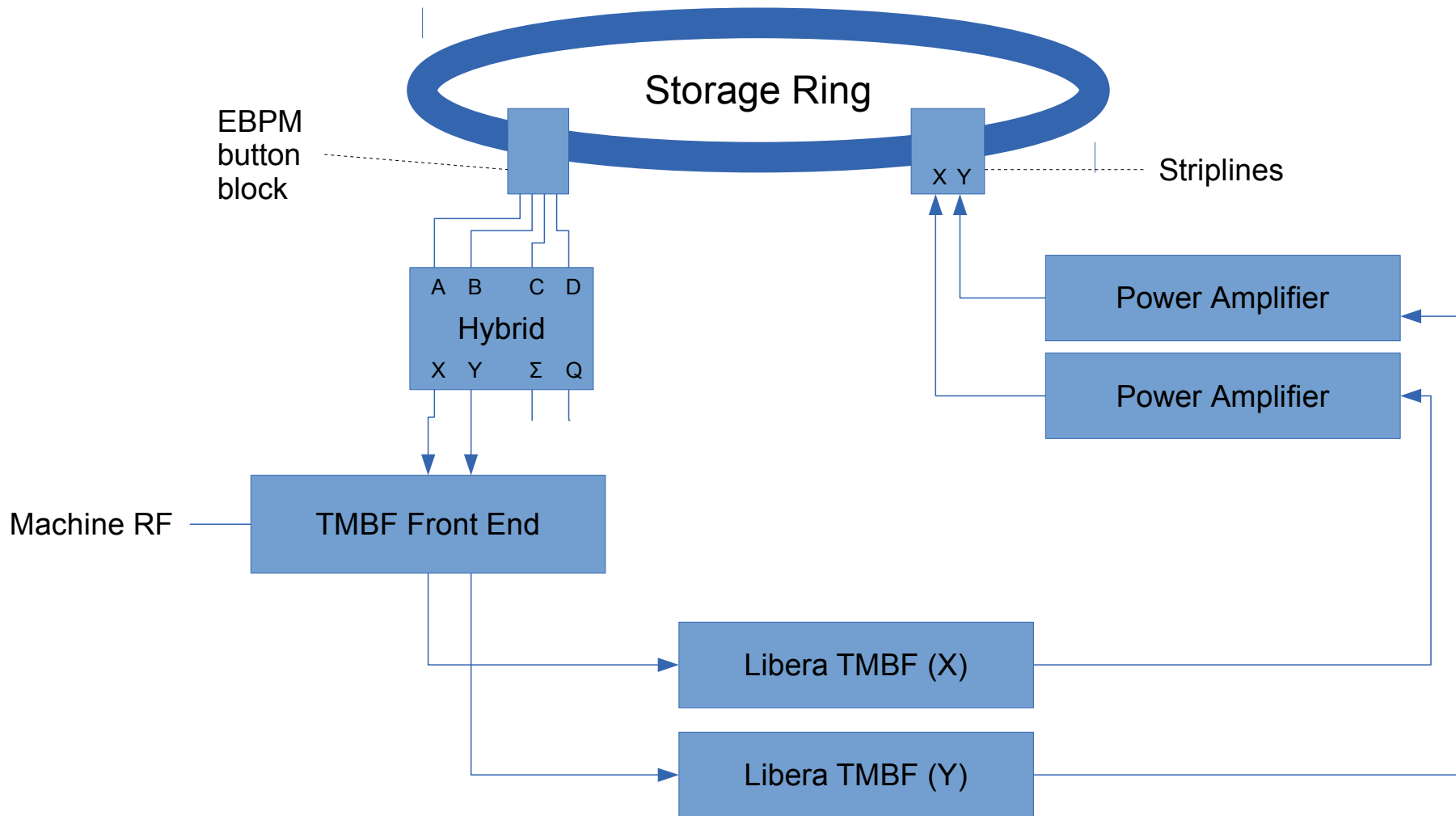


# Extending the Capabilities of the Transverse Multibunch Feedback Processor at Diamond

Michael Abbott, Diamond Light Source  
Libera Workshop 2014  
10<sup>th</sup> April 2014

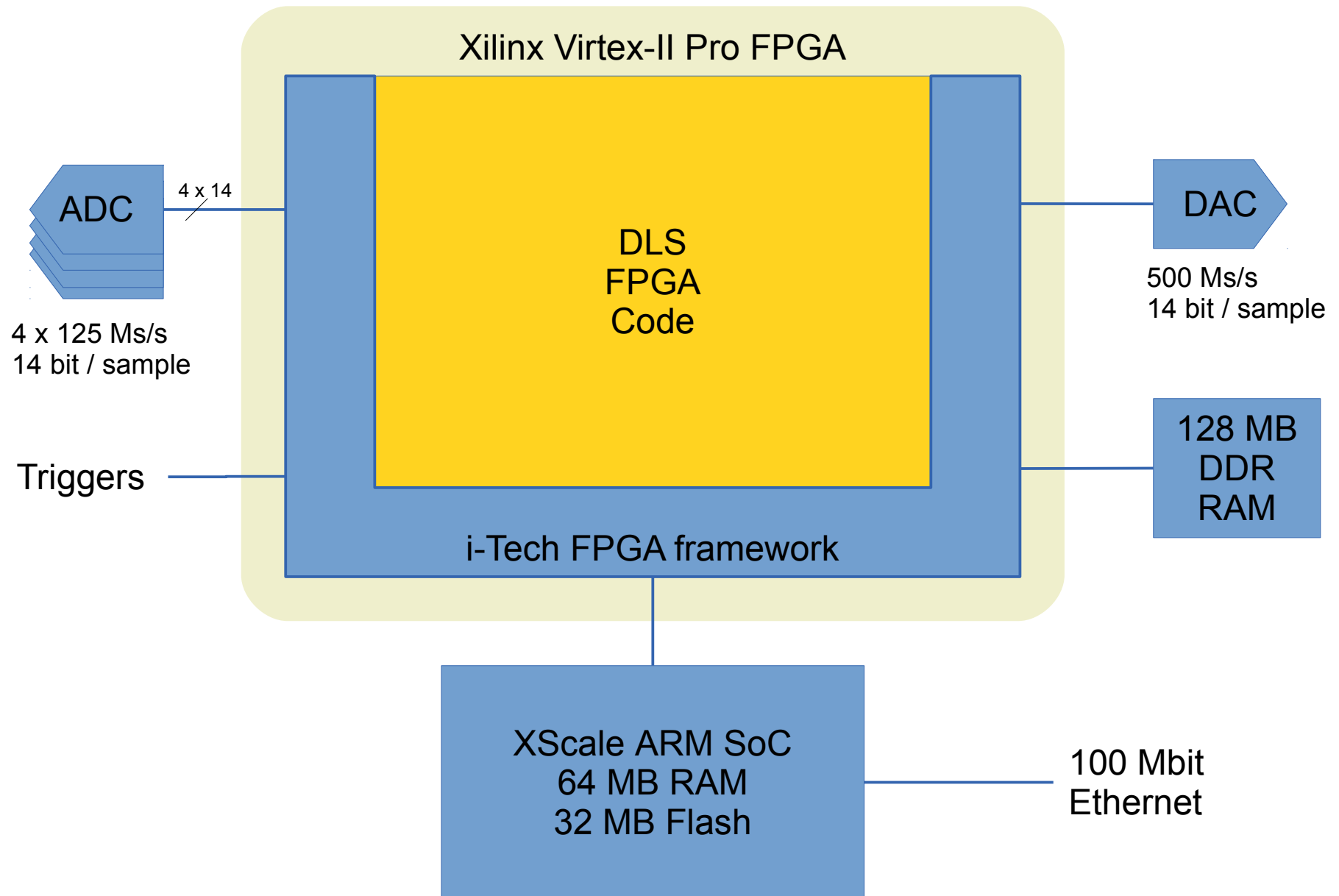
# TMBF in Context



# TMBF Processing Framework

- Button pickups for ABCD RF signals
- RF hybrids convert to X & Y
- Libera TMBF front end for signal conditioning
- 4 x 125 Ms/s 14-bit ADC in Libera TMBF
- FPGA performs processing and control
- 1 x 500 Ms/s 14-bit DAC in Libera TMBF
- Power Amplifier drives striplines to couple back to stored beam

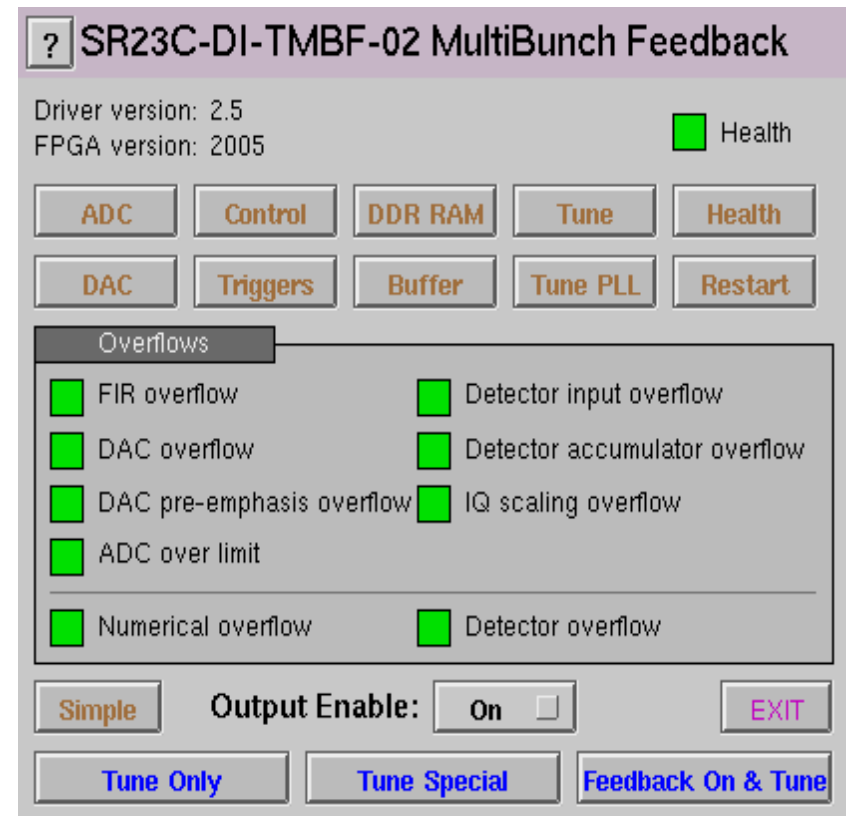
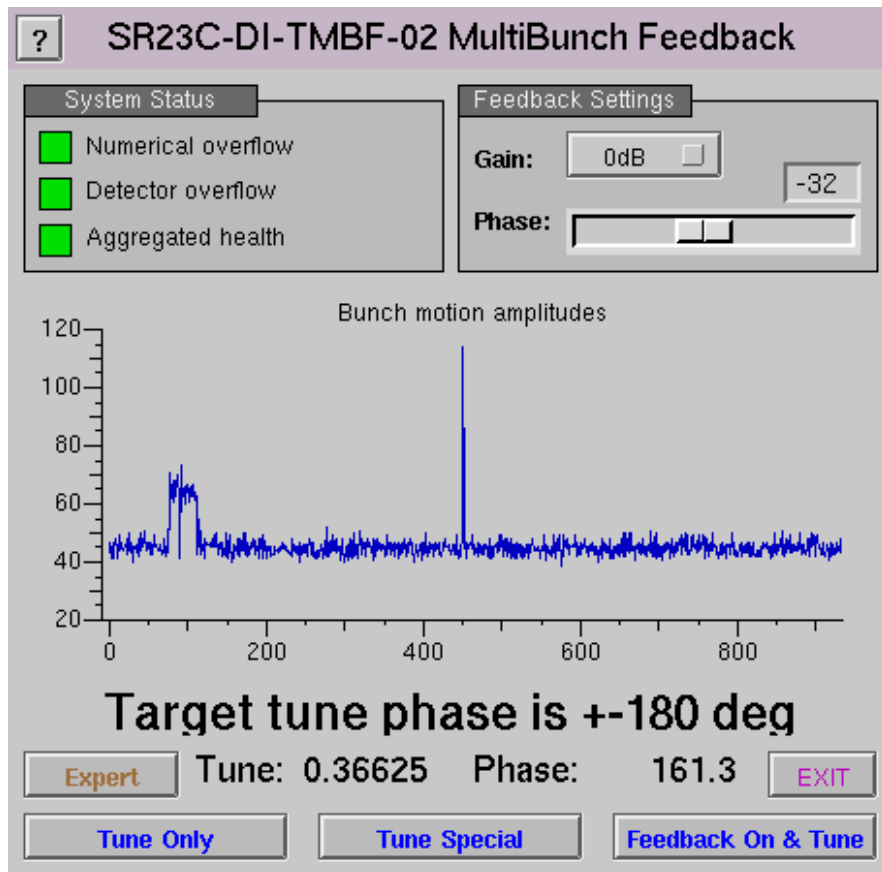
# Libera TMBF



# TMBF Applications

- Instability damping: simple tuned FIR to reverse phase for negative feedback. Works well in part due to very sharp resonance of beam at machine tune frequency.
- Tune measurement by tune excitation and detection
- Machine physics investigations: probing instabilities

# Top Level EPICS Control



# New TMBF Capabilities

- Bunch-by-bunch control of filter, output, gain.
- Three output sources: FIR, two NCOs
- Programmable sequencer
- Tune following PLL
- Output pre-emphasis filter, input compensation to come
- 2ns resolution triggers and synchronisation
- Compensation for all offsets and delays

# Architecture and Implementation

- Split between EPICS IOC (written in C) and FPGA (written in System Verilog)
- Offset and delay compensation in software where possible: 19 delay parameters automatically measured and compensated
- Libera platform provides SBC, DDR RAM, ADC, DAC. Everything else in DLS code



# 19 Compensated Delays

# DDR readout delays

DDR\_ADC\_DELAY = 230

DDR\_FIR\_DELAY = 1

DDR\_RAW\_DAC\_DELAY = 5

DDR\_DAC\_DELAY = 10

# Buffer trigger delays

BUF\_ADC\_DELAY = 229

BUF\_FIR\_DELAY = 0

BUF\_DAC\_DELAY = 9

# MinMax buffer delays

MINMAX\_ADC\_DELAY = 228

MINMAX\_DAC\_DELAY = 8

# Bunch selection offsets

BUNCH\_FIR\_OFFSET = 228

BUNCH\_GAIN\_OFFSET = 0

# Detector single bunch offsets and phase

# skews

DET\_ADC\_OFFSET = 1

DET\_FIR\_OFFSET = 232

# These two values represent actual phase

# delays in bunches relative to an

# aligned single turn.

DET\_ADC\_DELAY = 0

DET\_FIR\_DELAY = 12

# Tune following delays

FTUN\_ADC\_OFFSET = 3

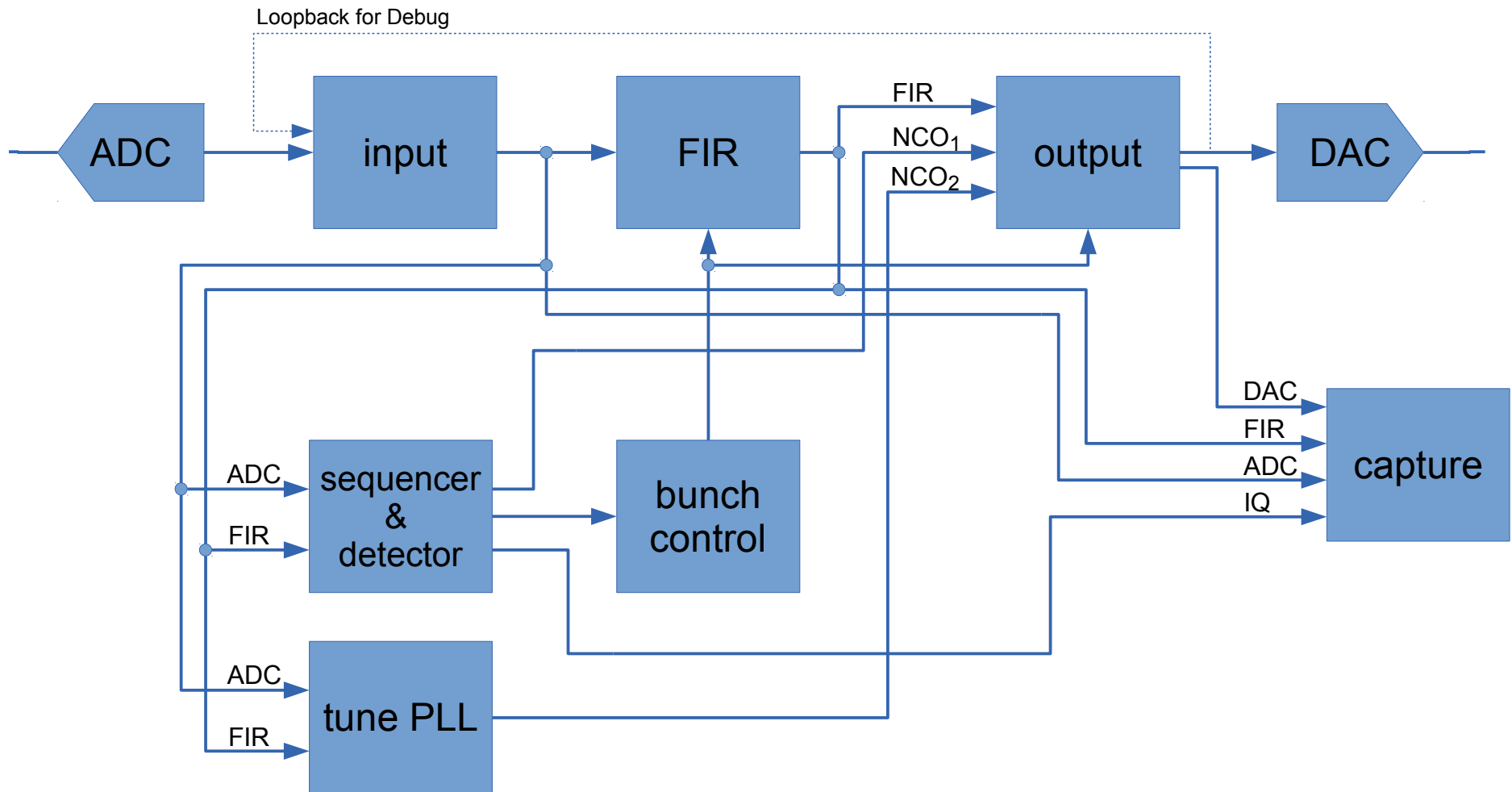
FTUN\_FIR\_OFFSET = 0

FTUN\_ADC\_DELAY = 928

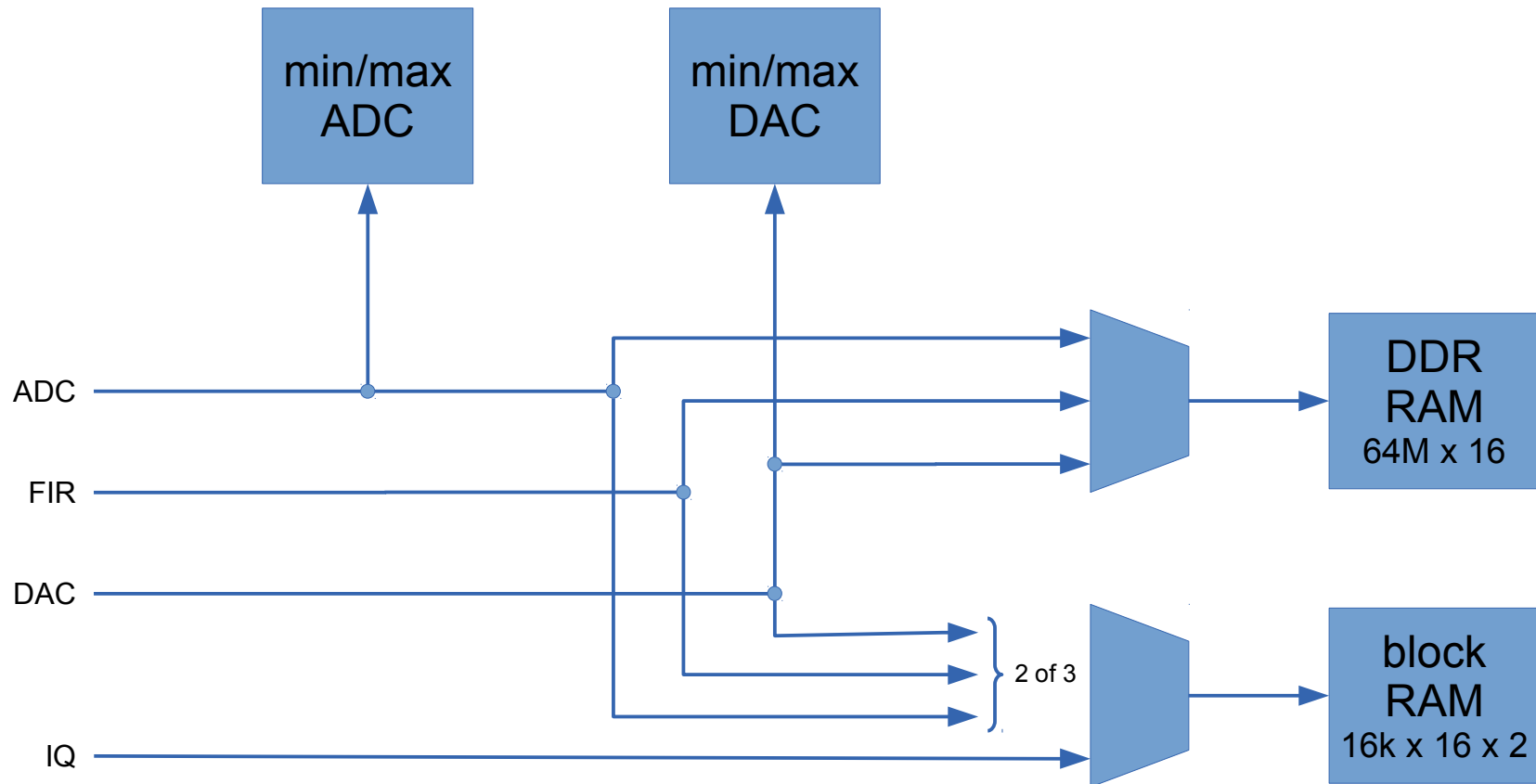
FTUN\_FIR\_DELAY = 940

All measured automatically when testing a new FPGA version

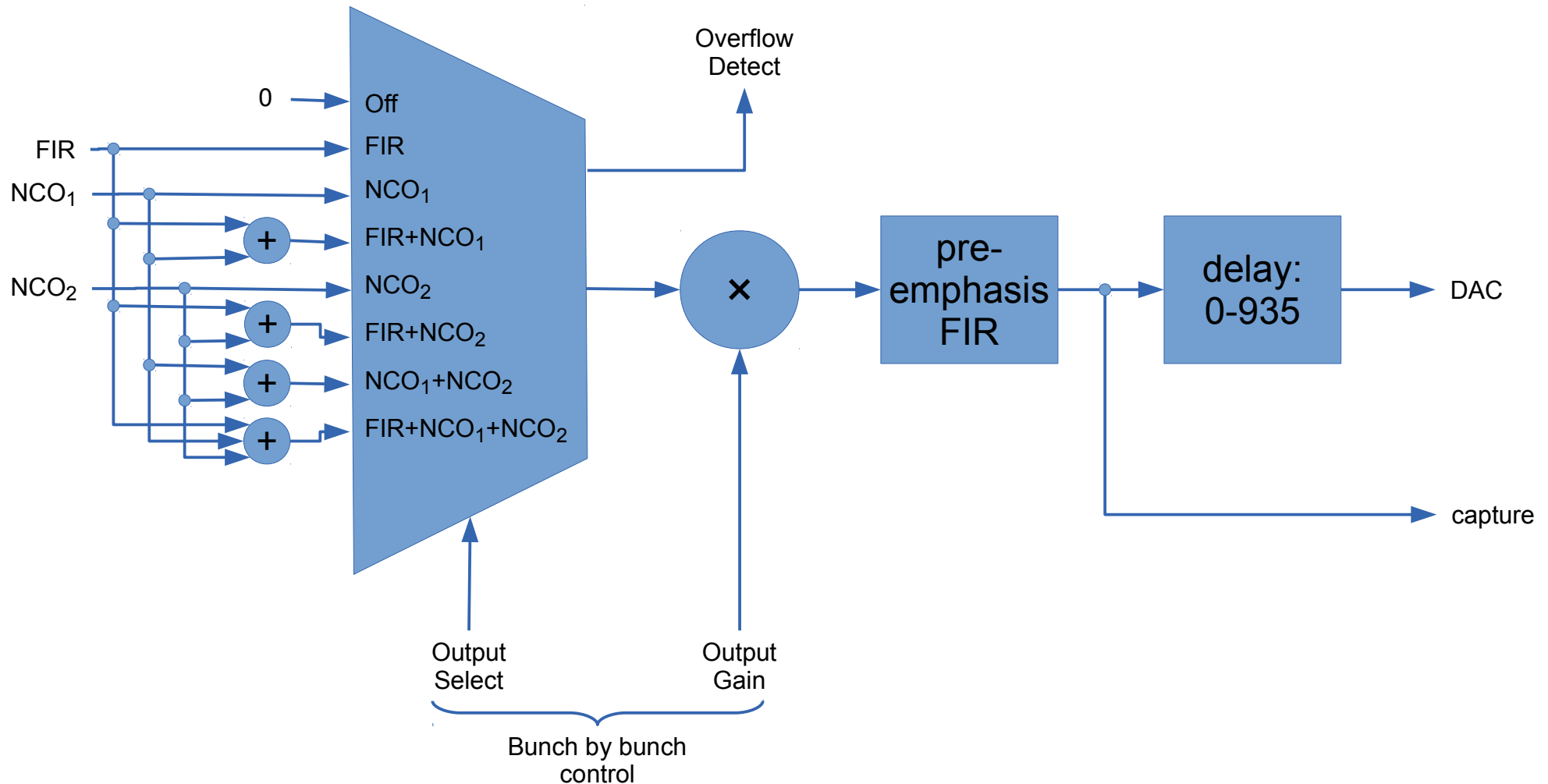
# FPGA System Overview



# Data Capture



# Output Control



# A Fragment of FPGA Code

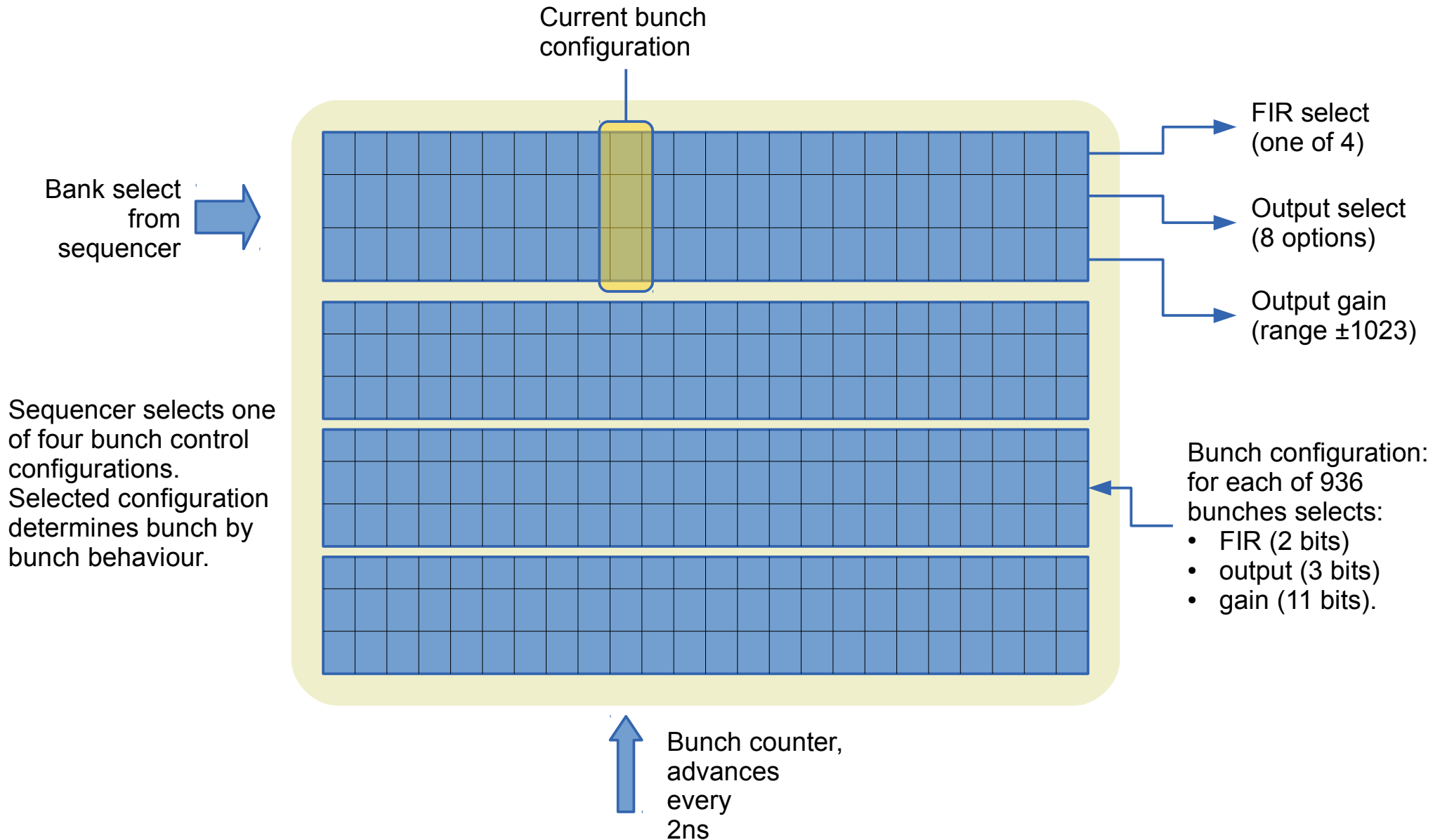
```
// DAC selection input multiplexer. The intermediate selection is 16 bits wide
// so we can detect overflow when adding up to three 14 bit values.
logic [7:0] [15:0] dac_out_mux = 0;
always_ff @(posedge adc_clk_i) begin
    dac_out_mux[0] <= 0;
    dac_out_mux[1] <= signed'(fir_dat_i);
    dac_out_mux[2] <= signed'(hom_0_dat_i);
    dac_out_mux[3] <= signed'(hom_0_dat_i) + signed'(fir_dat_i);
    dac_out_mux[4] <= signed'(hom_1_dat_i);
    dac_out_mux[5] <= signed'(hom_1_dat_i) + signed'(fir_dat_i);
    dac_out_mux[6] <= signed'(hom_1_dat_i) + signed'(hom_0_dat_i);
    dac_out_mux[7] <=
        signed'(hom_1_dat_i) + signed'(hom_0_dat_i) + signed'(fir_dat_i);
end
wire [15:0] dac_out_sel = dac_out_mux[out_mux_sel_i];

// Latch final result and detect overflow
logic [13:0] dac_mux_out = 0;
logic mux_overflow = 0;
always_ff @(posedge adc_clk_i) begin
    dac_mux_out <= dac_out_sel;
    mux_overflow <= ~&dac_out_sel[15:13] & |dac_out_sel[15:13];
end
```

# Bunch by Bunch Control

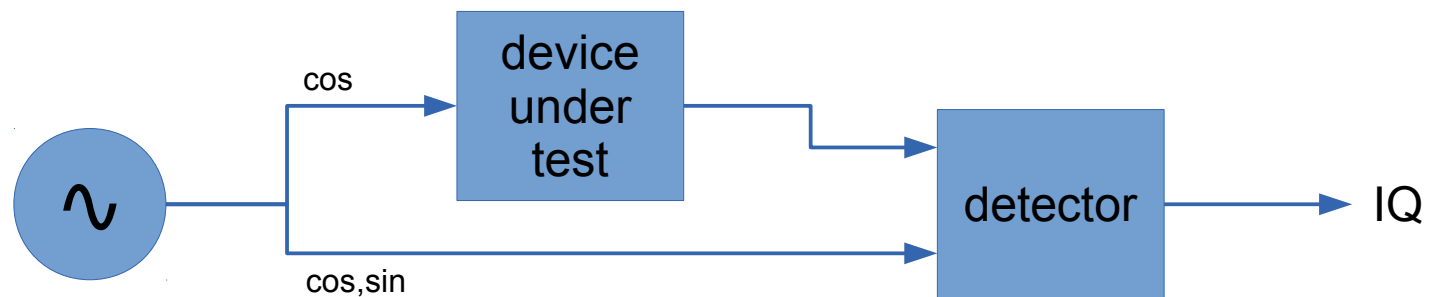
- Different feedback on different bunches, helpful for unusual fill patterns
- Precise control over tune sweeps (can measure up to four channels at once)
- Low risk user time experiments on a single bunch (nobody cares too much if we break one bunch!)

# Bunch Control



# Tune Measurement

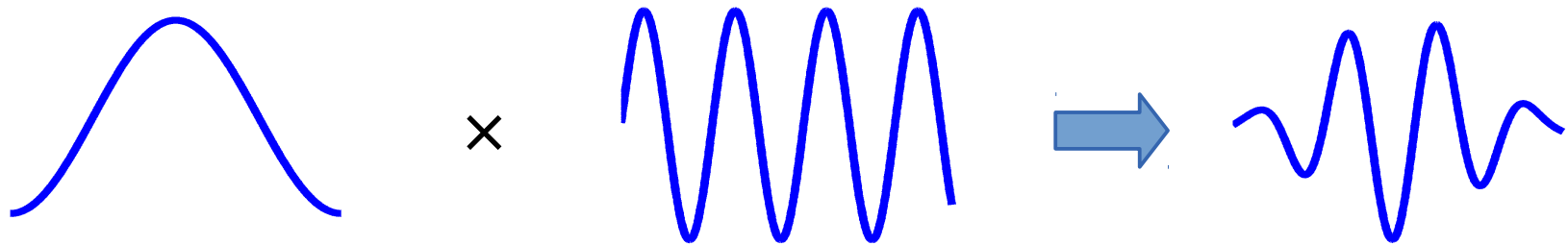
- Excite system under test with fixed frequency sin wave
- Measure response by mixing with original excitation: use both phases to get IQ response
- Sweep excitation frequency to get response over a frequency range



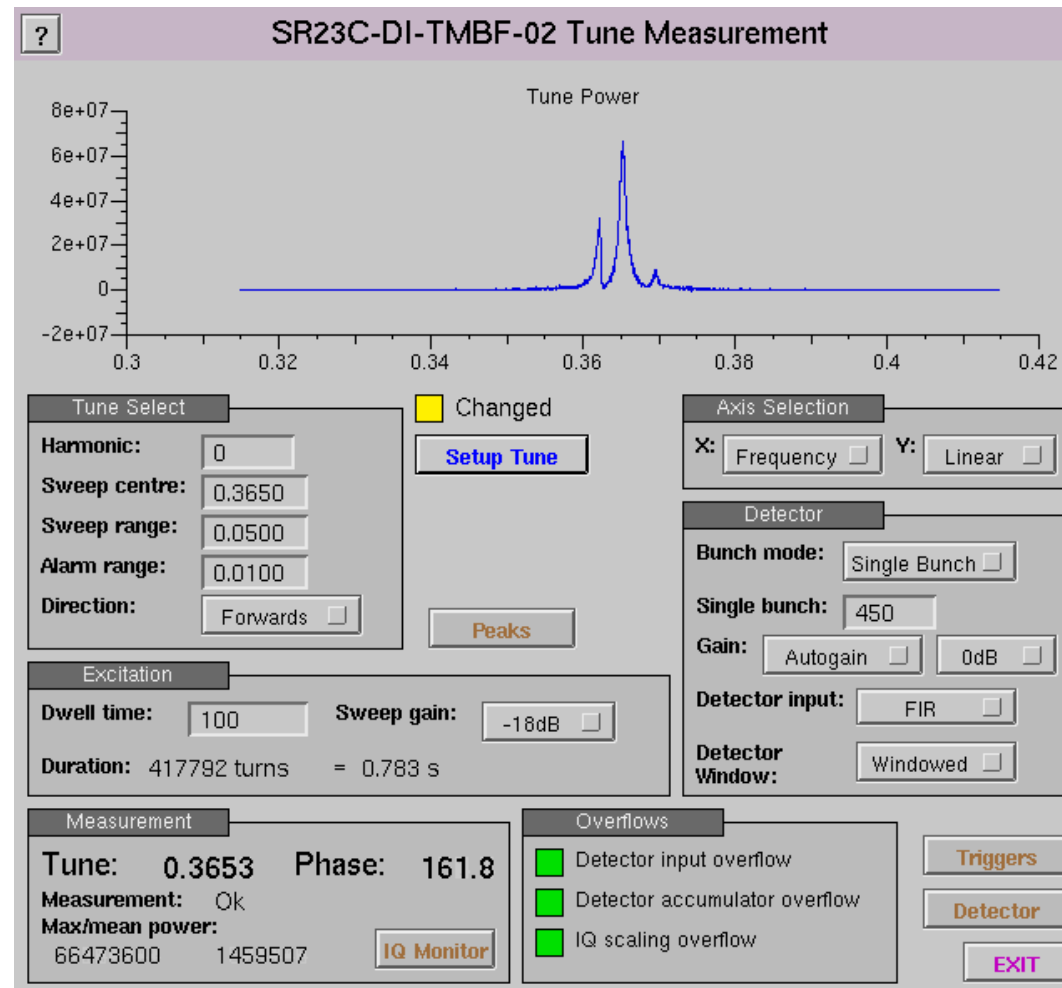


# Detector Features

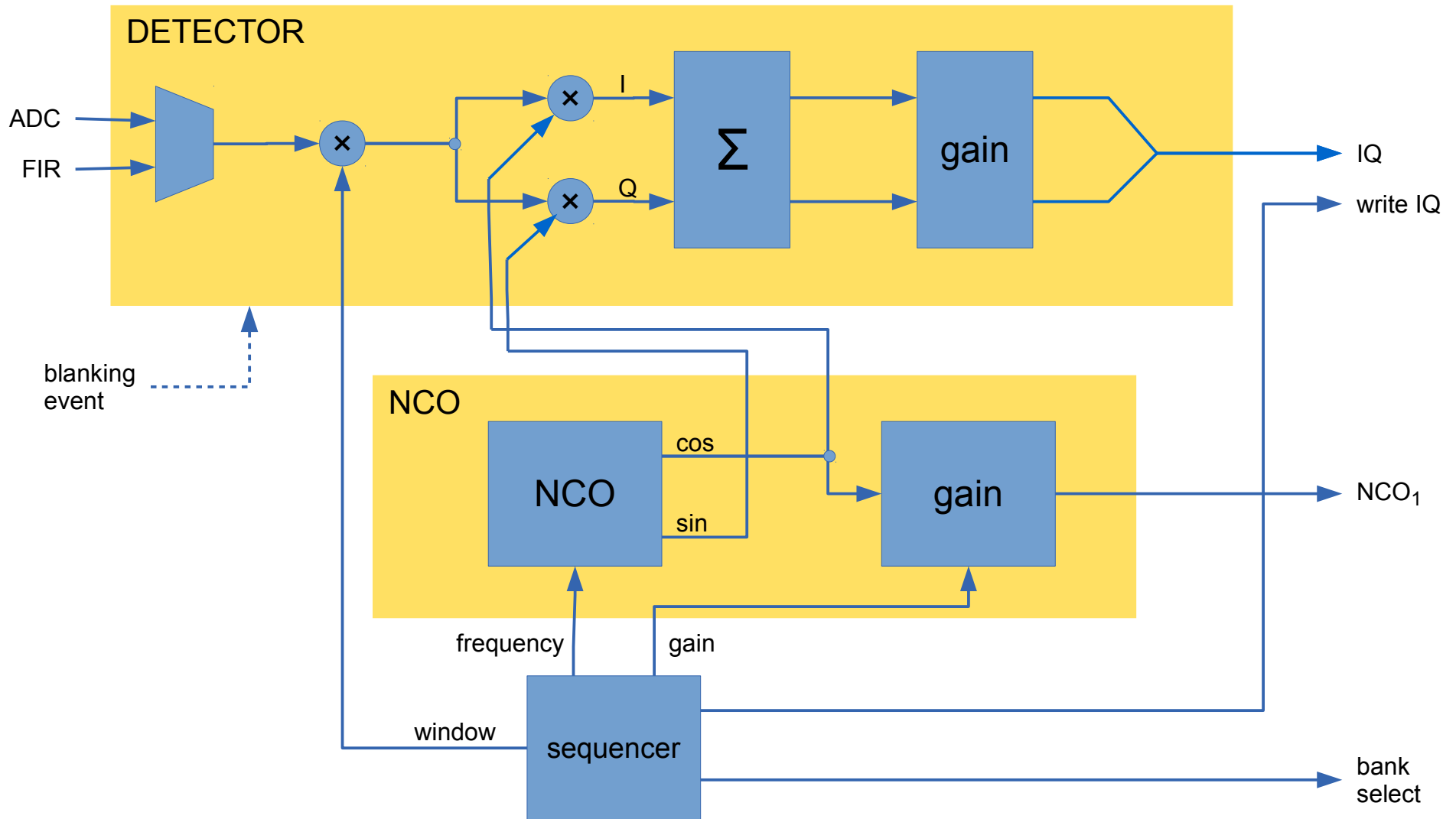
- Blanking signal. External signal during injection to suspend detector during injection to avoid spurious noise.
- Window. Windowing sine wave that doesn't fit into detector window controls leakage from adjacent frequencies:



# Tune Measurement Screen



# Sequencer and Detector

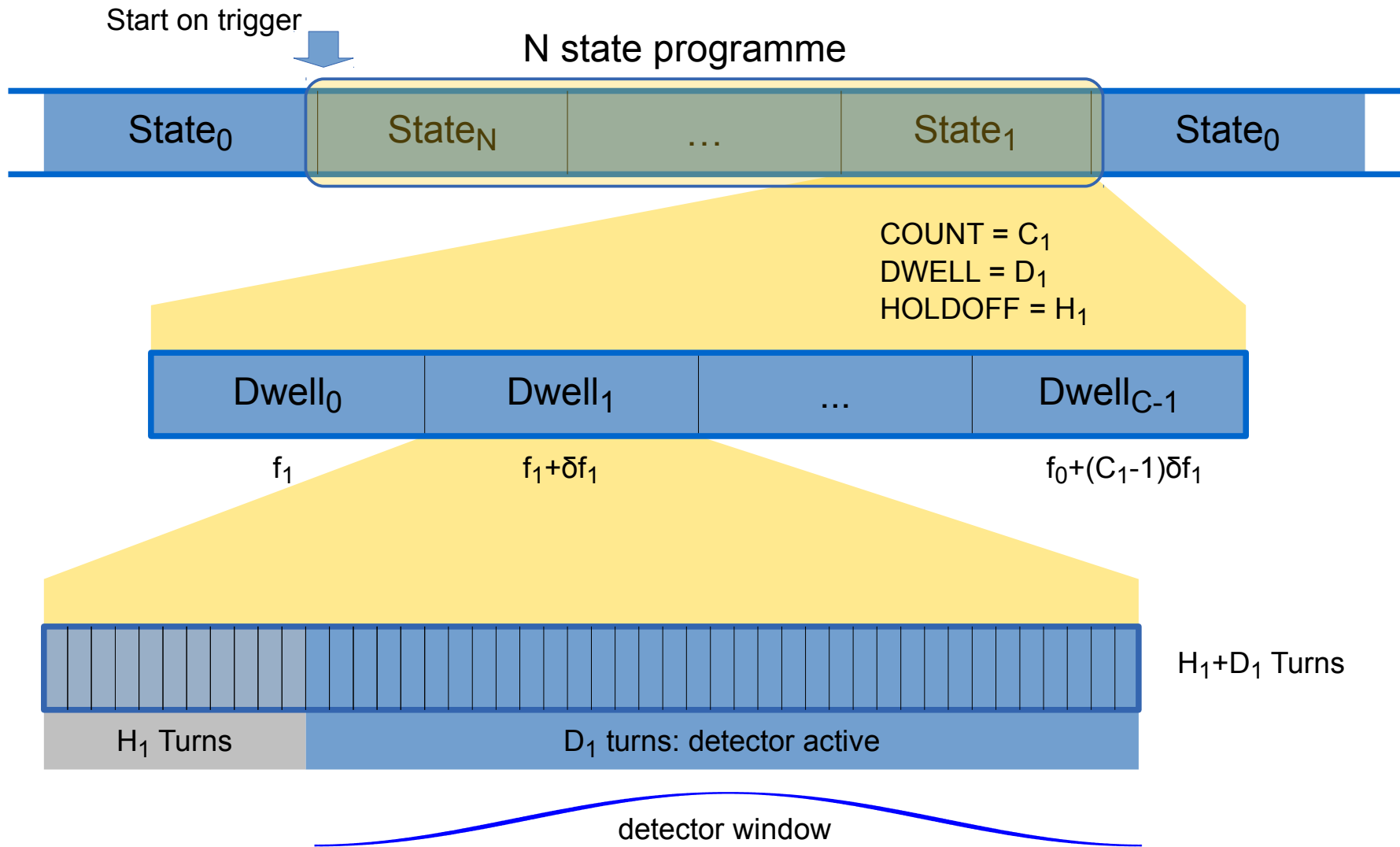


# Programmable Sequencer

For tune sweeps and timed feedback control experiments. For up to 7 sequenced states can select:

- Bunch configuration from one of four selections
- Frequency sweep for NCO
- Optional IQ data capture for sweep response
- Duration of state

# Sequencer Programme



# Sequencer Applications

- Tune sweep: one extra state with sweep
- Complex machine investigations. For example, grow/damp exploration of unstable modes, can perform the following experiment:
  - Running in feedback with standard feedback
  - Turn feedback off, excite one mode for 100 turns
  - Wait for natural growth of mode, eg 1000 turns
  - Run an alternate feedback filter for 1000 turns
  - Restore standard feedback, capture 1000 turns

# Implementing Grow/Damp

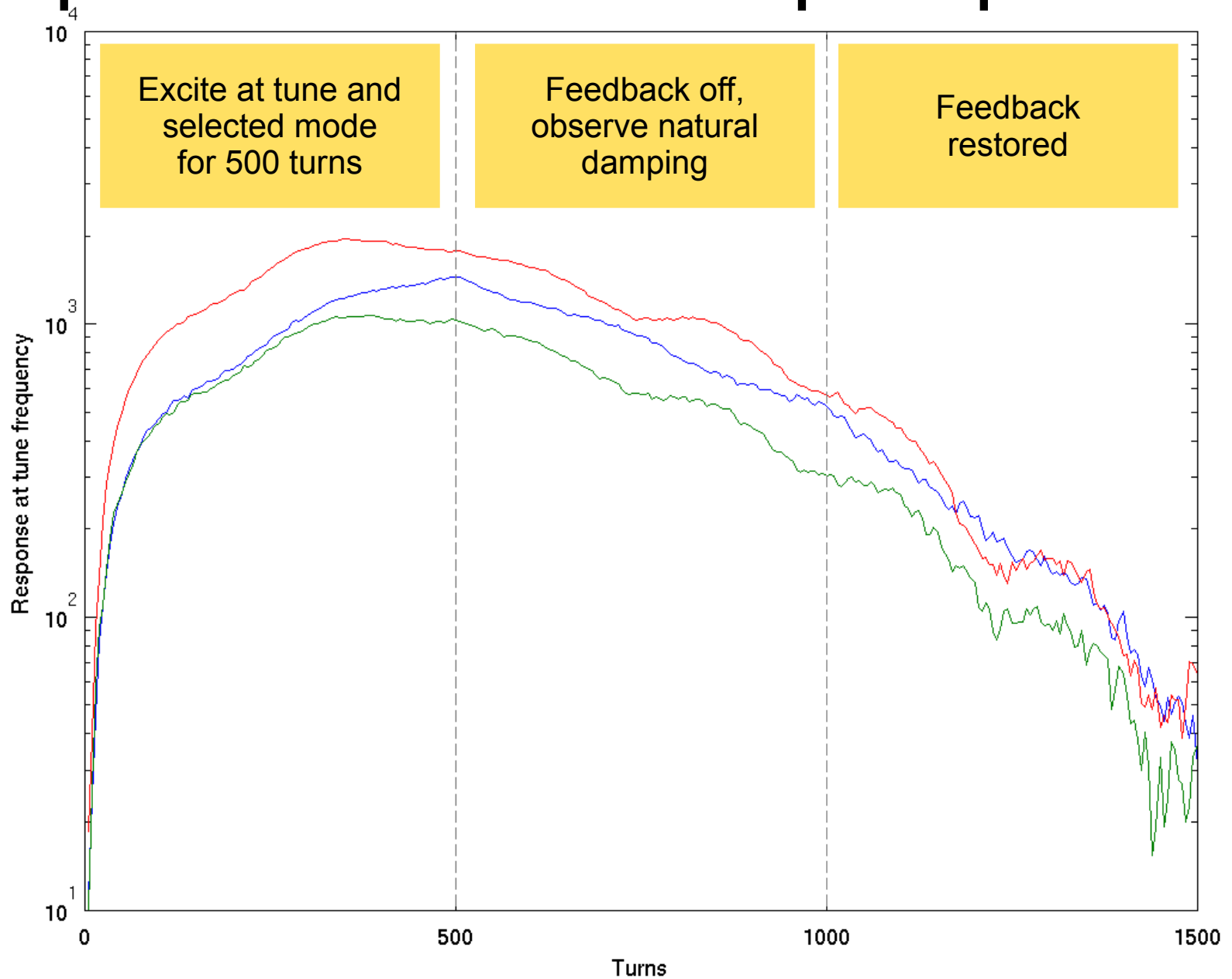
State #	FIR	Output	Duration	Dwell time	Bank #
0	Standard	FIR	-	-	0
4	-	Sweep NCO	100	1	1
3	-	Off	1000	1	2
2	Alternate	FIR	1000	1	3
1	Standard	FIR	1000	1	0
0	Standard	FIR	-	-	0

Here we use four out of seven available sequencer states, two of four available feedback filters, and all four available bunch control banks.

The result of this experiment is 3100 IQ sample points.

Can also simultaneously capture turn by turn data into the fast DDR buffer if desired, have room for up to 35,000 turns in the buffer, but readout is relatively slow, around 1,200 turns per second.

# Graph of a Grow/Damp Experiment





# Low Level Control Screens

**TS-DI-TMBF-01 Control Configuration**

**Sequencer**

	Sweep start	Sweep step	Sweep end	Magnitude	Bunch bank
1	37.75000	0.0000244	37.85000	-6dB	Bank 1
	4096	0	30	Off	Windowed
	Capture	Holdoff	Dwell time	Blanking	Data window
Start:	1	1	Window	Capture count: 4096	Steady state: Bank 0
	Stop	Reset Win		Duration: 122880 turns = 0.230 s	More States

**Bunch Bank**

Bank	FIR select	DAC out status	DAC gain
0	All #0	Off	1023
1	All #0	Sweep	1023
2	All #0	NCO	1023
3	All #0	Off (NCO @0)	1023

**FIR Control**

FIR	FIR freq/size	FIR phase	Gain
0	2 / 9	25	-42dB
1	4 / 10	180	
2	3 / 8	131	
3	3 / 9	58	

**NCO**

NCO freq: 0.00000 NCO gain: -12dB

**EXIT**

**TS-DI-TMBF-01 Sequencer Setup** **EXIT**

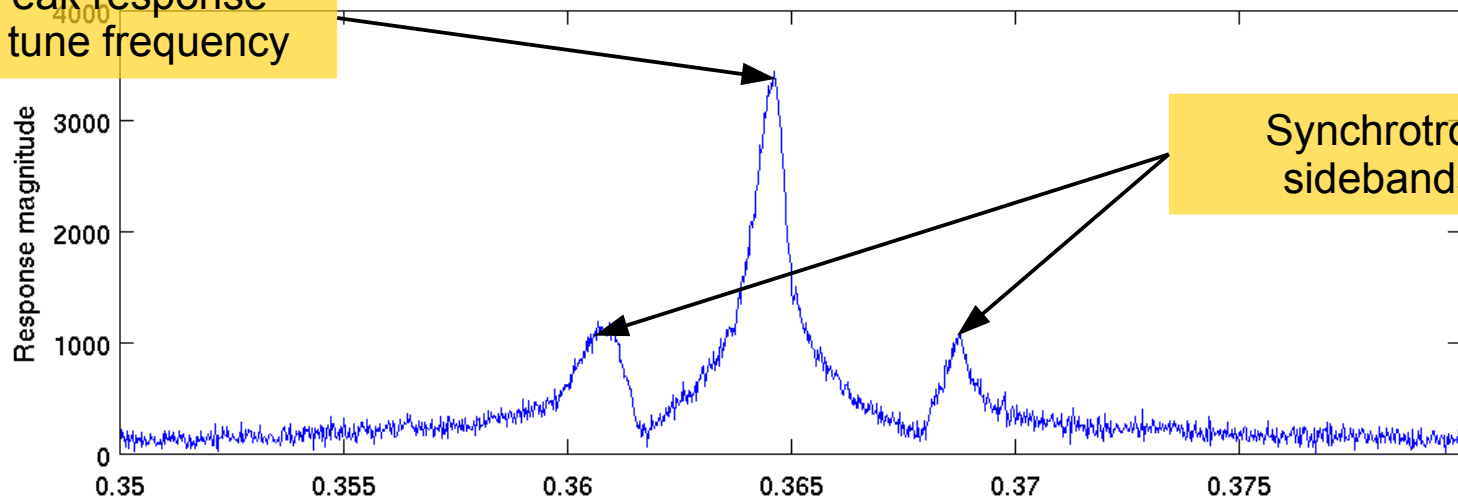
**Sequencer**

	Sweep start	Sweep step	Sweep end	Magnitude	Bunch bank
7	300.0000	0.0000000	300.0000	-6dB	Bank 1
	4096	1	100	Off	Disabled
	Capture	Holdoff	Dwell time	Blanking	Data window
6	6.00000	0.0000000	6.00000	-6dB	Bank 1
	4096	1	10000	Off	Disabled
	Capture	Holdoff	Dwell time	Blanking	Data window
5	1.00000	0.0000000	1.00000	-6dB	Bank 1
	4096	1	1000	Off	Disabled
	Capture	Holdoff	Dwell time	Blanking	Data window
4	4.00000	0.0000000	4.00000	-6dB	Bank 0
	2000	1	100	Off	Disabled
	Capture	Holdoff	Dwell time	Blanking	Data window
3	100.4500	0.0000000	100.4500	0dB	Bank 1
	1	1	10	Off	Disabled
	Discard	Holdoff	Dwell time	Blanking	Data window
2	100.4500	0.0000000	100.4500	0dB	Bank 1
	4096	0	200	Off	Disabled
	Discard	Holdoff	Dwell time	Blanking	Data window
1	37.75000	0.0000244	37.85000	-6dB	Bank 1
	4096	0	30	Off	Windowed
	Capture	Holdoff	Dwell time	Blanking	Data window

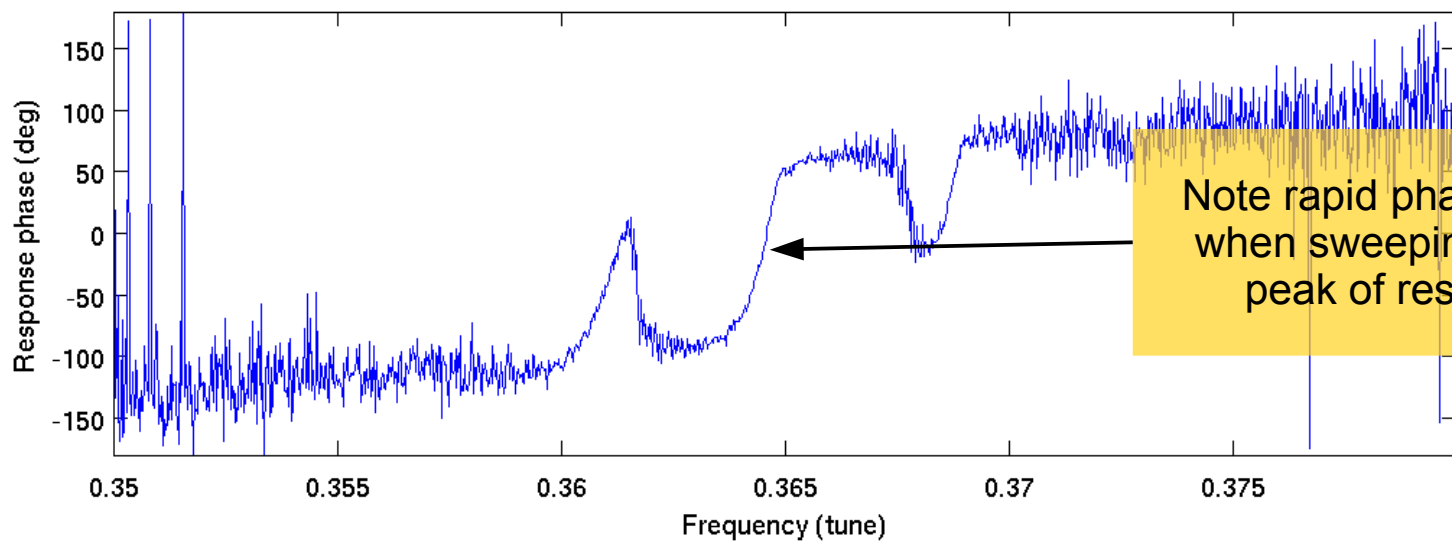
Start: 1 0 Window Capture count: 4096  
Stop Reset Win Duration: 122880 turns = 0.230 s  
Steady state: Bank 0

# Tune Measurement

Peak response  
at tune frequency



Synchrotron  
sidebands

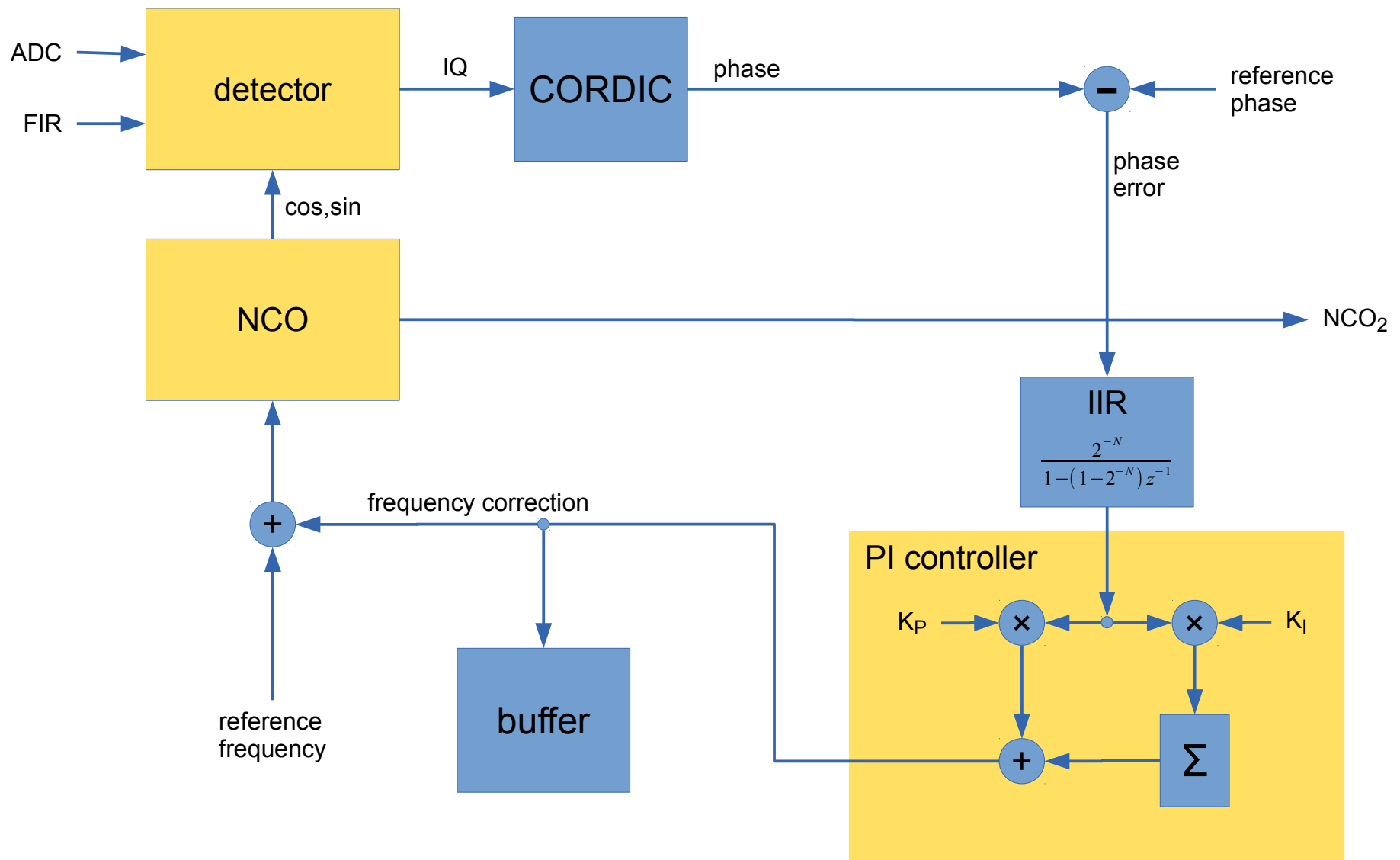


Note rapid phase change  
when sweeping through  
peak of resonance

# Tune Phase Locked Loop

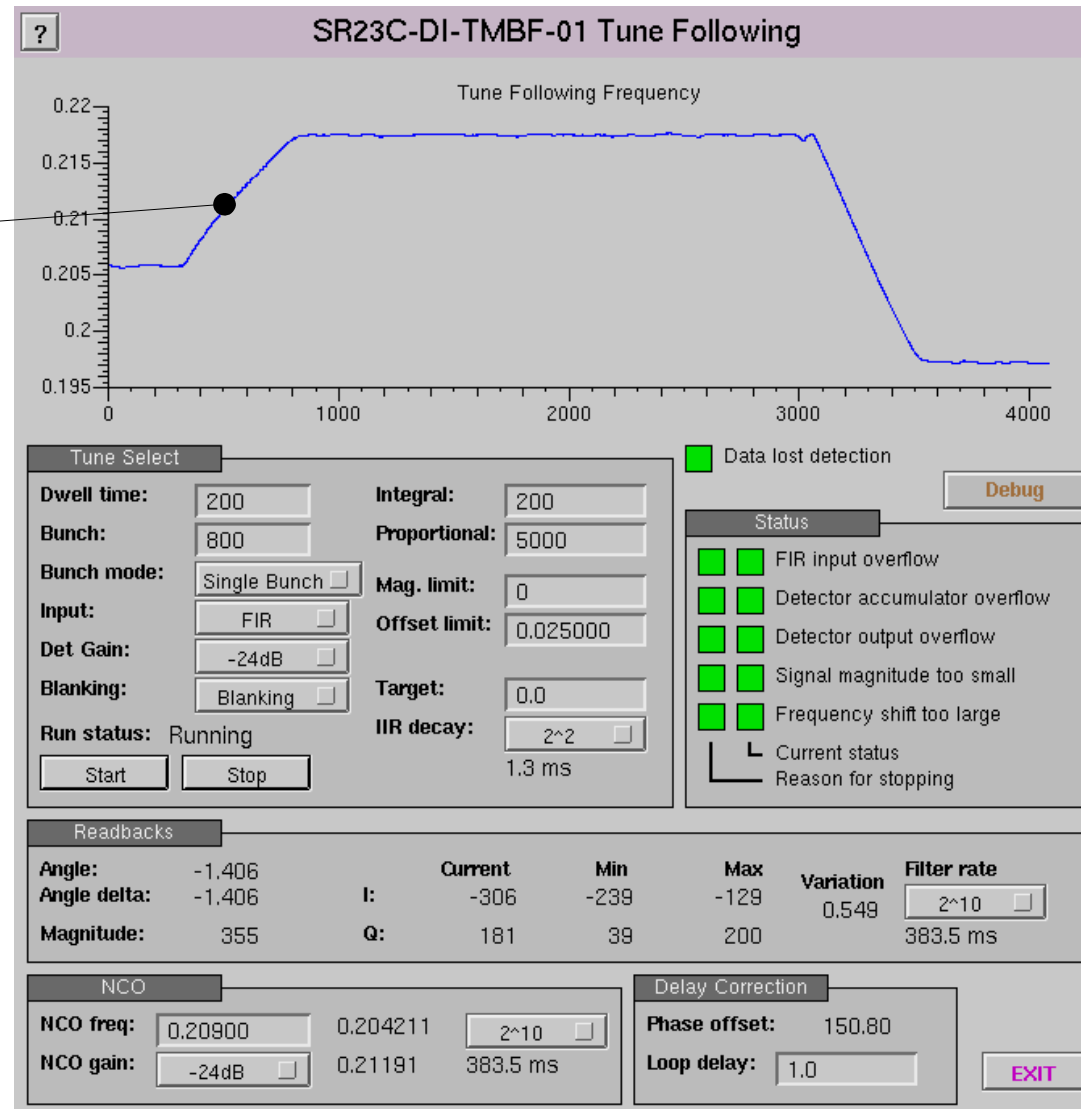
- Take advantage of rapid phase change through tune frequency peak
- Measure phase at, eg, 2.6 kHz (every 200 turns)
- Run simple controller to track frequency to keep phase at target value
- Result is high update rate tune measurement
- Can quickly measure tune width by stepping phase through  $\pm 45^\circ$

# Tune Phase Locked Loop



# Tune PLL Screen

BBA orbit changes  
cause changes in tune

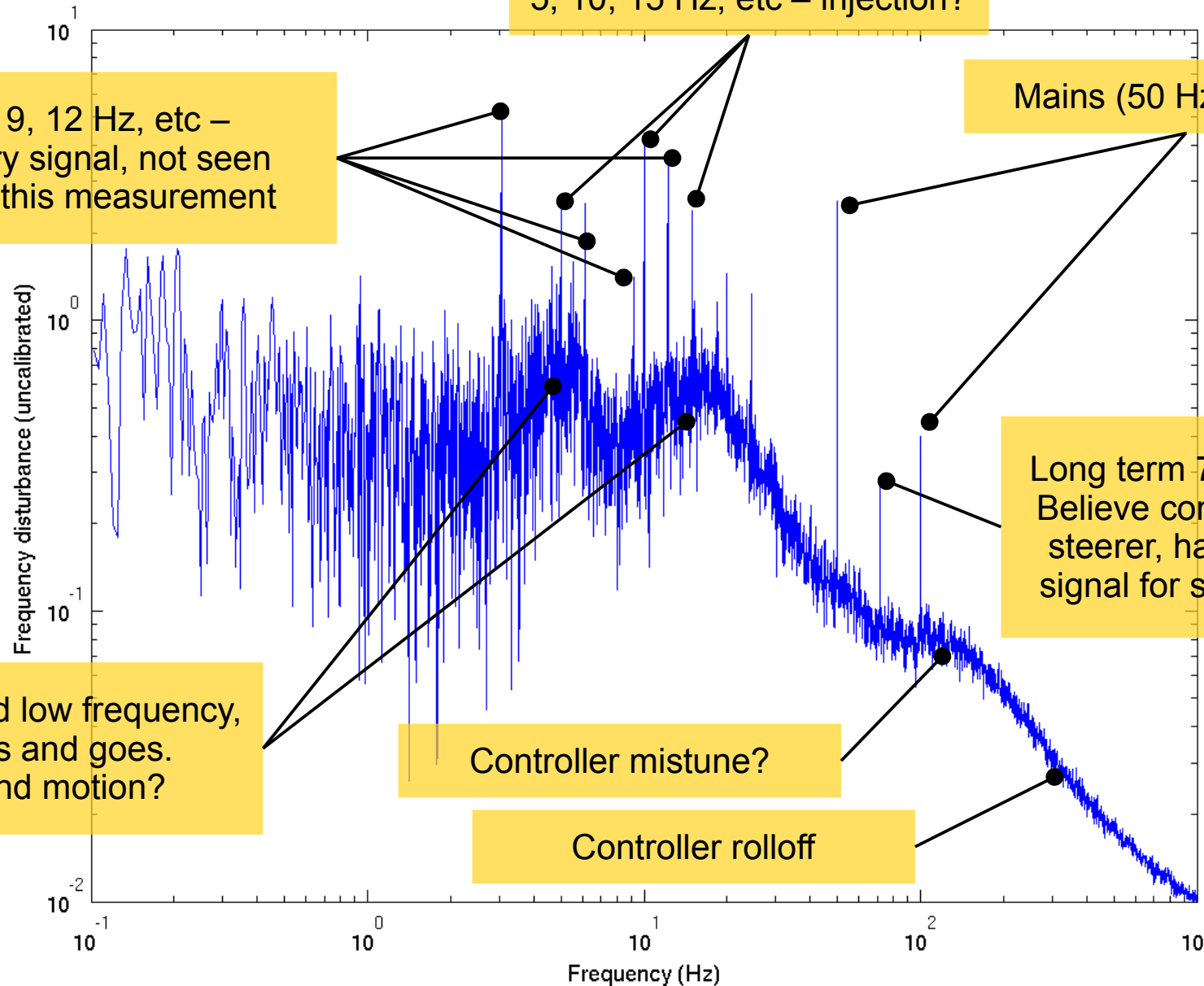


# First PLL Tune Measurements

5, 10, 15 Hz, etc – injection?

3, 6, 9, 12 Hz, etc –  
mystery signal, not seen  
before this measurement

Mains (50 Hz, 100 Hz)



Broadband low frequency,  
comes and goes.  
Ground motion?

Controller mistune?

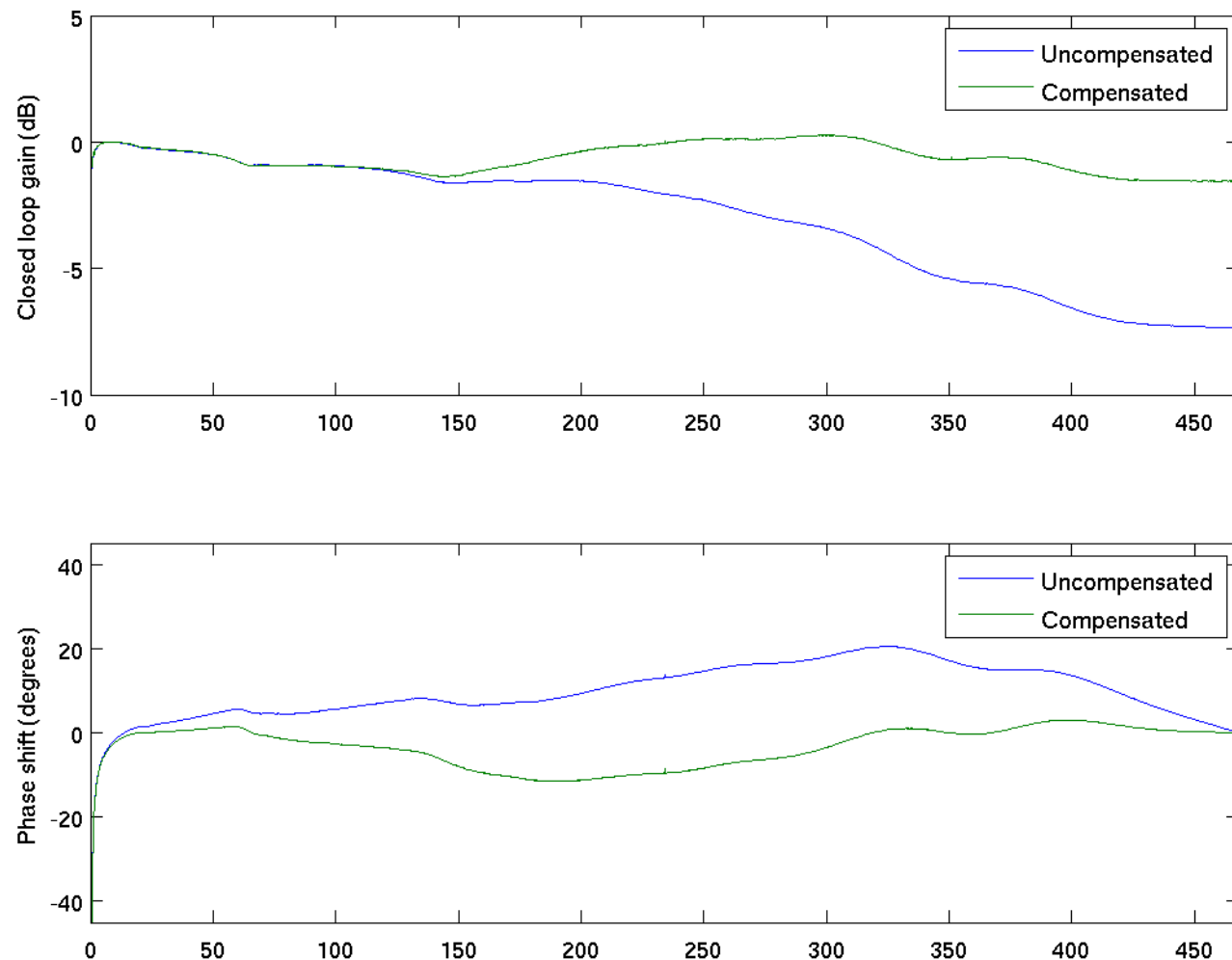
Controller rolloff

# Output Pre-Emphasis

- Loss of gain at high frequencies from DAC, Amplifier, ADC
- Can partially compensate with 3-tap FIR in DAC output
- Will also add 3-tap FIR to ADC input to compensate ADC high frequency droop

# Closed Loop Gain Compensation

Filter coefficients: +1.323 -0.485 +0.162

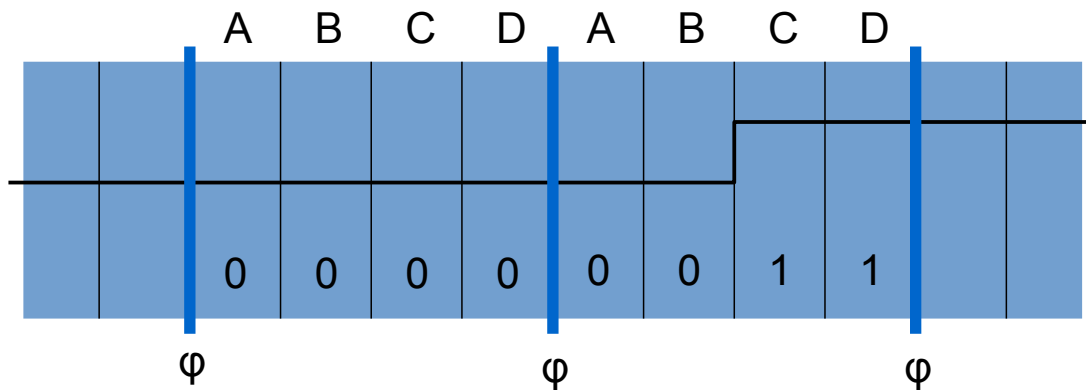
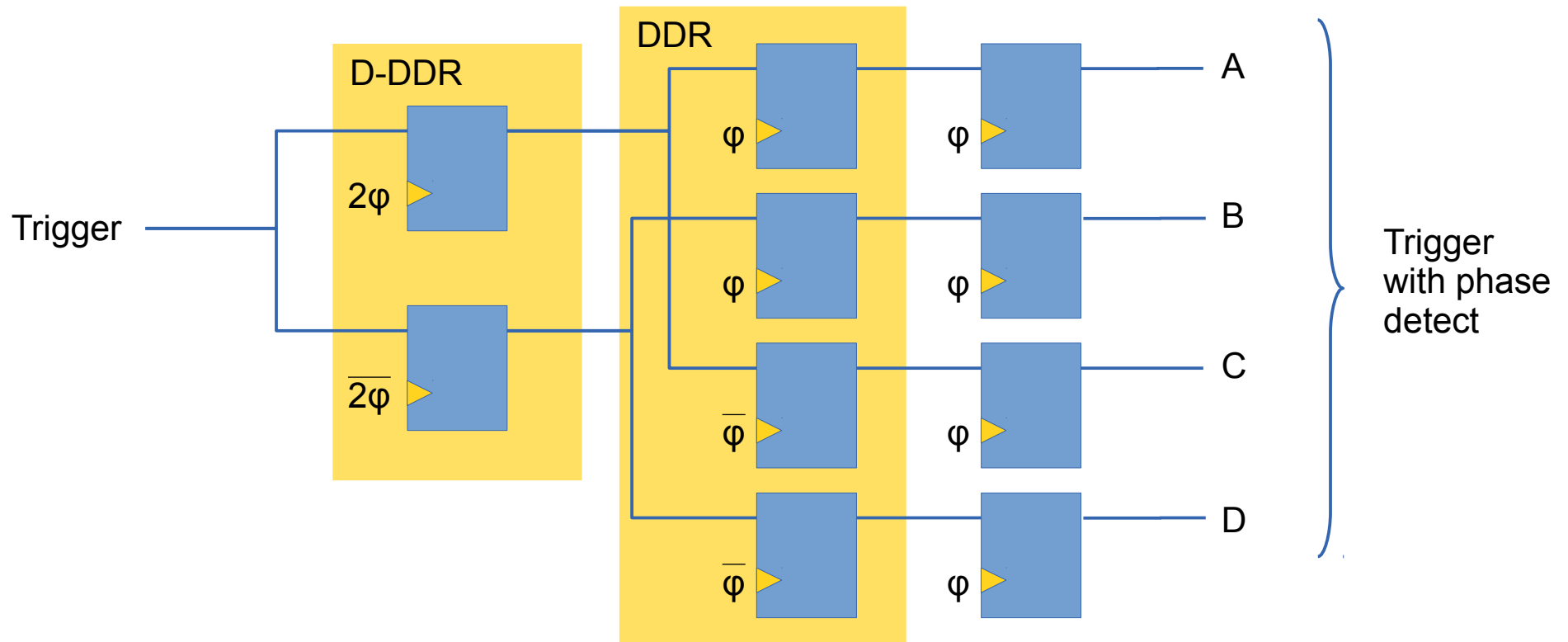




# 2ns Trigger Synchronisation

- FPGA runs with 8ns cycle: 4 machine bunches per tick
- Need to identify which bunch is synchronous with incoming trigger
- Need to identify trigger edge with 2ns precision

# Discovering Trigger Edge



Here we see ABCD go from 0000 to 0011 and can infer precise phase of trigger edge

# FPGA from a software perspective

- You need a helpful FPGA expert! Isa's help and advice was invaluable
- Completely different mind set: everything happens at once, not in sequence
- Open tools are practically nonexistent, entire philosophy is very bound to vendors
- System Verilog a horribly flawed language
- Timing constraints always a perpetual problem
- Solve problems with more pipelining!

# Conclusions

- Developing driver software and FPGA together: very satisfying and flexible
- Many many thanks to:
  - Isa Uzun for first version of FPGA and lots of advice
  - Guenther Rehm for detailed steering, guidance, many ideas, and in-depth evaluation
  - Graham Naylor for initial FPGA design
  - I-Tech for the Libera TMBF platform