

Booster Main Injector Phase Lock Divide-By-32 Phase Error Trajectory Method

March 11, 2013

C. Drennan

Table of Contents

Table of Figures	ii
I. Introduction	3
II. Background and System Specification	3
III. The Proposed Main Injector Phase Lock Controller.....	6
IV. Generation of the Phase Error Reference Trajectory Curve	8
V. Jitter in the starting frequency at the start of MI phase lock	11
V.1 A New Frequency Trigger	11
V.2 Phase Error Reference Trajectory Adjustment	14
VI. Setup for Bench Testing	17
VII. Implementation of the Adaptive Reference Curve	Error! Bookmark not defined.
VIII. Proposal for Online System Testing	21

Table of Figures

Figure II.1 Typical phase detector output	4
Figure II.2 Phase detector output just before beam extraction	5
Figure II.3 Phase error trajectories for the divide-by-32 phase detector	5
Figure III.1 The divide by 32 counters in the FPGA logic.....	6
Figure III.2 Output of the divide-by-32 phase detector	7
Figure III.3 The simplified phase lock controller block diagram	8
Figure IV.1 The initial upper and lower curves used to define the reference trajectory gain curve	9
Figure IV.2 Plots of the warping curve and the weighting “kappa” curve	10
Figure V.1.1 Triggering the MI Phase Lock process.	13
Figure V.2.1 Definition of the Update Interval	14
Figure V.2.2 Calculation of the Update Interval to compensate for the phase error slope	15
Figure V.2.3 Scope screen shot where the curve interval control is not being used.....	16
Figure V.2.4 Scope screen shot where the curve interval control is being used.	16
Figure VI.1 Test Bench Setup	18
Figure VI.2 Photo of the test setup.....	21
Figure VII.1 Simplified System Block Diagram.....	22
Figure VII.2 Shows the use of two Dual Phase Detector modules.....	23
Table VII. Digital Input on the Dual Phase Detector Module	25

I. Introduction

The Main Injector phase lock controls are designed to phase lock the Booster LLRF, and in turn the Booster beam bunches to be delivered to the Main Injector, to the MI RF reference. Phase lock is to be accomplished within a 3 to 4 milli-second interval just before beam extraction from the Booster to the Main Injector. There are significant limits on how we can manipulate the Booster LLRF without inducing synchrotron oscillations in the beam. The AC Damper feedback to the Phase Controller (radial position control loop) provides some dampening of these oscillations; however the induced oscillations must remain sufficiently weak so they can be damped to sufficiently low amplitude before beam leaves the Booster.

This note discusses a variation of the current MI phase lock controls. The basic controller layout is presented. There is a discussion on how the phase error reference trajectory curve was chosen and how it was made to be adjustable online. One source of variation in the system response is discussed and a method of dealing with the variation is proposed. This note also documents details on how the bench testing was carried out.

II. Background and System Specification

The system of electronics that currently implements the MI phase lock was designed and installed in the early 1980's. There are obsolescence issues with the circuit components and some lack of information on the various calibrations and fine compensations that make the system work. There is also a desire to be able to have more access to the systems configuration, to make the phase lock interval shorter, and to reduce the intermittent instabilities that have been observed.

During March and April of 2012 studies with the actual operating Booster were performed to test prototype replacements for the MI phase lock system. Many of the details on triggering and integrating the prototype system alongside the normal, operating system were worked out. A couple different approaches to phase locking the Booster RF to the MI RF were attempted in order to shorten the phase lock interval. It was found, however, that the Booster beam was very sensitive to changes in the RF frequency and would begin to oscillate if the phase error feedback slew rate exceeded a rather low threshold.

From these tests and observations made with the Booster we have some reasonable limits on the acceptable frequency/phase feedback control voltage for the phase lock to avoid inducing excessive oscillations in the beam. The limits on the feedback control voltage (and the corresponding frequency change in the DDS) were determined to be

Control Voltage Peak < 30 milli-Volts (1.5 kHz)

Control Voltage Slew Rate < 50 milli-Volts/milli-second (2.5 kHz / milli-second)

The existing MI phase lock control electronics uses a "divide-by-32" scheme along with a reference phase error trajectory that guides the phase error to zero. The system uses two phase detectors. The first one monitors the wrapping phase between the MI RF and the Booster LLRF. Figure I.1 below shows the typical phase detector output signal.

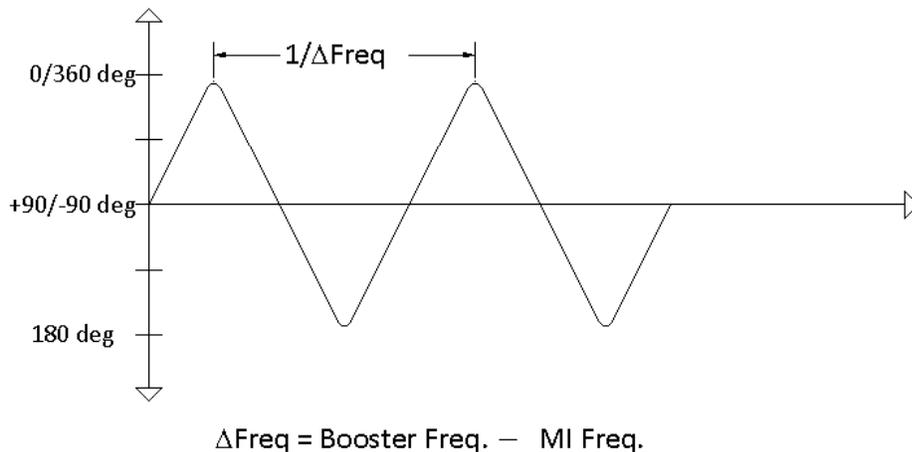


Figure II.1 Typical phase detector output

As the Booster LLRF frequency ramps towards the MI frequency the period of the triangle-wave output of the phase detector increases. Once a pre-determined Booster Frequency in the acceleration frequency ramp has been reached, a trigger is fired indicating the start of the MI phase lock cycle. The first phase detector notes when the slope of this signal switches from positive to negative and resets the divide by 32 high speed ECL counters that divide the Booster RF and the MI RF. The lower frequency outputs of the two counters are then compared using a second phase detector. The frequency difference between these two signals is 32 times smaller and hence the triangle-wave output of this second phase detector has a period that is 32 times longer.

Figure II.2 below shows the phase detector output when the Booster LLRF frequency is 8000 Hz different from the MI RF frequency 4 milli-seconds before extraction and then ramps to become equal to the MI RF frequency after the 4 milli-seconds.

Figure II.3 shows the output of the phase detector whose inputs are the divide-by-32 versions of our RF signals. You can see from the traces that, if we reset the divide-by counters and start the phase lock process at the appropriate frequency offset between the Booster RF and the MI RF, the phase approaches zero in a desirable manner in the normal course of the acceleration (in this case $dF=5600$ Hz). However many factors will cause the phase to deviate from this desired course. Therefore, we provide a reference trajectory, similar to the expected un-regulated phase error trajectory, to drive the phase error to follow the desired path.

Figure II.3 illustrates the variation in the un-regulated phase error trajectory for cases where the frequency difference between the Booster RF and the Main Injector RF varies at the point where the slope on the phase measurement changes.

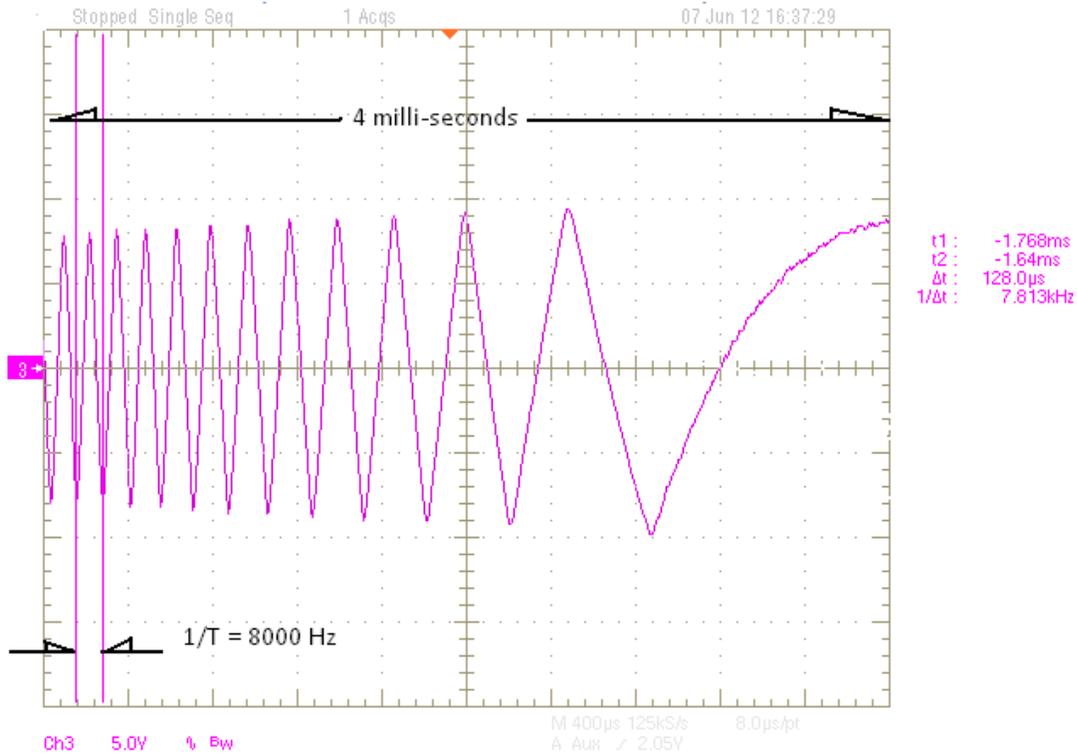


Figure II.2 Phase detector output just before beam extraction

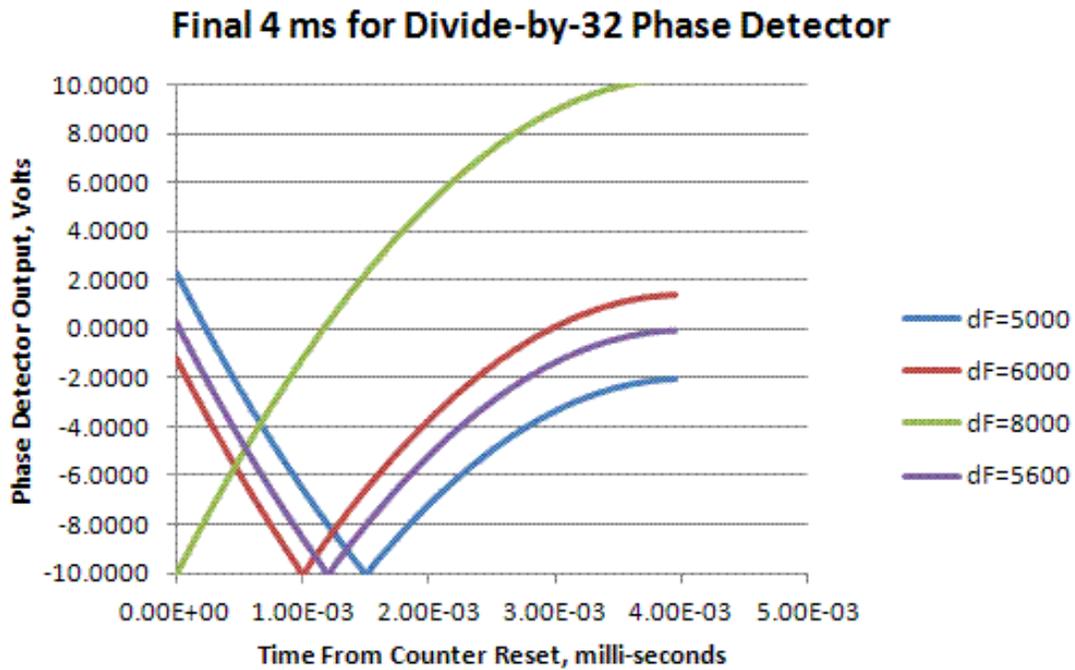


Figure II.3 Phase error trajectories for the divide-by-32 phase detector

III. The Proposed Main Injector Phase Lock Controller

The new version of the MI phase lock controls uses a similar divide-by-32. The new Dual Phase Detector NIM modules were designed and fabricated to be used in this application. This module uses LVDS receivers to square up the Booster and Main Injector RF signals and converts the signals to LVTTTL for input to an FPGA programmable logic chip. The timing logic that starts the MI phase lock process generates a “load counters” signal that holds the counters at zero until the desired time to start. The release of the load signal is synchronized to the RF signals to provide more consistent initial conditions.

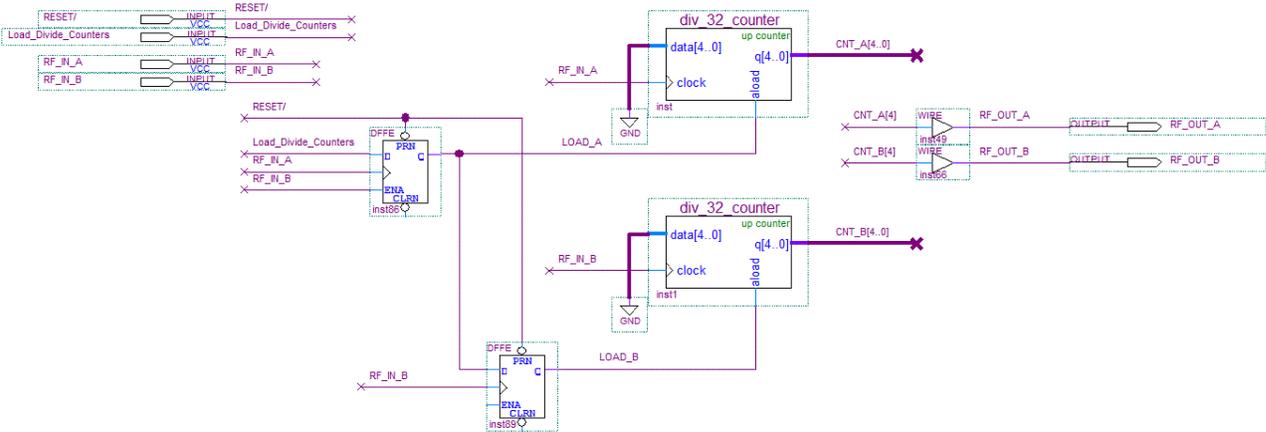


Figure III.1 The divide by 32 counters in the FPGA logic

The RF outputs of the divide by counters are transmitted back out the front panel to be taken back into the module at the phase detector inputs (phase detector B). The phase detector output signal for the divide by 32 RF signals contains both the difference frequency component, $F_{Booster} - F_{MI}$, and also the sum of frequencies term, $F_{Booster} + F_{MI}$, as is common with mixers. At input frequencies of 52MHz the sum of frequencies term was well out of the bandwidth of the following signal conditioning circuits. At the divide by 32 frequencies, around 3.25 MHz, the sum of frequencies term must be deliberately filtered out. This was done by adjusting the value of the filter capacitor on the AD8302 phase detector chip.

There is a transient interval in the divide by phase detector response following the release of the divide by counters. MI phase lock interval timing is setup to begin the reference trajectory and the phase lock feedback after the divide-by counters are started and then at the rising edge of the phase detector output.

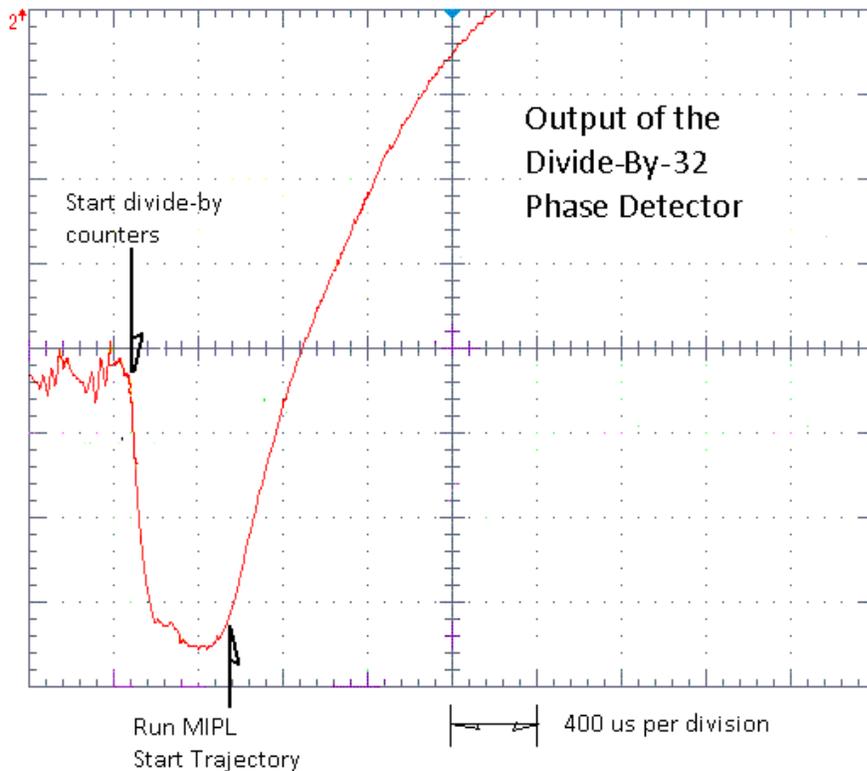


Figure III.2 Output of the divide-by-32 phase detector

The timing sequence that starts the MI phase lock interval is

1. Detect that the LLRF frequency has reach the target starting frequency.
2. Detect the next down slope of the normal RF phase detector, $F_{Booster} - F_{MI}$, and start the divide-by-32 counters.
3. Detect the rising edge of the divide-by-32 phase detector and start the trajectory curve and phase lock feedback.

The control feedback is determined from the difference between the divide-by -32 phase error and a reference trajectory curve. The reference trajectory curve is created using a gain curve from 1.0 to zero, which is applied to the initial phase error value at the start of the MI phase lock interval. That is, an initial phase error value is latched at the start of the interval and this trajectory gain curve drives this value to zero. Throughout the phase lock interval the measured phase error is subtracted from this reference trajectory and this difference is applied to a proportional gain term and an integral gain term. The two products are summed to become the control feedback value. Both the proportional and integral gain values are user adjustable through an MS Windows and USB interface.

The signal Run_MIPL triggers the MI phase lock interval. Note this signal in the controller block diagram, Figure III.3.

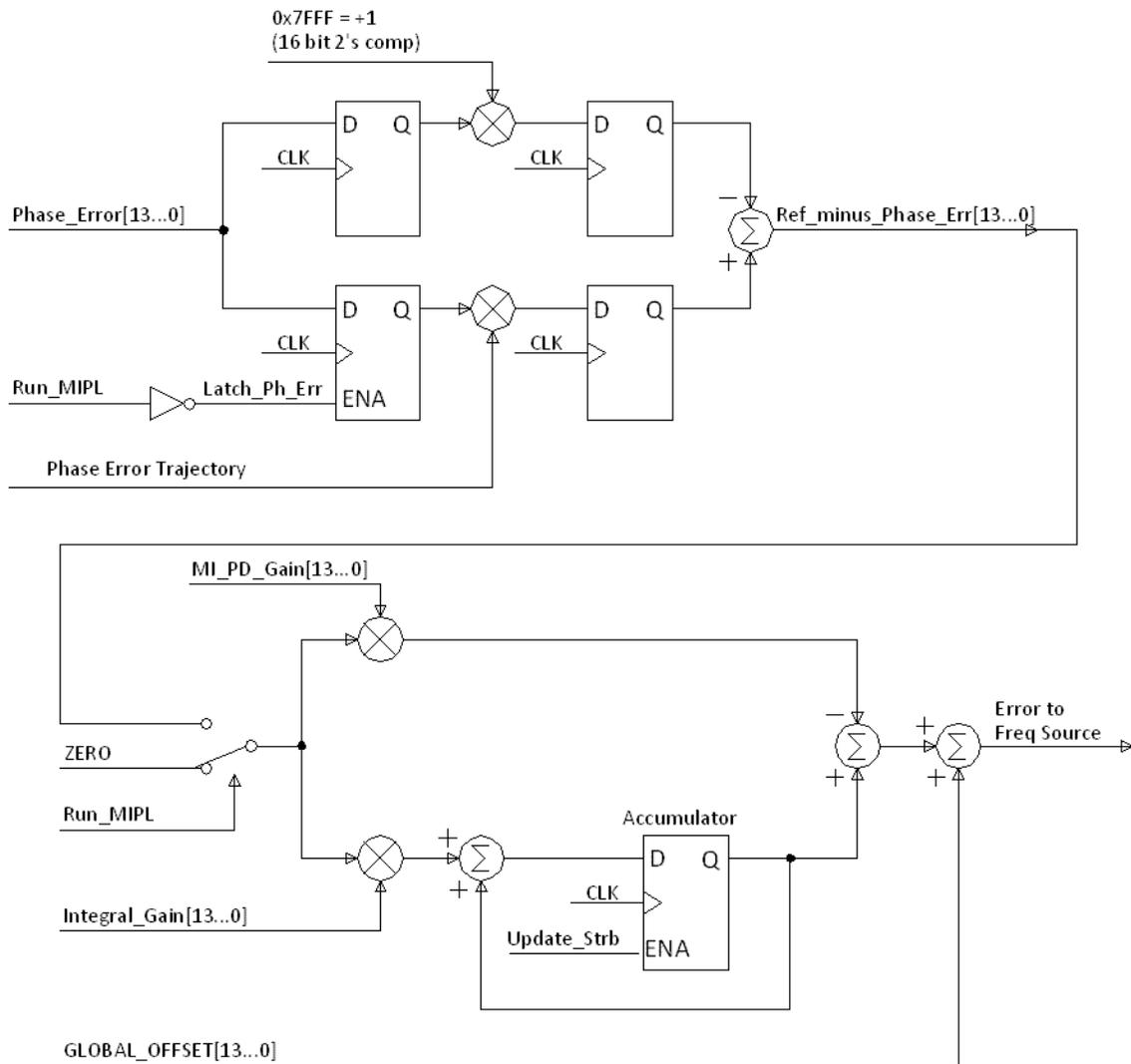


Figure III.3 The simplified phase lock controller block diagram

IV. Generation of the Phase Error Reference Trajectory Curve

Presenting a phase error reference trajectory that guides the divide-by-32 phase error to zero in a short amount of time (< 4 milli-seconds) and results in keeping the control feedback within the tight limits mentioned previously is not a simple matter. The curve needs to be a “natural” exponential type of curve, for which it was found very convenient, if not essential to be able to adjust or flex this curve on-line.

The reference trajectory curve used in the MIPL module is derived from two other curves as a weighted average of the two curves.

$$\text{Trajectory Curve}(n) = \alpha * \text{Upper_curve}(n) + (1 - \alpha) * \text{Lower_curve}(n)$$

The weighting term “alpha” is a settable parameter through the USB interface. The Upper Curve and Lower Curve are initially exponential functions like $f(x) = \text{Exp}(-N/\tau)$, where τ is a time

constant. Each curve is automatically scaled and offset so that the first value of the curve is always 1, and the 1024th value is always zero.

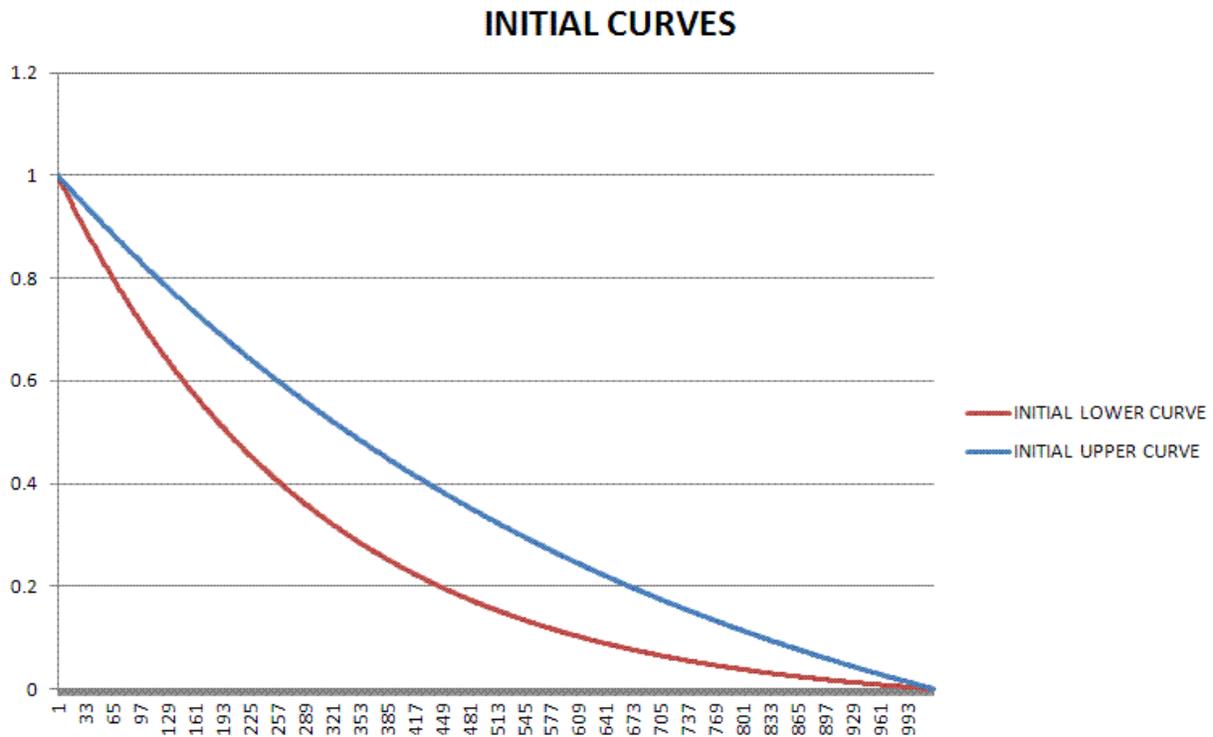


Figure IV.1 The initial upper and lower curves used to define the reference trajectory gain curve

Partly because of the scaling of the curves, the resulting upper and lower curves end fairly abruptly when they reach zero. There is a distinct discontinuity in the slope or d/dt of the curve at zero. This abrupt transition is difficult for the phase lock loop to follow and results in unnecessarily sharp changes in the control voltage at the end of the curve.

To smooth the transition to zero at the end of the curve we computed a weighted average between each curve, the Upper and Lower Curve, and a Warping Curve. The Warping Curve is also an exponential function with a very small time constant. Additionally the weighting term used for this average is not a scalar value but a curve. This "*kappa*" curve is also derived from an exponential function, but has been flip about the axis that connects its endpoints. This weighting curve gives more weight to the warping curve near the end of the curve, where the Upper and Lower curves typically approach zero.

$$\text{Warped Upper Curve}(n) = \text{kappa}(n) * \text{Upper_curve}(n) + (1 - \text{kappa}(n)) * \text{Warping_curve}(n)$$

$$\text{Warped Lower Curve}(n) = \text{kappa}(n) * \text{Lower_curve}(n) + (1 - \text{kappa}(n)) * \text{Warping_curve}(n)$$

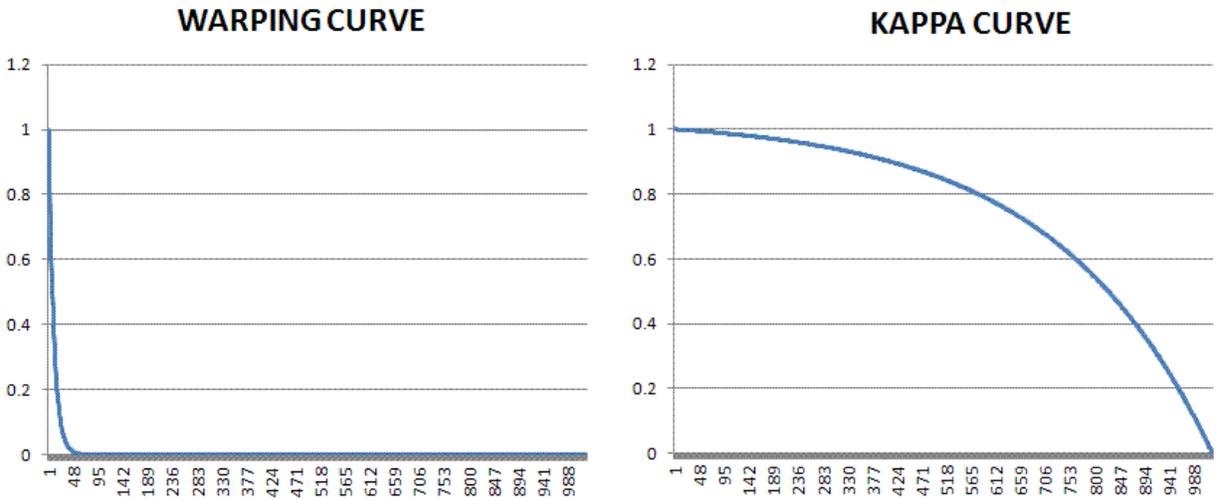


Figure IV.2 Plots of the warping curve and the weighting “kappa” curve

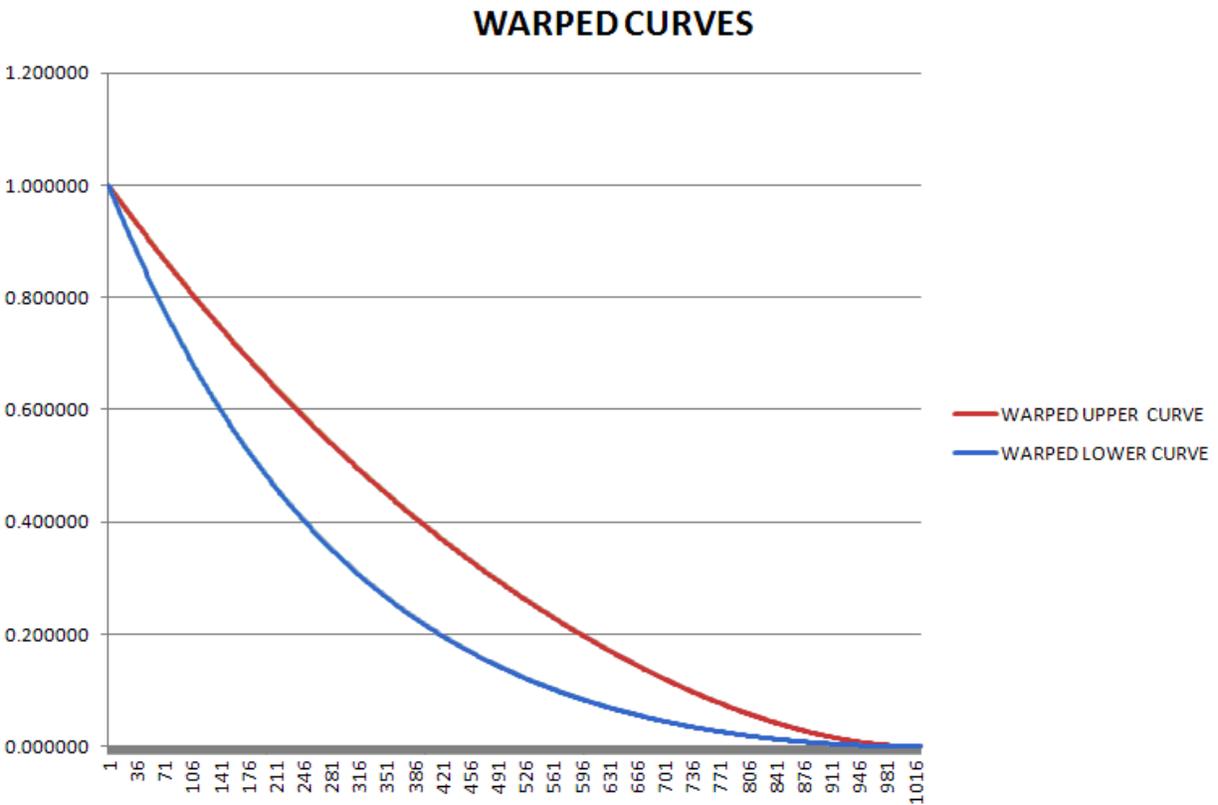


Figure IV.3 Plot of the final warped upper and lower curves that are installed into FPGA memory

The final reference trajectory curve is then the weighted average of the warped upper curve and the warped lower curve. The weighted average is done on-line in real time with the weighted upper and lower curves read from ROM memory in the FPGA, and the weighting term “alpha” is a user settable parameter through the MS Windows / USB interface.

$$\text{Trajectory Curve}(n) = \alpha * \text{Warped_Upper_Crv}(n) + (1 - \alpha) * \text{Warped_Lower_Crv}(n)$$

V. Jitter in the Starting Frequency at the Start of MI Phase Lock

Recall that in order to minimize the control manipulation required to phase lock Booster RF to MI RF, we have to provide a reference phase error trajectory that varies from the un-regulated phase error trajectory by a minimal amount. The curvature/slope of the un-regulated phase error trajectory varies depending on the Booster frequency at the start of the MI phase lock process (Run_MIPL).

The frequency of the Booster RF at Run_MIPL can vary due to the need to synchronize the phase lock control process with a down-slope of the phase error triangle wave before releasing the divide by 32 counters. The wrapping phase error triangle wave is not entirely synchronous to the frequency curve. For example, when the frequency curve reaches within 8k Hz of the MI RF frequency we begin looking for the next down-slope pulse before engaging the divide-by counters. When reaching this target frequency the down-slope pulse may follow immediately, or it may have just occurred and we will have to wait a full period of the 8k Hz triangle wave, 125 micro-seconds. A frequency curve that is changing 8k Hz in 4 milli-seconds can change as much as 250 Hz in this time. This does result in variation in curvature of the un-regulated phase error trajectory.

These variations in the curvature of the phase error versus a fixed slope reference phase error trajectory will result in control feedback that exceeds the beam limitations of keeping slew rates less than 50 Volts/second.

V.1 A New Frequency Trigger

So the current (normal) system for MI phase lock gets a trigger from the VXI-DDS LLRF Source when the sum of the Frequency Curve plus the Acceleration Phase Lock phase error reaches 52.8023 MHz, approximately 8 kHz below the MI RF frequency. Then the next down-slope of the phase error triangle wave triggers the start of the MIPL feedback. This process is illustrated in the top portion of Figure V.1.1. Note that the frequency can occur anywhere in the shaded region with respect to the wrapping phase signal. Since the Booster frequency continues to change, following the LLRF Frequency Curve, the difference in frequency between Booster and MI RF at the start of MI phase lock will vary. How much it varies depends on the delay between the frequency trigger and the Down Slope Pulse trigger (DSP output of the Timing Generator Module).

In order to adjust the MI phase lock trigger frequency in the “Test” system independent of the trigger frequency in the “Normal” system, the frequency difference between the MI and Booster RF is measured as a number of logic clock counts between zero crossings of the phase error triangle wave. The count is compared to a user programmed threshold corresponding to the target LLRF frequency.

The relationship between the counts threshold value and the related frequency difference threshold is

$$Counts_{Thresh} \left[\frac{clocks}{period} \right] = 80E6 \left[\frac{clocks}{second} \right] * (\Delta freq_{Thresh})^{-1} \left[\frac{second}{period} \right]$$

The lower portion of Figure V.1.1 illustrates the delays between the occurrence of the target frequency trigger at the VXI-DDS LLRF Source and the Down Slope trigger that starts the MI phase lock process. The count threshold is checked twice per period of the phase error triangle wave. Delay 2 is assumed to be a fixed interval until the Down Slope Trigger occurs, and is incurred if an over threshold condition is detected at Test 'A'. Likewise, if the over threshold condition is detected at Test 'B', Delay 1 is the interval before the trigger. These delays are expected to be constant for a given trigger LLRF frequency and LLRF frequency curve, following the trigger frequency.

In the controller logic, and in the figure, we can distinguish whether the over threshold condition was detected by Test 'A' or Test 'B'. If Test 'A' saw the over threshold condition, the logic will see the Up Slope indication before the Down Slope Trigger.

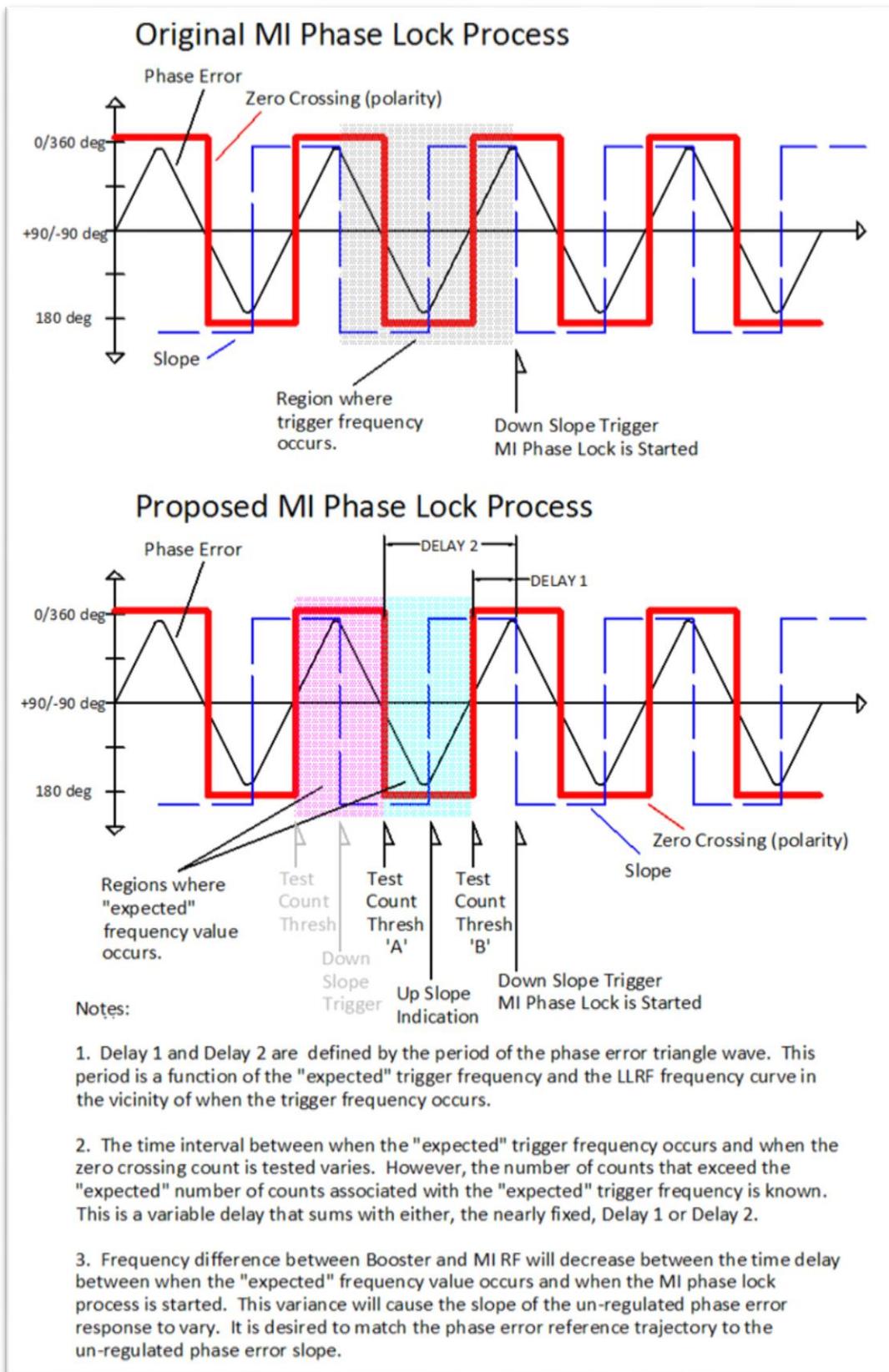


Figure V.1.1 Triggering the MI Phase Lock process.

V.2 Phase Error Reference Trajectory Adjustment

In order to minimize the control feedback required to phase lock the Booster RF to the MI RF we need to present a Phase Error Reference Trajectory that is similar in slope to the un-regulated response.

The slope of the Phase Error Reference Trajectory can be adjusted by changing the number of logic clocks that exist between updates of the curve values from memory.

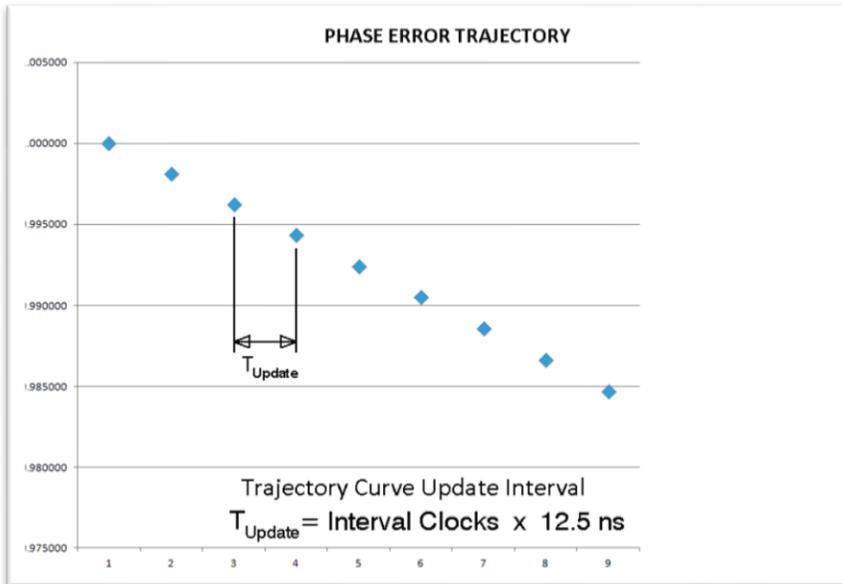


Figure V.2.1 Definition of the Update Interval

There is a variable delay between the occurrence of the target frequency and the occurrence of the Down Slope Trigger. The longer the delay, the closer the Booster RF frequency is to the MI RF frequency when we start the phase lock process, and the lower the slope of the phase error trajectory.

Using the value of the zero crossing count when the over threshold is detected, noting whether Test 'A' or Test 'B' detected the over threshold, and providing some user programmable parameters, we can adjust the update rate of the Phase Error Reference Trajectory to more closely match the un-regulated phase error response and minimize the control feedback necessary for phase lock. Figure V.2.2 illustrates the calculation of the Update Interval. We will use the "Additional Offset" if the over threshold condition was detected in Test 'A'.

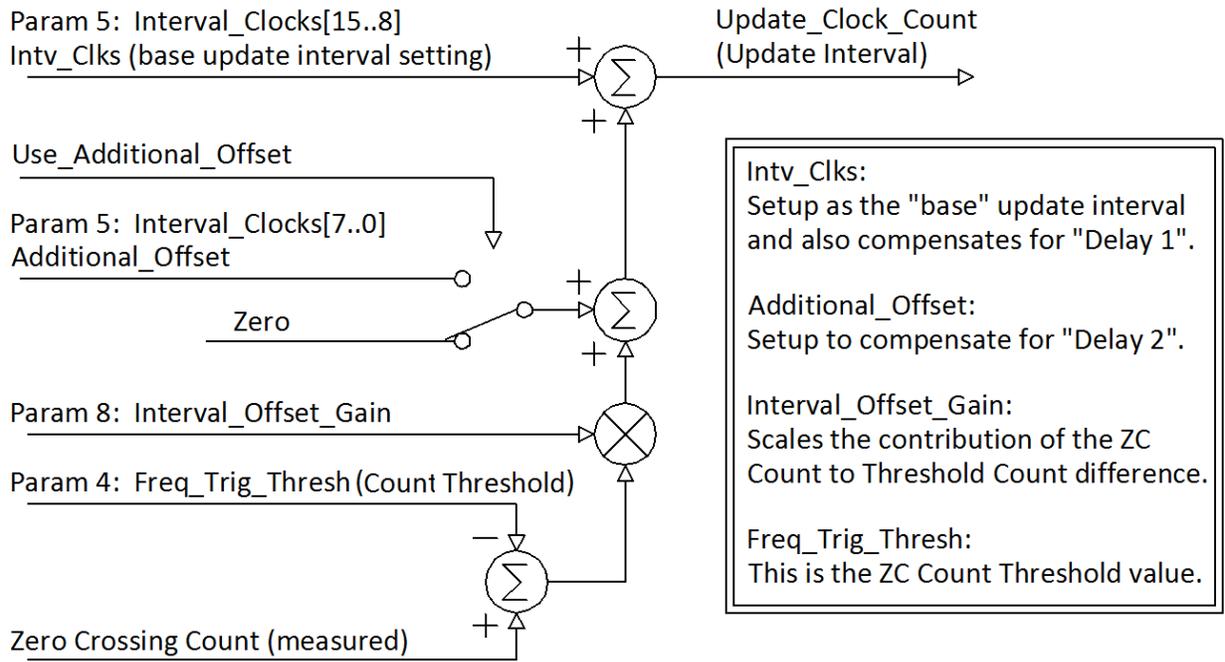


Figure V.2.2 Calculation of the Update Interval to compensate for the phase error slope

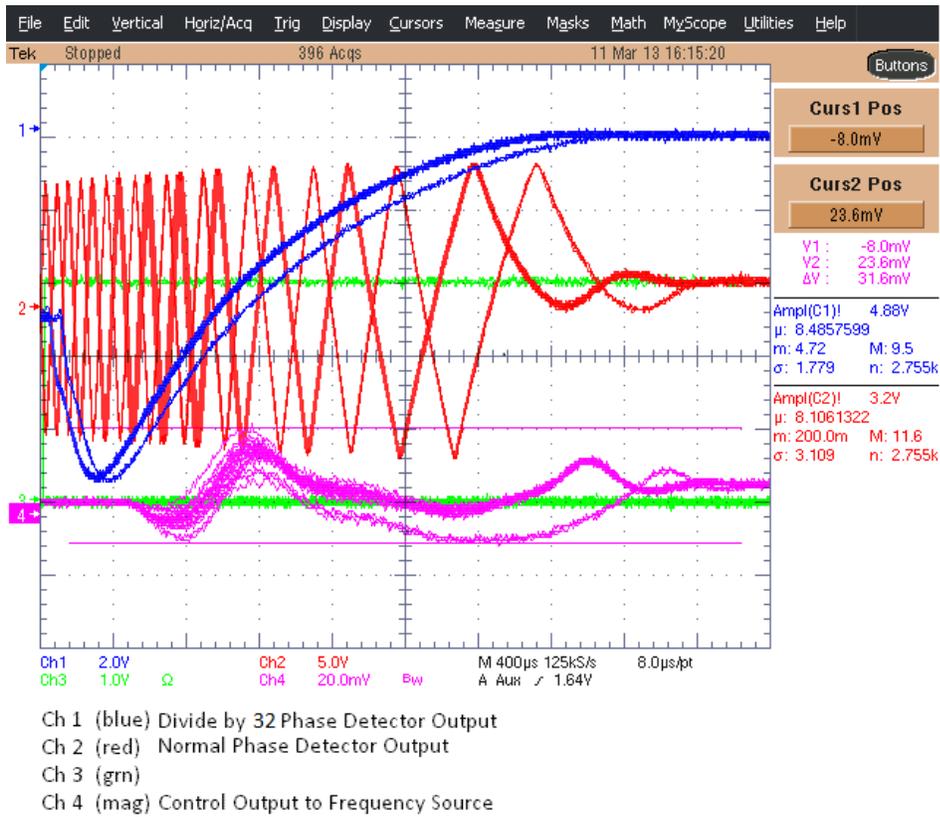


Figure V.2.3 Scope screen shot where the curve interval control is not being used.

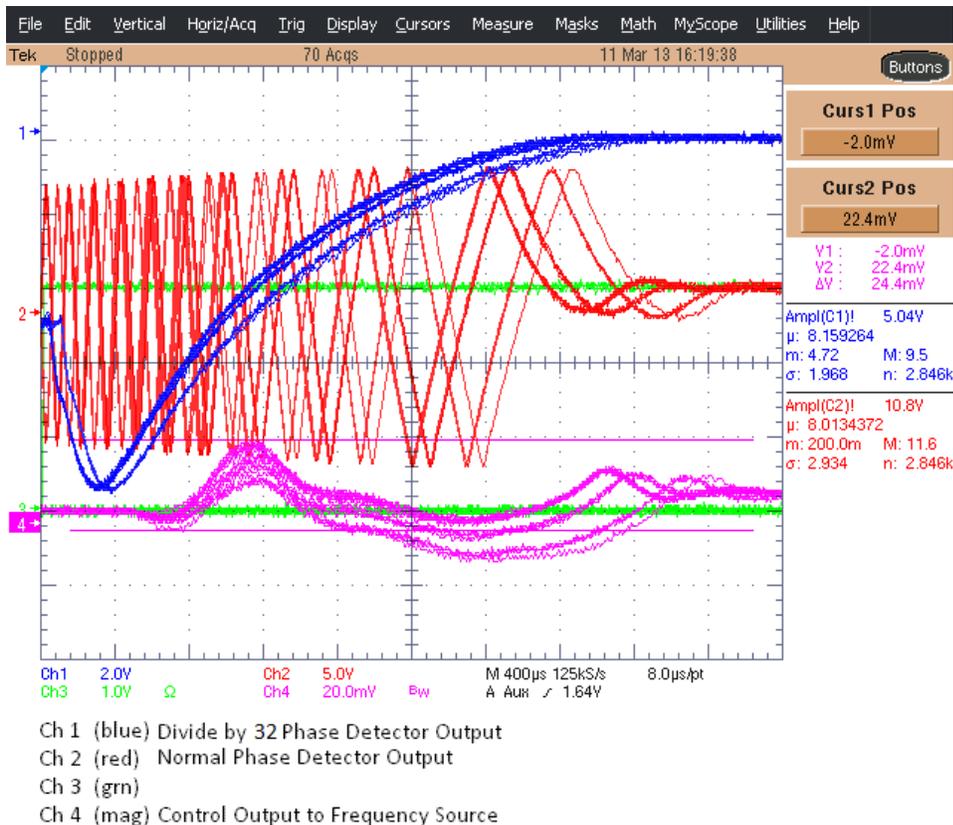


Figure V.2.4 Scope screen shot where the curve interval control is being used.

VI. Setup for Bench Testing

For testing on the bench we use the HP 8657B RF Generator. We set it up as the Booster LLRF at approximately 50 MHz with FM modulation. The phase control feedback and a bias curve that acts as the frequency curve are summed to manipulate the sources frequency at 50 kHz/Volt (or 1 kHz / 20 milli-Volt).

The frequency curve signal is intended to sweep the RF generator 8 kHz in the last 4 milli-seconds of what would be the Booster acceleration cycle. This provides the essential “base” phase response between the Booster RF and the MI RF.

The frequency curve is generated by a specially programmed Dual Phase Detector Module. This module takes the control feedback from the phase controller module and sums it with a frequency curve it is playing out from memory in its FPGA. USB interface parameters for this special Frequency Curve Module can be set to control the Frequency Curve Scale (Parameter 9) and the time duration of the curve (Parameter 5). The Phase Controller feedback is applied into the External Input (front panel AUX) and is summed with the frequency curve at the output summing amplifier.

The frequency curve can be setup by removing the control feedback and monitoring the effect of the curve on the HP8657B RF generator output by looking at the output of the normal RF phase detector, $F_{Booster} - F_{MI}$, output. With the Frequency Curve Trigger as a time reference, the period of the phase detector triangle wave output at different time offsets from the trigger can be made. The magnitude of the final frequency change can be adjusted with USB Interface parameter 9, and the time between the Frequency Curve Trigger and the final frequency value can be adjusted with parameter 5. The range of frequencies generated also depends on the base frequency setting of the RF generator.

The Tektronix AWG520 is a two channel arbitrary waveform generator setup to produce two identical 50 MHz sine waves. By using one channel for the normal phase detector comparison and the other channel as input to the divide-by-16 counters and phase detector, we can adjust the phase between these two references as we could with cable length adjustments. The phase between these can be adjusted to produce a final phase difference between the Booster RF and MI RF near zero.

Figure V.1 shows the cabling of the signal generators and the two NIM modules. An additional trigger input is necessary for the Phase Controller Module at T1 and T3 and the Frequency Curve Module at T4. This is a 10 ms TTL level pulse that occurs at a reasonable rate, between 15 Hz and 0.5 Hz.

When seeing a rising edge at T3, the FPGA code for the Phase controller fixes the input switches for the output summing amplifier so that only the FPGA DAC output is connected. All other switches are open. For the Frequency Curve Module the FPGA DAC and the External Input signals are connected and the other two switches are open.

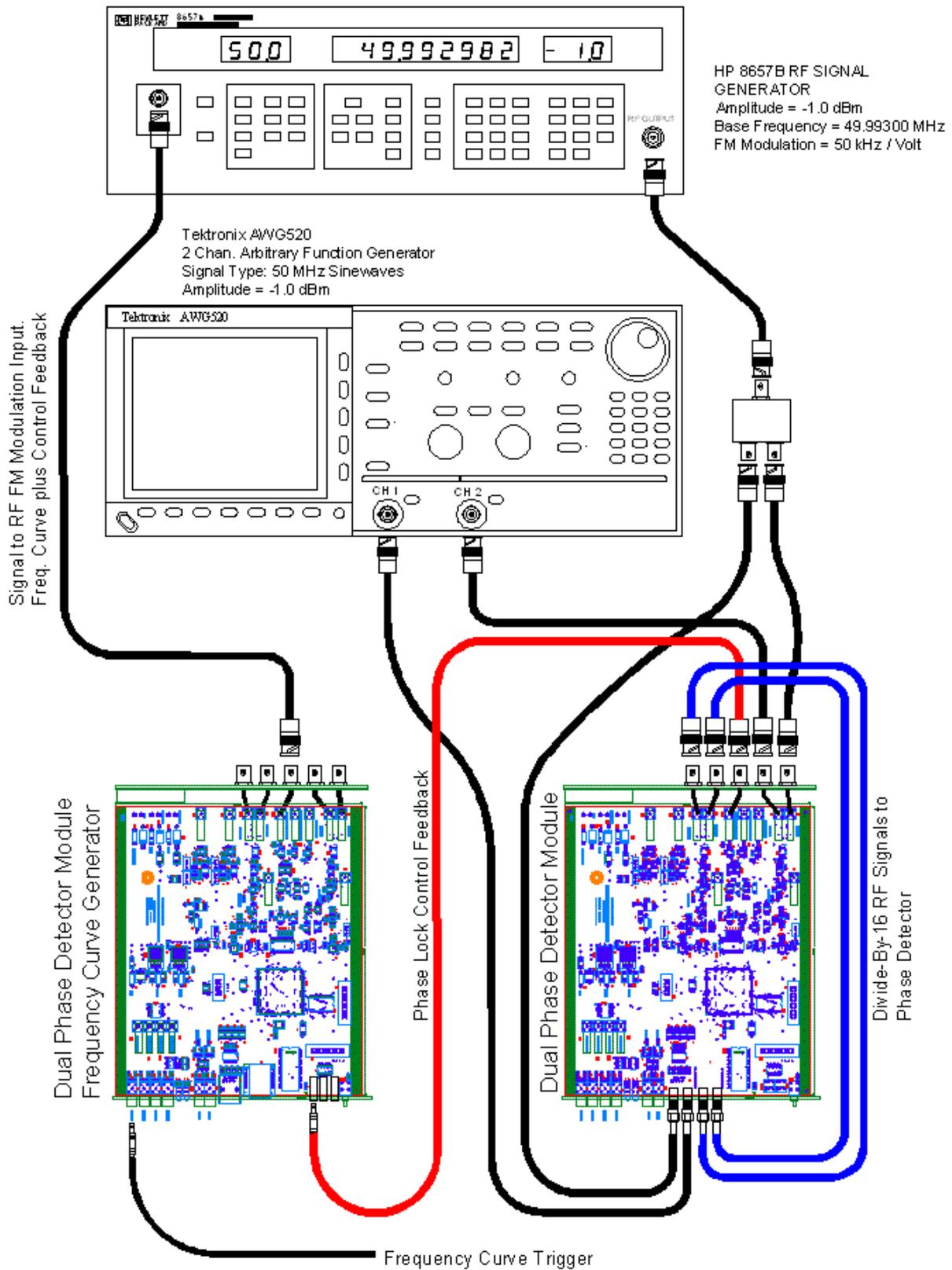


Figure VI.1 Test Bench Setup

Table VI.1 USB Interface Parameters as they apply to the MIPL Phase Controller module

Param#	Typical Value In Hex	Number Format	Description (Phase Controller Module)
0	0x0800	12 Bit Offset Binary Value (0x0800 = zero)	BRF to MI RF PD Gain (NOT USED)
1	0x0800	12 Bit Offset Binary Value (0x0800 = zero)	BRF to MI RF PD Offset
2	0x0E00	14 Bit Offset Binary Value (0x2000 = zero)	div-by 32 MI PD Proportional Controller Gain
3	0x0740	12 Bit Offset Binary Value (0x0800 = zero)	div-by 32 MI PD Offset
4	0x1180	16 Bit Unsigned Integer	Internal Frequency Trigger Threshold. Number of clocks between zero crossings of the normal PD triangle wave output.
5	0x9020	Dual 8 Bit Unsigned Integer	Bits [15..8] MIPL Transition Interval. Number of 12.5 ns clocks between memory curve updates. Bits [7..0] Alternate Slope Offset. This is the base offset that is applied to the MIPL transition interval if the slowest set of reference trajectories is desired. See the explanation on the adaptive reference curve, Section V.2.
6	0x1F40	14 Bit Offset Binary Value (0x2000 = zero)	div-by 32 MI PD Controller Integral Gain
7	0xA000	16 Bit Unsigned Integer	MODE Control Word
8	0x2A00	14 Bit Offset Binary Value (0x2000 = zero)	Interval Offset Gain. This is a gain between [+1 ... -1] that is applied to the difference between the number of zero crossing counts in the PD zero crossings that triggered the Internal Frequency Trigger and the Frequency Trigger Threshold setting. See the explanation on the adaptive reference curve, Section V.2.
9	0x2000	14 Bit Offset Binary Value (0x2000 = zero)	Global FPGA DAC Output Offset
10	0x2800	16 Bit Unsigned Integers	Filter delay term for the slope indication on the Divide-By PD. Value is the number of 12.5 ns clocks that the slope state signal must be stable for before a change of state (up, down) is registered.
11	0xB000	16 Bit Offset Binary Value (0x8000 = zero)	Curve Weighting Factor, Alpha. Used to create the reference trajectory gain curve from the Upper and Lower curve memory values. Values below 0x8000 (zero) do not make sense.

Table VI.2 USB Interface Parameters as they apply to the Frequency Source module

Param#	Typical Value In Hex	Description (Phase Controller Module)	Description (Freq. Source Module)
0	0x0800	12 Bit Offset Binary Value (0x0800 = zero)	Accel PD Gain (NOT USED)
1	0x0800	12 Bit Offset Binary Value (0x0800 = zero)	Accel PD Offset (NOT USED)
2	0x2000	14 Bit Offset Binary Value (0x2000 = zero)	MI PD Controller Gain (Set to zero)
3	0x0800	12 Bit Offset Binary Value (0x0800 = zero)	MI PD Offset (Set to zero)
4	0x0000	16 Bit Unsigned Integer	Internal Frequency Trigger Threshold (NOT USED)
5	0x0500	16 Bit Unsigned Integer	Freq Curve Duration Setting. Number of 12.5 ns clocks between memory curve updates.
6	0x2000	14 Bit Offset Binary Value (0x2000 = zero)	MI PD Controller Integral Gain (Set to zero)
7	0xA000	16 Bit Unsigned Integer	MODE Control Word
8	0x0020	14 Bit Offset Binary Value (0x2000 = zero)	Frequency Curve Trigger Delay
9	0x2308	14 Bit Offset Binary Value (0x2000 = zero)	Frequency Curve Amplitude Adjustment.
10	0x0604	Dual 8 Bit Unsigned Integers	Bits [15..8] Filter delay term for the slope (NOT USED) Bit[7..0] Integral controller gain stage rate adjustment. (NOT USED)
11	0x3FFF	14 Bit Offset Binary Value (0x2000 = zero)	(NOT USED).

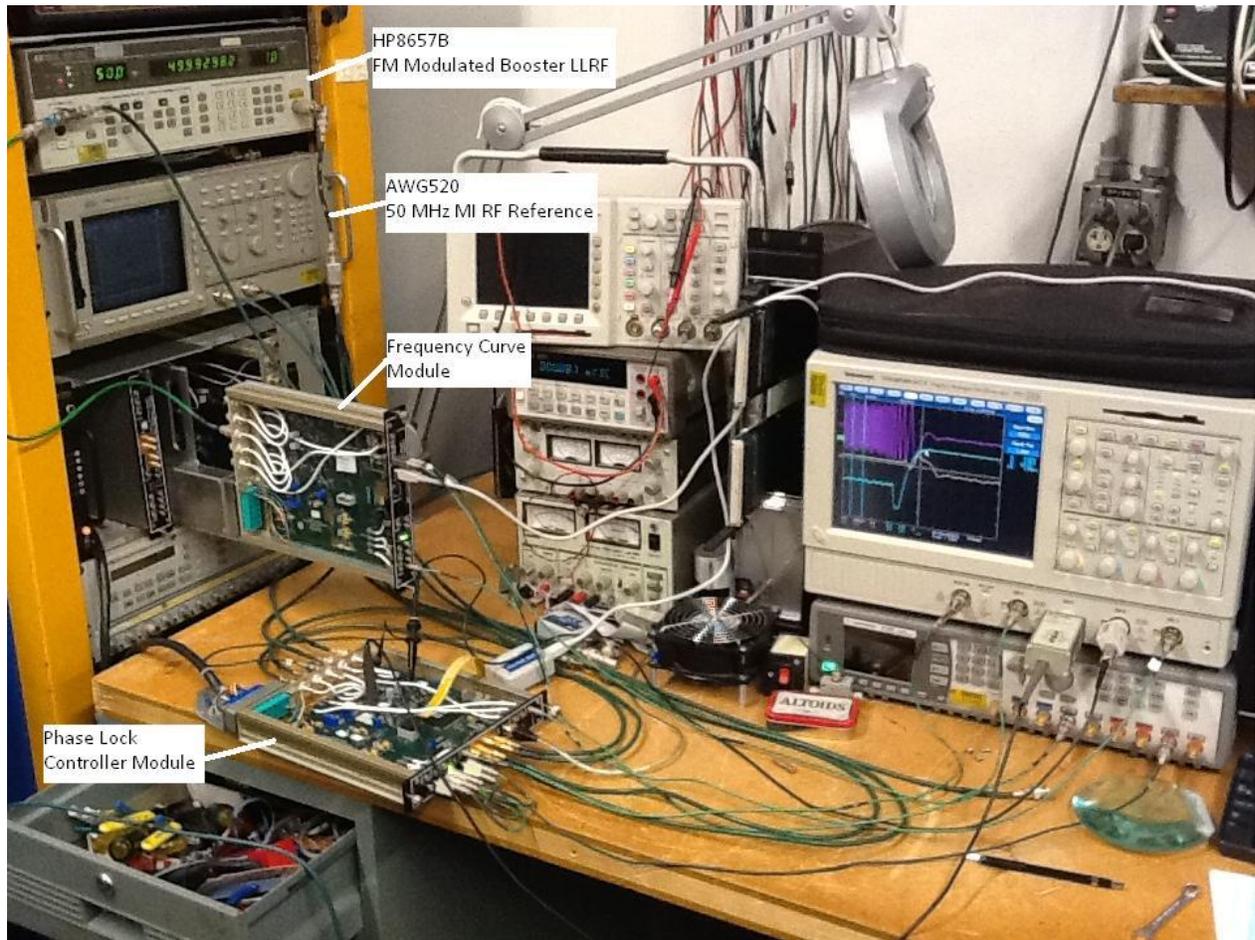


Figure VI.2 Photo of the test setup

VII. Proposal for Online System Testing

For testing the new MI phase locking method and electronics with the Booster and actual beam, we need to be able to switch between the “Normal” system and the “Test” system. Figure VII.1 is a simplified block diagram indicating where the switches appear. The first switch allows switching between the “Normal” acceleration phase lock phase error feedback out of the Mode Controller and the “Test” phase error feedback generated by the Prototype Dual Phase Detector Module. Both the normal and the test error feedback passes through the Program Generator so that both signals are treated the same way during the normal MI phase lock that is managed by the Program Generator.

TeV Clock, Booster Reset events are chosen for the CAMAC 377 timer trigger signal that controls the switch. The ACNET timer trigger that controls the Acceleration Phase Lock Normal/Test Switch is B:PHDSW. This trigger will fire just before the Booster cycle begins and the switch will connect the “Test” system in to control the Booster. The switch will switch back to the “Normal” position after 40 ms.

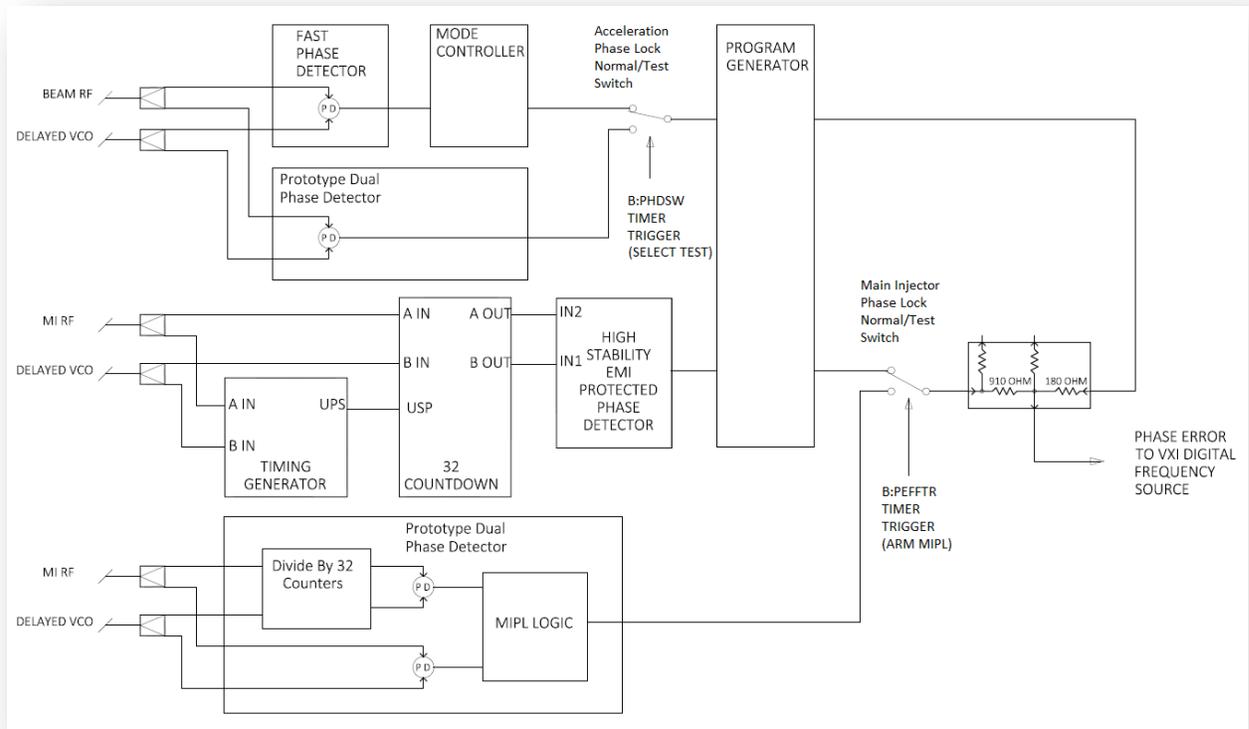


Figure VII.1 Simplified System Block Diagram

The second switch allows switching between the “Normal” MI phase lock system and the “Test” MI phase lock system. This is affected by the ACNET timer trigger device B: PEFFTR. When using the “Test” MI phase lock system we also must be using the “Test” acceleration phase lock system. In addition to switching the MI phase error feedback signal we also must prevent the “Normal” MI phase lock system from triggering so that the acceleration phase error feedback signal is not affected by the Program Generator. The acceleration phase error signal passes straight through the Program Generator for the whole cycle in this mode.

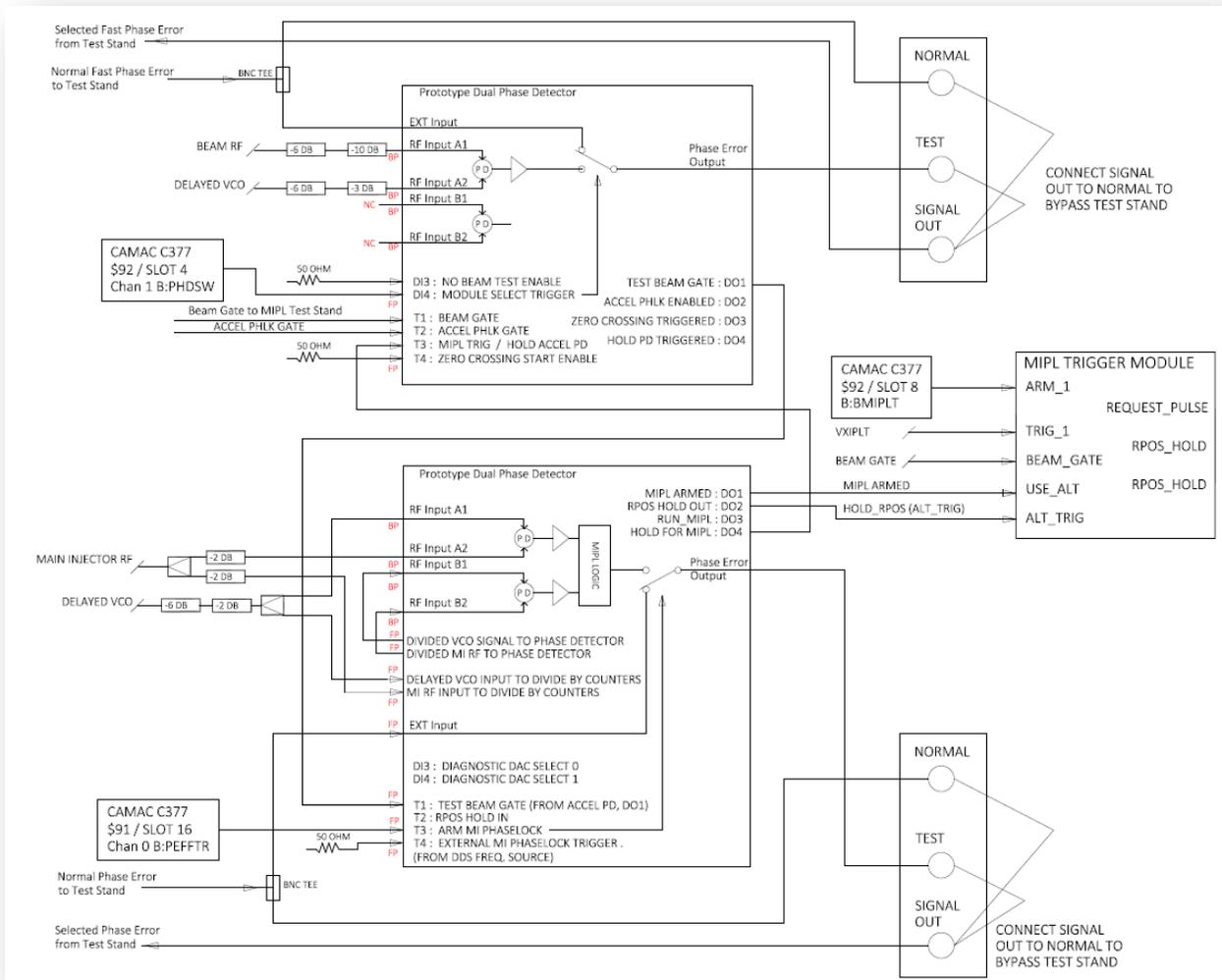


Figure VII.2 Shows the configuration and cabling of two Dual Phase Detector modules.

In Figure VII.2 shows more of the details of how the “Test” system is tied in with the “Normal” system. One of the important features is to note is how the signals “Normal Fast Phase Error” and the “Normal MIPL Phase Error” are split with a BNC Tee to bring one copy to the Prototype Dual Phase Detector to be switched by the appropriate ACNET timer trigger, and a second copy is brought over to a test stand bypass panel so that the entire test system can be easily bypassed. Figure VII.3 is a photo of the bypass panels shown in the bypass mode. To connect the Test system in the BNC connections are moved to the port labeled “TEST”.



Figure VII.3 The Test system bypass panels shown in the bypass mode.

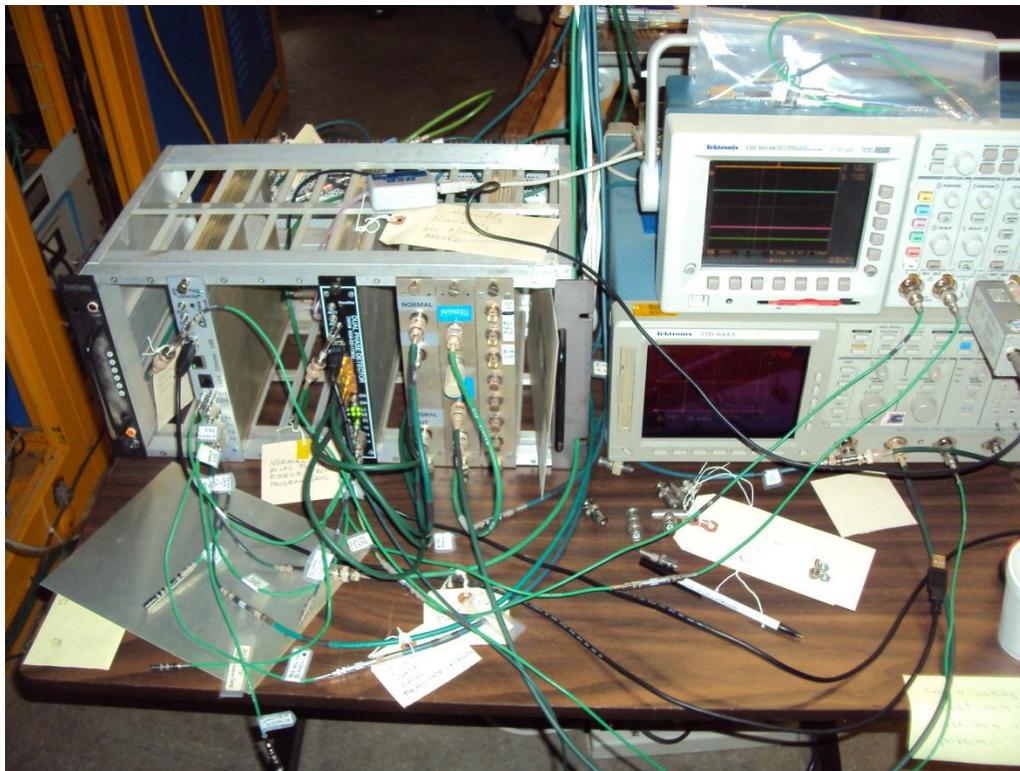


Figure VII.4 Photo of the Test system behind the BLLRF racks.

Table VII. Digital Input on the Acceleration Phase Lock, Dual Phase Detector Module

Front Panel Input	FPGA Signal Name	Description
DI3	Booster_Test_Event	This is a select Booster reset (TCLK) trigger that will generate a substitute “Test Beam Gate” out of the module to gate the LLRF Frequency Source when no beam is present. It does this by outputting the received ACCEL_PHLK_GATE as the Beam Gate. When this trigger does not pulse, the Test Beam Gate is a copy of the BEAM_GATE_IN. This function was created for testing phase lock with the Frequency Source without beam in the Booster.
DI4	Module_Select_Trigger	This is the Booster reset trigger that selects the test system to control acceleration phase lock and MI phase lock. When it does not pulse before a Booster cycle the normal LLRF system phase error signals are used.
T1	BEAM_GATE_IN	This gate is high when beam is in the Booster and gates output from the acceleration phase detector and is a requirement for the MI phase lock to run.
T2	ACCEL_PHLK_GATE	This gate is also required for acceleration phase lock and MI phase lock to be enabled. This signal turns on after an operator settable delay (B:VPLON) and turns off after another settable delay (B:VPLOFF).
T3	ARM_MI_PHASELOCK	This is a select Booster reset (TCLK) trigger that arms the MI phase lock “test” system to be triggered by MI_PHASELOCK_TRIGGER.
T4	MI_PHASELOCK_TRIGGER	This is the trigger from the LLRF Frequency Source that fires when the LLRF frequency reaches a specified value to start the MI phase lock process.

VII.1 MIPL Trigger Module

A new module was put together to replace the existing “Gate Sync Module”. The normal function of the Gate Sync Module is to allow the MI Phase Lock system to be armed by the ACNET timer trigger B:BMIPLT. Once armed, the frequency trigger VXIPLT out of the LLRF DDS-VCO VXI module will generate the “Request Pulse” to the Timing Generator which starts the “Normal” MI phase lock process.

The replacement module is called the MIPL Trigger Module. It allows switching between two modes of operation, Figure VII.2. The first mode is the normal mode which performs just as the Gate Sync Module did. The second mode of operation set by making the signal USE_ALT active (high) prevents the Request Pulse from being generated and additionally provides the HOLD_RPOS signal that in the normal mode is generated by the Program Generator in the normal MI phase lock system.

Note that in the normal system, the ARM 1 signal B:BMIPLT occurs only a few milliseconds before the Frequency Trigger is expected and hence the 10 ms is sufficiently long to wait for the TRIG 1 (VXIPLT) to arrive.

