

# Booster to Main Injector Phase Lock Controller

---

July 20, 2015

C. Drennan

## Table of Contents

Table of Figures .....	ii
I. Introduction .....	1
II. History and System Development.....	1
III. Basics of the Divide by 32 Phase Lock Method.....	3
III.1 Motivation for the Chosen Method.....	3
III.2 MI Phase Lock Timing.....	3
III.3 MI Phase Lock Control Feedback Generation.....	8
IV. Generation of the Phase Error Reference Trajectory Curve.....	10
V. Jitter in the Starting Frequency at the Start of MI Phase Lock .....	12
V.1 Frequency Trigger Details .....	13
V.2 Phase Error Reference Trajectory Adjustment .....	15
VI. Extraction Notch Adjustment (still under study, October 2015) .....	16
VII. Setup for Bench Testing .....	19

## Table of Figures

Figure III.2.1 Typical phase detector output.....	3
Table III.2.1 Logic signals used and/or produced by the MI phase lock process.....	4
Figure III.2.2 Typical start of MI Phase Lock scope screen capture. ....	5
Figure III.2.3 Start of MIPL where the use of the alternate reference trajectory slope is signaled.....	6
Figure III.2.4 The logic schematic of the divide by 32 counters.....	7
Figure III.2.5 Phase error trajectories for the divide-by-32 phase detector .....	7
Figure III.2.6 A scope screen capture of a full MIPL interval done in the lab, on the bench. ....	8
Figure III.3.1 The simplified phase lock controller block diagram .....	9
Figure IV.1 The initial upper and lower curves used to define the reference trajectory gain curve .....	10
Figure IV.2 Plots of the warping curve and the weighting “kappa” curve.....	11
Figure V.1.1 Triggering the MI Phase Lock process. ....	14
Figure V.2.1 Definition of the Update Interval .....	15
Figure V.2.2 Calculation of the Update Interval to compensate for the phase error slope .....	16
Figure VI.1 MI phase lock with an offset in the reference trajectory for plus RF Bucket correction. ....	17
Figure VI.2 MI phase lock with an offset in the reference trajectory for no RF Bucket correction.....	17
Figure VI.3 MI phase lock with an offset in the reference trajectory for minus RF Bucket correction. ....	18
Figure VII.1 Test Bench Setup .....	20
Table VII.1 USB Interface Parameters as they apply to the MIPL Phase Controller module .....	21
Table VII.2 USB Interface Parameters as they apply to the Frequency Source module.....	23
Table VII.3 Test Interface selection of different IO options using Parameter 7. ....	24

## 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.

In order to phase lock in the time allowed and minimize oscillations in the Booster a phase difference, reference trajectory method is used. This method and the associated electronics are discussed in this document.

## II. History and System Development

The system of electronics that implemented the MI phase lock from the early 1980's until January of 2014. The electronic components for this system had significant obsolescence issues and there was a lack of knowledge with regard to the various calibrations and fine compensations that made the system work.

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.

By February of 2013 it was determined that the divide by 32 phase error reference trajectory method used by the original electronics was as good or better a method as any we could achieve within the scope of the modifications and testing methods Booster operations would allow.

A four channel DDS RF generator module prototype had been developed, in 2009 and 2010 that could adjust both the frequency of the RF output signals and the phase of the RF signal independently. It is expected that this approach could provide the means to quickly and gently perform the phase lock. However, this would also replace a larger portion of the Booster LLRF controls than the Booster department was comfortable with at the time. Trying to splice candidate electronics into the delicately balanced Booster LLRF while it is running 24 hours / 7 days a week is very challenging. Replacing only the MI phase lock electronics and testing and developing the new modules in parallel with the operational system, had a smaller impact on Booster operations.

In by the end of 2012, one of the new Dual Phase Detector NIM modules was adapted and programmed to implement the divide by 32 phase error reference trajectory method [reference]. By February of

2013 the module was spliced into the Booster LLRF for performing studies with actual beam extracted to the 8 GeV dump. A new Acceleration Phase Lock phase detector was needed to work in conjunction with the MI Phase Lock controller. The Acceleration Phase Lock phase detector is used in the basic control loop that maintains the phase lock between the Booster beam RF and the DDS (VCO) LLRF reference. This module was tested in parallel with the existing Booster controls through the summer and by fall of 2013 testing began with the new MI phase lock system. By December, beam phase locked by the new system was being delivered to the MI and evaluated.

In January of 2014, the new MI phase lock and new Acceleration phase detector electronics was switched in to run the Booster LLRF.

The new MI phase lock system provided a small improvement in the variation in the phase of the beam injected into the MI. The larger benefit at this point was that we had a system that could be supported and we could investigate methods to reduce extraction beam losses and the MI injection phase variations further.

At the end of 2014, a final version of the MI Phase Lock Controller and the Acceleration Phase Lock Detector modules were designed. These modules had both previously been implemented using hardware kluges and firmware variations on Dual Phase Detector modules.

Two potential improvements for the MI phase lock system have been identified and are being targeted for implementation with this revision. The first is a phase error bandwidth adjust circuit for more stably holding the Acceleration phase error at the start of MI phase lock. The second is a fix addressed at eliminating jitter in the extraction notch at extraction. Both of these modifications will be discussed in this document

### III. Basics of the Divide by 32 Phase Lock Method

#### III.1 Motivation for the Chosen Method

The MI phase lock control electronics uses a “divide-by-32” scheme along with an adjustable reference phase error trajectory that guides the divide RF phase error to zero. One of the strongest motivations for using this method is that the Booster beam phase can develop an unmanageable oscillation if the LLRF frequency is changed too rapidly.

From tests and observations made with the Booster we have determined 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 LLRF frequency reference) were determined to be

$$\text{Control Voltage Peak Deviation} < 30 \text{ milli-Volts (1.5 kHz)}$$

$$\text{Control Voltage Slew Rate} < 50 \text{ milli-Volts/milli-second (2.5 kHz / milli-second)}$$

Note the conversion between the feedback Volts applied to the LLRF Frequency Reference module and the change in the RF frequency out is 1000 Hz per 20 mV.

#### III.2 MI Phase Lock Timing

The system uses two phase detectors. The first one monitors the wrapping phase between the MI RF and the Booster LLRF. Figure III.2.1 below shows the typical phase detector output signal.

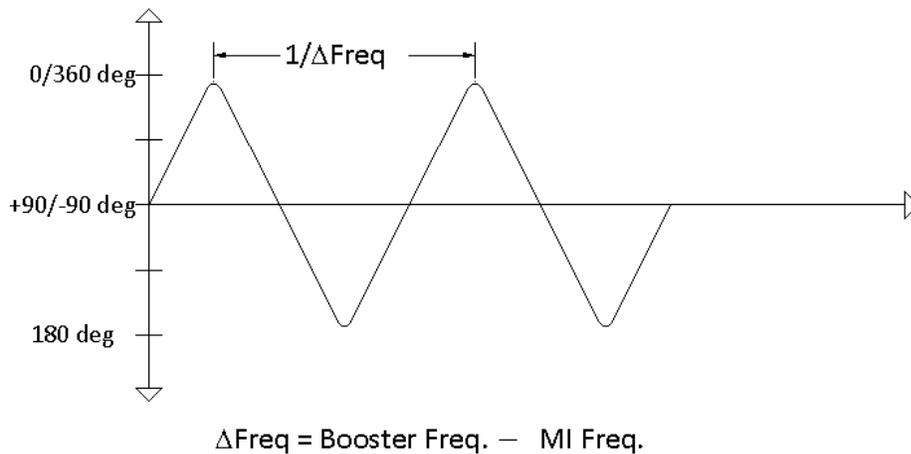


Figure III.2.1 Typical phase detector output

As the Booster LLRF frequency ramps towards the MI frequency the period of the triangle-wave output of the phase detector increases. The frequency difference between the Booster RF and the MI RF is measured using this first phase detector output by counting the number of 12.5 ns logic clocks that occur between zero crossings of the digitized signal. Once a pre-determined difference frequency has been reached, a trigger is fired indicating the start of the MI phase lock cycle. Several other logic signals are derived from this Booster RF versus MI RF phase detector signal. These signals are used by the MIPL Timing state machine to start the MIPL process in a repeatable, consistent way.

The signals used for MIPL Timing are listed and defined in Table III.2.1, and Figure III.2.2 and Figure III.2.3 show scope screen captures labeling the signals and showing how they are used in sequencing.

Table III.2.1 Logic signals used and/or produced by the MI phase lock process.

Signal	Description
Polarity	This signal is set <b>High</b> when the polarity of the phase detector signal is <b>positive</b> and <b>Low</b> when it is <b>negative</b> . The rising and falling edges of this signal correspond to the zero crossings of the phase detector signal.
SlopeA	This signal is set <b>High</b> when the phase detector signal slope is <b>positive</b> and <b>Low</b> when it is <b>negative</b> . The rising and falling edges of this signal generate the 'up_trig' and 'dn_trig' signals. This signal has some filtering that prevents jitter around the positive and negative peaks of the triangle wave.
Freq_Trigger	This pulse is fired when the number of 12.5 ns logic clocks between zero crossings of the phase detector signal first exceeds the set threshold count. This signal initiates the MI phase lock sequence.
up_trig	This is a single clock wide pulse that fires on the <b>rising edge</b> of "SlopeA". It is used to determine whether the difference frequency threshold trigger fired during the negative portion of the phase detector signal or the positive portion.
dn_trig	This is a single clock wide pulse that fires on the <b>falling edge</b> of "SlopeA". It is use in synchronizing the start of the MI phase lock reference trajectory.
Hold_For_MIPL	This signal fires on the first negative to positive zero crossing after the "Freq_Trigger" fires.
Load_Div_Counters	This signal is set <b>Low</b> during the start up sequence for the MI phase lock process. When the signal is set low the divide by 32 counters start running producing a square wave output at the Booster RF or MI RF frequency divided by 32.
Use_Alt_Slope	This signal indicates that the Phase Error Reference Trajectory should have a lower slope. It will go active when the difference frequency threshold during the negative slope portion of the phase detector signal. Usage is explain in detail in a later section.
Run_MIPL	Once the MIPL Timing state machine reaches the end of its sequence, this signal goes active signaling the start of the Phase Error Reference Trajectory.

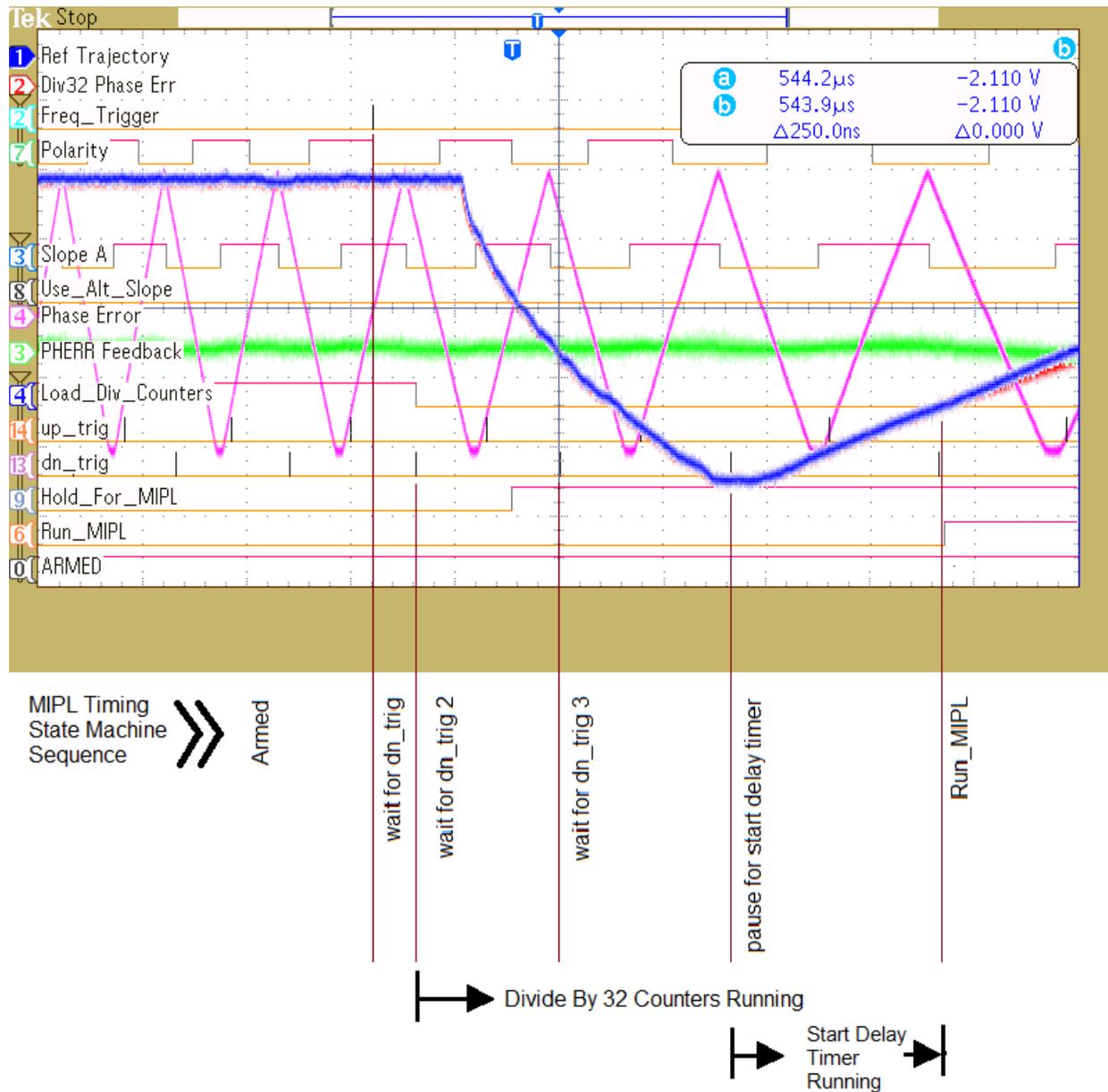


Figure III.2.2 Typical start of MI Phase Lock scope screen capture.

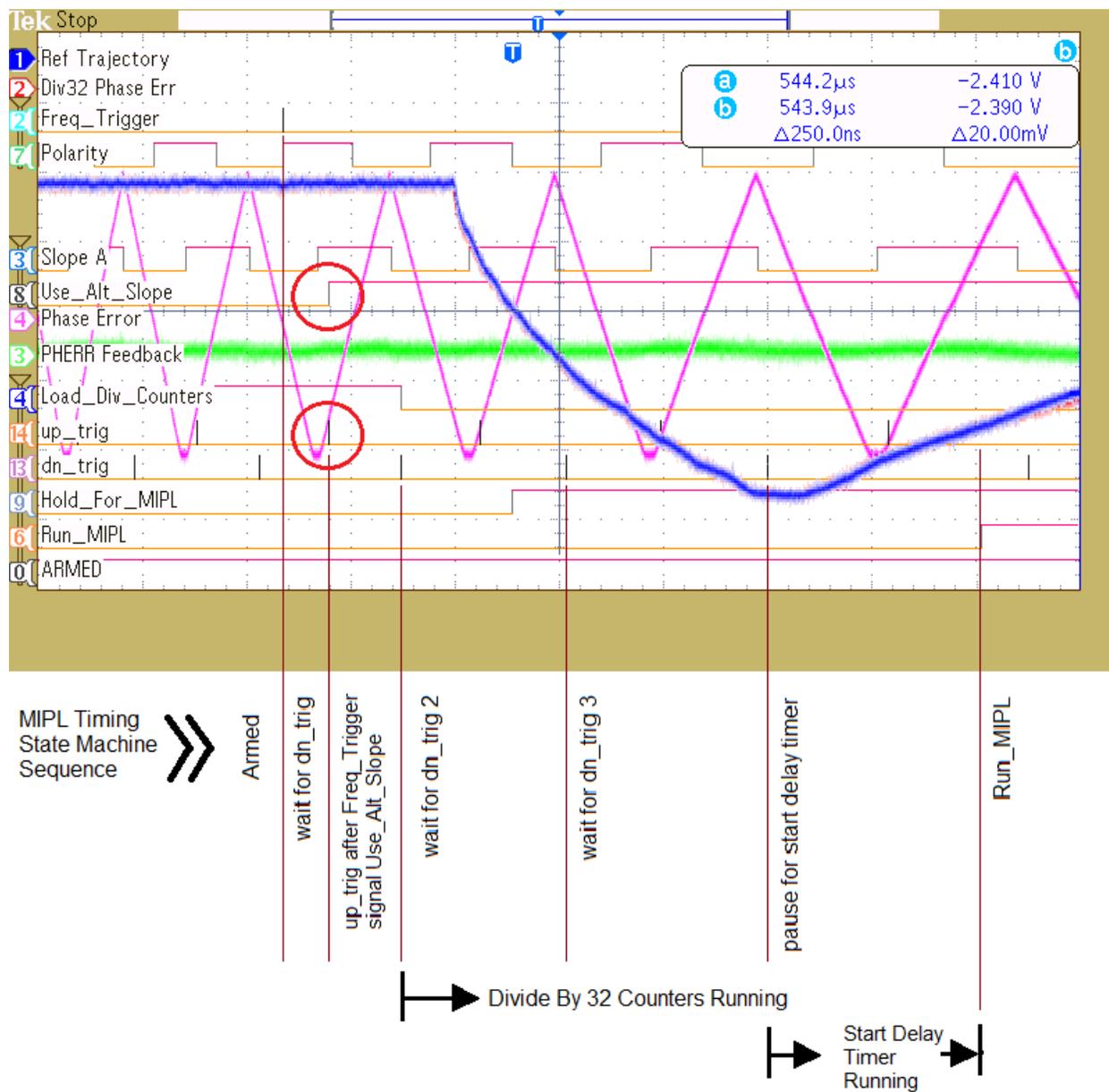


Figure III.2.3 Start of MIPL where the use of the alternate reference trajectory slope is signaled.

Note in Figure III.2.3 above the point where the divide by 32 high speed counters that divide the Booster RF and the MI RF are released from reset (Load\_Div\_Counters). The logic schematic for the counters is shown in Figure III.2.4. 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 III.5 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 before extraction.

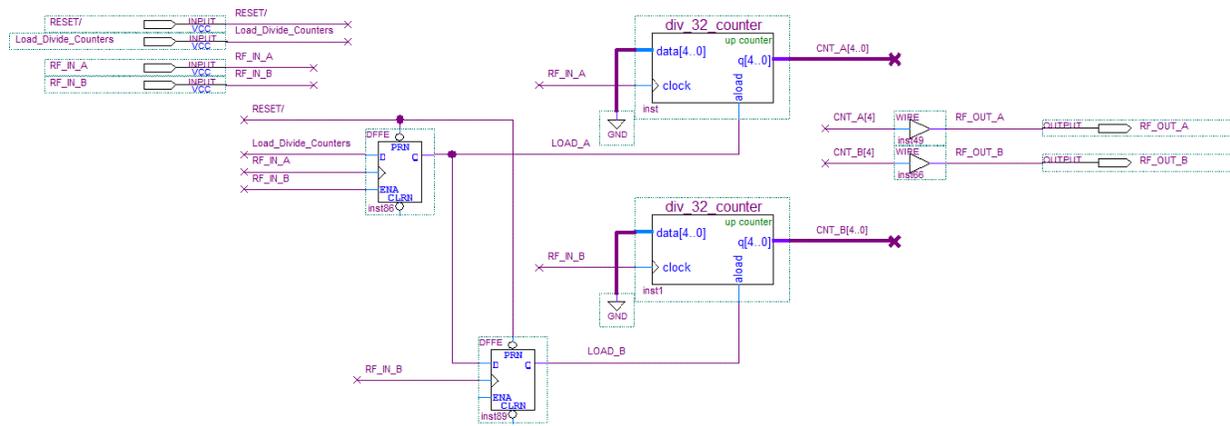


Figure III.2.4 The logic schematic of the divide by 32 counters.

Figure III.2.5 shows the output of the phase detector whose inputs are the divide-by-32 versions of the 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.

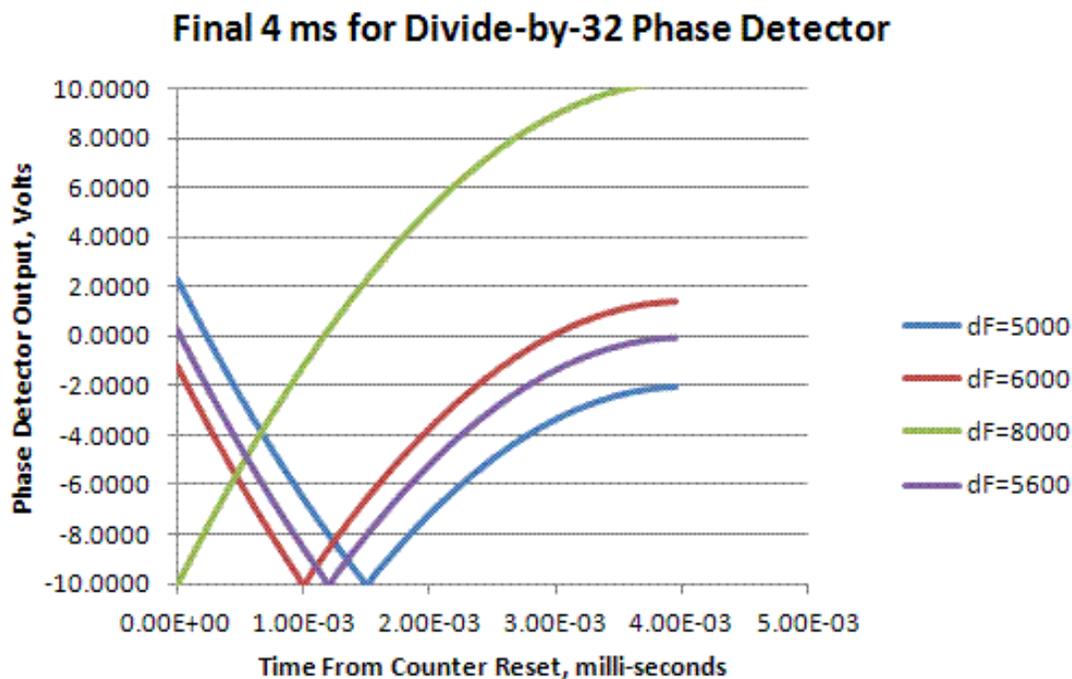


Figure III.2.5 Phase error trajectories for the divide-by-32 phase detector

Figure III.2.6 is another scope screen capture showing the Divide by 32 Phase Error, in red on channel 2, and the Phase Error Reference Trajectory, in blue on channel 1, all the way through to phase locked. The magenta trace on channel 4 is the Booster RF to MI RF phase error, as before, and in green on channel 3 is the phase error feedback to the FM modulated sinewave generator used on our lab bench. The setup of the testing in the lab is covered in a later section.

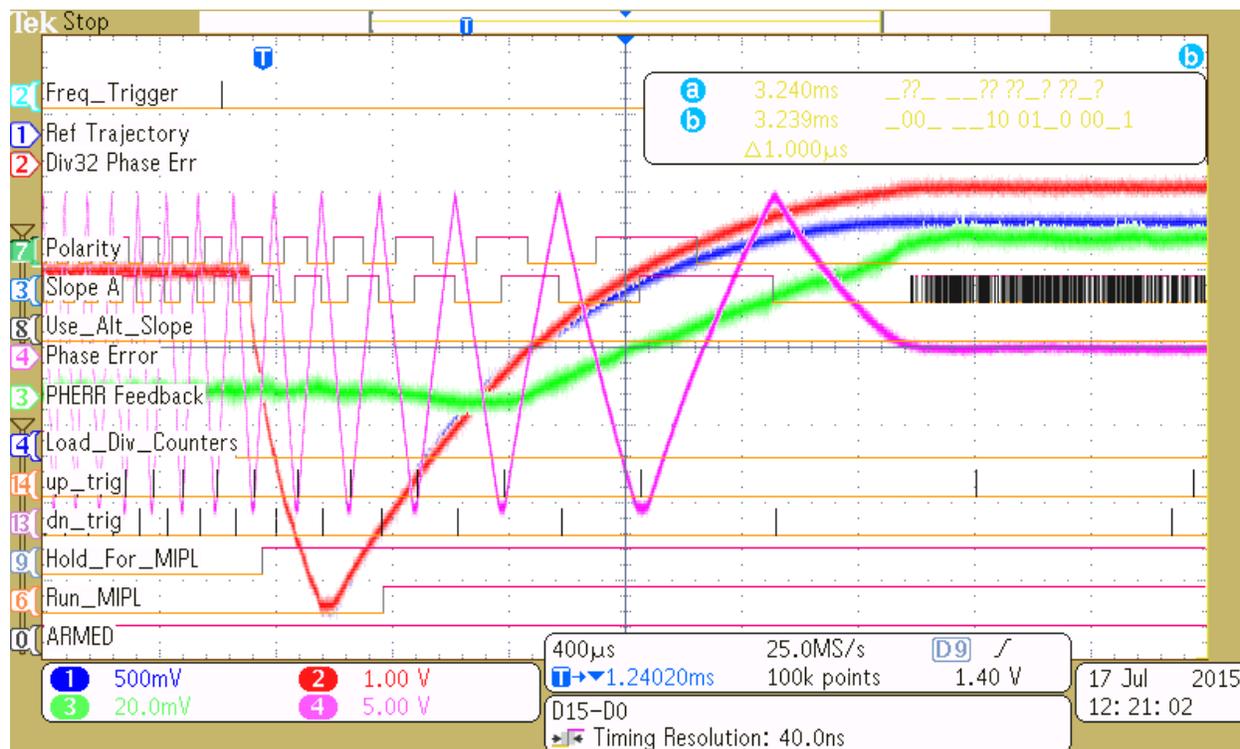


Figure III.2.6 A scope screen capture of a full MIPL interval done in the lab, on the bench.

### III.3 MI Phase Lock Control Feedback Generation

The voltage feedback to the LLRF Frequency Reference module modulates the LLRF frequency about a predetermined frequency trajectory of the frequency curve. During the acceleration cycle, all but the last 4 milli-seconds when MI phase lock occurs, the Acceleration phase detector feeds back the phase error between the LLRF reference and a beam bunch pickup signal let's call, the beam RF. At the start of the MI phase lock interval the Acceleration phase detector feedback must be held fixed and the Radial Position control loop must be disabled. These are the two control loops that modulate the LLRF frequency and phase, respectively. These are not allowed control once the MI phase lock starts. Beam stability is sensitive to how and at what value the Acceleration phase detector feedback is held at.

The phase lock feedback to the LLRF Frequency Reference is basically the error between the divide by 32 RF phase error and the phase error reference trajectory. The divide by 32 phase error signal is digitized and processed using a programmable FPGA and the feedback voltage is produced by the FPGA algorithms and writing a digital to analog converter (DAC).

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 the Reference Trajectory Offset value. By tracking and then latching the phase error to be the starting point of the reference trajectory we do not produce a sudden difference between the two and output a glitch on the feedback.

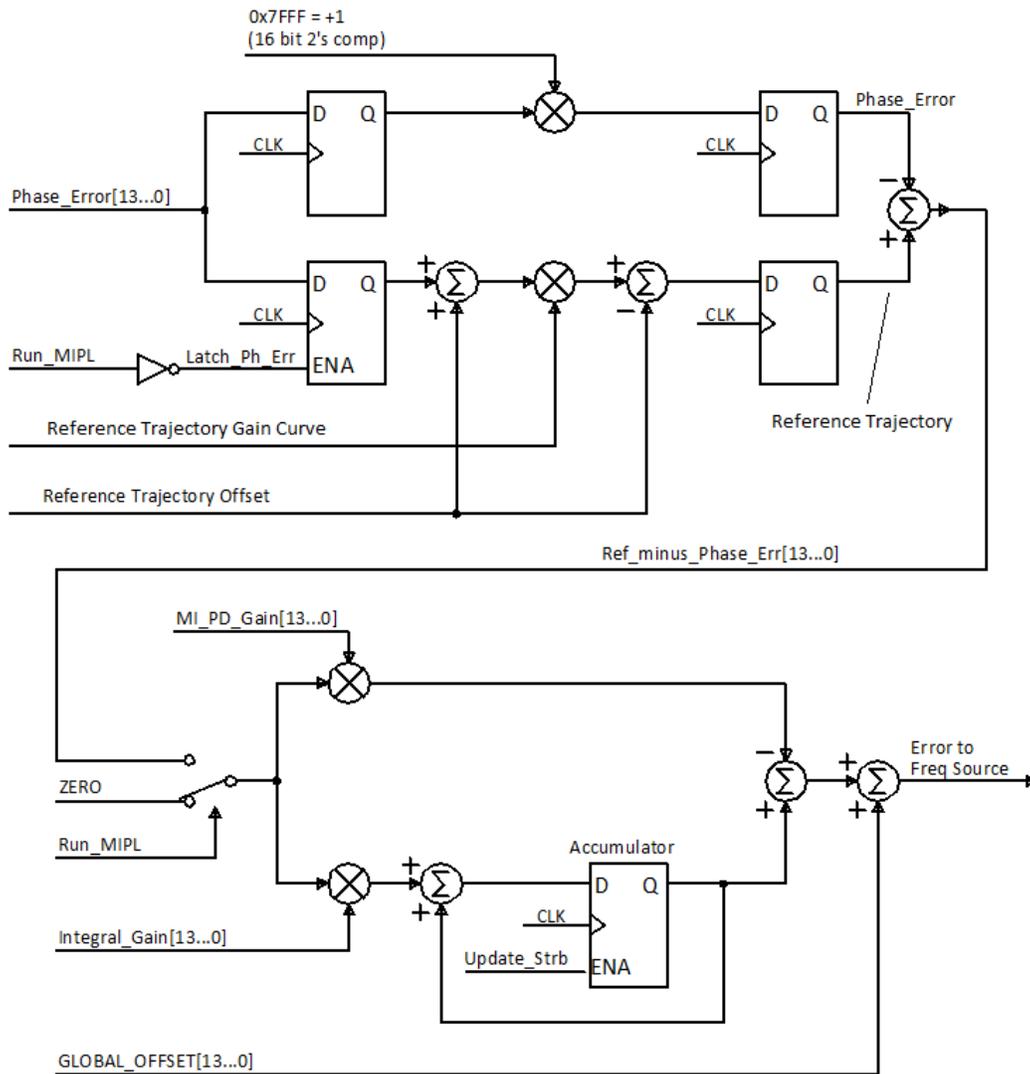


Figure III.3.1 The simplified phase lock controller block diagram

Referring to the block diagram of Figure III.3.1, we see at the top left where the signal Run\_MIPL causes the Phase Error to be latched in the path that produces the reference trajectory. Initially the Gain Curve has a value of 1 and the Offset value is both added to and subtracted from the reference trajectory equally. As the gain curve approaches zero, the initial phase error component of the curve and the additive offset approach zero leaving only the negative Offset. The Offset can be set to a negative or positive value by the user.

Throughout the phase lock interval the measured phase error is subtracted from the reference trajectory and the 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.

#### 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 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  $f(x) = \text{Exp}(-N/\text{tau})$ , where *tau* 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.

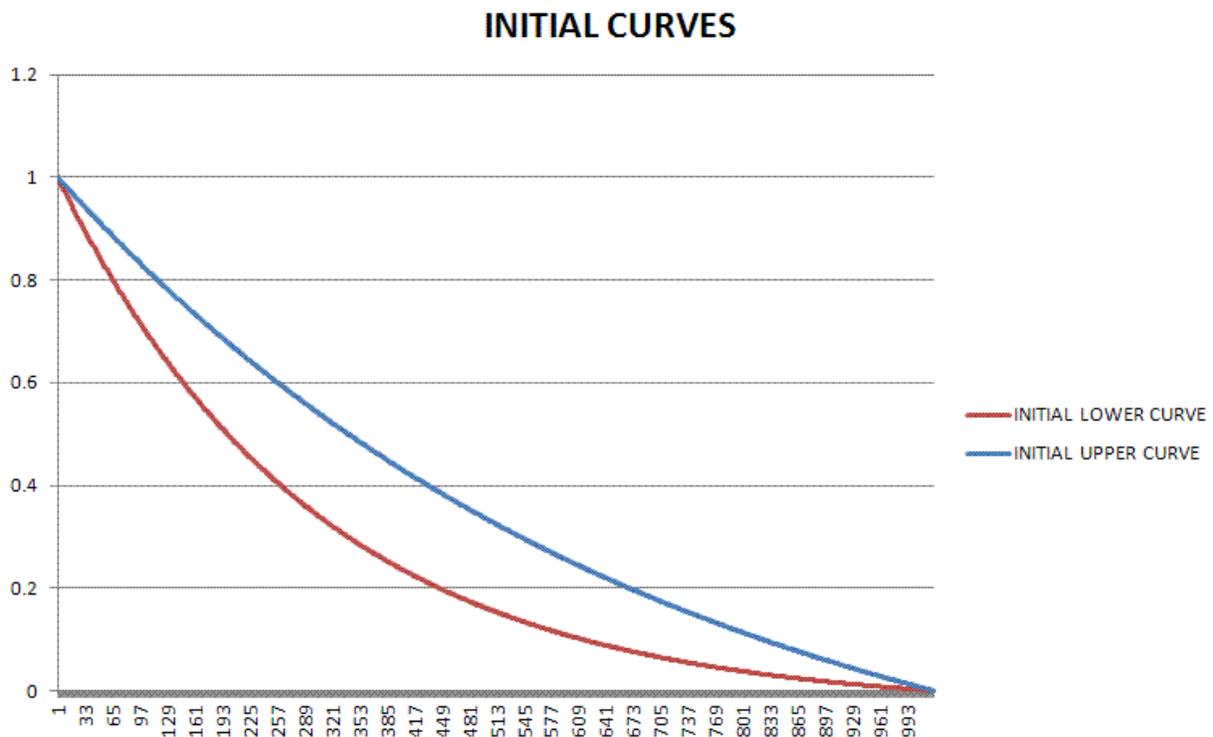


Figure IV.1 The initial upper and lower curves used to define the reference trajectory gain curve

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.

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

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

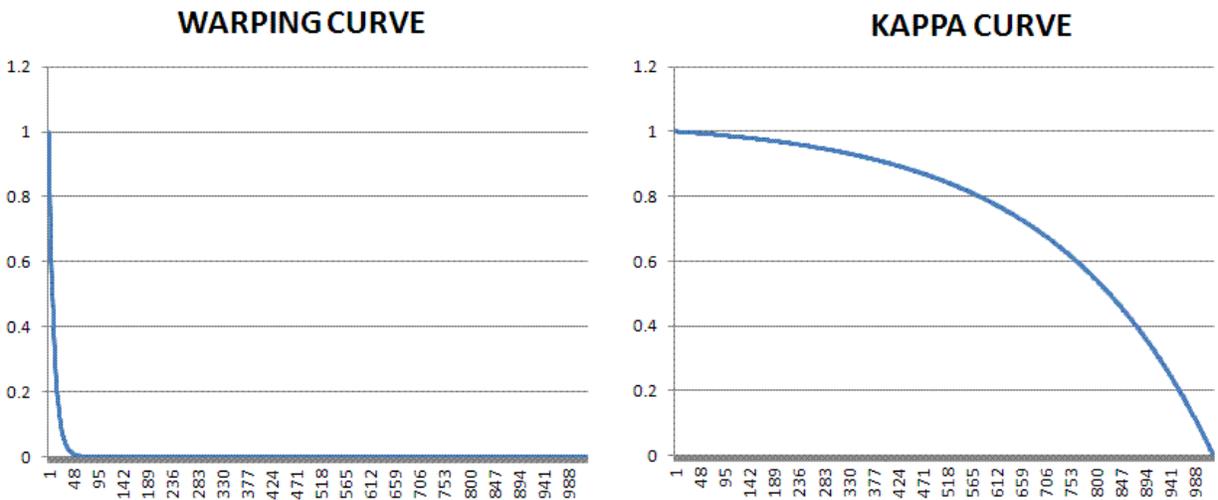


Figure IV.2 Plots of the warping curve and the weighting “kappa” curve

## WARPED CURVES

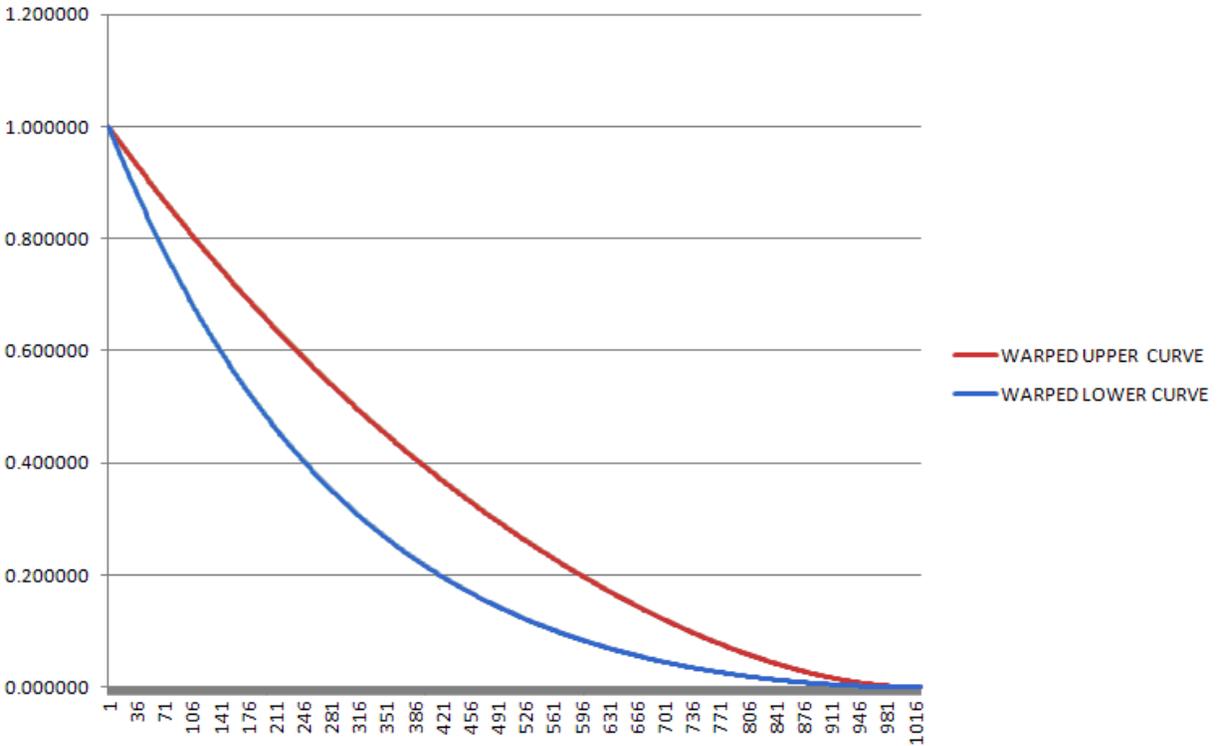


Figure IV.3 Plot of the final warped upper and lower curves that are installed into FPGA memory

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 Frequency Trigger Details

The older system for MI phase lock got a trigger from the VXI-DDS LLRF Frequency Reference module when the sum of the Frequency Curve plus the Acceleration Phase Lock phase error reached 52.8023 MHz, approximately 8 kHz below the MI RF frequency. Then the next down-slope of the phase error triangle wave triggered the start of the MIPL feedback.

The new system follows the same approach except that the new system determines the MI phase lock Frequency Trigger by counting the 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

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

Figure V.1.1 compares details of this timing process for both the old MIPL system and the new. 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 varied depended on the delay between the frequency trigger and the Down Slope Pulse trigger (DSP output of the old Timing Generator Module).

The lower portion of Figure V.1.1 illustrates details of the new MIPL system. 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.

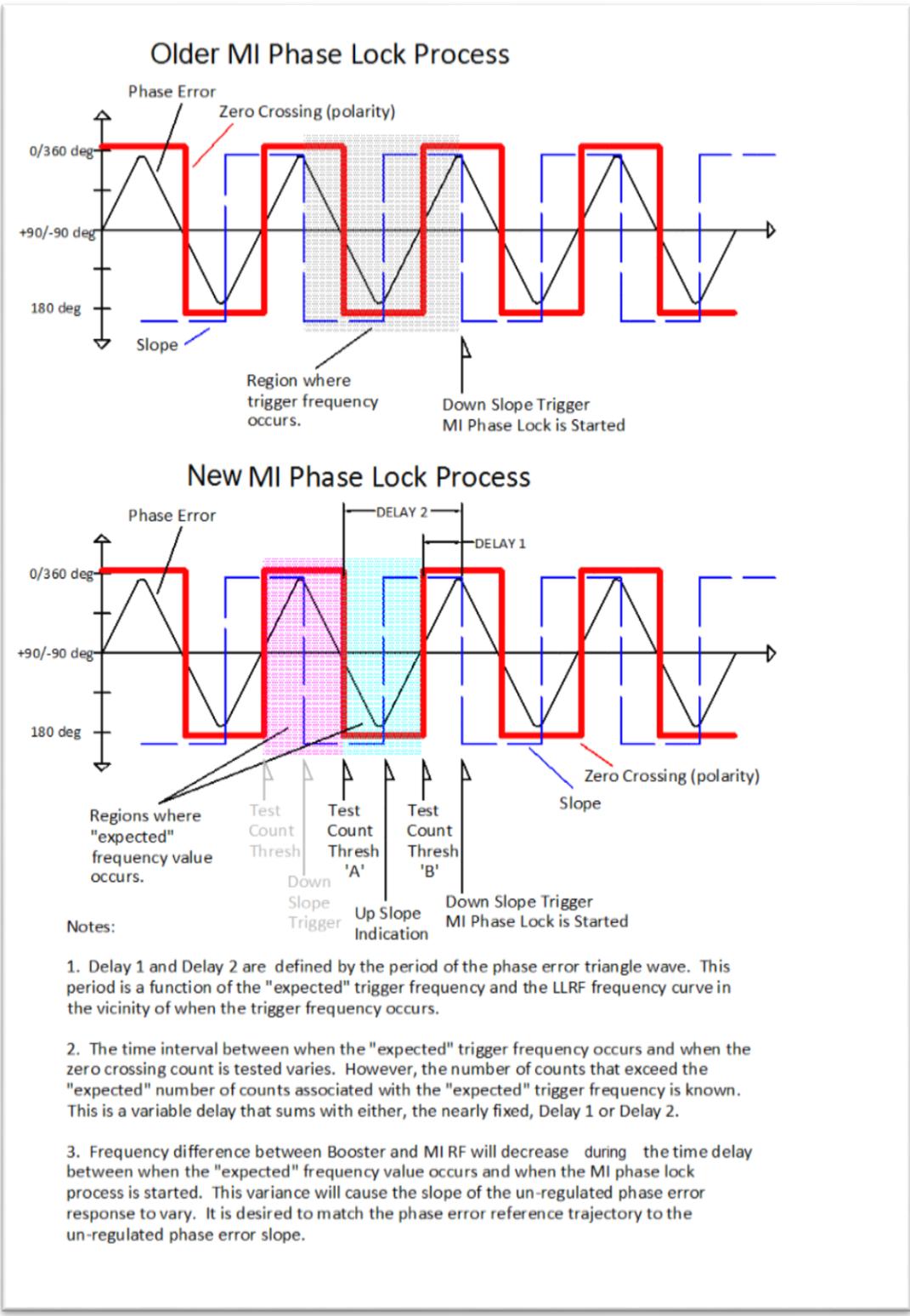


Figure V.1.1 Triggering the MI Phase Lock process.

## 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.

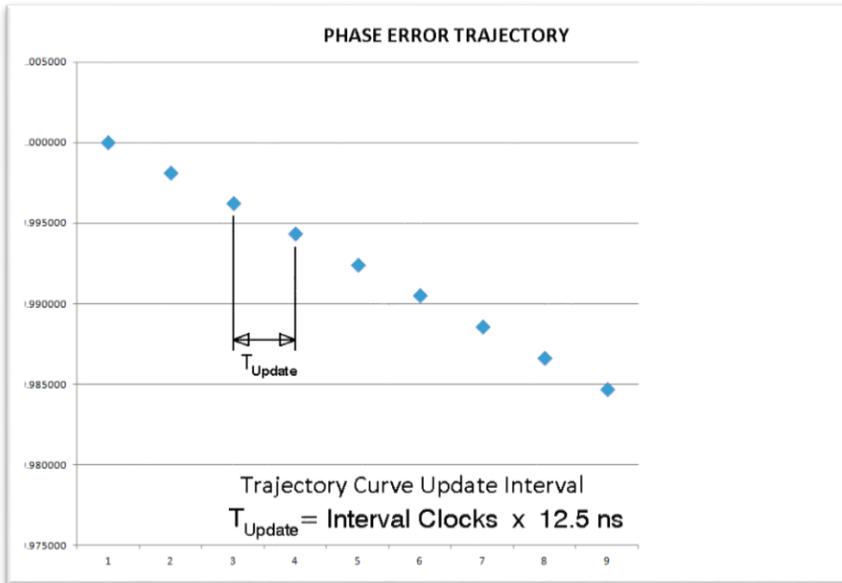


Figure V.2.1 Definition of the Update Interval

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'.

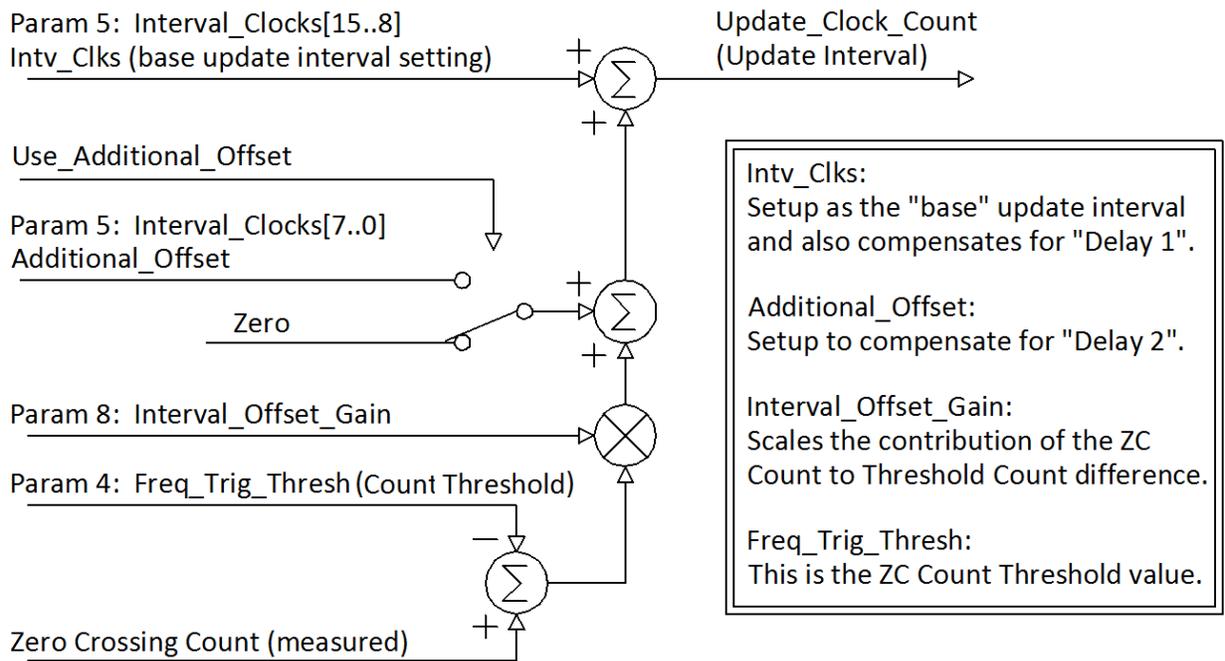


Figure V.2.2 Calculation of the Update Interval to compensate for the phase error slope

## VI. Extraction Notch Adjustment (still under study, October 2015)

A difficulty with the current MI phase lock method is that, at the start of the process the LLRF Radial Position control (RPOS) is disabled and the beam radial position is allowed to drift. This and possible some other effect we are not yet sure about, cause the extraction notch whose position in the orbit had been controlled by the Cogging Control system, to deviate from its expected position by plus or minus one RF bucket (one RF sinewave period).

In order to correct the notch position error we must first, before phase error reference trajectory starts, predict whether and in which direction the notch may be off. We hope to control the total number of degrees that the Booster RF slips in phase with regard to the MI RF between the time we disable the RPOS control (Hold\_For\_MIPL) and phase locked. The phase slipping is precisely represented by the Booster RF to MI RF phase error signal.

The total degrees of phase slippage can be changed by changing the offset at which the phase error reference trajectory (or equivalently the divide by 32 phase error) settles. We will have three trajectory offsets; one for the minus RF bucket correction, one for the plus RF bucket correction, and the case where no correction is desired. The important thing is that the phase difference that the Booster RF to MI RF phase error flattens out at, at phase locked, is the same for each of the three variations. Details of how the phase error reference trajectory is generated and how the offsets are handled was covered in Section III.3.

In the figures below are shown the three offset variation we will be adjusting for.

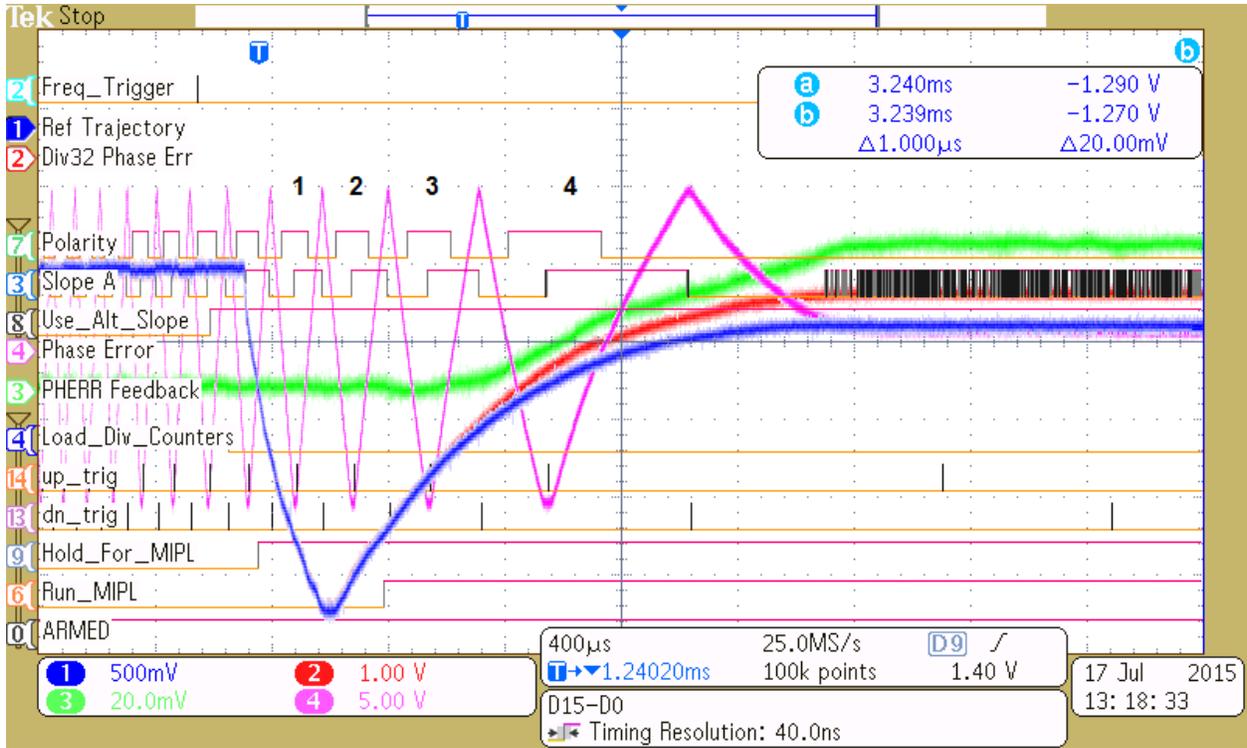


Figure VI.1 MI phase lock with an offset in the reference trajectory for plus RF Bucket correction.

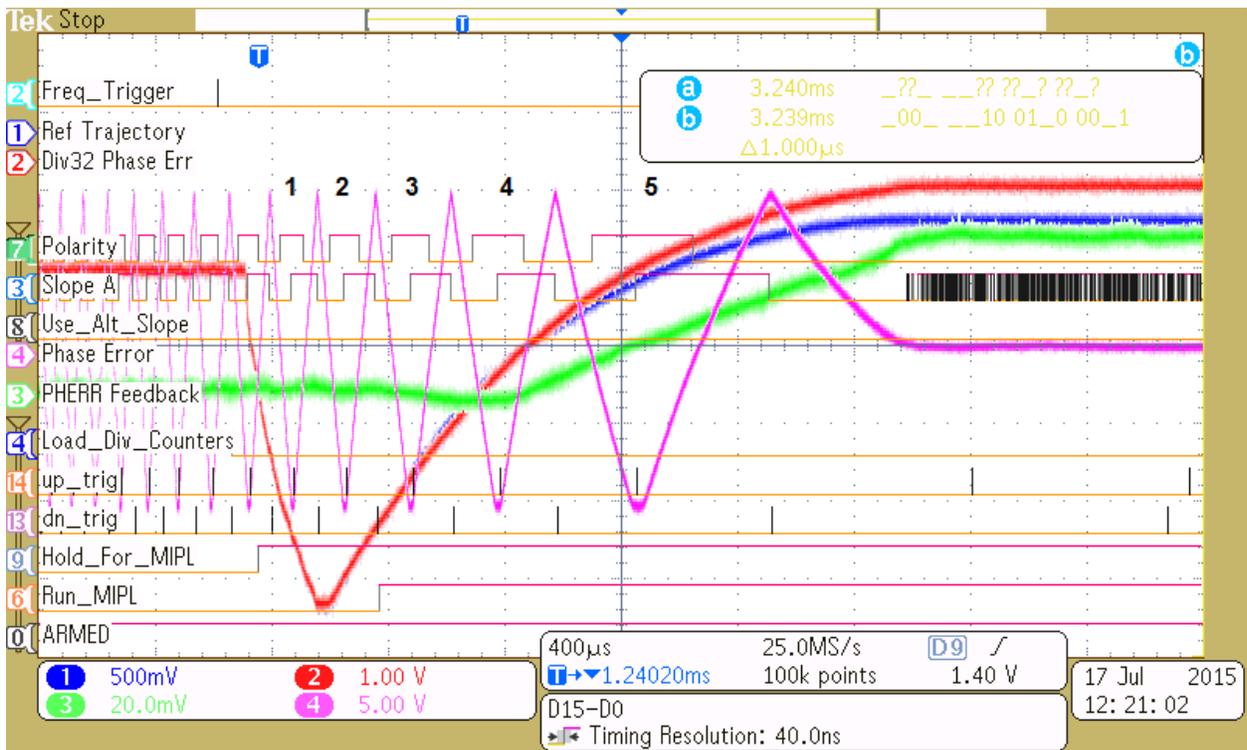


Figure VI.2 MI phase lock with an offset in the reference trajectory for no RF Bucket correction.

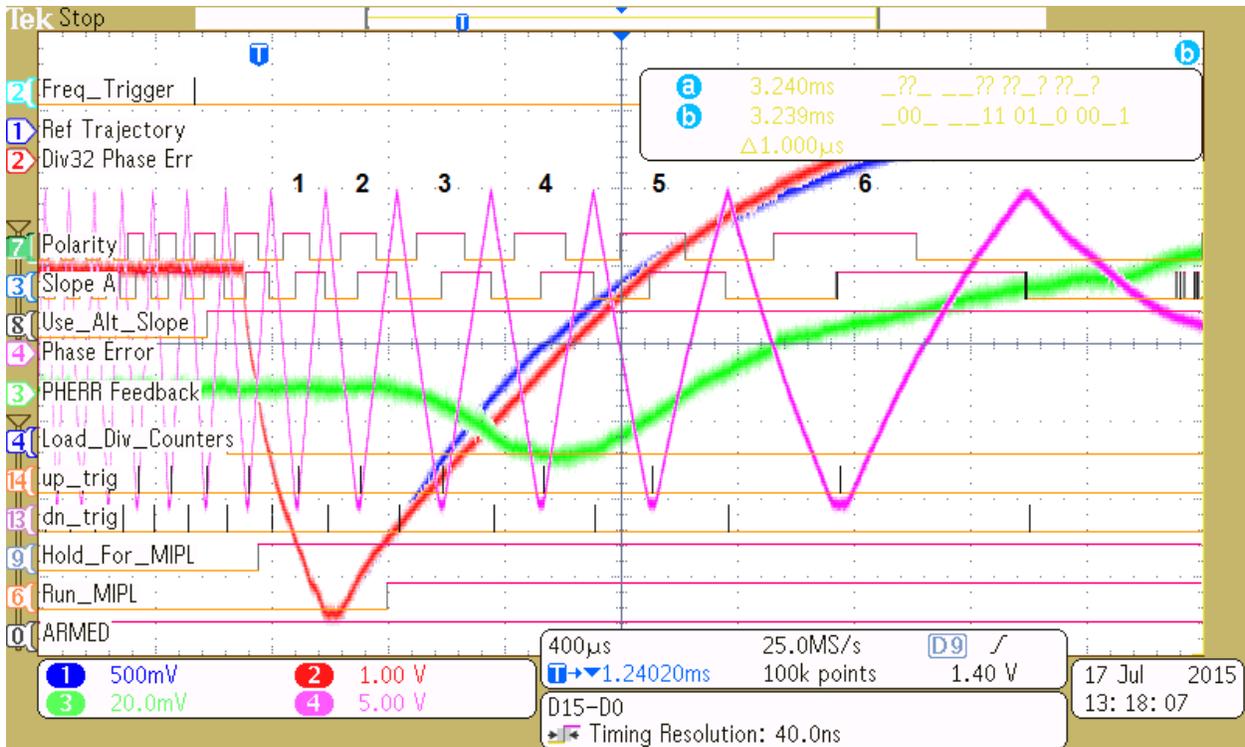


Figure VI.3 MI phase lock with an offset in the reference trajectory for minus RF Bucket correction.

In order to detect which case we have before the reference trajectory has to start is being studied also. The current proposal is to count the Booster RF cycles between BDOT, the start of Booster acceleration, and the occurrence of Hold\_For\_MIPL signal. It is hoped that this will have some nominal value that always results in the extraction notch being in the right place. One count too many will signal one error case and one count too few the other error case.

## VII. Setup for Bench Testing

For testing on the bench we use a Tektronix AFG3102C Arbitrary Function 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 Tektronix AFG3102C Arbitrary Function 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-32 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 VII.1 shows the cabling of the signal generators and the two NIM modules.

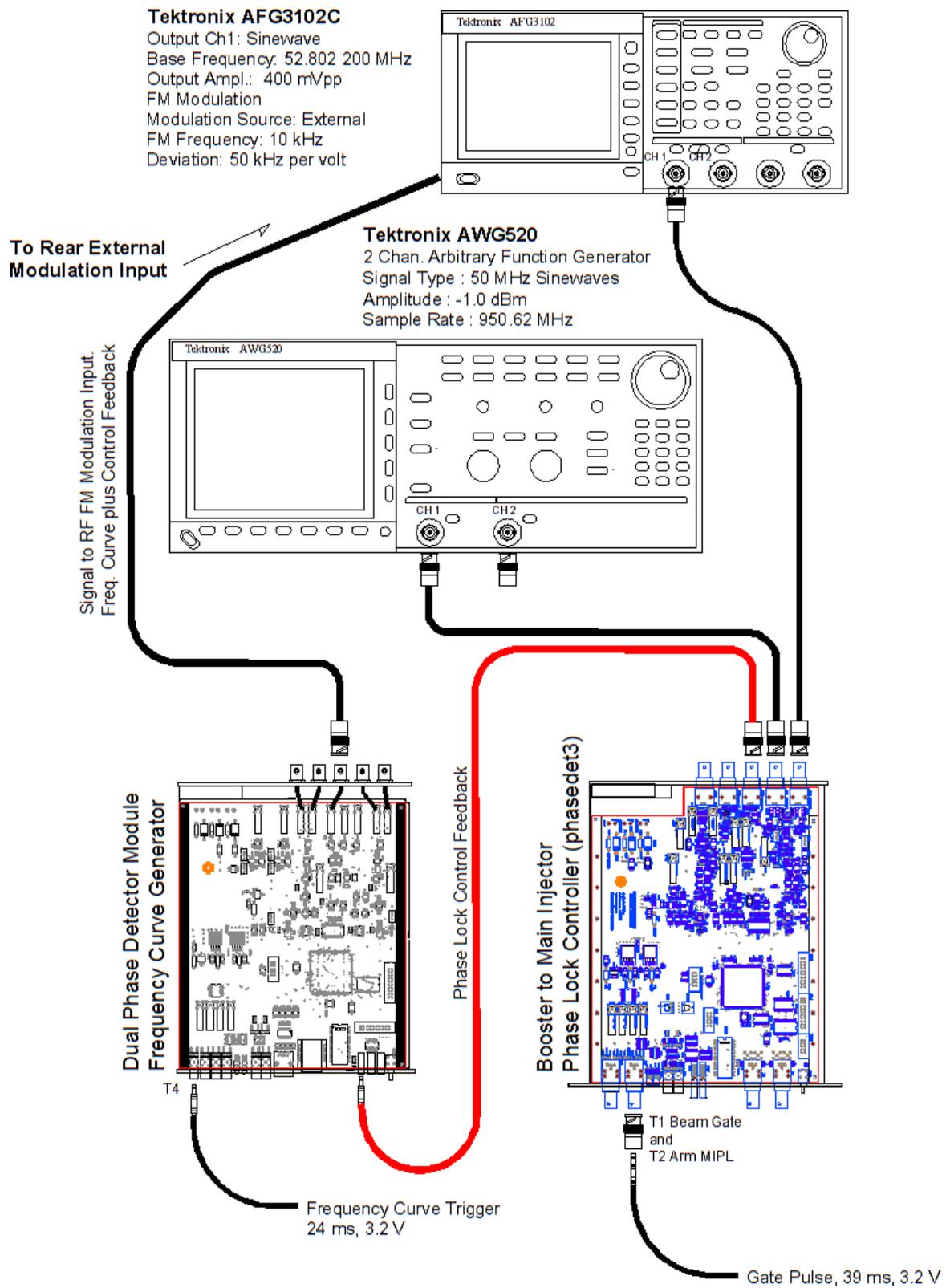


Figure VII.1 Test Bench Setup

Table VII.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	0x0103	Two, 8 Bit Count Values	A number of 12.5ns clocks that delay the first edge of the div-by 32 RF outputs. <b>A_DLY = Param_0[4..0] (example: 0x03)</b> <b>B_DLY = Param_0[12..8](example:0x01)</b>
1	0x0800	12 Bit Offset Binary Value (0x0800 = zero)	BRF to MI RF PD Offset . Effects the signal into ADC_1.
2	0x1800	14 Bit Offset Binary Value (0x2000 = zero)	div-by 32 MI PD Proportional Controller Gain. Applied to the div-by 32 phase trajectory tracking.
3	0x0000	12 Bit Offset Binary Value (0x0800 = zero)	div-by 32 MI PD Offset. Effects the signal into ADC_2.
4	0x12F0	16 Bit Unsigned Integer	Internal Frequency Trigger Threshold. Number of clocks between zero crossings of the normal PD triangle wave output.
5	0x8A10	Dual 8 Bit Unsigned Integer	<b>BASE Intvl Clk Setting for no Cog Adjustment Bits [15..8] MIPL Transition Interval.</b> Number of 12.5 ns clocks between memory curve updates.  <b>Bits [7..0] Alternate Slope Offset.</b> 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	0x2200	14 Bit Offset Binary Value (0x2000 = zero)	ADC zero crossing threshold for determining polarity and frequency difference.
7	0x8035	16 Bit Unsigned Integer	MODE Control Word. See table VI.3.
8	0x3000	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	0x8220	16 Bit Offset Binary Value (0x8000 = zero)	<b>BASE Phase Offset.</b> Used to adjust the final phase in the no Cogging adjustment mode.
10	0x4000	16 Bit Unsigned Integers	Delay between final “Down Trigger” and the start of the Reference Trajectory.
11	0x8000	16 Bit Offset Binary Value (0x8000 = zero)	DPOT Control Word.

12	0xB010	Dual 8 Bit Unsigned Integer	<p><b>PLUS Intvl Clk Setting for Cog Adjustment Bits [15..8] MIPL Transition Interval.</b> Number of 12.5 ns clocks between memory curve updates.</p> <p><b>Bits [7..0] Alternate Slope Offset.</b> 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.</p>
13	0x7978	16 Bit Offset Binary Value (0x8000 = zero)	<p><b>PLUS Mode Phase Offset.</b> Used to adjust the final phase in the PLUS Cogging adjustment mode.</p>
14	0x6010	Dual 8 Bit Unsigned Integer	<p><b>MINUS Intvl Clk Setting for Cog Adjustment Bits [15..8] MIPL Transition Interval.</b> Number of 12.5 ns clocks between memory curve updates.</p> <p><b>Bits [7..0] Alternate Slope Offset.</b> 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.</p>
15	0x8B12	16 Bit Offset Binary Value (0x8000 = zero)	<p><b>MINUS Mode Phase Offset.</b> Used to adjust the final phase in the MINUS Cogging adjustment mode.</p>

Table VII.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	0x0280	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	0x2200	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).

Table VII.3 Test Interface selection of different IO options using Parameter 7.

Bits	Description	
[2..0]	<p>Selects the signal to output through the auxiliary DAC 1 output AO1 when bit[15] = '1'</p> <p>Aux DACs have an offset binary coding:            0x0000 = -10V            0x8000 = 0V            0xFFFF = +10V</p>	<p>[0,0,0] =&gt; ADC_1_DATA[15..2] (PD-1 phase error)</p> <p>[0,0,1] =&gt; Ref_minus_Pherr[13..0] (before gains are applied)</p> <p>[0,1,0] =&gt; Delta_T_Mon_OB[13..0] (Difference between PD-1 zero crossing times and the Freq_Trig_Threshold, Param 4)</p> <p>[0,1,1] =&gt; Integrator Output[13..0]</p> <p>[1,0,0] =&gt; Delta_T_Mon_OB[13..0] (Difference between PD-1 zero crossing times and the Freq_Trig_Threshold, Param 4)</p> <p>[1,0,1] =&gt; ADC_2_DATA[15..2] (PD-2 div-by 32 phase error)</p> <p>[1,1,0] =&gt; FreqTrig_Delta_T[15..2] (2s comp version of Delta_T_Mon)</p> <p>[1,1,1] =&gt; Method2_FeedBack[13..0] (MIPL PHerr Feedback)</p>
[3]		Used in module test only
[6..4]	<p>Selects the signal to output through the auxiliary DAC 2 output AO2 when bit[15] = '1'</p> <p>Aux DACs have an offset binary coding:            0x0000 = -10V            0x8000 = 0V            0xFFFF = +10V</p>	<p>[0,0,0] =&gt; ADC_2_DATA[15..2] (PD-2 div-by 32 phase error)</p> <p>[0,0,1] =&gt; Ref_minus_Pherr[13..0]</p> <p>[0,1,0] =&gt; Integral Gain Ramp[13..0]</p> <p>[0,1,1] =&gt; Ref_Trajectory[13..0]</p> <p>[1,0,0] =&gt; ADC_1_DATA[15..2] (PD-1 phase error)</p> <p>[1,0,1] =&gt; Ref_Curve_Intv_Clks[15..2] (Interval clocks applied to the reference curve after including adaptive adjustments)</p> <p>[1,1,0] =&gt; Delta_T_Mon_OB[13..0] (Difference between PD-1 zero crossing times and the Freq_Trig_Threshold, Param 4)</p> <p>[1,1,1] =&gt; Ramped Gain Monitor[13..0]</p>
[7]		Used in module test only
[11..8]		Used in module test only
[13..12]	Mode Select	<p>[0,0] =&gt; default assignments</p> <p>[0,1] =&gt; Base module assembly testing assignments</p> <p>[1,0] =&gt; default assignments <u>except</u> using alternate Digital Outputs.</p> <p>[1,1] =&gt; default assignments</p>
[14]	Ena_ZC_Start	Used in module test only
[15]	Select_Alt_DAC	Selects the multiplexed signals chosen by bits[2..0] and bits[6..4] to drive Aux DAC 1 and Aux DAC 2 respectively. When Low, the ADC 1 and ADC 2 values are selected.