Fermilab Low Energy Linac Low Level RF Upgrade Engineering Notes and Operational Guide

7/23/2009

AD/RF/LLRF

Low Energy 201.25 Linac MHz Upgrade

E. Cullerton, B. Chase, P. Joireman, V. Tupikov, P. Varghese, D. Klepec, T. Butler, L. Allen, M. Kucera

Table of Contents

Low Energy 201.25 MHz Linac Upgrade

E. Cullerton#, B. Chase#, L. Allen, B. Barnes,J. Branlard, T. Butler, P. Joireman, D. Klepec, M. Kucera, P. Varghese, V. Tupikov

4/15/09

Abstract:

An upgrade to the low energy 201.25 MHz Linac at Fermilab is described in this paper. The upgrade to the low energy Linac system has been implemented to improve cavity vector regulation and reduce beam losses. The upgrade includes an adaptive feed forward system for beam loading compensation, a digital phase feedback system for the RF, a digital phase comparator for cavity tuning, and a low level RF system that is digitally controlled and referenced to a very stable 805 MHz reference line. The adaptive feedforward system is designed to improve transient beam loading for the first few microseconds of beam. The phase feedback system and cavity tuning system are digital systems with ACNET parameter control. The new low level RF system is designed to improve the phase stability and replace the inter-tank phase system that controls the synchronous phase of the RF. The main components of this new system are housed in two VXI modules, located in a VXI chassis at each station of the low energy Linac. One VXI module is referred to as the Multi-Channel Field Controller (MFC) Board[1], and the other VXI module is referred to as the RF Board[2]. The MFC Board uses a modern Digital Signal Processor (DSP) and Field Programmable Gate Array (FPGA) to implement the digital control system, and the RF Board is responsible for analog RF signal processing and other analog signal processing. A slot 0 controller is housed in each VXI chassis to interface the MFC Board to the control system[3]. An operations manual is available for daily operation issues[8].

1 System Overview

A simplified block diagram of the old system is shown in figure 1.1. The layout of the block diagram is similar to the physical layout of the real system. The main components of the system are the cavity, the 7835 power amplifier, the driver cabinet, the modulator, and the NIM modules located in the rack at the front of each station.

The cavity is an Alverez drift tube accelerating cavity, with a motorized tuning slug to control the resonant frequency. The 7835 PA is a 5MW cathode driven RF tube amplifier. The 7835 PA provides the high power RF to the cavity for accelerating beam. A 9-inch coax line delivers RF power to the cavity. There are directional power couplers on the 9-inch line to provide forward and reverse power signals. The Driver Cabinet consists of a series of medium power solid state amplifiers and high power tube amplifiers to amplify the low level RF to approximately 175 kWatts to drive the 7835 Power Amplifier. The modulator controls the anode current of the 7835 PA, and the modulator can deliver up to 10 MW of power and it consists of power FETs and large switch tubes. In the rack at each station, a 5 wide NIM module called Phase/Frequency control, controls the phase and resonant frequency of the cavity. Also in the rack is a 1 wide NIM module, called the Gradient Regulator, which controls the gradient program that is sent to the Modulator. Within the Modulator, there is a current feedback loop that is controlled by the Modulator Regulator.

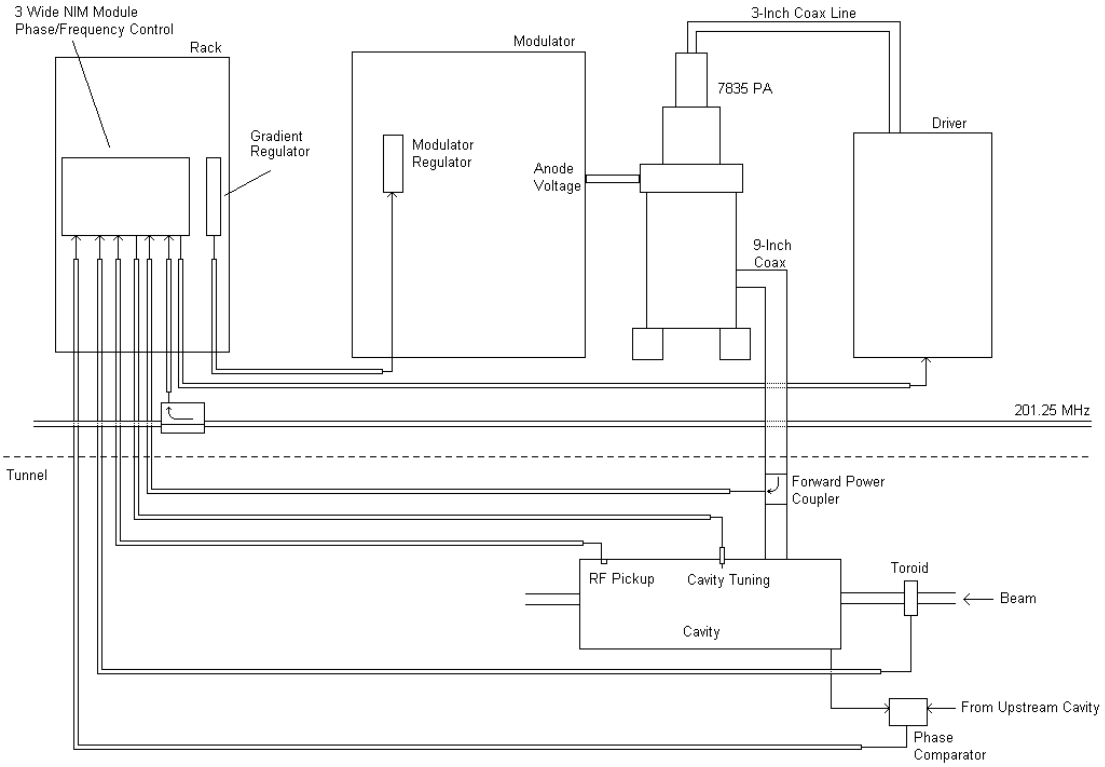


Figure 1.1: Simplified block diagram of the old system

The main components of the upgrade are an adaptive feed forward system to improve beam loading compensation, a new digitally controlled phase feedback system that replaces the analog RF phase feedback, a RF system locked to a reference line that replaces an inter-tank phase regulation system, and a digital phase comparator that replaces an analog phase comparator of the cavity tuning control system.

The goal of the upgrade is to improve the amplitude and phase regulation of the RF as beam is accelerated through the cavity. The system must compensate for beam loading while ensuring that the RF amplitude and phase remain constant. An analysis of the low energy linac beam loading compensation is detailed in [4]. The specification given was to regulate the amplitude of the RF gradient in the cavity to 0.2%.

The first limitation of the old system was that the modulator could not increase the RF power of the 7835 PA fast enough to compensate for beam loading. As a result, the first few microseconds of beam are of poor quality and chopped off before transfer to the booster. A picture of the RF cavity gradient of station 5 is shown in figure 1.2. As seen in the figure, the amplitude of the RF deviates by 1% while beam is being accelerated through the cavity. This number varies from station to station, but the figure shows data from station 5, which had the best regulation of all.

A second limitation of the old system was that a toriod signal, which detected beam amplitude in the beam line, was responsible for increasing the modulator voltage. The signal from the toriod to the RF station has a delay associated with the cable lengths and position in the tunnel. A diagram of the toriod is included in figure 1.1. This delay further increased the amount of time for the modulator to increase its voltage to

compensate for beam loading. The solution to this problem was to incorporate a feed-forward system that programs the modulator to start increasing its voltage before beam arrives. This new feed-forward system includes adaptive learning and will be described in section 1.1.

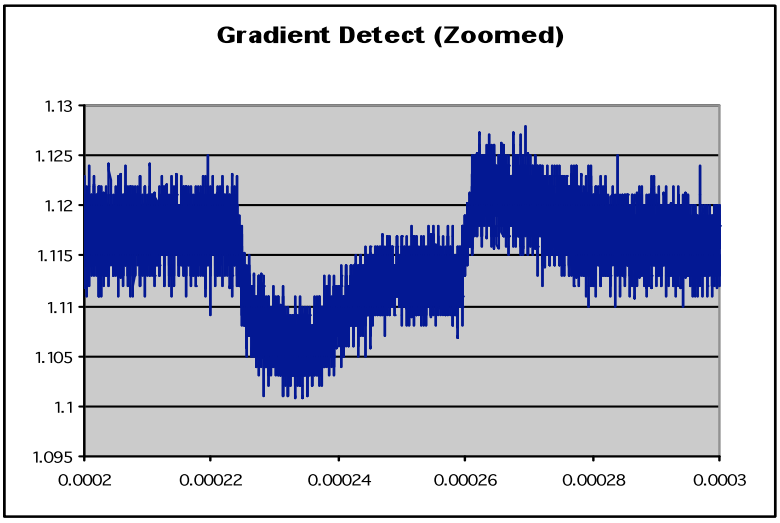


Figure 1.2: Plot of the RF gradient in the cavity of the old system for station 5.

A third limitation of the old system was its phase feedback configuration. In the old system, there was no way to adjust the feedback gain of the system while the machine was operational. A potentiometer used to adjust the gain was located inside the 5 wide Phase/Frequency NIM module, requiring the system to be shutdown in order to make adjustments to feedback gain. The new system uses a digitally controlled phase feedback system with parameters that are adjustable through ACNET while the system is operational. The new phase feedback system will be discussed in section 1.2.

The fourth limitation of the old system was the inter-tank RF phase system. The inter-tank system used phase comparators between each RF cavity to keep the phase of the 201.25 MHz RF synchronous with the accelerating beam. This phase is referred to as the synchronous phase. When the synchronous phase of one cavity was adjusted, the synchronous phases of all the other low energy linac cavities needed adjustment when using this inter-tank system. The result was a RF system at each station that was inter-dependent on other stations of the low energy linac. The upgrade eliminates this interdependency and regulates the phase of the RF phase at each station using a phase stable reference line, which will be described in section 1.2. It addition, the old system used a phase loop around the entire low energy linac that ensured the synchronous phase of the low energy linac was aligned with the synchronous phase of the high energy linac. This phase loop is not needed in the new system and was removed after the new system was installed.

The final component of the upgrade simply replaces an analog phase comparator of the cavity tuner control system. The function of the cavity tuner is to maintain the resonant frequency of the cavity. It does this by comparing the phase of the forward power with the phase of the signal coming out of the cavity, and keeping this phase constant. This ensures that the resonant frequency of the cavity is held constant. The

new upgrade replaces the analog phase comparator with a digital phase comparator. The digital phase comparator allows the user have some control of the phase comparator using ACNET parameters. The main reason this was added to the upgrade was to completely eliminate the need for the 5 wide Phase/Frequency control NIM module. The new digital phase comparator will be discussed in section 1.3

1.1 Feed Forward

The feed forward system works by injecting a signal into the gradient regulator to compensate for beam loading. The feed forward signal comes from a DAC on the MFC board. The gradient regulator controls the shape of the voltage for the modulator. As mentioned previously, the modulator voltage needs to be increased just before beam time to increase the RF power delivered to the cavity by the 7835 Power Amplifier. The problem with the old system was that the modulator voltage did not increase early enough or fast enough, and the beam quality suffered as a result. The new feed forward system solves this problem by increasing the voltage slightly ahead of the arrival of beam in the cavity to make sure that the RF power is correct when beam gets to the cavity. A diagram of the old toriod signal in relation to the new feed forward pulse is shown in figure 1.3.

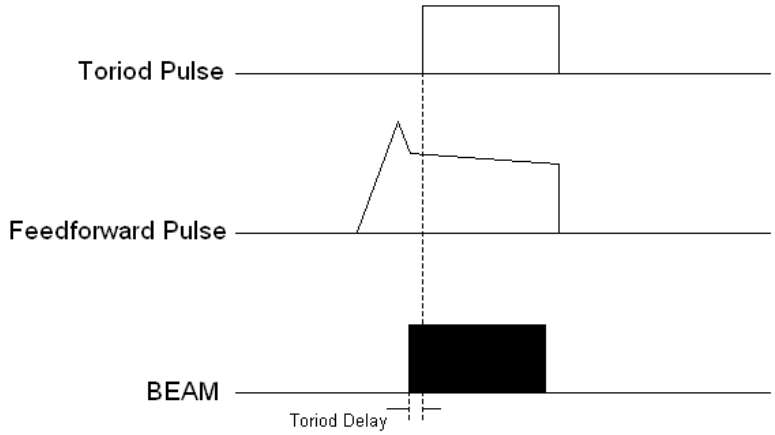


Figure 1.3: Toriod and Feed Forward.

A schematic of the gradient regulator is shown in figure 1.4. The gradient regulator is housed in a 1-wide NIM module in the rack at each low energy linac station. As seen in the figure, the gradient regulator, in the old system, summed the gradient program with the toroid signal and subtracted the detected gradient from the cavity. The gradient regulator is the summing junction of the amplitude feedback loop to control the gradient waveform. The new system replaces the toroid signal with the feed forward signal. There are gain adjustments, compensation capacitors, and compensation resistance on this input to add shape to the signal, but they are not used with the new feed forward system. The shaping capacitors are taken out of the circuit by switching front panel DIP switches to the OFF position on the front panel of the NIM module. All shaping and gain of the feed forward signal will be done using the digital system.

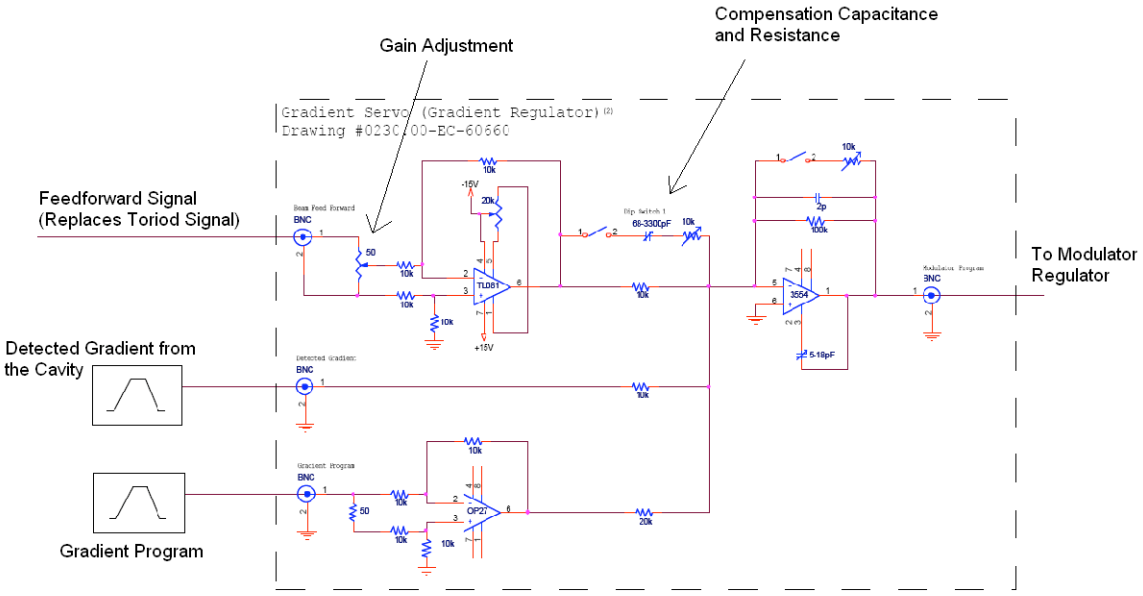


Figure 1.4: Gradient regulator schematic showing the feed forward input.

The feed forward system can be used in manual mode or adaptive mode. The difference between manual mode and adaptive mode is that the amplitude of the feed forward waveform is constantly adjusting in adaptive mode. In adaptive mode, the amplitude of the feedforward is stored in separate tables for three beam types, which are HEP, NTF, and Studies. The manual parameters are the same for each beam type. The manual parameters of the feedforward waveform is a slope on the front edge of the feedforward waveform, a spike on the front edge of the feedforward waveform, the amplitude of the feedforward waveform, and the slope of the amplitude. A diagram of all the feed forward parameters is shown in figure 1.5.

The adaptive correction will correct for amplitude errors that arise from such things as switch tube aging, RF tube aging, cavity Q variation, beam current changes, and other parameters that could affect the amplitude of the RF. In adaptive mode, the system samples the gradient from the cavity and calculates an amplitude error value, which is applied as a weighted amplitude correction on the next beam pulse.

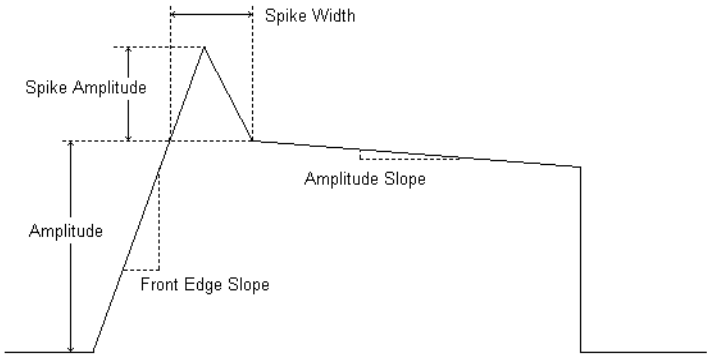


Figure 1.5: Parameters of feed forward waveform.

Timing information for the feed forward system is received by the MFC board from the control system. A simplified diagram of the timing information is shown in figure 1.6. The main timing signal from the control system is referred to as the Pre-Pulse signal. The Pre-Pulse signal contains information about when the beam will arrive, the beam type, and when to shut off the feed forward pulse. As seen in figure 1.6, the Beam-Qualifier portion of the Pre-Pulse tells the system that beam will arrive **t** seconds later, the width of the beam-qualifier tells the system what type of beam is coming, and the Chopper portion of the Pre-Pulse gives information about when to shut off the feed forward pulse.

The toriod signal has a different purpose in the new system. In the new system, the toriod signal is sent to the MFC board and used to provide a safety check for the feed forward system. The MFC board looks at the toriod signal to determine if beam is present a few microseconds after the feed forward is applied to the RF. If beam is not present the MFC board will shutdown the feed forward pulse. This is important so that RF power is not sent to the cavity while beam is not present, because a sustained increase of RF power may result in sparking across 9-Inch coax transmission line, which may damage the 7835 tube. With this safety check, only a few microseconds of RF power will be applied, and sparking will be avoided. As stated above, the Pre-Pulse signal also contains information about the beam type. The pulse width of the beam-qualifier portion determines what type of beam it is. The possible beam types are HEP, NTF, or Studies. This information enables the feed forward algorithm to use beam type specific tables to control the amplitude of the feed forward signal. A more detailed description of the timing system can be found in [5], and a more detailed description of the adaptive feed forward algorithm can be found in [6].

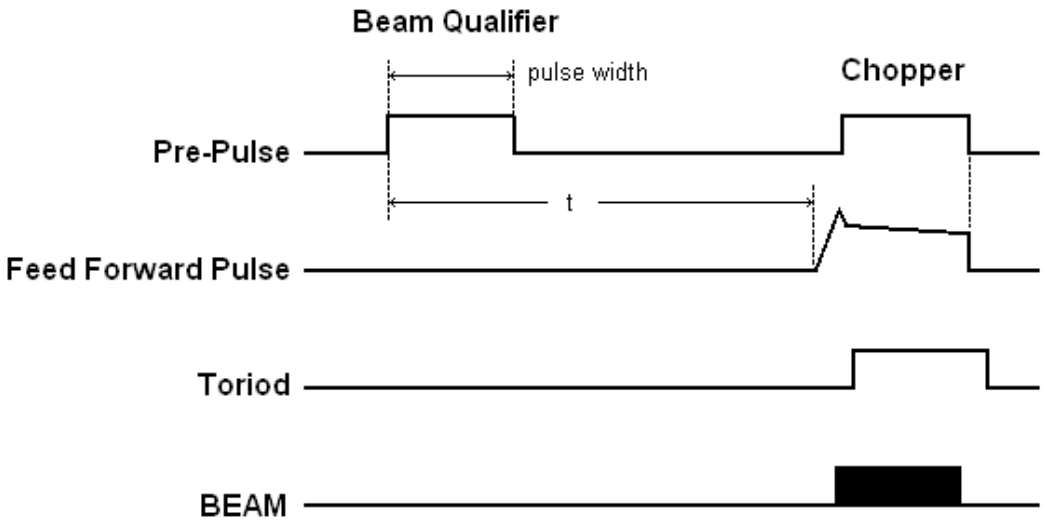


Fig 1.6. Timing information about the beam.

1.2 Inter-Tank Phase and Phase Feedback Upgrade

The old system used an inter-tank phase comparator system to keep the phase of the 201.25 MHz RF synchronous with the accelerating beam. This phase is referred to as the synchronous phase, and for the low enrgy linac, the synchrounous phase is -32 degrees. A diagram of the inter-tank phase comparator system is shown in figure 1.7.

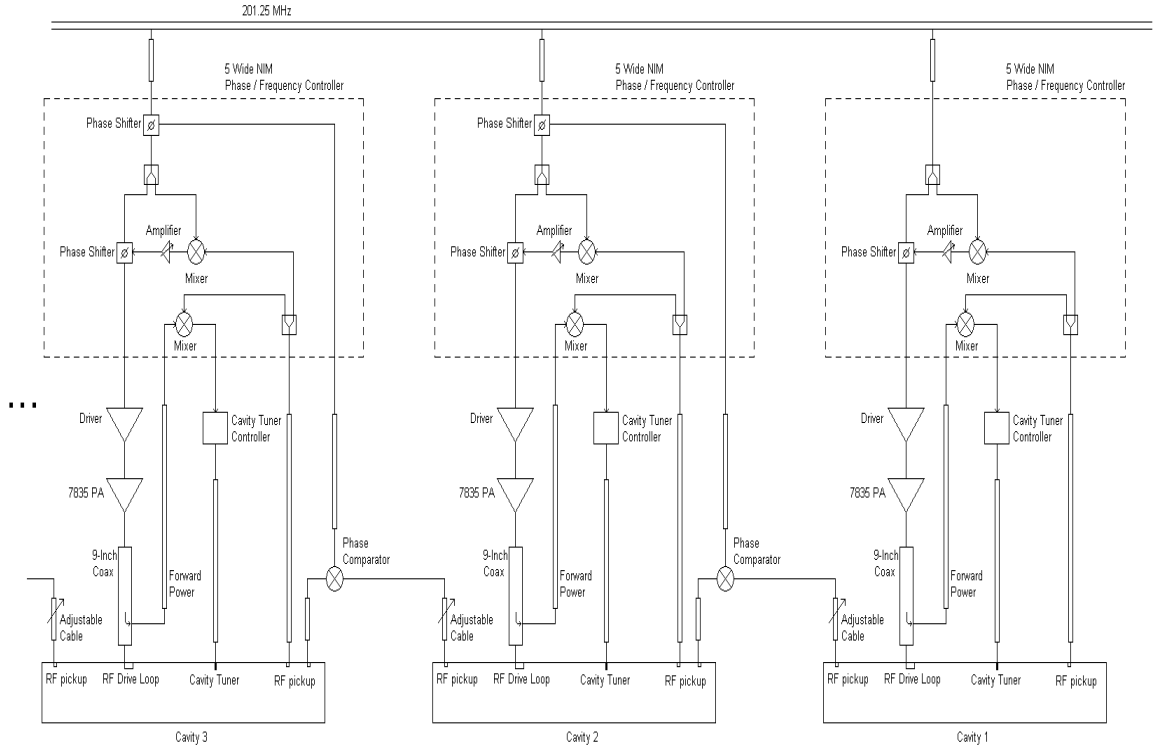


Fig 1.7. Old RF system

As seen in the figure, the phase of the RF in accelerating cavity n is compared to the phase of the RF in accelerating cavity n+1 using a phase comparator mixer. When the phase of each accelerating cavity is the correct synchronous phase, the output of the phase comparator is zero. There are adjustments to the cables that can be made to achieve this, as seen in the figure. Over time the phase of the RF may drift due to changes in RF amplification or other parameter that can cause the phase of the RF to change. These errors will create an error voltage on the phase comparator which sent to a phase shifter in the RF signal path to keep the phase at the correct synchronous phase. The drawback of this system is that if the phase of one station drifts, it causes every cavity down stream of it to make corrections to its phase. This produces ripple effect that is constantly adjusting itself.

The phase feedback of the old system was handled by a separate system and is also shown in figure 1.7. The phase feedback system was housed in the 5 wide NIM module. This analog system has a feedback gain and bandwidth defined by the amplifier and phase shifter. Changing the feedback gain requires dismantling the 5 wide NIM module that the amplifier is housed, and adjusting a potentiometer.

The new system uses a phase stable 805 MHz reference line that runs the length of the entire Linac, which was installed as part of an upgrade to the high energy 805 MHz

Linac[11]. Using this reference line allows each station to adjust its phase independently. A diagram showing the new system is shown in figure 1.8. The reference line resonates at 805 MHz, which is the operational frequency of the high energy Linac. This presents some design challenges, since the low energy Linac operates at 201.25 MHz. Ideally, a reference line resonating at 201.25 MHz would be used, but installing such a line proved to be disruptive to operations. The solution that is used is to divide the 805 MHz signal from the reference line by 4 to produce the 201.25 MHz at each station. Details are discussed in [2].

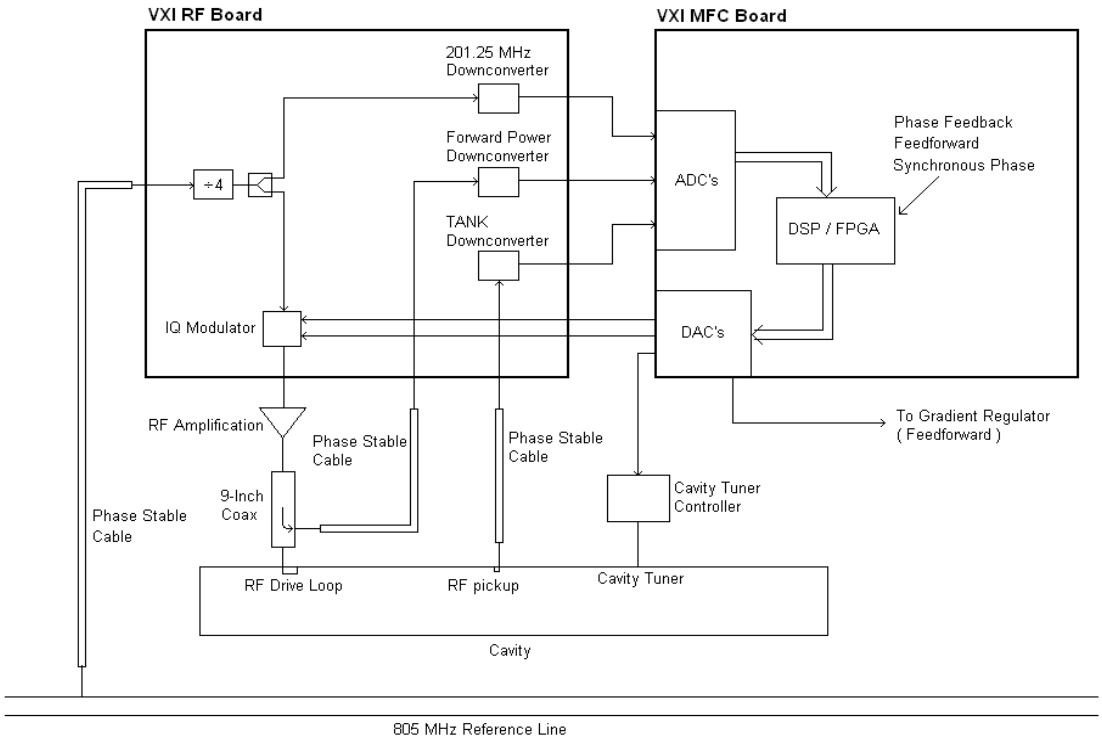


Fig 1.8. Upgrade system.

The 805 MHz Reference line is located in the tunnel and a phase stable cable is used to route the signal upstairs to the VXI RF Board at each station. The 805 MHz signal is divided by 4 to produce the 201.25 MHz signal for the RF system. The 201.25 MHz signal is split into two signals. The signal that is used to drive the cavity is referred to as the fanout. One signal is used to drive the cavity and one signal is used to provide a reference for phase comparison. The reference is downconverted to an IF frequency and is sent to one of the ADC's on the MFC board.

Another signal is taken from a RF pickup on the cavity and brought upstairs to the RF Board. This signal is also downconverted to an IF frequency and sent an ADC on the MFC board. This signal is referred to as the fanback. The reference signal and the cavity signal are used in a phase feedback system to hold the phase in the cavity constant

with respect to the 805 MHz reference line. ACNET parameters are used to control the fanback phase, fanout phase, and phase feedback gain. The advantage of this system is that the phase of the cavity RF can be adjusted independently without affecting the upstream cavities of the linac, unlike the inter-tank phase system in the old system. The phase can also be rotated through 360 degrees because the phase of the RF is controlled by an IQ modulator on the RF board. The new phase feedback system is described in more detail in [10]. The reference line downconverter, cavity downconverter, and IQ modulator is described in more detail in [2].

1.4 Cavity Tuning

The components of the old cavity tuning system are included in figure 1.7. The system uses an analog phase comparator to keep the phase shift of the RF through the cavity constant. The new system replaces the analog phase comparator with a digital phase comparator. The new system includes ACNET parameters for loop gain and phase comparator offset.

2 Installation

Installation of the new upgrade hardware is described in this section. A diagram of all the new cables and connections is shown in figure 2.1.

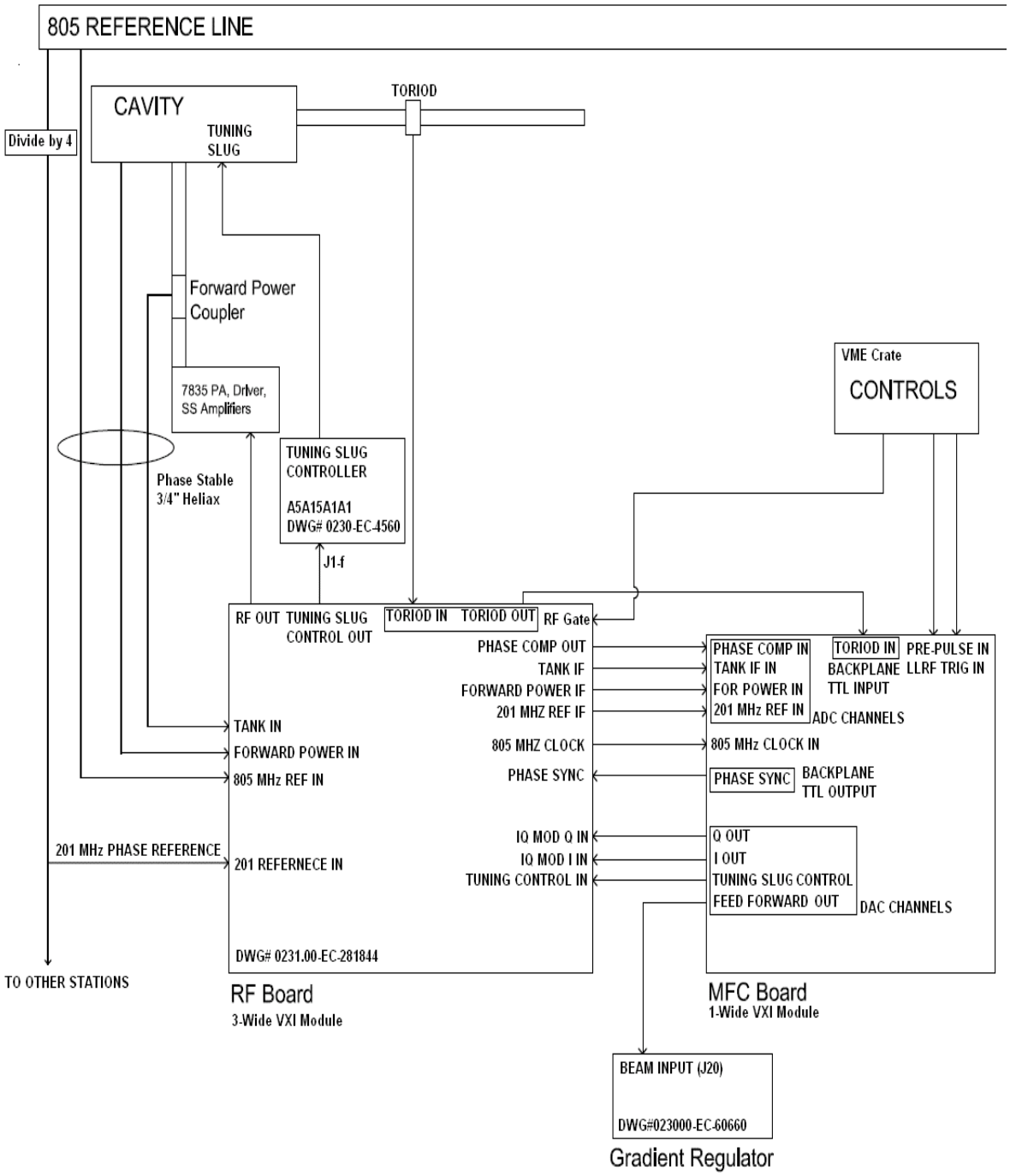


Figure 2.1: All cable connections of the upgrade.

A picture of the rack setup is shown in figure 2.2. The picture shows the VXI chassis mounted in the rack. The VXI chassis is a C size 13-slot Agilent model E1401B, capable of providing 900 Watts of power. The VXI chassis has lockable front panel for access to the slot0 controller, MFC board, and RF board. Another picture of the VXI modules is shown in figure 2.3. The slot 0 controller, the MFC Board, and the RF Board can be seen in the photo.

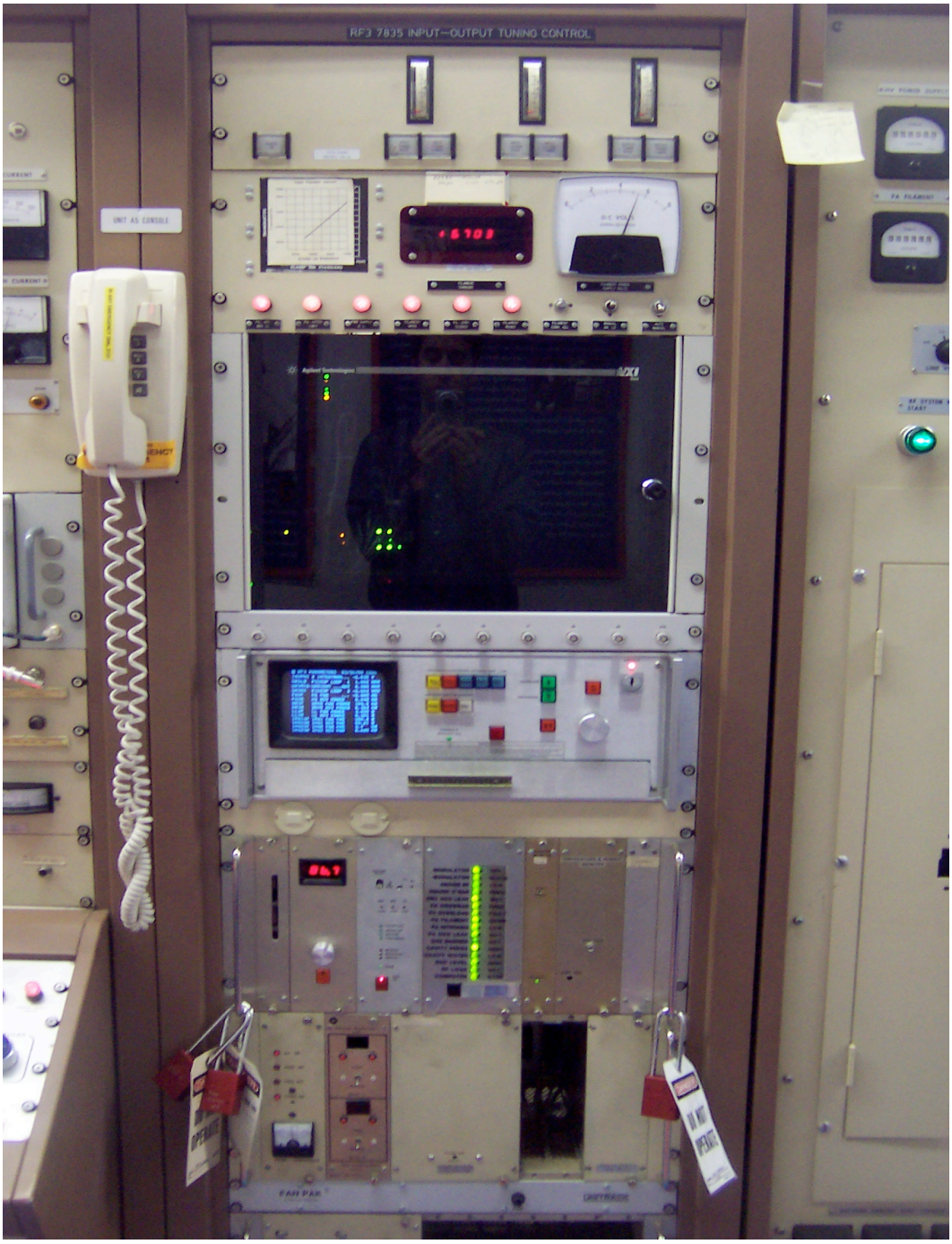


Figure 2.2: Picture of the rack setup housing the VXI chassis at each station.

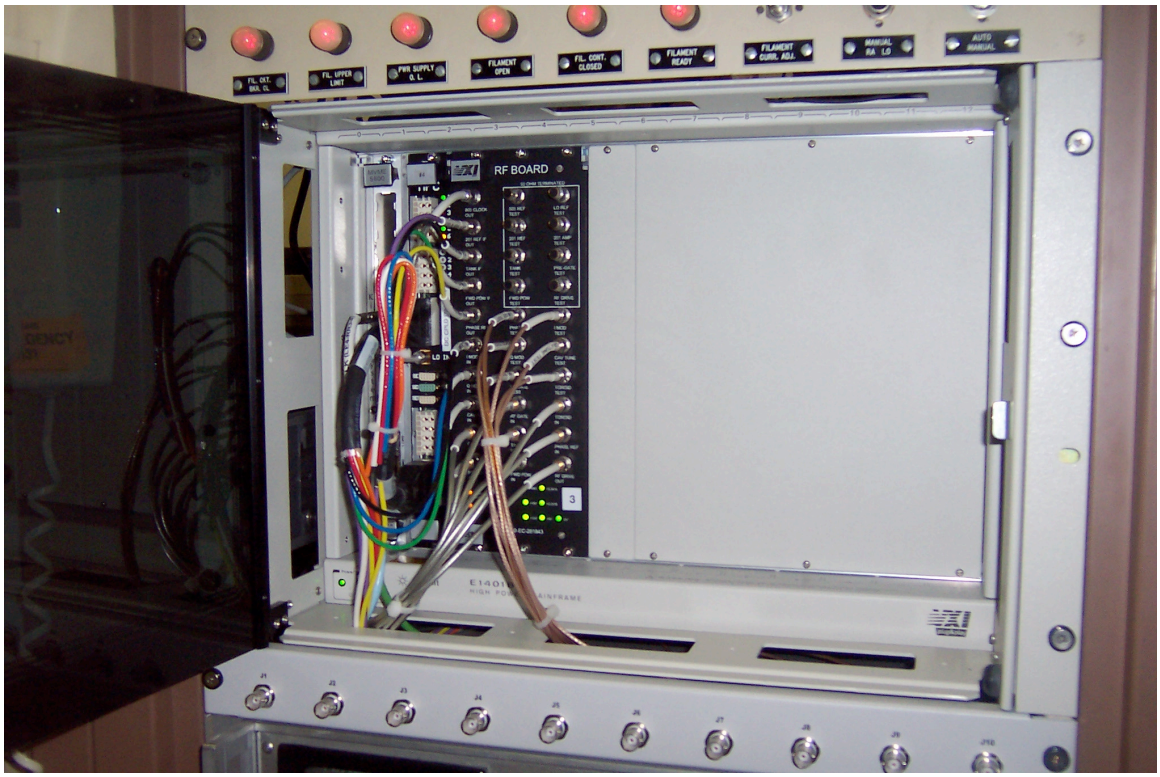


Figure 2.3: Picture of the VXI modules.

Four phase stable cables were pulled from the tunnel to the rack. The cable is heliax 3/8 Inch phase stable cables, model # xxx. Three of the cables provide for the TANK signal, 805MHz Reference line, and Forward Power signals. A fourth cable was pulled as a spare. A picture of the cables connecting to the backside of the VXI chassis is shown in figure 2.4. From the back of the VXI chassis, shorter lengths of cables are connected from the cables to the RF Board and MFC Board.

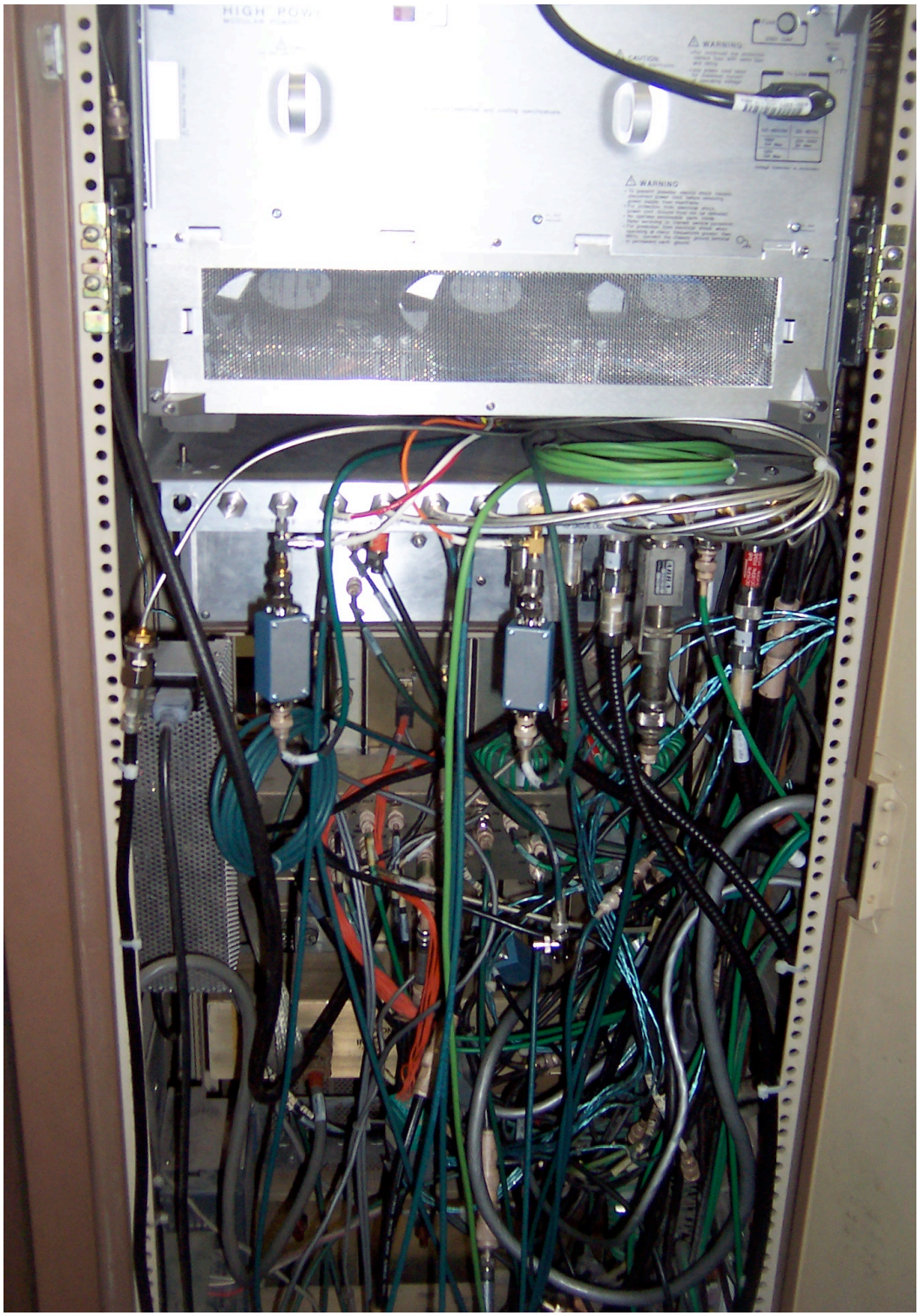
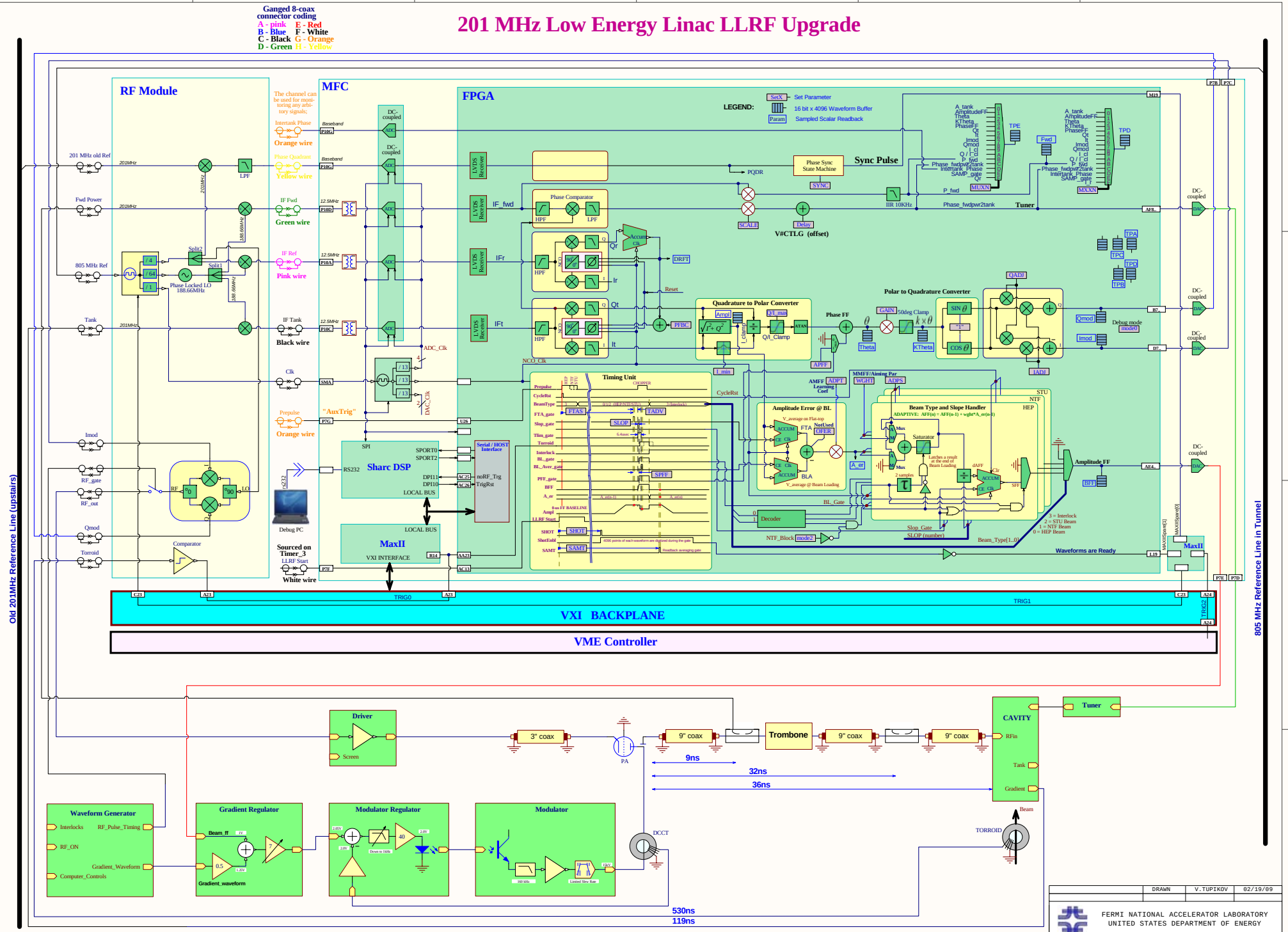


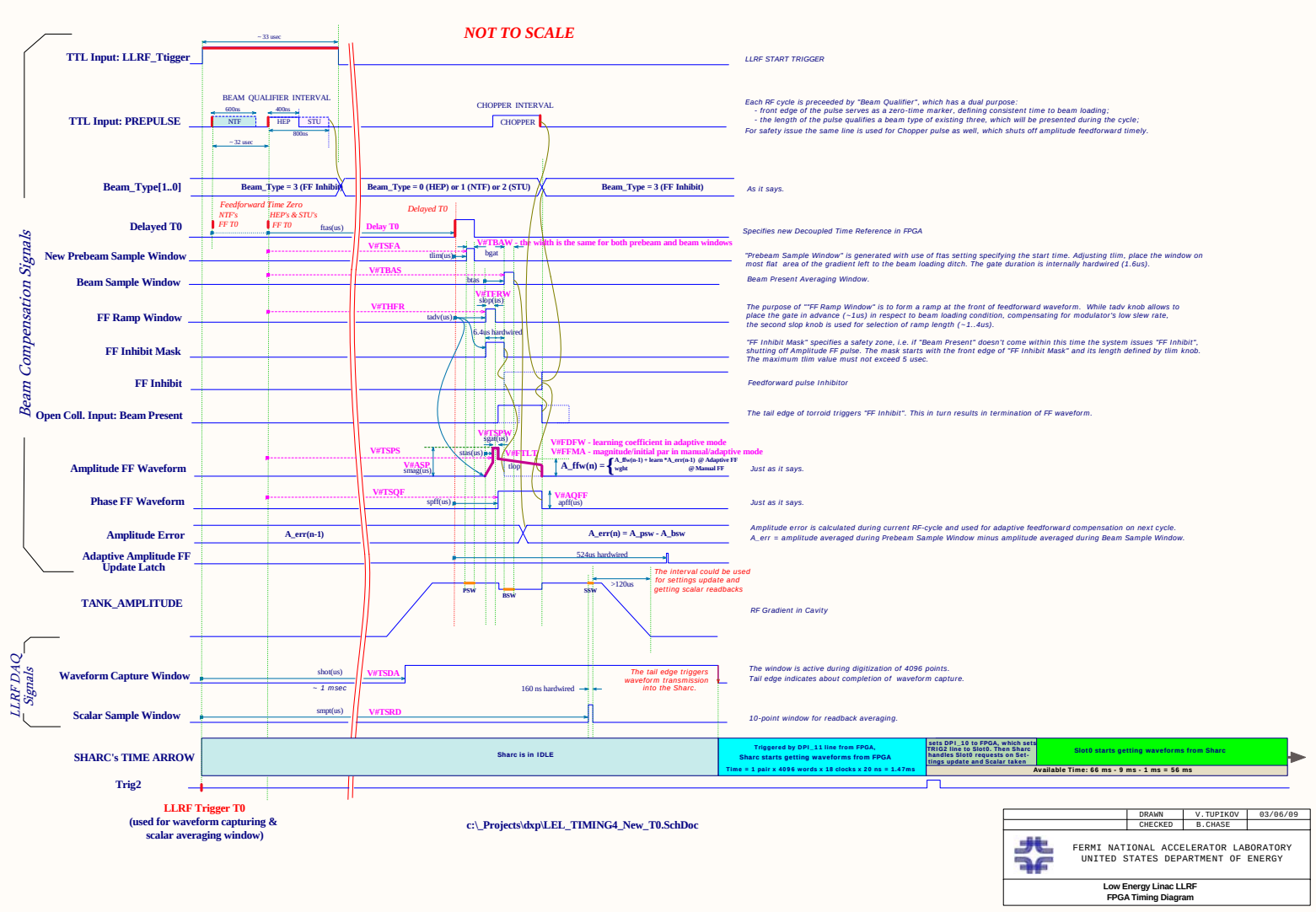
Figure 2.4: Picture of the cable connections to the VXI chassis.

References

[1] Varghese, P. et al., 2007, “Multichannel Vector Field Control Module For LLRF Control of Superconducting Cavities”, Fermilab note.
[2] Cullerton, E., 2009, “VXI RF Board”, Fermilab note.
[3] Joireman, P., 2009, “Linac Low Level Upgrade Controls/Front-end Interface”, Fermilab note.)
[4] Cullerton, E., 2009, “Beam Loading Compensation For the Fermilab Low Energy Linac”, Fermilab note.
[5] Butler, T., 2009, “Linac low Energy LLRF timing System”, Fermilab note.
[6] Tupikov, 2009, “MFC FPGA Program for the low energy Linac”, Fermilab note.
[7] Paper about the High Energy Linac Upgrade (Brian Chase, Ralph Pasquinelli)
[8] Cullerton, E., 2009, “Low Energy Linac Operational Guide”, Fermilab note.

201 MHz Low Energy Linac LLRF Upgrade




NOT TO SCALE
Beam Compensation Signals
LLRF DAQ Signals
TTL Input: LLRF_Ttigger
~ 33 usec
LLRF START TRIGGER
BEAM QUALIFIER INTERVAL
600ns
400ns
NTF
HEP
STU
800ns
~ 32 usec
CHOPPER INTERVAL
CHOPPER
TTL Input: PREPULSE
Each RF cycle is preceeded by "Beam Qualifier", which has a dual purpose:
- front edge of the pulse serves as a zero-time marker, defining consistent time to beam loading;
- the length of the pulse qualifies a beam type of existing three, which will be presented during the cycle;
For safety issue the same line is used for Chopper pulse as well, which shuts off amplitude feedforward timely.
Beam_Type[1..0]
Beam_Type = 3 (FF Inhibit)
Beam_Type = 0 (HEP) or 1 (NTF) or 2 (STU)
Beam_Type = 3 (FF Inhibit)
As it says.
Feedforward Time Zero
NTF's FF T0
HEP's & STU's FF T0
Delayed T0
Delay T0
ftas(us)
Delayed T0
Specifies new Decoupled Time Reference in FPGA
New Prebeam Sample Window
V#TSFA
tlim(us)
V#TBAW - the width is the same for both prebeam and beam windows
bgat
"Prebeam Sample Window" is generated with use of ftas setting specifying the start time. Adjusting tlim, place the window on most flat area of the gradient left to the beam loading ditch. The gate duration is internally hardwired (1.6us).
Beam Sample Window
V#TBAS
btas
Beam Present Averaging Window.
V#TFRW
slop(us)
FF Ramp Window
V#THFR
tadv(us)
6.4us hardwired
The purpose of ""FF Ramp Window" is to form a ramp at the front of feedforward waveform. While tadv knob allows to place the gate in advance (~1us) in respect to beam loading condition, compensating for modulator's low slew rate, the second slop knob is used for selection of ramp length (~1..4us).
FF Inhibit Mask
"FF Inhibit Mask" specifies a safety zone, i.e. if "Beam Present" doesn't come within this time the system issues "FF Inhibit", shutting off Amplitude FF pulse. The mask starts with the front edge of "FF Inhibit Mask" and its length defined by tlim knob. The maximum tlim value must not exceed 2 usec.
FF Inhibit
Feedforward pulse Inhibitor
Open Coll. Input: Beam Present
The tail edge of torroid triggers "FF Inhibit". This in turn results in termination of FF waveform.
V#TSPW
sgat(us)
V#FDFW - learning coefficient in adaptive mode
V#FFMA - magnitude/initial par in manual/adaptive mode
V#TSPS
stas(us)
V#FTLT
tlpp
A_ffw(n) = { A_ffw(n-1) + learn *A_err(n-1) @ Adaptive FF ; wglst @ Manual FF }
V#ASP
smag(us)
Amplitude FF Waveform
Just as it says.
V#TSQF
spfl(us)
V#AQFF
apfl(us)
Phase FF Waveform
Just as it says.
Amplitude Error
A_err(n-1)
A_err(n) = A_psw - A_bsw
Amplitude error is calculated during current RF-cycle and used for adaptive feedforward compensation on next cycle.
A_err = amplitude averaged during Prebeam Sample Window minus amplitude averaged during Beam Sample Window.
Adaptive Amplitude FF Update Latch
524us hardwired
The interval could be used for settings update and getting scalar readbacks
PSW
BSW
SSW
>120us
TANK_AMPLITUDE
RF Gradient in Cavity
Waveform Capture Window
shot(us)
~ 1 msec
V#TSDA
160 ns hardwired
The tail edge triggers waveform transmission into the Sharc.
The window is active during digitization of 4096 points.
Tail edge indicates about completion of waveform capture.
Scalar Sample Window
smpt(us)
V#TSRD
10-point window for readback averaging.
SHARC's TIME ARROW
Sharc is in IDLE
Triggered by DPI_11 line from FPGA, Sharc starts getting waveforms from FPGA
Time = 1 pair x 4096 words x 18 clocks x 20 ns = 1.47ms
sets DPI_10 to FPGA, which sets TRIG2 line to Slot0. Then Sharc handles Slot0 requests on Settings update and Scalar taken
Slot0 starts getting waveforms from Sharc
Available Time: 66 ms - 9 ms - 1 ms = 56 ms
Trig2
LLRF Trigger T0
(used for waveform capturing & scalar averaging window)
c:_Projects\dxp\LEL_TIMING4_New_T0.SchDoc
DRAWN V.TUPIKOV 03/06/09
CHECKED B.CHASE
FERMI NATIONAL ACCELERATOR LABORATORY
UNITED STATES DEPARTMENT OF ENERGY
Low Energy Linac LLRF
FPGA Timing Diagram

Low Energy Linac Operational Guide

Ed Cullerton 4/15/09

1. System Calibration

This section describes the phase control and phase calibration of the Low Energy Linac.

A diagram of the new low level linac phase control system is shown in figure 1.

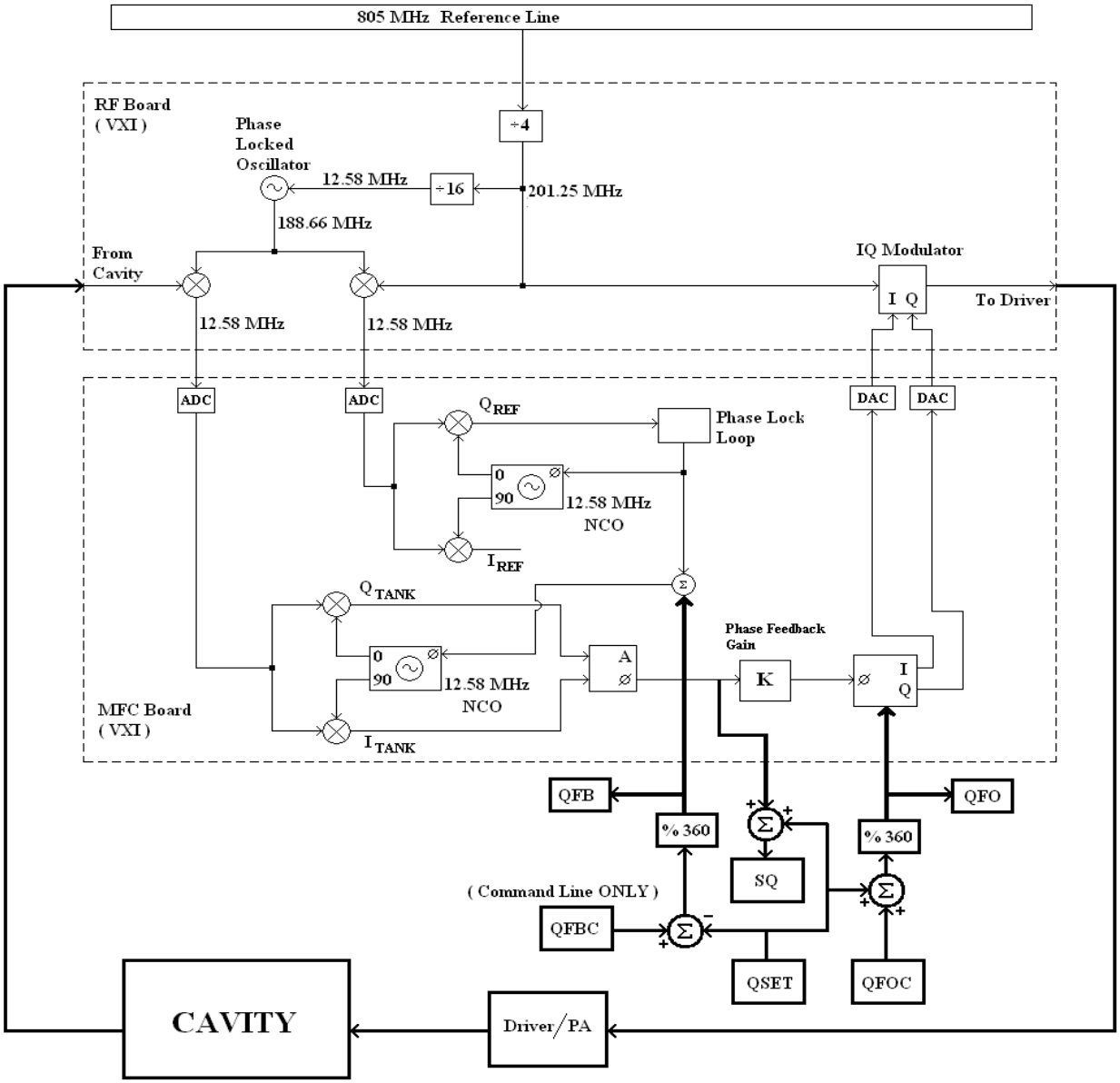


Figure 1.

The parameters used for phase control and readback are:

- L:V#QSET – This parameter allows the user to rotate the phase of the RF inside the cavity. The nominal setting is -32 degrees, which coincides with the synchronous phase angle described in common accelerator literature.
- L:V#SQ – This is a readback for the phase. This readback will show any drift or change in the RF phase in the cavity relative to the 805 MHz reference line. This read back should be data logged to observe long term phase stability.

The parameters used for phase calibration are:

- L:V#QFBC – This is the phase fanback calibration. This parameter can only be adjusted through the use of a terminal command line, and is reserved expert users. This parameter should not need any adjustment after initial commissioning of the system.
- L:V#QFOC -This is the phase fanout calibration. This parameter should only be adjusted when there is significant change of the phase in the RF path to the cavity.

Calibration of the system due to changes in RF path (fanout)

When calibrating the fanout, the phase gain should be set to zero (L:V#QGN=0). This opens the phase feedback loop so that adjustments to the fanout phase can be made. A requirement for the phase feedback system is that Qtank ~=0.

Qtank can be observed using the scalar readback L:V#STKQ, or the Qtank waveform may be observed on a D27 plot.

The fanout phase calibration (L:V#QFOC) should be adjusted so that Qtank ~= 0. Once this requirement is fulfilled the phase feedback loop may be closed again(L:V#QGN).

Adjusting the 7835/Driver input and output tuning

Care should be taken when adjusting any component that may change the phase in the RF drive path. A large phase deviation may cause the phase feedback system to go unstable. It is recommended that the feedback gain be turned down to a small value when adjustments are being made (-3 or less).

After adjustments are complete, it is necessary to calibrate the fanout phase. Follow the steps outlined in the section “calibration of the system due to changes in RF drive path”

2. Adjusting Amplitude Feed Forward

Initial adjustment procedure:

Make sure adaptive feed forward is OFF. (L:V#MDSW[8]=0)

Set the feed forward spike amplitude (L:V#ASP) to zero.

Set the feedforward ramp width (L:V#TFRW) to a small value. (0.1) Zero values are not allowed.

Set the feed forward tilt (L:V#FTLT) to zero.

Set the manual feed forward amplitude (L:V#FFMA) to 0.4.

Set the gain potentiometer on the front panel of the Gradient Regulator to its lowest setting so that no beam loading compensation is applied

Start beam.

Increase the gain potentiometer on the front panel of the Gradient Regulator and check to see if there is beam loading compensation.

If no beam loading compensation is observed, adjust the start time (L:V#THFR) of the feed forward pulse.

Once the start time is correct, beam loading compensation should be observed.

Adjust the potentiometer again so that a good level of beam loading compensation is observed.

Once this gain potentiometer is set, it should not be adjusted again.

Operational adjustments:

Make adjustments to the following parameters until a satisfactory beam loading profile is observed.

L:V#TFRW – Sets the total rise time of the ramp on the front edge of the feed forward pulse.

L:V#THFR – Sets the start time of the feed forward pulse.

L:V#FFMA –Sets the amplitude of the feed forward in manual mode.

L:V#ASP –Sets the amplitude of the feed forward spike.

L:V#TSPW – Sets the width of the feed forward spike.

L:V#FTLT –Sets the amount of tilt during the flat top portion of the feed forward pulse.

The final adjustment to the feed forward should be the chopper shut off. The shut off of the feed forward pulse can be delayed by adjusting L:V#TCHP.

3. Adjusting Phase Feed Froward

Make adjustments to the following parameters.

L:V#TSQF – Sets the phase feed forward start time.

L:V#AQFF – Sets the phase feed forward amplitude.

4. ACNET Parameters

L:V#QMNI - (EXPERT) Sets the minimum value of I in the Q/I function of the phase calculation of the FPGA code.

L:V#QMQI –(EXPERT) Sets the maximum value of the Q/I function of the phase calculation of the FPGA code.

L:V#QSYN – (EXPERT) Resets the phase quadrant lock system. Set to 1 to reset. Value will return to zero after setting.

L:V#SQUA – Readback of phase quadrant indicator. A value around +3 volts indicates the phase is in the correct quadrant.

L:V#ARF – (EXPERT) Sets the level of the 201.25 MHz RF out of the RF Board.

L:V#SGRD – Readback of the cavity gradient.

L:V#TSRD – Sets the time for all scalar readbacks.

L:V#TSDA – Sets the start time for waveform data capture.

L:V#STAT – Readback of slot 0 status. (4 indicate good condition)

L:V#TCHP – (EXPERT) Sets the stop time of the feed forward pulse relative to the chopper portion of the Pre-Pulse.

L:V#TODL – (EXPERT) Sets the timing offset used by the FPGA code. Set to 970 us.

L:V#TSFA – (EXPERT) Sets the start time for gradient flat top averaging. (Used to calculate beam loading error)

L:V#THFR – (EXPERT) Sets the start time of the feed forward pulse.

L:V#TSPS – (EXPERT) Sets the start time of the feed forward spike.

L:V#TBAS – (EXPERT) Sets the start time for measuring the gradient during beam loading.

L:V#TFRW – (EXPERT) Sets the total rise time of the ramp on the front edge of the feed forward pulse.

L:V#TSPW – (EXPERT) Sets the width of the feed forward spike.

L:V#FTLT – (EXPERT) Sets the amount of tilt during the flat top portion of the feed forward pulse.

L:V#TCHP – (EXPERT) Sets the stop time of the feed forward pulse relative to the chopper portion of the Pre-Pulse.

L:V#ASP – (EXPERT) Sets the amplitude of the feed forward spike.

L:V#FFMA – (EXPERT) Sets the amplitude of the feed forward in manual mode.

L:V#FDFW – (EXPERT) Sets the adaptive feed forward weighting value. A Y/N indicator indicates whether adaptive mode is ON or OFF.

L:V#TBAW –(EXPERT) Sets the width of the gradient averaging during beam.

L:V#QFB – Readback of the fanback phase setting to the FPGA.

L:V#QFO – Readback of the fanout phase setting to the FPGA.

L:V#QGN – (EXPERT) Sets the phase feedback gain.

L:V#STKQ – Readback of the Q component of the TANK signal from the cavity.

L:V#STKI – Readback of the I component of the TANK signal from the cavity.

L:V#SGRD – Readback of the gradient amplitude.

L:V#QSET – (EXPERT) Sets the accelerating phase angle in the cavity.

L:V#QFOC – (EXPERT) Sets the fan out phase.

L:V#SQ – Readback of the cavity phase.

L:V#TSQF – (EXPERT) Sets the phase feed forward start time.

L:V#AQFF – (EXPERT) Sets the phase feed forward amplitude.

L:V#QFBC – Readback of the fanback phase setting.

L:V#SMQ – Readback of the Q value applied to the IQ modulator.

L:V#SMI – Readback of the I value applied to the IQ modulator.

L:V#SRPE – Readback of the reference line phase locked loop.

L:V#WAMX – Sets which waveform is sent through the multiplexer for channel A data acquisition.

L:V#WBMX – Sets which waveform is sent through the multiplexer for channel B data acquisition.

L:V#CTLG – (EXPERT) Sets the cavity tuner loop gain.

L:V#COFF – (EXPERT) Sets cavity tuner offset.

L:V#MODE – Readback of the decimal value of the binary switch settings.

L:V#MDSW[1] – Sets the data acquisition sample rate.

L:V#MDSW[2] – NTF beam loading compensation. (1=OFF 0=ON)

L:V#MDSW[5] – Freeze waveform data acquisition.

L:V#MDSW[7] – Trigger waveform data acquisition only during beam.

L:V#MDSW[8] – Turn ON or OFF adaptive beam loading compensation. (1=ON 0=OFF)

L:V#WDGA – Waveform data acquisition parameter.

L:V#WDGB – Waveform data acquisition parameter.

Low Energy Linac Beam Loading Compensation

Ed Cullerton 4/15/09

Shown in figure 1 is a spreadsheet of the low energy Linac beaming loading calculations. The spreadsheet includes cavity model data, cavity Q calculations, power for each station, optimum cavity detuning angle, and beam energy at each station.

Measured data of the cavity coupling was taken in the 2006 shutdown, and measured data of cavity impedance was taken from [1]. This data is used in the spreadsheet to determine if the actual coupling coefficients agree with designed coupling coefficients. The spreadsheet concludes that station 3 and station 5 had coupling coefficients that did not agree with the original design values. Since the data was taken, station 3 has been adjusted to its design value of coupling and improved performance was observed at the station. Station 5 has not been adjusted to the original design value as of this time.

	A	B	C	D	E	F	G	H	I	J	K	L
1	**Cavity Calculations**											
2	Ed Cullerton			Data form	s11 measurement of cavity							
3	12/13/2006			Data from	1970 Procedings Linac Conference Table II, p22							
4												
5	Beam current	45	mA									
6	Syncrounous Phase Angle	-32	degrees									
7	Generator Frequency	201.24063	MHz									
8												
9	**Station #**		**1**		**2**		**3**		**4**		**5**	
10	Unloaded Q		63134		65047		61921		59627		54567	
11	Proton Energy IN		0.75	MeV	10.42	MeV	37.54	MeV	66.2	MeV	92.6	MeV
12	Proton Energy OUT		10.42	MeV	37.54	MeV	66.18	MeV	92.6	MeV	116.5	MeV
13	Effective Shunt impedance	first cell	27	Mohm/m	53.5	Mohm/m	44.6	Mohm/m	35	Mohm/m	29.6	Mohm/m
14		last cell	47.97	Mohm/m	44.8	Mohm/m	35.2	Mohm/m	28.5	Mohm/m	25	Mohm/m
15		Average	37.485	Mohm/m	49.15	Mohm/m	39.9	Mohm/m	31.75	Mohm/m	27.3	Mohm/m
16	Cavity Length		7.44	m	19.02	m	16.53	m	16.68	m	15.58	m
17	Total effective shunt Impedance (measured)		278.8884	Mohm	934.833	Mohm	659.547	Mohm	529.59	Mohm	425.334	Mohm
18												
19	Power Delivered to the Beam		0.43515	Mwatts	1.2204	Mwatts	1.2888	Mwatts	1.188	Mwatts	1.0755	Mwatts
20	Cavity Voltage		11402651.8	Volts	31979308.87	Volts	33771659.52	Volts	31130300.67	Volts	28182355.53	Volts
21												
22	Shunt impedance		103.55	Mohm	334.16	Mohm	255.8	Mohm	210.48	Mohm	172.4	Mohm
23	Power Dissapted in Cavity Wall		0.627814911	Mwatts	1.53021935	Mwatts	2.229329528	Mwatts	2.302108561	Mwatts	2.303495253	Mwatts
24	Power Dissapted in Cavity Wall (Measured)		0.61	Mwatts	1.38	Mwatts	2.245	Mwatts	2.48	Mwatts	2.49	Mwatts
25	Optimum Beta		1.693118294		1.79753272		1.578111035		1.516048643		1.46689916	
26	Measured Beta		1.683		1.762		1.108		1.505		1.085	
27	Effective Impedance (Check)		278.8723993	Mohm	934.8235336	Mohm	659.4808027	Mohm	529.5779185	Mohm	425.2934153	Mohm
28	Shunt impedance (Check)		103.5559413	Mohm	334.1633838	Mohm	255.8256767	Mohm	210.4848018	Mohm	172.4164517	Mohm
29												
30	Optimum Detuning Angle		9.136101859	Degrees	10.10073817	Degrees	7.97632918	Degrees	7.30335288	Degrees	6.744827127	Degrees
31	Loaded Q		23442.71328		23251.56004		24017.97253		23698.66741		22119.67188	
32	Optimum Cavity Frequency		201.2413203	MHz	201.2414009	MHz	201.241217	MHz	201.2411742	MHz	201.241168	MHz
33	Beam Voltage magnitude		3416586.873	Volts	10583710.09	Volts	8843402.515	Volts	7467866.023	Volts	6246147.064	Volts
34	Beam Voltage Phase		189.1361019	Degrees	190.1007382	Degrees	187.9763292	Degrees	187.3033529	Degrees	186.7448271	Degrees
35	p1 (for calculations)		1.54111237		1.60064686		1.469900003		1.428803845		1.39484317	
36	Generator Power		1.062964911	Mwatts	2.75061935	Mwatts	3.518129528	Mwatts	3.490108561	Mwatts	3.378995253	Mwatts
37	Reflected Power		0	Watts	0	Watts	0	Watts	0	Watts	0	Watts
38												
39	Cavity Model L		1.297149814	uH	4.062845966	uH	3.267127943	uH	2.791717903	uH	2.498681222	uH
40	Cavity Model C		0.482188161	pF	0.153948682	pF	0.191443658	pF	0.22404527	pF	0.250320539	pF
41	Cavity Model Transformer		1105.977771		1928.207762		1800.513946		1666.340874		1533.145945	
42												
43	**How to determine the original design values**											
44	1. Set the beam current (B5) to a random number											
45	2 a. Adjust the shunt impedance (C22) so that the value is equal to the shunt impedance check cell (C28).											
46	2 b. Also use the effective impedance check cell to see if the effective shunt impedance matches the measured data.											
47	3. See if the optimum Beta (C25) matches the measured Beta (C26). It is best to check station 1,2 & 4 for this, due to coupling changes in 3 & 5.											
48	4. Check the power dissapated (C23) in the cavity walls with the measured data (C24) to make sure the numbers are reasonable											
49	5. If the number are not close, change the beam current and repeat.											
50	6. Once an acceptable beam current and shunt impedance is found, the design can be changed by simply changing the beam current. The shunt impedance should not be altered.											
51												
52	45 mA - original design current											
53	Sunt impedance values		103.55		334.16		255.8		210.48		172.4	
54												

Figure 1. Beam loading compensation calculations.

A MATLAB program has also been written to help visualize the voltage vectors inside the cavity. Figure 2 shows the MATLAB program. The data in the fields are for station 5.

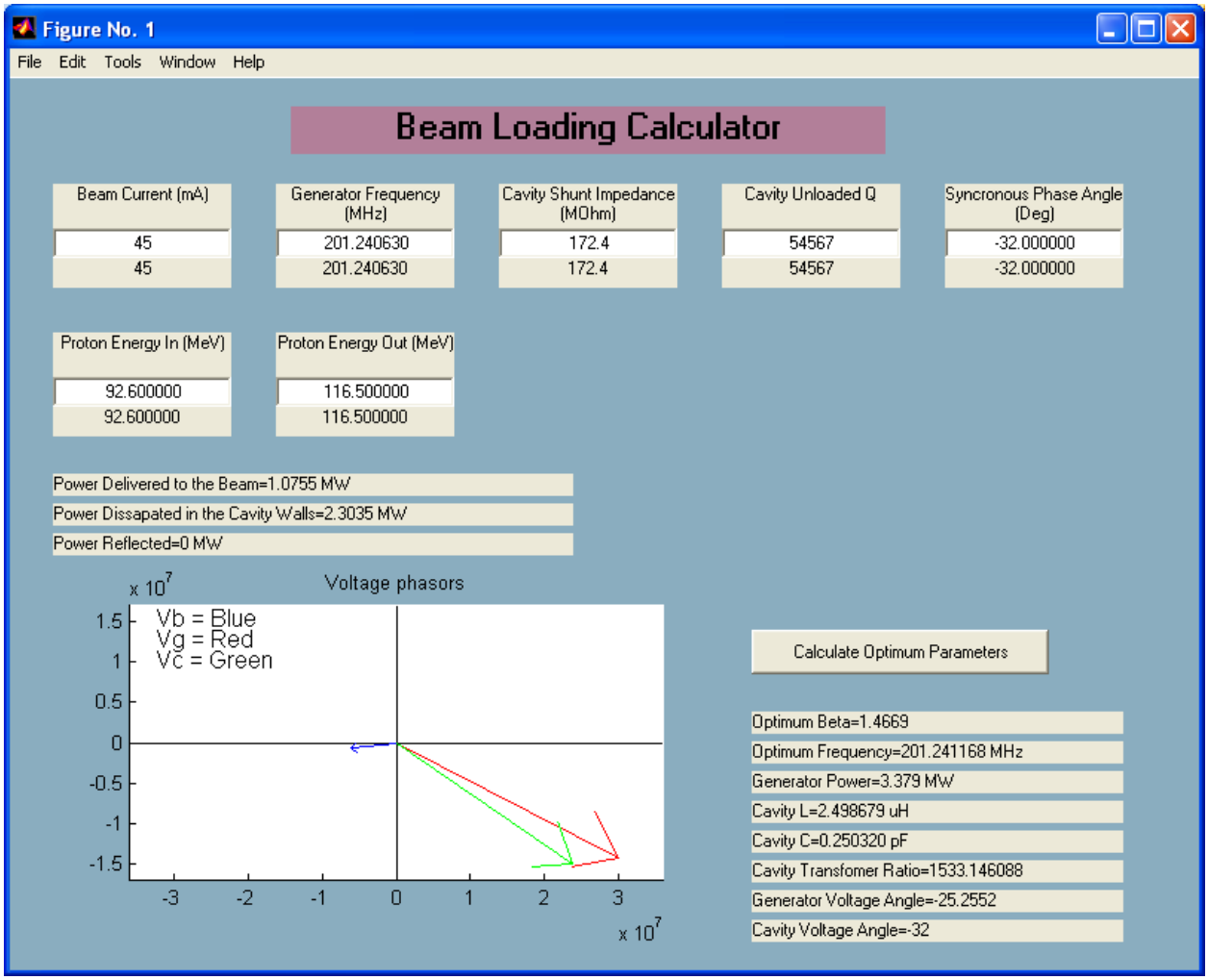


Figure 2. MATLAB program to visualize cavity voltage vectors.

[1] 1970 Proceedings, Linac Conference, Table II, p22.

Low Energy Linac VXI RF Board

Ed Cullerton#, Brian Chase, Ralph Pasquinelli

4/15/09

Abstract:

The VXI RF Board for the low energy Linac upgrade is described in this paper. The RF Board is responsible for all the analog RF signal processing associated with the upgrade. The RF signal processing includes clock division, downconversion, IQ phase modulation, RF gating, phase locked oscillator, phase comparison, filtering, and low level amplification. Other analog signal processing includes comparator circuits, voltage gain and offset circuits, test point buffers, and clock divider reset circuitry. This paper will describe in detail each component of the RF Board.

1. VXI RF Board Overview

A block diagram of the VXI RF Module is shown in figure 1.1. Each section of the block diagram will be covered in this overview. More detailed sections of the official schematic will be shown to clarify the details. The official schematic is drawing# 0231.00-EC-281843.

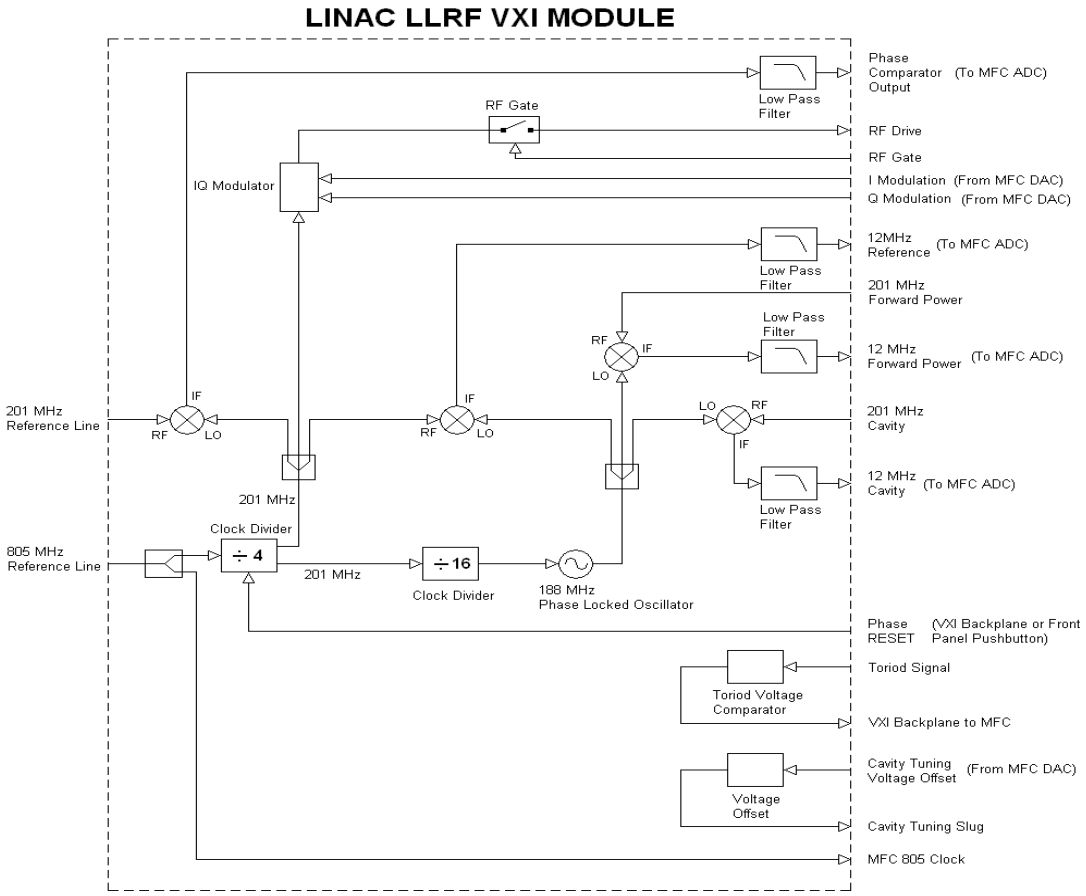


Figure 1.1: Block Diagram of the VXI RF Board.

1.1 RF Drive

All the components in the signal path to the RF Drive output will be described in this section. A block of the RF Drive path is shown in figure 1.2.

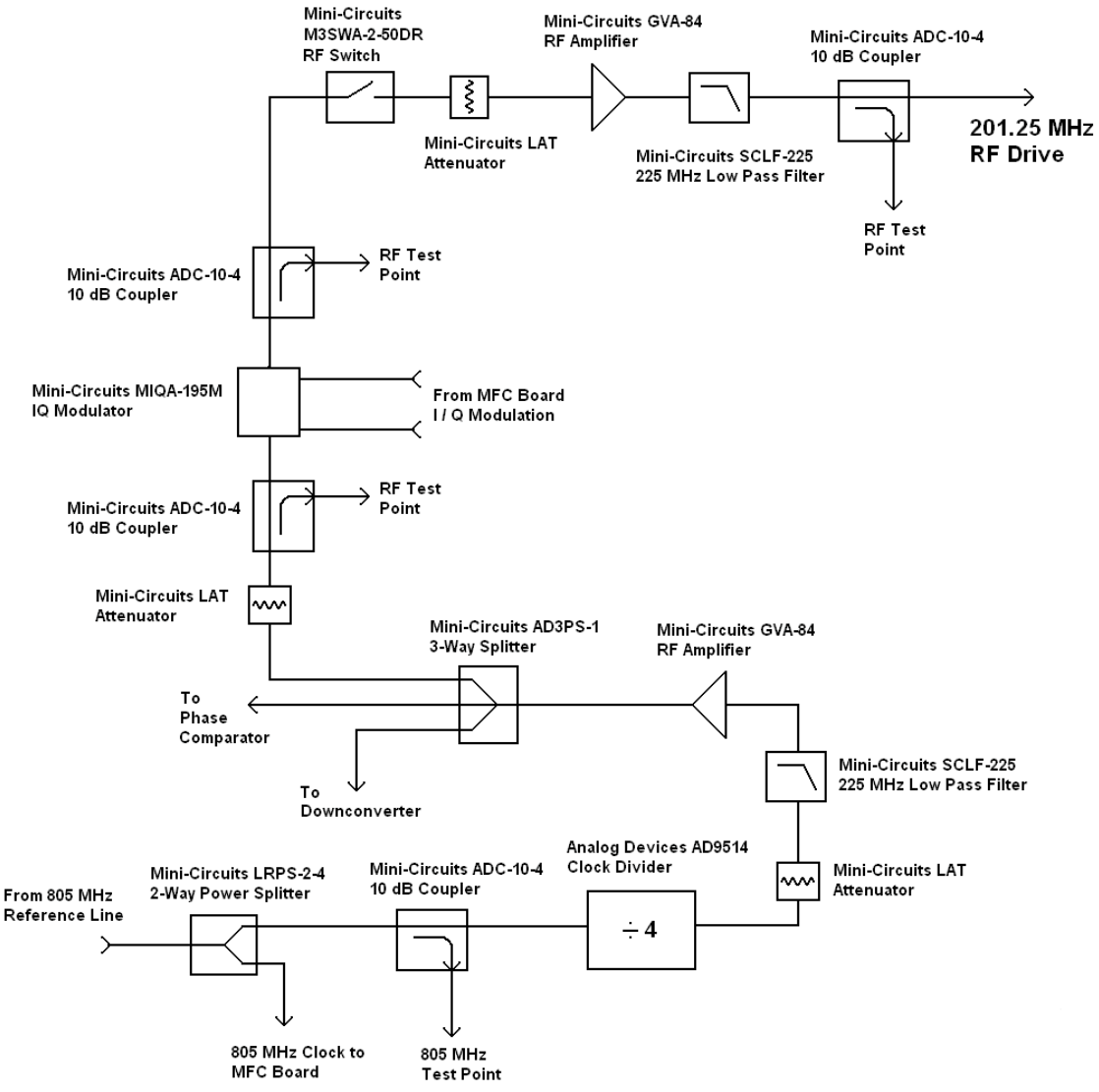


Figure 1.2: Block diagram of the RF Drive signal path.

The new system uses the 805 MHz Reference Line as a phase stable reference for the 201.25 MHz RF that drives the accelerating cavity. The 805 MHz reference line was installed during the high energy linac upgrade[1]. The 805 MHz signal that is supplied to the RF module is split into two signals. One signal provides a clock signal for the MFC board[2], and the other signal is sent to a clock divider to provide the 201.25 MHz RF Drive.

The 805 MHz signal is divided by 4 to generate the 201.25 MHz RF Drive signal for the accelerating cavity. Details of the clock divider circuitry will be described in detail in section 1.3.

After the 805 MHz signal has been divided down to 201.25 MHz, the signal is passed through an attenuator, a low pass filter, and a RF amplifier as seen in the block diagram in figure 1.2. The output of the clock divider is a square wave, and by passing it through a 225 MHz low pass filter, only the 201.25 MHz sine wave will be on the output. The filter also eliminates the power in the higher order frequency components that may push the GVA-84 RF amplifier into compression. The attenuator is included to control the output level of the RF amplifier, and it also serves to attenuate reflections of the high frequency components blocked by the low pass filter.

A portion of the official schematic detailing the circuitry is shown in figure 1.3. The Mini-Circuits GVA-84 RF (U57) amplifier has a gain of 24 dB, a 1dB compression point of 20.5 dBm, operates from DC to 6 GHz, and runs off a single 5V power supply. A Mini-Circuits ADCH-80A RF Choke (U58) is needed to supply the 5V power to the output pin of the amplifier. DC blocking capacitors are placed on the input and output of the amplifier (C31 and C32), and power supply filtering capacitors are placed on the power supply (C33 and C34). Further down the signal chain is a Mini-Circuit MIQA-195M IQ modulator (U30) to provide phase modulation to the RF Drive. The input to the IQ modulator is designed for 10 dBm od signal power, so the output of the amplifier must be adjusted accordingly. Before the signal gets to the IQ modulator, it passes through a 3-way splitter, an attenuator, and a 10 dB coupler. The 3-way splitter is a Mini-Circuits AD3PS-1 power splitter (U22), the attenuator is a Mini-Circuits LAT-5 5 dB attenuator (U28), and the 10 dB coupler is a Mini-Circuits ADC-10-4 (U29). The 3-way splitter splits the 201.25 MHz to provide for a phase quadrant detector that checks the phase quadrant of the RF Drive, and the splitter also provides the 201.25MHz that will be downconverted and sent to the MFC Board as the 201.25 MHz reference. On each output of the 3-way splitter is an attenuator. These attenuators are placed there to improve the isolation of the 3-way splitter, and they should be as high as the circuit will allow. The isolation of the Mini-Circuits AD3PS-1 3-way splitter is 35 dB.

The maximum output that the GVA-84 amplifier will output is about 20 dBm without going into compression, so that allows 10 dB of attenuation before the signal gets to the IQ modulator. The 3-way splitter has about 5.2dB of attenuation, so a 5dB attenuator is placed before the IQ modulator. There is a 5dB attenuator and an 8 dB attenuator are used on the other two output ports of the splitter, giving a much improved port to port isolation. The 10dB coupler placed before the IQ modulator aloows for a test point to check the power level at the input to the IQ modulator. If the power level is not correct, the attenuator before the 225 MHz low pass filter should be adjusted (U55).

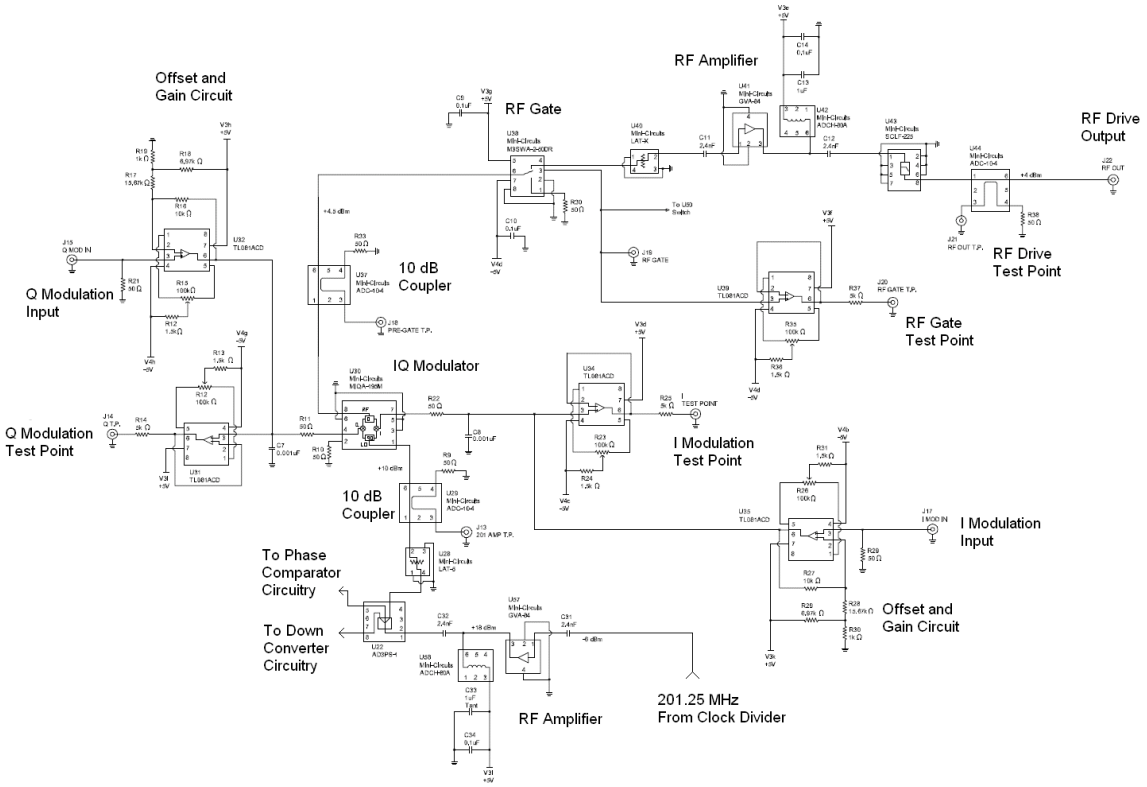


Figure 1.3: Portion of the official schematic including the RF Drive Path

The IQ modulator is capable of providing phase and amplitude modulation. For this system, only phase modulation is used. The magnitude and phase linearity of the IQ modulator was measured and the results are shown in figures 1.4 and 1.5. The measured results show that the magnitude is linear up 0.3 volts on either the I or Q input. The phase linearity was measured with an I/Q magnitude vector of 0.32 volts. This corresponds to a vector magnitude of 80% if full scale is 0.4 Volts. This is a good operational point to be used in an operational system.

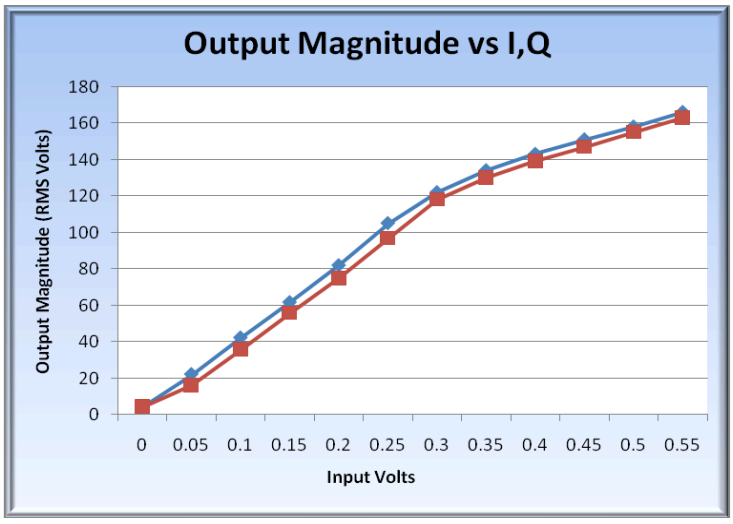


Figure 1.4: Magnitude linearity of the I/Q Modulator.

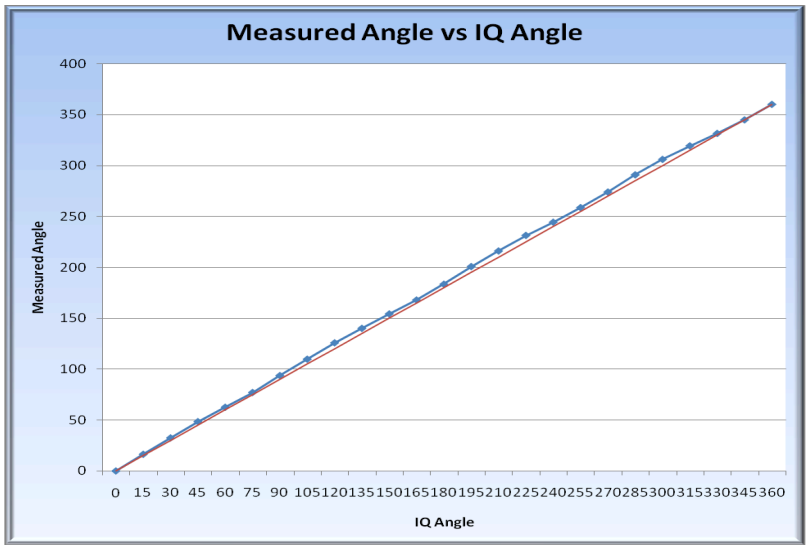


Figure 1.5: Phase linearity of the I/Q Modulator

Two output DACs from the MFC Board drive the I and Q ports of the I/Q modulator[3]. The DAC outputs are 50 Ohms and are unipolar outputs that range from 0 to 500mV. Since the DAC's are unipolar and the I/Q modulator is bipolar, an offset circuit is needed to translate the signal. The zero position of the DAC outputs is 250 mV which translates to an output range of +/- 250mV. The I/Q modulator has a linear range of +/- 400 mV. An OP-AMP circuit (U32 and U35) and was designed to provide an offset of 250 mV, and a gain of 1.6. The input of the circuit is terminated with a 50 Ohm resistor to match the impedance of the DAC output. There is also a buffer amplifier (U31 and U34) to provide a high impedance test point on the front panel of the module.

It is important that the each port of the I/Q modulator be terminated in 50 Ohms at 201.25 MHz. A 50 resistor with shunt capacitor (R11/R22 and C7/C8) provides a 50 Ohm path to ground for the 201.25 MHz on the I and Q ports of the I/Q Modulator. The value of the shunt capacitor should be chosen carefully because a high value shunt capacitor can create a low frequency pole in the phase feedback loop. After the I/Q modulator there is a 10 dB coupler for testing (U37).

The gating of the 201.25 MHz RF Drive is done with a Mini-Circuits M3SWA-2-80DR switch(U38). The switch is a single pole double throw RF switch with a minimum RF isolation of 53 dB. The switch terminates the RF signal to a 50 Ohm resistor (R30) when the switch is off to prevent reflections. The switch control is a TTL gate to turn ON and OFF the switch and is provided by the Linac. A buffer amplifier (U39) provides a high impedance test point for the RF gate on the front panel of the module.

After the RF gating, the signal passes through an attenuator (U40), a RF amplifier (U41), and a 225 MHz low pass filter (U43). The RF amplifier is the same Mini-Circuits GVA-84 RF amplifier that was used earlier in the signal chain. The attenuator should be chosen so that the amplifier does not compress. The low pass filter suppresses any higher order frequency components that may be on the RF Drive. Finally, a 10 dB coupler (U44) provides a front panel test point for the RF Drive.

1.2 Downconversion

A block diagram of the downconverters are shown in figure 1.6. There are three RF signals in this system that are downconverted to a 12 MHz IF and sent to the Analog-to-Digital converters (ADC's) of the MFC board. The three signals are the 201.25 MHz signal that comes from the cavity (TANK), the 201.25 MHz signal that drives the cavity (Forward Power), and the 201.25 MHz RF Drive that comes from dividing the 805 MHz reference line. The IF frequency is 12.577538 MHz, which is 805 MHz divided by 64.

Three mixers are used to down convert the RF signals to the IF frequency. This means that an LO frequency is needed that is offset from 201.25 MHz by 12.577538 MHz, resulting in a LO frequency of 188.663066 MHz. This system uses a phase locked oscillator to provide the LO for down conversion. The phase locked oscillator uses a 12.577538 MHz reference signal that is phase locked to the 805 MHz reference line. The phase locked oscillator is a Wenzel VHF-PLO series oscillator.

To generate a 12.577538 MHz signal that is phase locked to the 805 MHz reference line, the 201.25 MHz RF drive that was generated by the 805 MHz reference line is divided by 16. The phase of the 12.577538 MHz reference does not matter as long as it locked to the phase of the 805 MHz reference. This is true because the LO is split into three, and the phase of all three down converters will remain relative to each other. Details of the clock divider circuitry will be discussed in section 1.3.

Figure 1.7 shows a portion of the official schematic drawing that has the downconverter circuitry. Only the LO amplifier and the TANK down converter is shown in this schematic. The mixers used for the circuit are Mini-Circuits model# JMS-1MH (U6). The LO input to the mixer is +17 dBm, the RF input has a 1 dB compression point of +14 dBm, and the conversion loss is about 6.5 dB. The ADC inputs of the MFC Board can handle 2 Volts peak to peak, which correspond to +4 dBm, so the IF output of the downconverter mixer is about +3 dBm, which is close to full scale of the ADC. In order to achieve +3 dBm at the IF output, the RF input of the mixer needs to be about +10 dBm. This is well below the 1 dB compression point of the mixer to maintain linearity. It is very important that this signal remain linear due to the processing of small variations of the signal at during gradient flat top, which should be close to full scale of the ADC. The 6.5 dB of conversion loss puts the IF output at around +3 dBm. A Mini-Circuits PLP-15 Low pass Filter (U7) is used to filter out the frequency components above the 12.577538 MHz IF.

The output of the Wenzel phase locked oscillator provides the LO for each of the downconverters. The output power of the PLO is about + 13 dBm. Since the mixers require a +17 dBm LO, the signal must be amplified. A Mini-Circuits HELA-10D RF amplifier is used (U1). The amplifier has a gain of 11 dB and a 1 dB compression point of +30 dBm. For this circuit, the input to the amplifier is +13 dBm, so the output is around +24 dBm, which is well below the 1 dB compression point. After the amplifier, the signal is split three ways using a Mini-Circuits AD3PS-1 (U4) to provide the LO for each downconverter. The splitter has an insertion loss of 6.5 dB, leaving 17.5 dBm at the output. The amplifier proved to have a slightly higher gain than 11 dB, so a 1 dB pad was placed on each output port of the splitter. This slightly reduces reflections from the mixer and improves port to port isolation.

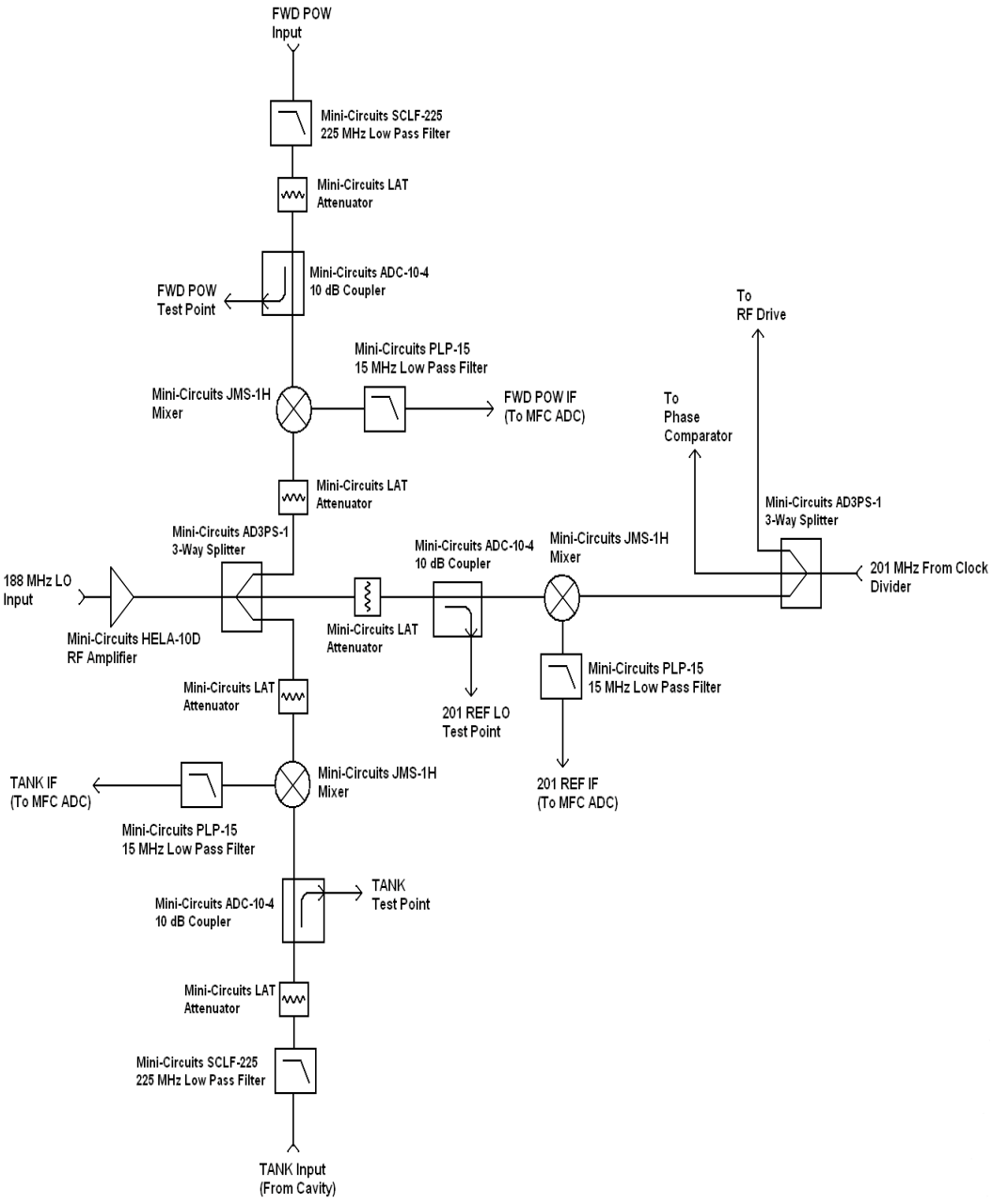


Figure 1.6: Downconversion block diagram.

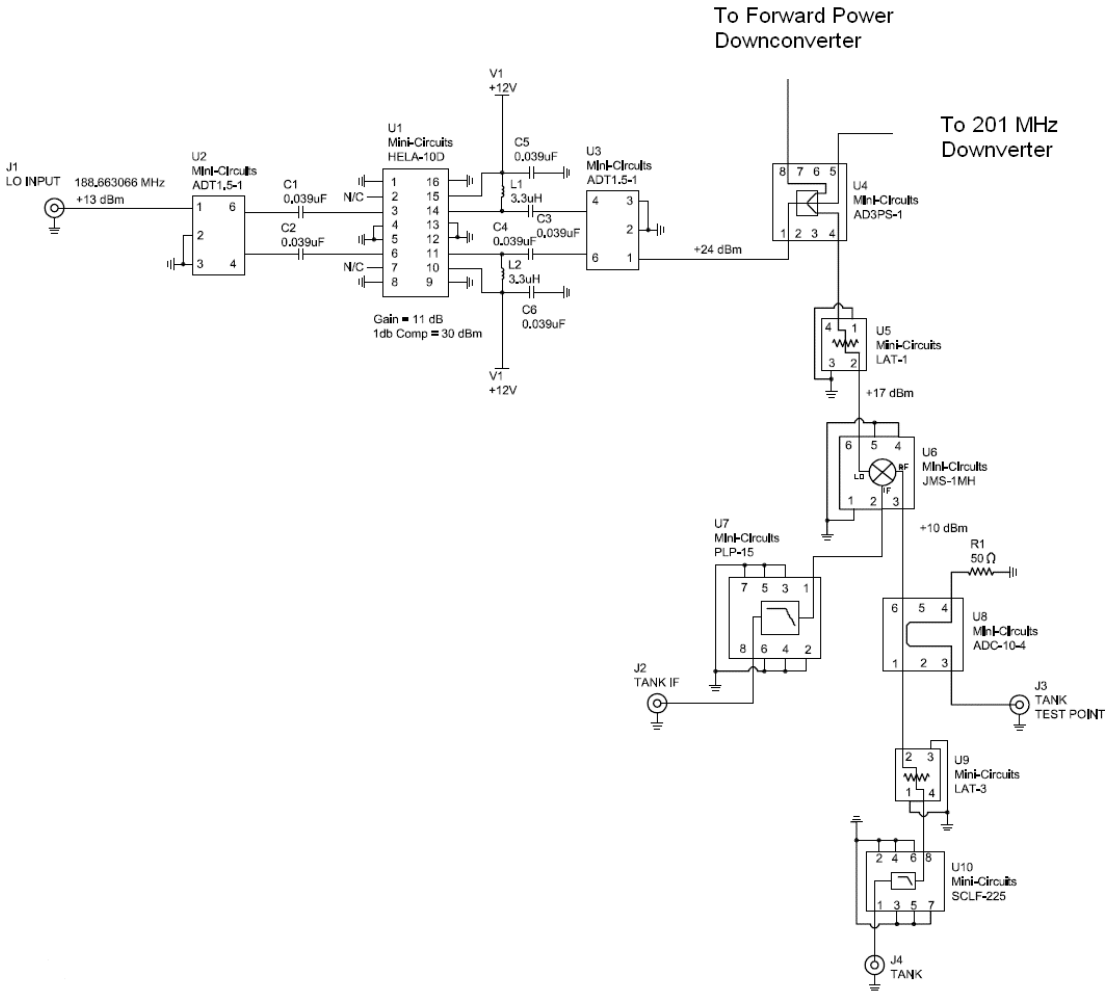


Fig. 1.7: Schematic of the down converter circuitry

1.3 Clock Divider and phase comparator circuit

A block diagram of the clock divider circuit is shown in figure 1.8, and sections of the official schematic drawing are shown in figures 1.9 and 1.10. The 805 MHz input comes from the phase stable reference line located in the tunnel. The function of this clock divider circuitry is to provide a 201.25 MHz signal and a 12.577538 MHz signal, both phase locked to the 805 MHz reference line. The circuit also ensures that the 201.25 MHz signal is in the correct phase quadrant after the divide by 4 operation.

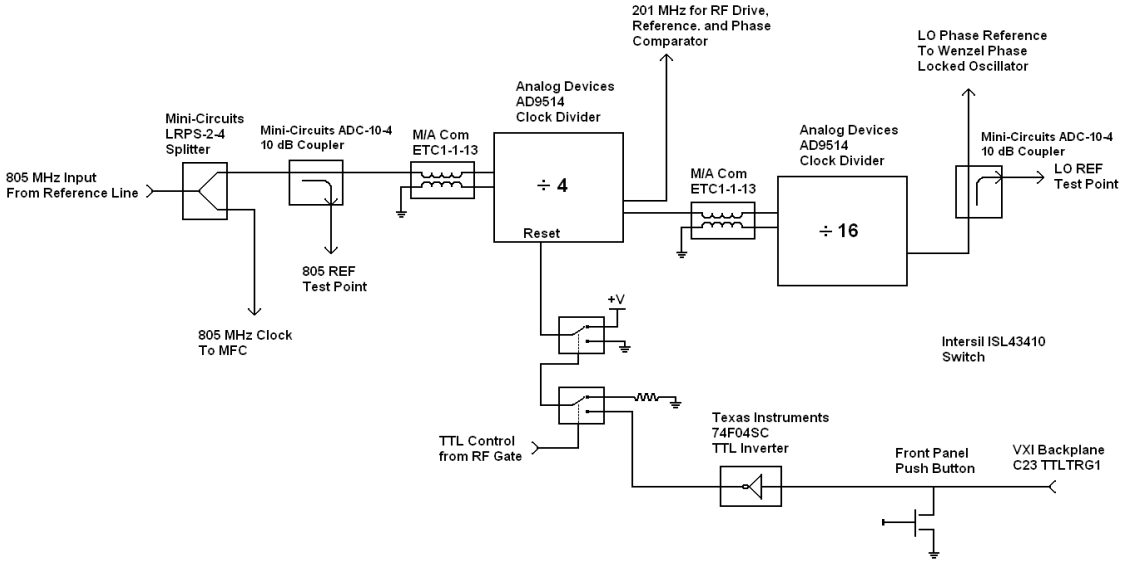


Figure 1.8: Clock Divider Block Diagram.

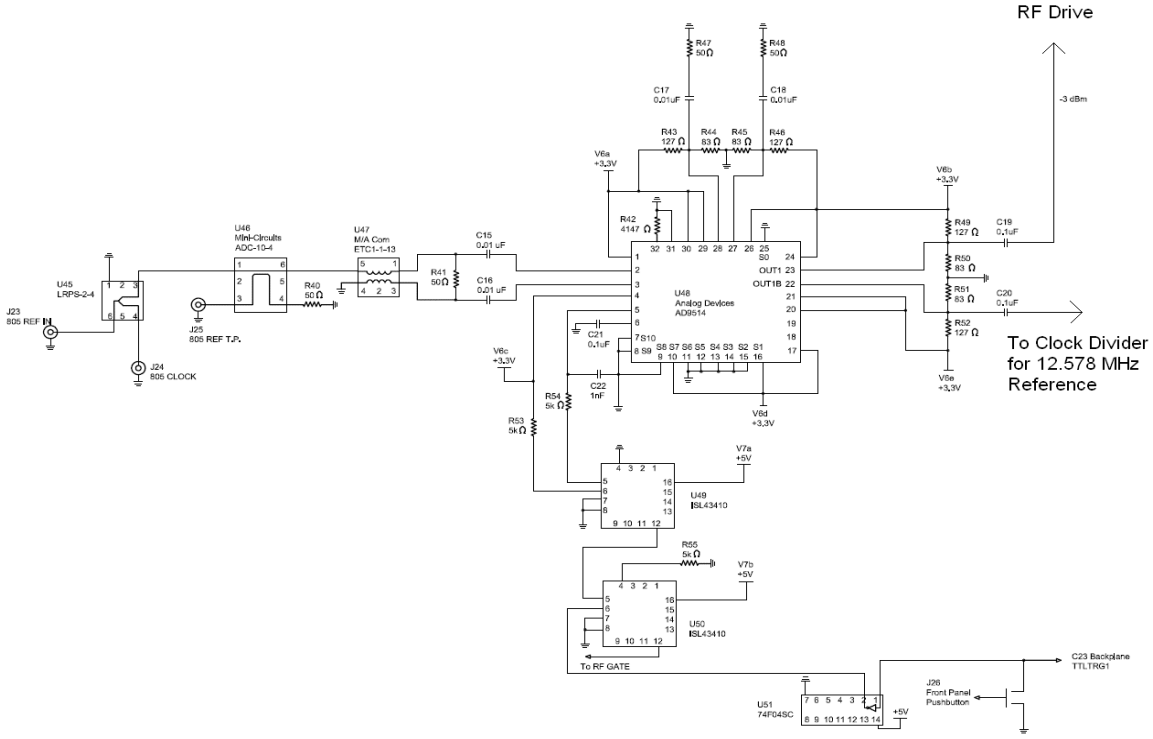


Figure 1.9: Section of the official schematic (Divide by 4 circuit).

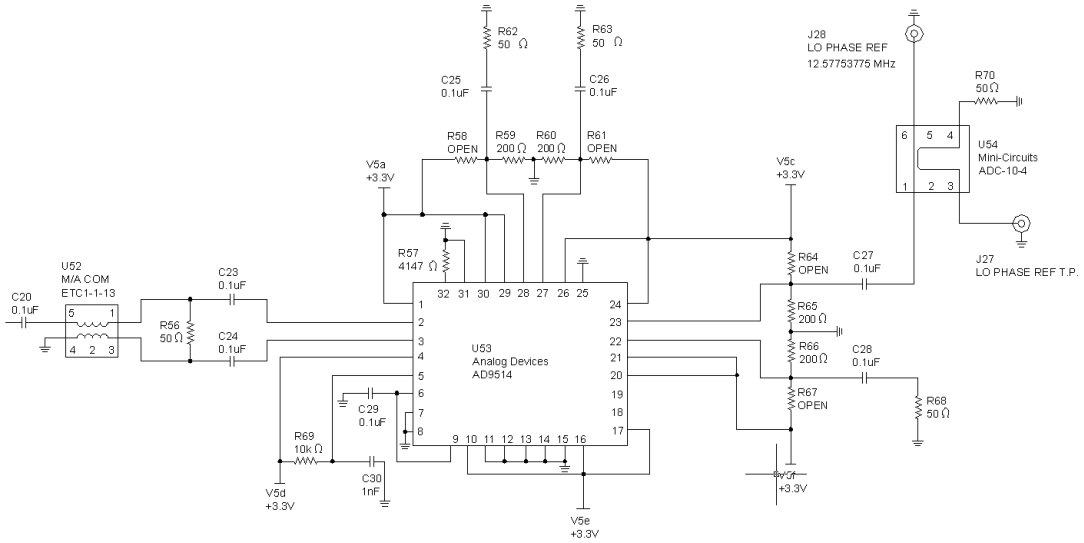


Figure 1.10: Section of the official schematic (Divide by 16 circuit).

Directly after the 805 MHz Input (J23), the signal is split using a Mini-Circuits LRPS-2-4 (U45) to provide the 805 MHz clock source for the MFC board (J24). Following the splitter, there is a Mini-Circuits ADC-10-4 (U46) 10 dB coupler to provide a test point before the clock divider (J25). A M/A Com ETC1-1-13 (U46) 1:1 transformer is used to convert the single ended input to a differential input before it goes to the Analog Devices AD9514 clock divider (U48). A 50 Ohm loading resistor (R41) and DC blocking capacitors (C15 and C16) are placed on the input of the AD9514 to provide impedance matching and dc blocking. The AD9514 (U48) is programmed to divide the 805MHz input by 4 to provide a 201.25MHz signal. There are 3 differential outputs on the AD9514 chip, and there are 11 pins (s0 – s10) for programming the 3 outputs (OUT0, OUT1, and OUT2). OUT0 and OUT1 are the LVPECL outputs, and OUT2 is a LVDS/CMOS output. The programmable pins can be set to one of 4 levels (0, 1/3, 2/3, 1). Level 0 is 0V (ground), level 1/3 is 1.1V, level 2/3 is 2.2V, and level 1 is 3.3V. By default, if a pin is not connected, the level is 1/3. Pin 6 (VREF) of the AD9514 provides the 2/3 level. As seen on the schematic, the circuit uses only the differential outputs of OUT1. The outputs are labeled OUT1 and OUT1B in the manufacturers' datasheet (Pins 23 and 24). The programmable pin settings for a divide by 4 ratio to divide the 805 MHz to 201.25 MHz are listed in the table below.

Using the pins to program the outputs, OUT0 and OUT1 were both turned ON. Both outputs are turned ON because there is no option to turn on only OUT1 or OUT2 with a voltage level of 960mV. There are possibilities of turning on only one of the outputs, but the voltage levels are reduced. For this circuit, a higher voltage is preferred to achieve higher power levels with minimal use of amplification. Because OUT0 is not being used, each differential line of OUT0 is terminated in 50 Ohms (R62 and R63).

PIN	Level
s0	0
s1	1
s2	0
s3	0
s4	0
s5	0
s6	0
s7	1
s8	0
s9	0
s10	0

S0 sets a delay on OUT2, and since OUT2 is not being used, S0 is set to level 0. S1 and S2 set which outputs are used and specifies the voltage level of the outputs. S1 is set to 1, and S2 is set to level 0. This turns outputs OUT0 and OUT1 on with a level of 960 mV and turns off OUT2. S3-S6 refers to outputs and delays that are not used, so they are all set to level 0. S7 and S8 set the divide ratio of OUT1, which is set to a divide ratio of 4, with S7 at level 1 and S8 at level 0. S9 and S10 refer to outputs that are not used and are set to level 0. Each differential line of the output (OUT1 and OUT1B) must be matched to 50 Ohms, and the data sheet indicates that a 200 Ohm resistor tied to ground (R50 and R51) on each differential output will provide a good match to 50 Ohms. DC blocking capacitors are also needed at the outputs (C19 and C20).

When dividing the 805 MHz Reference by four, the phase of the 201.25 MHz signal relative to the 805 MHz can be in one of four positions, as shown in figure 1.11. The AD9514 has a sync pin that restarts the outputs a few cycles after the sync pin is pulled low and returned to a high state. When the sync pin is pulled low and released, the phase can be in one of the four phase quadrants. The sync pin is set high for normal operation.

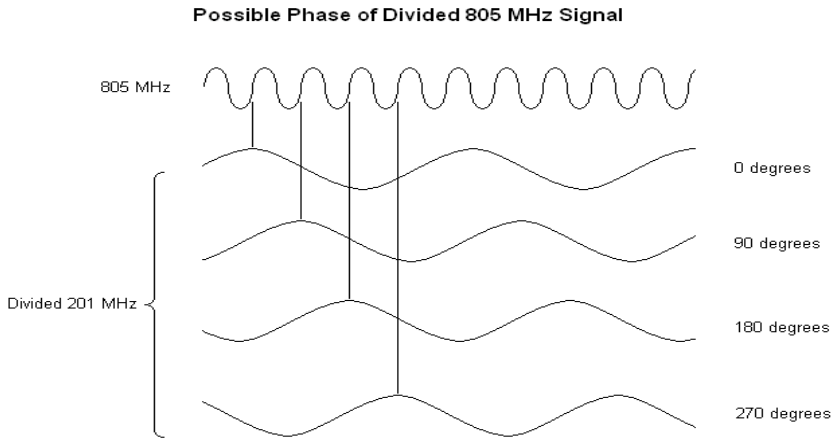


Figure 1.11: Possible phase of the 201.25 MHz after dividing the 805 MHz reference.

It is important that the relative phase of all 5 low energy stations be constant. In order to make sure the relative phase of each station is constant a 201.25 MHz reference line is compared to the 201.25 MHz from the 805 MHz using a phase comparator. A diagram of the reference line and the phase comparator circuit is shown in figure 1.12. The 201.25 MHz reference line is daisy chained down the low energy linac using 5/8" heliax cable. The phase drift of the daisy chained 201.25 MHz reference is not important because it is only used as a phase quadrant detector. The voltage output of the phase comparator operates in a positive voltage window that allows for drift of the daisy chained 201.25 MHz reference line.

The output of the phase comparator will be checked by the MFC board to ensure that the phase is in the correct quadrant[3]. If the phase is not in the correct quadrant, the MFC board will issue a command to reset the clock divider until the phase is in the correct quadrant. There is also a front panel pushbutton that allows a user to manually reset the phase. A safety feature is included that prevents any reset signal to the clock divider to occur while the RF is ON. This prevents interruption of the RF drive while the PA is at full power. There is also a front panel LED that indicates that the phase is in the correct quadrant.

A more detailed schematic of the phase comparator circuit is shown if figure 1.13. The 201 MHz reference line input (J10) is passed through a Mini-Circuits SCLF-225 Low Pass filter (U25) before it goes to the phase comparator. The phase comparator is a Mini-Circuits JMS-1 Mixer (U24), and the output of the phase comparator goes through a Mini-Ciruits PLP-1.9 low pass filter (U26). The output is sent to the MFC board so that it can be determined if the 201.25 MHz signal is in the correct phase quadrant. It was decided that the output of the phase comparator will be maximum when the 201.25 MHz signal is in the correct phase quadrant. The output voltage of the phase comparator also sent to an OP-Amp buffer (U27) to a front panel test point. And lastly, the output of the buffer is applied to a front panel LED to provide a visual indication that the phase is in the correct quadrant.

If the phase is not in the correct quadrant, the sync pin must be pulled low to reset the clock divider. The circuitry that is responsible for this operation is shown in figure 1.8 and 1.9. The command to pull the sync pin low is issued by the MFC board, or by pressing the front panel pushbutton (J26). The command from the MFC board is sent on the backplane of the VXI crate (Pin C23). The VXI backplane is active low and the pushbutton is tied directly to the back plane to pull it low as well. The backplane is input to a 74F04 TTL inverter (U51), which is fed to a two section switch circuit. Both switches are double pole single throw ISL model# ISL43410. The first switch (U50) is responsible for not allowing the sync pin to be pulled low while RF is driving the PA. The control pin of the switch comes from the TTL control voltage of the RF gating circuit. The second switch (U49) is controlled by the low pull of the backplane or the pushbutton when there is no RF present on the PA. This switch removes the 3.3 Volts from the sync pin, and when the low pull from the backplane is released the clock divider will reset.

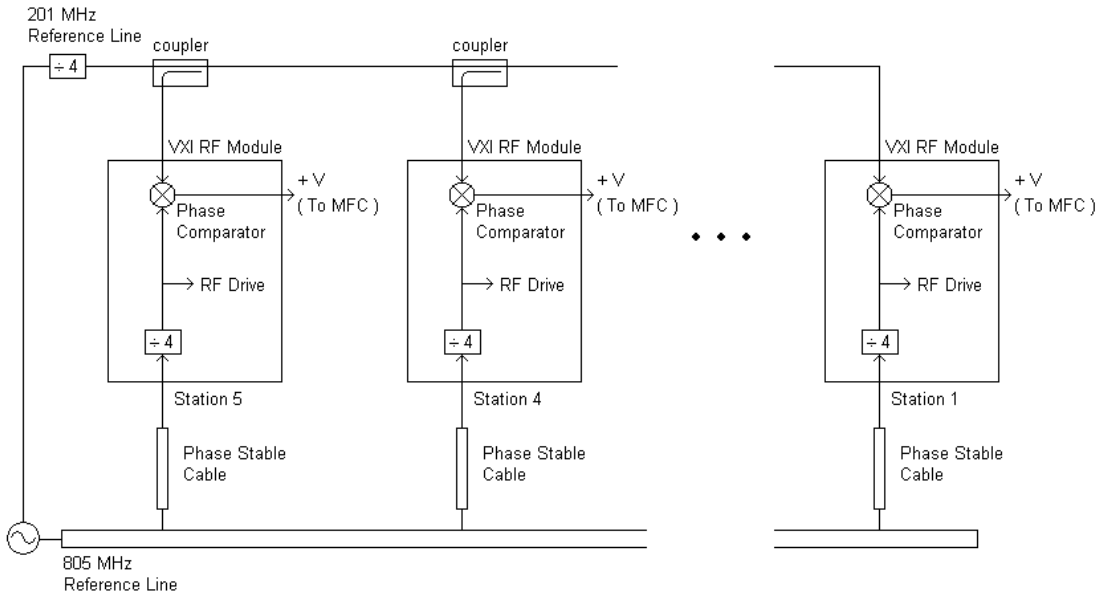


Figure 1.12: Phase comparator system to keep the relative phase constant.

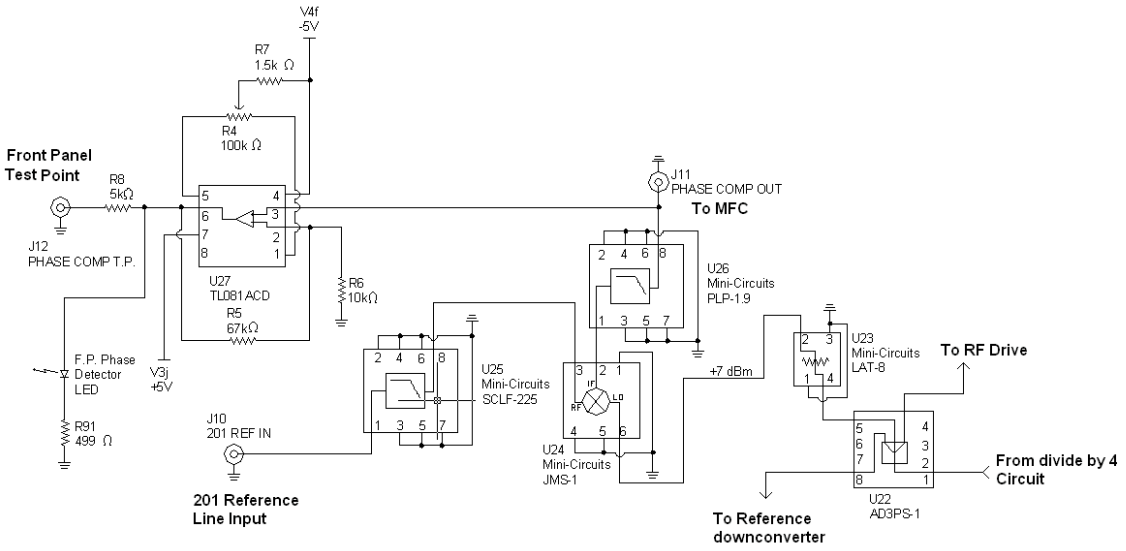


Fig 1.13: Phase comparator circuitry

The divide by 16 circuit used to generate the 12.577538 reference for the LO phase locked oscillator is shown in figure 1.10. One of the differential outputs of the divide by four circuit (U48, OUT1) is used to provide the RF Drive source, the 201.25 MHz for the phase quadrant detector, and 201.25 MHz reference. The other differential output (U48, OUT1B) is sent to another AD9514 clock divider (U33) to provide the

12.577538 MHz reference for the LO phase locked oscillator. A second clock divider is needed because the divider can only divide by 32. (12 MHz = 805 / 64). The input and output circuitry are the same as the divide by four circuit, and the programmable pins are set for a divide by 16 operation. A 10 dB coupler is placed on the output to provide a front panel test point for the reference signal (U54 and J27).

1.5 Toriod Voltage Comparator

The purpose of the toriod comparator is to let the MFC board know whether or not beam is present is the cavity. The output of the comparator is a TTL level pulse that is sent to the backplane of the VXI crate (Pin A23). A schematic of the toriod comparator is shown in fig 1.14. A Texas Instruments LM339DR2 (U59) is used for the comparator. A Texas Instruments TL081ACD OP-Amp provides a buffer for a front panel test point, and a Texas Instruments 74f38SC (U61) provides a CMOS driver to send the signal to the VXI backplane and on to the MFC.

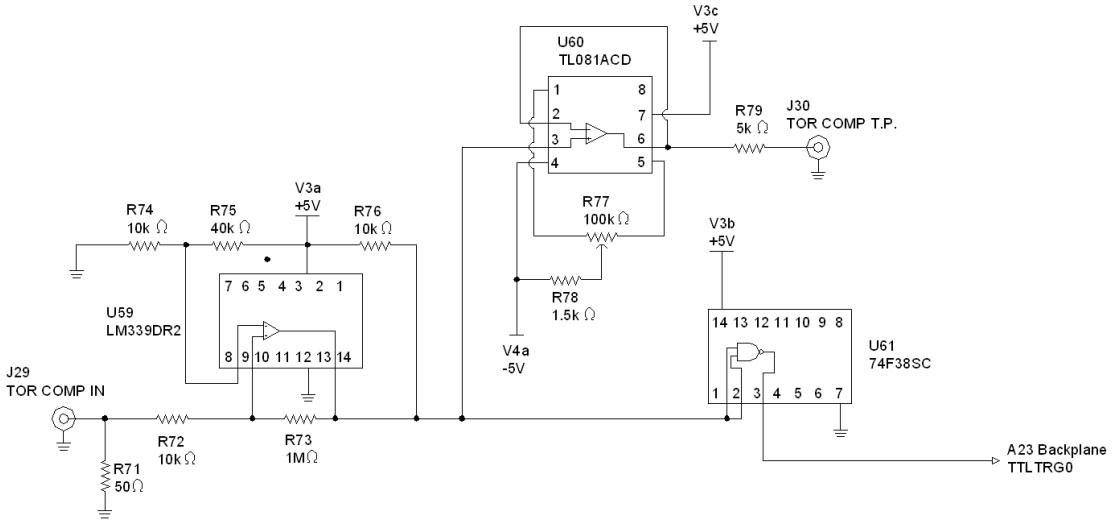


Figure 1.14: Toriod voltage comparator circuit.

1.6 Cavity Tuner Control Circuit.

One DAC output on the MFC board controls the tuning of the cavity. The output of the DAC has a range from 0 to 0.5 Volts, but a bi-polar voltage is needed to drive the cavity tuner controller. A schematic of the circuit used to convert the signal to bipolar is shown in figure 1.15. The same circuit is used for the I and Q DAC's. The output of the circuit has a voltage range of +/- 0.4 Voltage, which is sufficient for the cavity tuner controller.

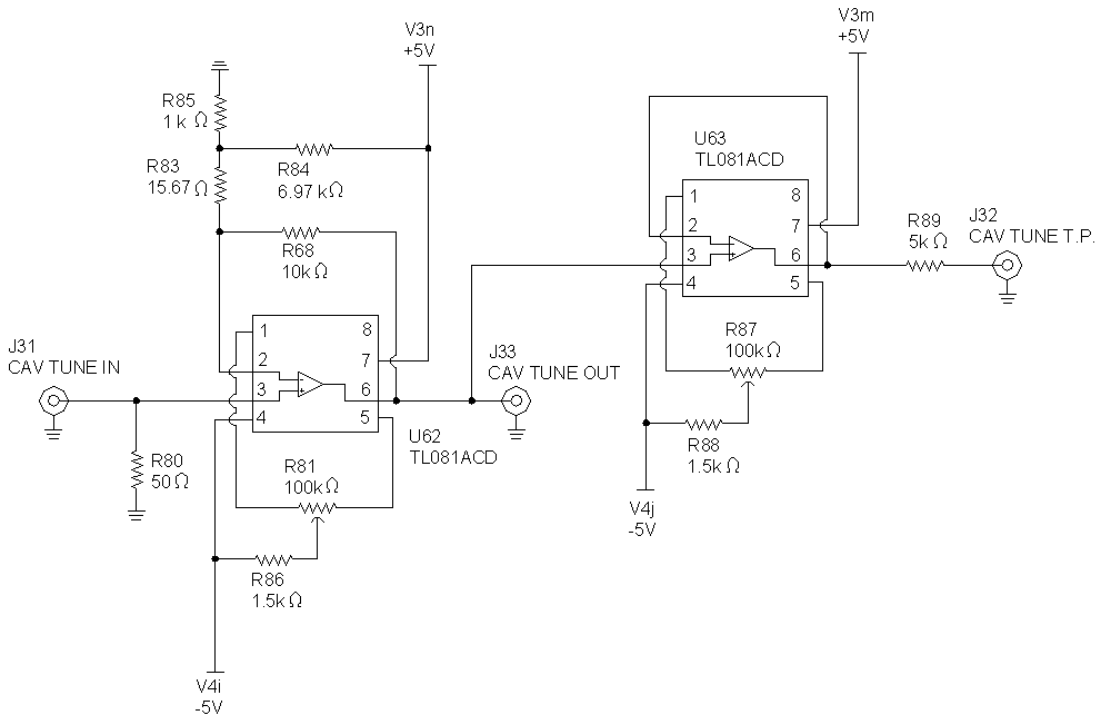


Figure 1.15: Cavity tuner control circuit.

2 Material and Layout

The material used for this board is Rogers 4030B, 30 mil, ½ ounce copper. The layout was generated using a program called CAM350, and each layer of the board was exported to a Gerber. A picture of the layout is shown in figure 2.1. The board has three layers (top, middle, and bottom). The top layer is populated with all the RF and analog components, the middle layer is a ground plane, and the bottom layer is for power. There are two 96 pin DIN connector on the left side of the board to connect to the VXI chassis backplane. The DIN connector provides power and access to the VXI backplane.

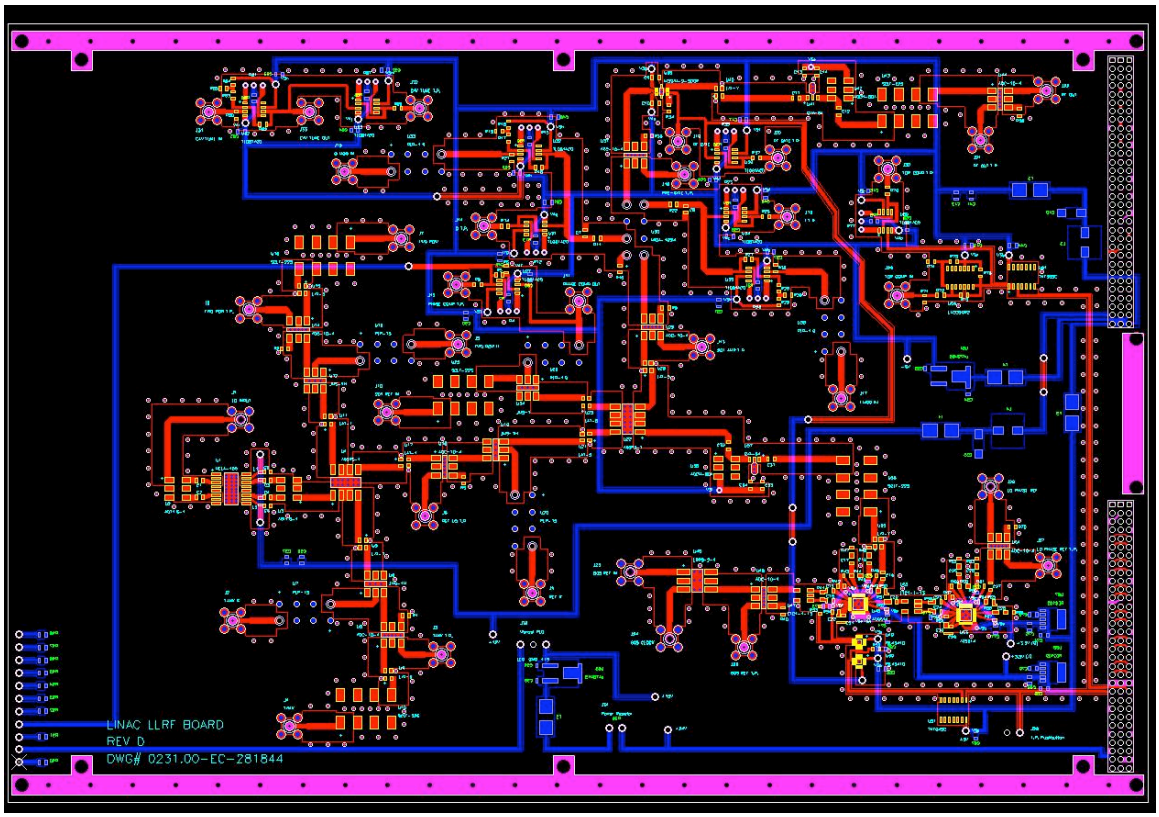
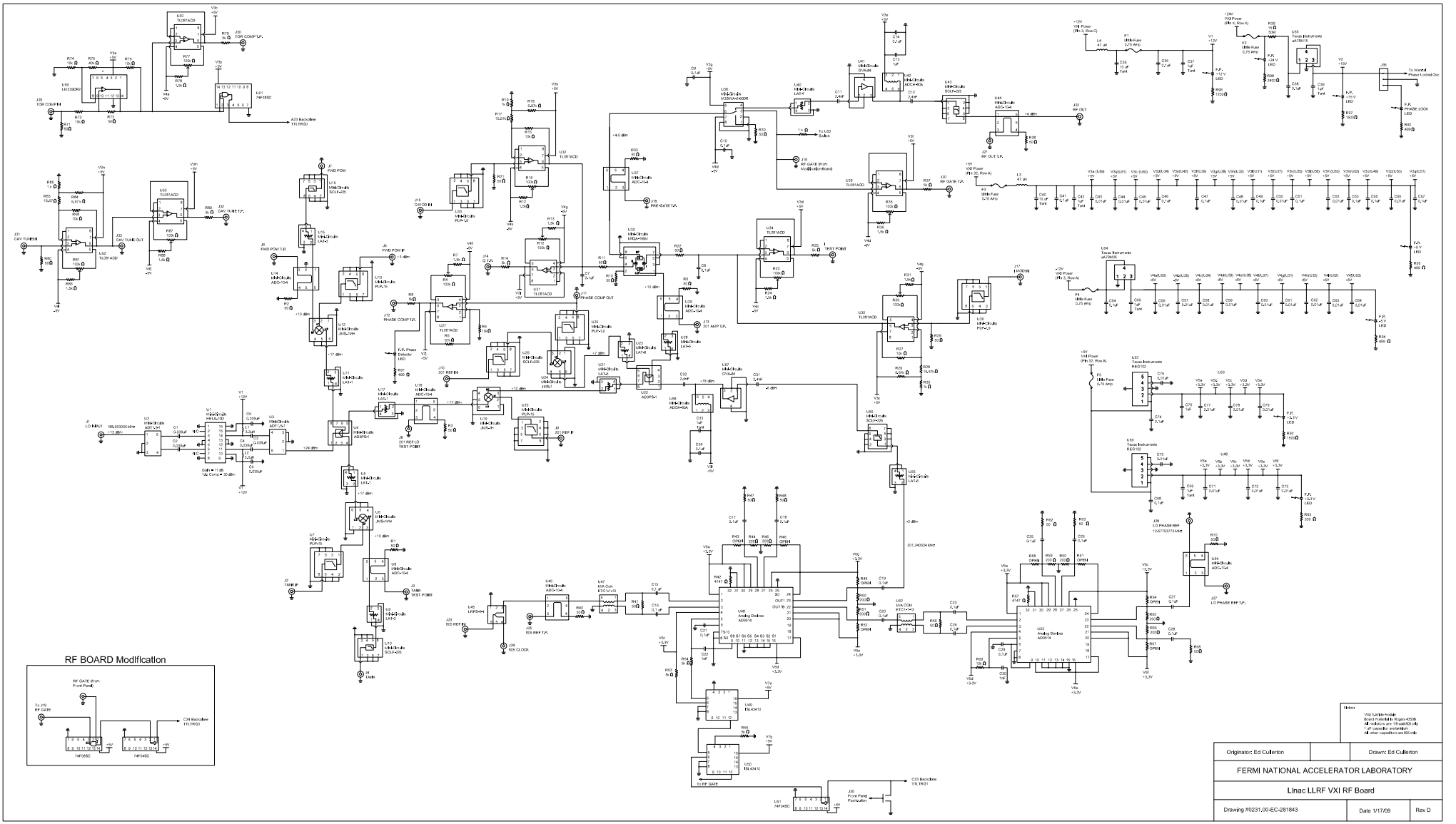


Figure 2.1: VXI RF board layout.

References

[1] Paper about the High Energy Linac Upgrade (Brian Chase, Ralph Pasquinelli)
[2] Varghese, P. et al., 2007, "Multichannel Vector Field Control Module For LLRF Control of Superconducting Cavities", Fermilab note.
[3] Tupikov, 2009, "MFC FPGA Program for the low energy Linac", Fermilab note.



RF BOARD Modification

Notes:
VXI 1x4 Mite module
Board material is Rogers 4003B
All resistors are 1/8 watt 5% only
1 µF capacitor are tantalum
All other capacitors are 50 volt

Originator: Ed Cullerton		Drawn: Ed Cullerton
FERMI NATIONAL ACCELERATOR LABORATORY		
LInac LLRF VXI RF Board		
Drawing #0231.00-EC-281843	Date 1/17/09	Rev D

Low Energy Linac LLRF Upgrade Digital Signal Processing

SRC: LEL_FPGA_Lib_10.doc FPGA version #188cyc1

Brian Chase, Ed Cullerton and Vitali Tupikov, *FNAL\RF\LLRF Embedded System Group*

***Abstract*—The paper is devoted to FPGA code for the low energy Linac upgrade. The code is responsible for the entire digital signal processing, interfacing with onboard Sharc DSP-master and for data acquisition associated with the upgrade. The digital signal processing includes NCO frequencies synthesis with phase-locked reference LO, IF signals downconversions, filtering, phase feedback and amplitude feed-forward loops handling, tuner controller feedback loop management, filtering, gain and resolution control, beam type decoding and synchronization. Interface and data acquisition parts include combined Sharc-FPGA serial/parallel interface, waveform and scalar buffers, and signal/test point multiplexers. The details on each FPGA component will be described further within the paper.**

***Index Terms* — beam compensation, low level RF, Digital Signal Processing.**

TABLE OF CONTENTS

The work is done under Fermilab Proton Linac Upgrade Plan

I. Introduction

THE described Altera's Quartus II Library is a collection of schematic parts that may be used in larger design. The collection is based on a code developed for Low Energy Linac LLRF Upgrade [1] and implemented in MFC board Cyclone II FPGA [3]. A block diagram of FPGA code is shown in the Appendix C at the end of the paper. Appendix B shows a Timing Diagram, which might be helpful in understanding of signal flows and sequencing. More detailed sections of the schematic will be shown and described to clarify the meanings in corresponding paper paragraphs.

II. Triggers and Timing

A. Triggers

There are only two triggers coming to the system and getting available in FPGA. They are "LLRF_START" and "PREPULSE" (see *Fig. 1* and two top signals on Timing Diagram in Appendix B). If the first trigger is a single pulse presented in the system during each RF cycle, the second one is composite signal consisting of Beam Qualifying and Chopper parts. The part decoding is done with use of AND-ing of "LLRF_START" with "PREPULSE". The AND-ed portion is treated as the Beam Qualifier and the rest is the Chopper. The triggers spec can be found in [7] or in [8].

B. Timing Generator Module

The Timing Generator module shown in *Fig. 1* consists of two subunits 'LLRF_StartTim" and "PrepulsseTim". Though both are responsible for generation of timing signals, there is still a functional difference between them. See for description below in corresponding subsections.

FPGA block diagram in Appendix C presents the module in simplified timing diagram form. However such representation gives an impression about signal relation between timing set--parameters, trigger lines with other inputs, and outputs. For more detail see Timing Diagram in Appendix B.

1) "LLRF_StartTim"

The unit provides signals necessary for LLRF system data acquisition and waveform capturing. The process is driven by 15 Hz Linac's event through LLRF_Start trigger.

Internal Submodules:

- "Gating";

Clock requirements:

- 62 MHz;

Inputs:

- LLRF_START - as it says;
- PREPULSE - beam qualifier part of Prepulse line;
- CLK - FPGA clock;
- SAMT[15..0] - defines start time of scalar parameters averaging (Acnet parameter V#TSRD) from LLRF_Trig_T0 in 4*(clock samples);
- SHOT[15..0] - defines start time of waveform capture (Acnet parameter V#TSDA) from LLRF_Trig_T0 in 4*(clock samples);
- RESET - system reset;
- PREPULSE_CAPT_SEL – binary switch set by Sharc-DSP selecting mode of waveform capturing. When it is high the capture is triggered by PREPULSE_TRIG, i.e. during beam present condition. When it is low, the waveforms are captured by LLRF_START trigger;
- PREPULSE_TRIG

Outputs:

- LLRF_Rst - reset pulse triggered by any LLRF_START trigger;
- SAMPL_GATE - scalar readbacks averaging window;
- LLRF_CAPT - initiated by each LLRF cycle;
- CAPT_START - initiated by LLRF_CAPT but gated depending on selected event LLRF_START or PREPULSE. In other words, the pulse is issued each LLRF pulse when PREPULSE_CAPT_SEL is low and only at PREPULSE event when PREPULSE_CAPT_SEL is high;

2) "PrepulseTim"

The unit serves as a feedforward compensation arbiter generating for these purposes all required timing windows (gates) and actuators. Within the unit the beam type decoding is performed as well. It is done by decoding the length of PREPULSE signal (see description of "Beam_Qualif_Decod" subunit).

At input it takes the system clock, LLRF and beam qualifier triggers plus a set of time parameters, which had been written by DSP into FPGA registers either on power up or on request from Slot0 controller. Within the module the parameters are turned into actuators and gates, and then connected through output ports to specific FPGA subsystems.

Internal Submodules:

- "Beam_Qualif_Decod";
- "Gating";

Clock requirements:

- 62 MHz;

Inputs:

- LLRF_TRIG_START - as it says;
- PREPULSE - beam qualifier part of Prepulse line;
- CLK - FPGA clock;
- FTAS[15..0] - defines new time reference base (vxworks startup script parameter value =970);
- TADV[15..0] - time delay in clock samples (Acnet parameter V#TFHR) from new time reference base to: ff ramp start; ff inhibit mask start;
- SLOP[15..0] - ff ramp width in clock samples;
- TLIM[15..0] - prebeam sample window start in clock samples from new time reference base;
- BTAS[15..0] - beam sample window start in clock samples from ff ramp start;
- BGAT[15..0] - beam and prebeam sample windows width in clock samples;
- STAS[15..0] - ff spike window start in clock samples from ff ramp start;

- SGAT[15..0] - ff spike windows width in clock samples;
- SPFF[15..0] - phase ff start in clock samples new time reference base;
- TCHP[15..0] - time delay in clock samples from CHOPPER start to beam shut off;
- CHOPPER - decoded chopper portion of Prepulse trigger line;
- SYS_RST - system reset;

Outputs:

- CYCLE_RESET - triggered by LLRF_START and issued each RF cycle;
- BEAM_TYPE[1..0] - beam type bus: HEP=0, NTF=1, STU=2, INTRL=3;
- FTA_GATE - prebeam average window on flat-top;
- SLOP_GATE - amplitude ff ramp window;
- BL_GATE - amplitude ff pulse window without ramp;
- TLIM_GATE - beam inhibit window;
- BL_AVER_GATE - average window at beam-loading;
- STAS_GATE - amplitude ff spike window;
- PFF_GATE - phase ff pulse window;
- BEAM_CUTOFF - beam cutoff pulse triggered by CHOPPER (not used any longer!!!)
- HEP_BEAM_ON - HEP beam type;
- NTF_BEAM_ON - NTF beam type;
- STU_BEAM_ON - STU beam type;

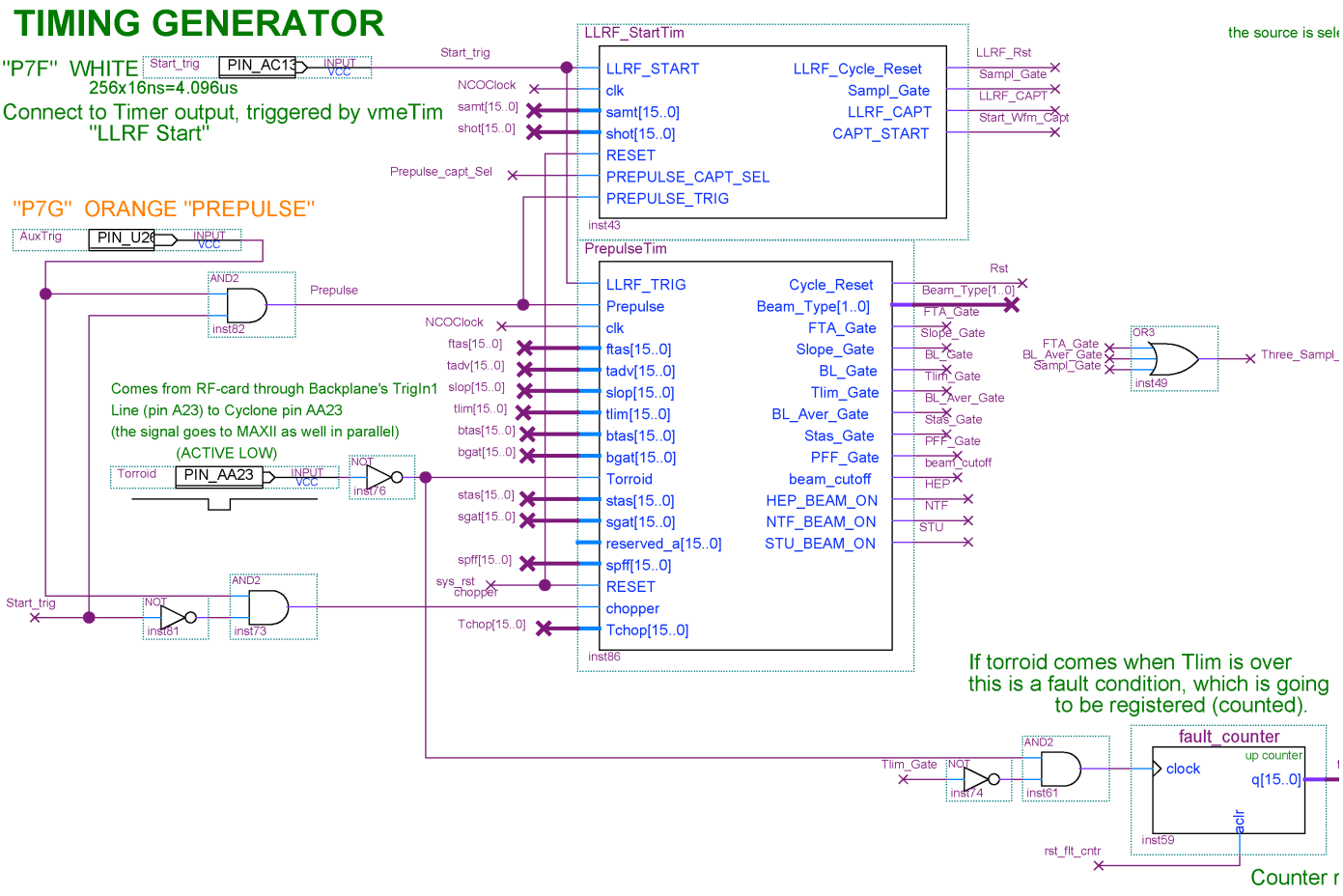


Fig. 1 Timing Generator Module.

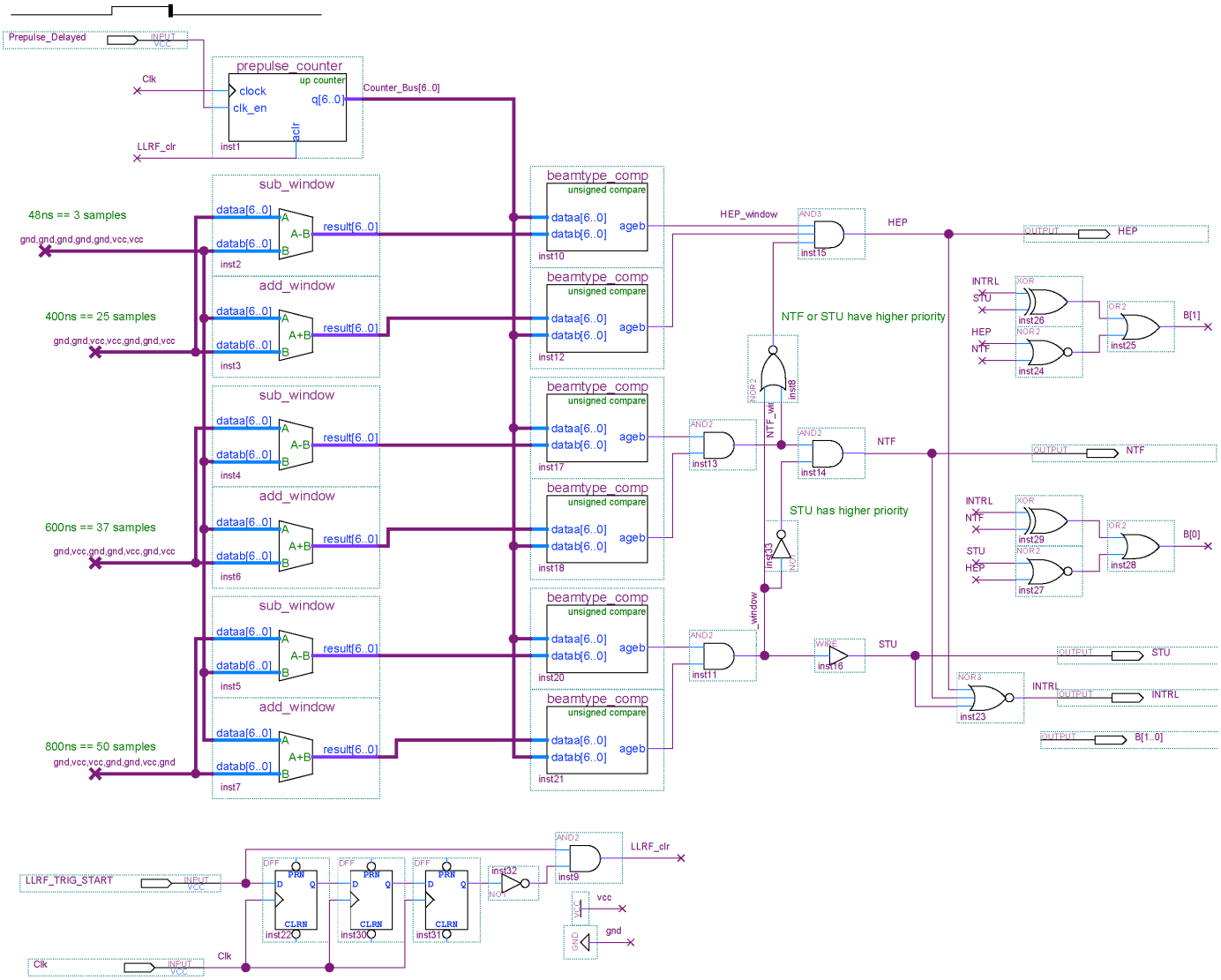


Fig. 2 Implementation of Beam Qualifier Decoder.

3) "Beam_Qualif_Decod"

The unit (*Fig. 2*) decodes the length of Prepulse_Delayed signal in accordance with Table 1 and forms two types of outputs:

- Enable lines for each beam type including INTRL (or No Beam) and
- 2-bit bus with truth table as shown in Table 2 below.

'Prepulse' signal is used as prepulse_counter clock enable input, allowing measuring the beam qualifier duration in clock samples (Clk input is 62MHz FPGA system clock). The timer is initially cleared by 'LLRF_clr' pulse generated from 'LLRF_TRIG_START' input.

Once 'Prepulse' changed level from high (active) to low, the prepulse_counter output keeps result on Counter_Bus[6..0]. The value is compared against 3 pairs of numbers forming min and max limits for each beam type windows. The comparison results then wired to HEP, NTF, STU and INTRL and through coding circuit to B[1..0] output ports.

TABLE 1: Prepulse width decoding into Beam Type

Type	Prepulse_Delayed Time
HEP	368 ns < T < 432 ns
NTF	560 ns < T < 632 ns
STU	768 ns < T < 832 ns
INTRL	Neither of the above

TABLE 2: Truth Table

HEP	NTF	STU	INTRL	B1	B0
0	0	0	1	0	0
0	0	1	0	0	1
0	1	0	0	1	0
1	0	0	0	1	1

Internal Submodules:

- None;

Clock requirements:

- 62 MHz;

Inputs:

- LLRF_TRIG_START - as it says;

- Prepulse_Delayed - beam qualifier part of Prepulse line;
- CLK - FPGA clock;

Outputs:

- HEP_BEAM_ON - HEP beam type;
- NTF_BEAM_ON - NTF beam type;
- STU_BEAM_ON - Study beam type;
- INTRL_BEAM_ON - beam Interlock type;
- Beam_Type[1..0] – beam type b

III. Receiver and Downconverters

A. LVDS Receiver Building Block

One channel of LVDS receiver consists of two odd and even 6-bit shift registers and one output 12-bit latch (see *Fig. 3*). Each of registers shifts serial data in by different fronts of ADC's data clock, presented on circuit input as ports *ser_clk* and *ser_clkn*. The result is latched by coming from ADC *ser_frame* signal as it shown in *Fig. 3*.

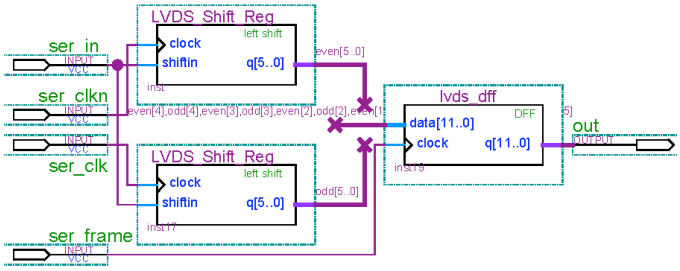


Fig. 3. Building block of LVDS Receiver.

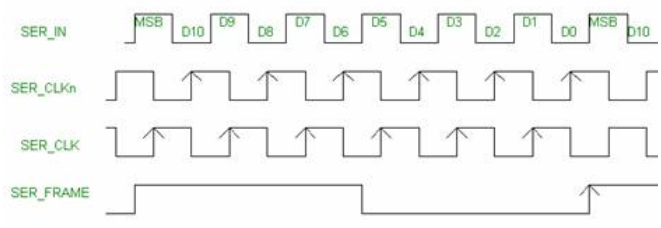


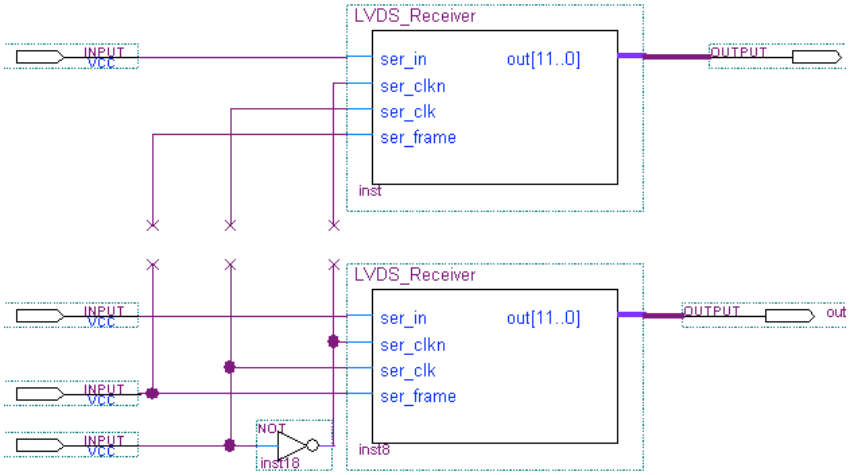
Fig. 4. Waveform diagram of data clocking.

Since the frame strobe coming from ADC has the frequency rate equaled to the top project system's clock then the LVDS receiver output can be easily pipelined into the system. The frequency rate of *ser_frame* is six times lower than *ser_clk* (*ser_clkn*) rates, which is defined by AD9222 chip architecture. To keep output data valid the followings have to be satisfied:

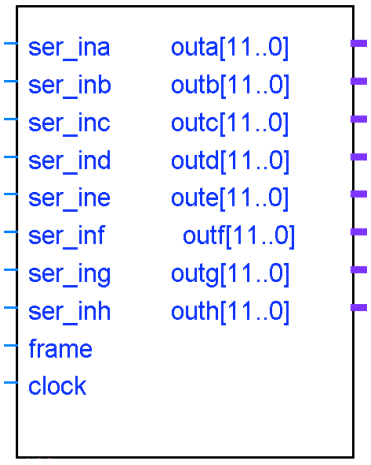
- the delays on both *ser-in* lines coming to odd and even shifters have to be equal;
- *ser_clk* and *ser_clkn* have to be reversed in phase. This is done on top level design file with use just one inverter for one ADC IC.

B. LVDS Receivers (wrapper)

Encapsulating receiver building blocks into LVDS_Receiver and combining eight of them together as it's shown in *Fig. 5*a we will build a complete receiver set for one ADC chip AD9222. After wrapping this schematic the AD9222 LVDS Receivers graphical representation will look like it is shown in *Fig. 5*b.



a)



b)

Fig. 5. a) AD9222 LVDS Receivers; b) AD9222 LVDS Receivers Wrap;

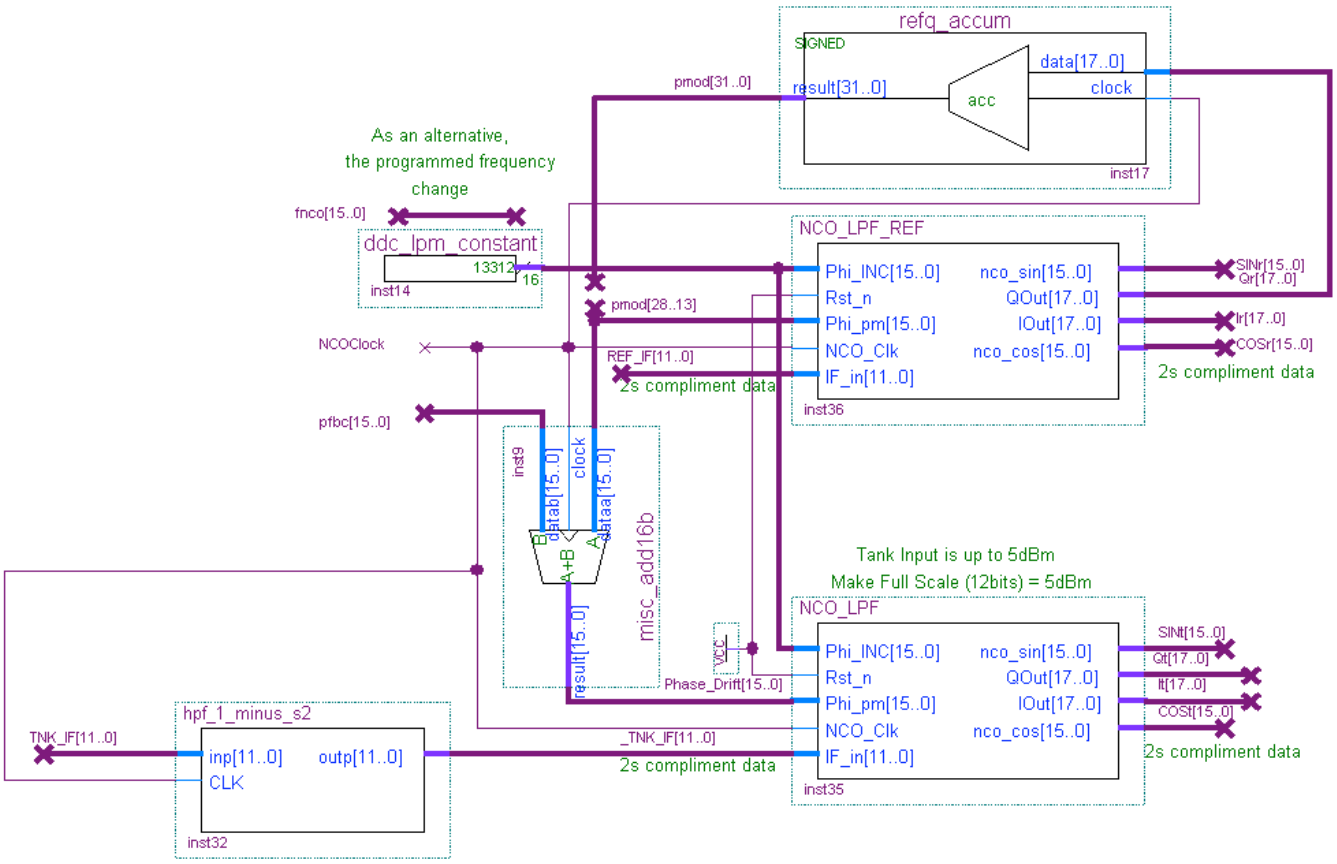


Fig. 6. IF_ref and IF_tank DDCs

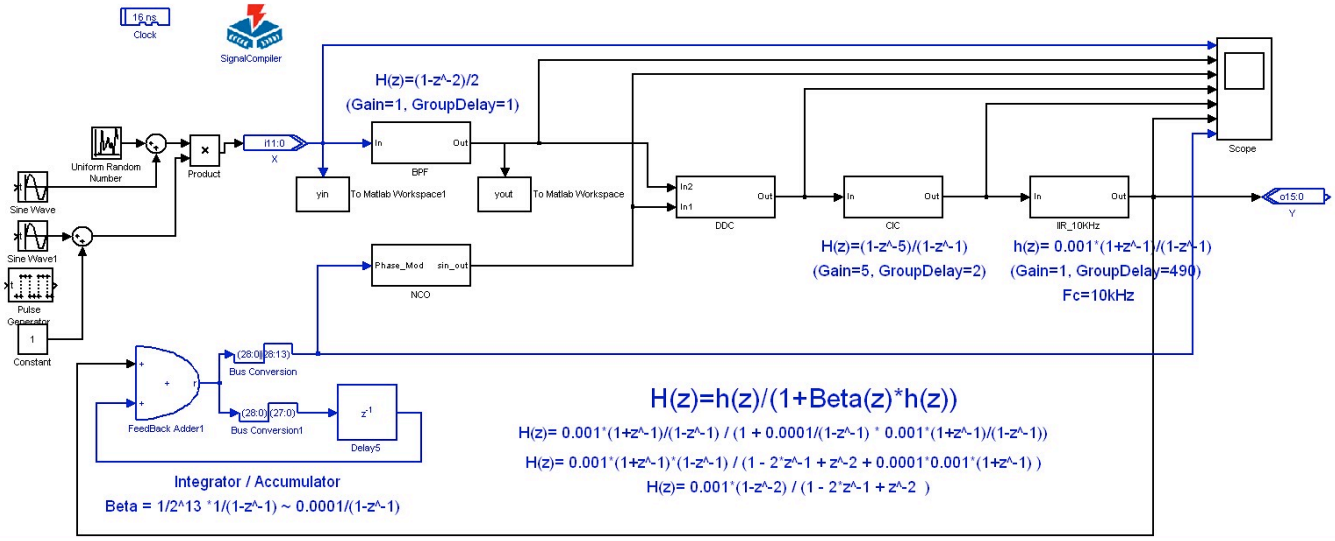


Fig. 7. DSP-Builder simulation diagram of PLL loop.

C. Intermediate Frequency Down-converters

1) Reference Line Phase-Locked IF Down-converter

The DDC tracks phase change in coaxial cable, which pulled from the tap on the 805-MHz reference line in the Linac tunnel to each station. Having the coax type, geometry and placement completely identical to other coax carrying tank IF to adjacent MFC board input, we can state that phase changes in both of them due to temperature and any other ambient conditions must be with a good approximation the same.

The phase tracking mechanism is based on the feedback

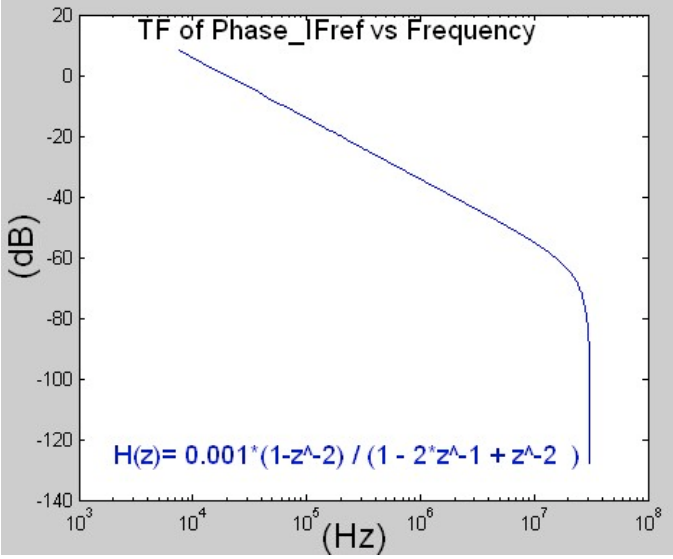
scheme, when DDC's Q-output is applied through integrator (accumulator) to the phase control input of NCO (see for DDC internal structure details in followed section). The scheme could be thought as a feedback loop with zeroed phase set point, which in turn eliminates a need in summing junction. The loop tries to keep Q-out zeroed all a time, so any phase change in reference coaxial cable is compensated at NCO phase input through the loop gain. The same phase error signal summed with input phase setting *pfbc* goes to IF_tank DDC phase control input (see *Fig. 6*).

Schematic diagram of PLL loop simulation in Matlab's is shown in *Fig. 7*. As it is seen the controller transfer function consists of two parts: CIC-filter and IIR-filter with transfer functions presented in Z-transform $H_{CIC}(z) = (1-z^{-5})/(1-z^{-1})$ and $H_{IIR}(z) = 0.001 \cdot (1+z^{-1})/(1-z^{-1})$ respectively. So total

$$H(z) = 0.001 \cdot (1-z^{-5}) \cdot (1+z^{-1})/(1-z^{-1})^2 =$$

$$= 0.001 \cdot (1-z^{-5}) \cdot (1+z^{-1})/(1-z^{-1})^2$$

the gain transfer function of reference line DDC (*Fig. 8*) at 0-crossover bandwidth is about 20 kHz.



2) Tank IF Down-converter

To prevent DDC's Local Oscillator (LO) leakage to output, IF_tank signal passes a simple HPF filter before entering the DDC (*Fig. 6*)

3) Digital Down-Converter Realization

The schematics for both Reference Line and Tank IF DDCs are similar, besides their cutoff frequencies on output of IIR filters (10kHz and 1 MHz for reference line and tank gradient signals respectively). The DDC (*Fig. 7*) consists of:

- Altera's IP-MegaCore function of numerically controlled oscillator (NCO);
- Mixing multipliers (I & Q networks);
- IIR elliptical 1st order filters (I & Q networks);
- x1.4 Scaler (I & Q networks);
- CIC filters, notching down second LO harmonic (I & Q networks);

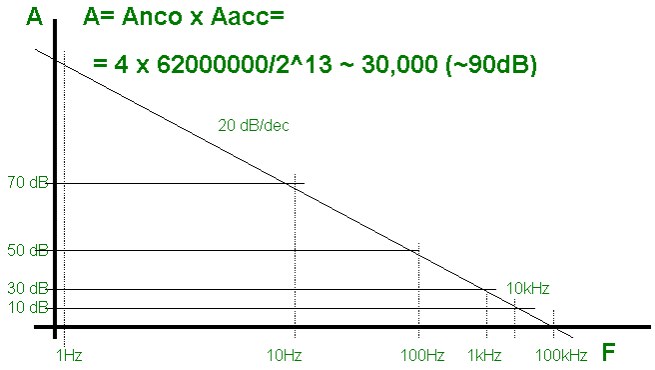


Fig. 8. Loop gain transfer function for phase locked reference line DDC;

NCO provides sine and cosine signals, which go to I- and Q- multipliers where they are mixed with IF inputs resulting in baseband and 2nd LO harmonic on multiplier outputs. The followed IIR filter rejects frequencies higher than 10kHz (reference line) or 1 MHz (tank gradient). Unfortunately, even after filtering the 2nd LO harmonic is still relatively high: about -40dBm. It will be taken care of by second dedicated 2nd order boxcar (or CIC) filter, which notches the harmonic by approximately 60dB down. And at last, the only block that left in signal path is the scaler, which adjusts an output dynamic range to full scale, i.e. the input IF signal of 1.1Vpp has to be transformed on DDC's I and Q outs into full range 18-bit words.

IV. Quadrature to Polar Converter

Clear that in a direct RF feedback system there is no a Quadrature to Polar Converter (QPC). Unfortunately, it is not a case for Low Energy Linac (LEL) upgrade. The direct RF feedback system was sacrificed for the sake of less manpower hungry dual system consisting of phase feedback and amplitude feedforward parts. From one hand, the approach doesn't require right away taking care of existing interlock system, but in the same time it leaves the room for future maneuvers, when it can be affordable, to turn the current LLRF upgrade into more functionally advanced direct RF system.

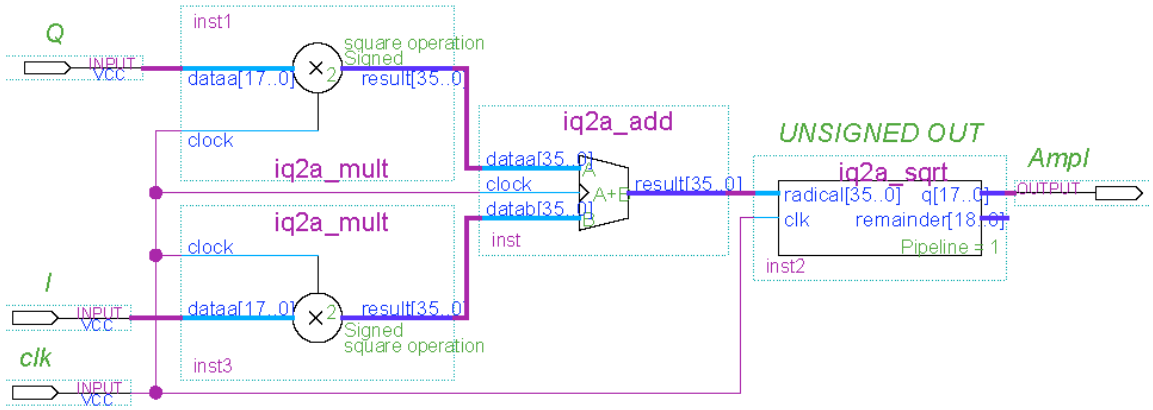


Fig. 9. Square root of sum of squares as an amplitude converter. The total latency is 3 samples;

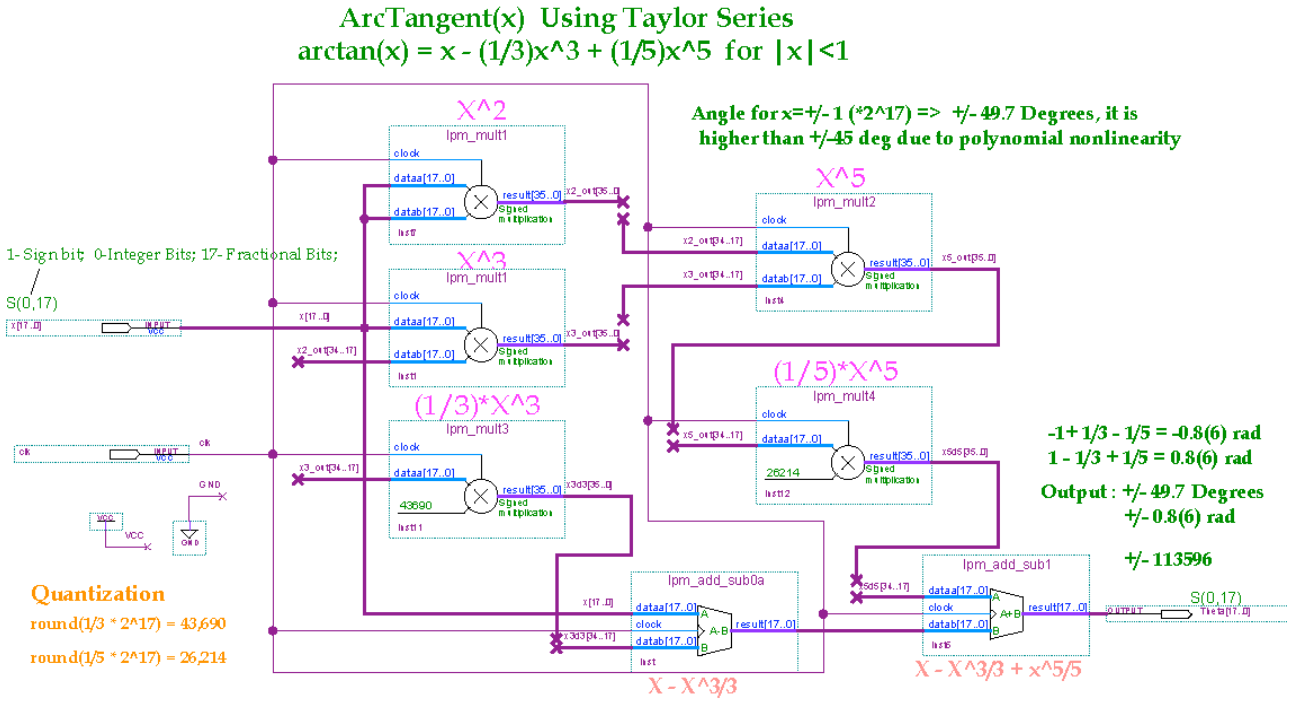


Fig. 10. Arctangent function implementation. The total latency is 5 samples;

I and Q outputs from gradient downconverter are used on QPC inputs, where they are converted to both polar amplitude and phase components. Actually, QPC consists of two independent parts: amplitude converter and phase converter, described below.

A. Amplitude

The amplitude conversion is a straight forward and follows the formula $A = \sqrt{I^2 + Q^2}$ (see *Fig. 9*).

B. ATAN Polynomial

To make a conversion from quadrature *I* and *Q* components to polar phase θ we have to calculate arctangent function $\theta = \arctan(Q/I)$. An elegant **CO**ordinate **R**otation **DI**gital **C**omputer or CORDIC *atan*-function algorithm [4] was disregarded due to its relatively long latency in contrast to polynomial algorithm. The provision of N-bit resolution in CORDIC's *atan* case requires N processing iterations (samples), while 18-bit implementation of polynomial algorithm in FPGA gives just 5 sample delay. This is already more than 3 times faster, but nevertheless there is a possibility to increase the ratio even higher – up to 5, by using two clock edges in pipeline. The polynomial formula used for *atan* is:

$$\theta = \arctan(Q/I) = (Q/I) - \frac{(Q/I)^3}{3} + \frac{(Q/I)^5}{5},$$ where *Q/I* is a division product.

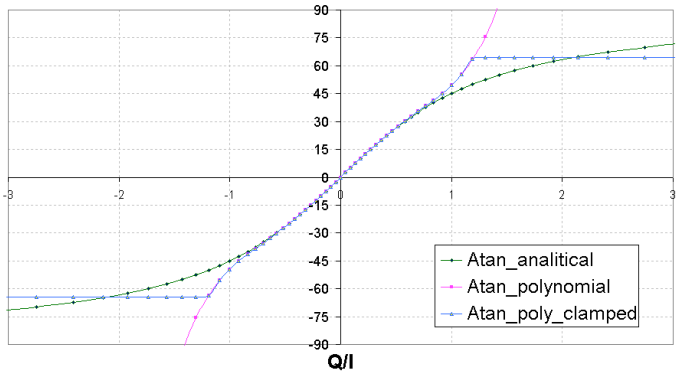


Fig. 11. Comparison of analytical, polynomial and polynomial-clamped arctangent functions;

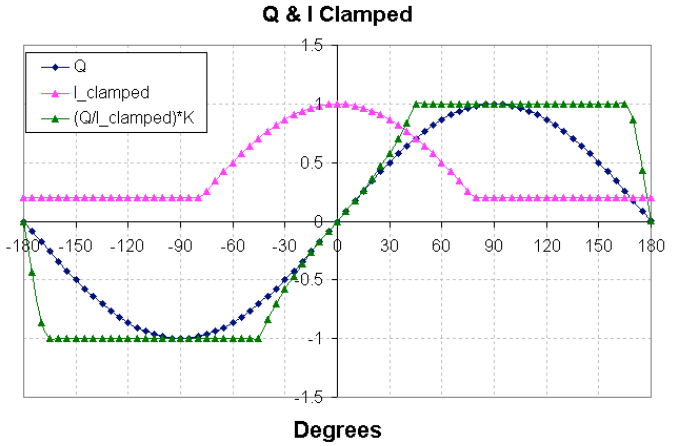


Fig. 12. I and Q are shown for sinusoidal baseband signal. I is clamped at the bottom, that prevents from division by zero. Q/I is clamped by 1, providing phase output after atan function within ±45°;

As it is seen in *Fig. 11*, there is a negligible difference between polynomial (pink) and analytical (green) arctangent curves on the interval ±45°. Beyond this interval the polynomial diverges significantly from analytical counterpart, but applied *Q/I* clamper (saturator) keeps the phase value within reasonable range, preventing it from overflowing. For example, setting the clamp to 1.3 extends the phase dynamic range up ±75°, though on the expense of nonlinear region. That gives extra headroom for feedback, which has to keep phase during beam on/off conditions below 0.4°. The purpose and implementation of the clampers is discussed in next paragraph. The FPGA implementation of polynomial is shown in *Fig. 10*, where there are 18-bit unitless data-in and sampling clock on input and 18-bit data-out (in radians) on output. Both in and out ports are presented in signed fixed point notation U(0,17) (see Appendix A for detail).

1) Imin and Q/I Clampers

To limit polynomial atan function nonlinear region, two clampers I_clamp and Q/I_clamp are used (*Fig. 12*). First one simply clamps I to positive number larger than zero. The number is set by user (usually this is 0.1) and prevents the further division Q/I from overflow. The result of division goes to IQ_Clamp, where its value gets saturated.

V. Transmitter

Each of four transmitters consists of 14-bit parallel output bus, connected to dedicated DAC channel. Output data is clocked out from FPGA by NCO clock and latched in into DAC by synchronous clock with adjustable phase coming from clock distribution IC

A. Multiplexed Signal Outputs

By default, the system starts on power-up with preselected output channel set Imod, Qmod, BeamFF and TunerControl directed to DACs. However, the built-in flexibility allows the monitoring usage of the same four DACs by means of multiplexing a lot of other internal parameters. To employ the feature, LSB or "debug" bit of *mode* register has to be set. Parameter group consisting of 2 signals is selected by *muxN* register. How it can be done with a total number of DACs is shown in Table 1. Regardless of "debug" bit state the selected group of signal is all a time routed to Test Point (1 through 4) waveform buffers, which allows capturing during system normal operation, i.e. at not "debug" mode (see *Fig. 13*). Each DAC_out function consists of two 16-bit multiplexers 8 to 1. The double size of generic function is defined by use on MFC board a dual DAC. So, in DAC_out function there are:

- two 18-bit inverted MSB output ports, which go to dual DAC inputs, providing offset binary data;
- two 18-bit output ports provide signed (twos complement) data to Test Point Waveform buffers;
- all output data is latched on output dff-triggers.

Table 1 Parameter groups selected by muxN.

	DAC_out 1		DAC_out_2	
muxN	**TP1**	**TP2**	**TP3**	**TP4**
0	REF_IF	Tank_IF	SINr	COSr
1	Qr	Ir	Qt	It
2	Prepulse	HEP_gate	NTF	STU
3	FTA_gate	Slope_gate	Torroid	A_tank
4	Tlim_gate	BL_Aver_gate	Start_Snapshot	FF_Interlock
5	Aerr	Thetat	Sample_gate	Pulse_End
6	Rst	HEP_out	NTF_out	STU_out
7[(a)]	Qmod	Imod	Fwd_Power	FF_out

[(a)] *This is power-up default value.*

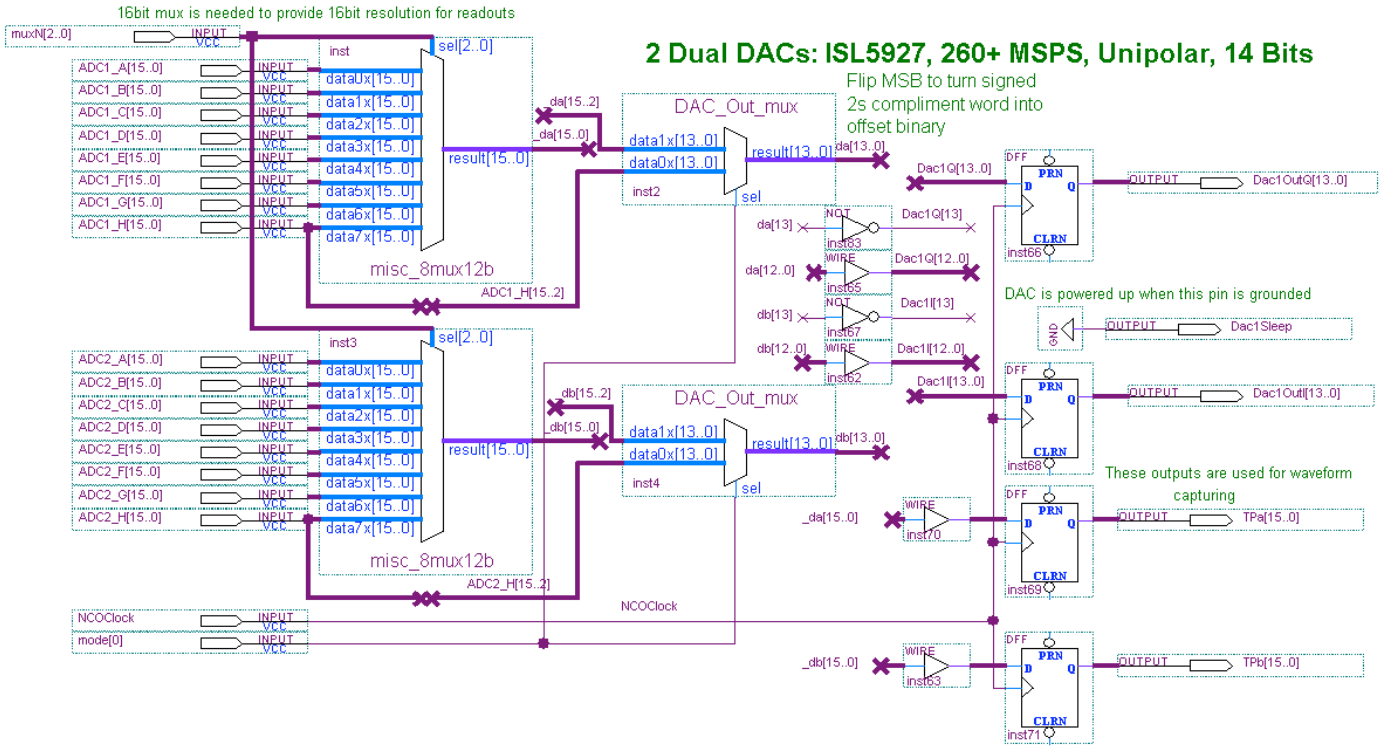


Fig. 13. The core of DAC_out function, the wrapped version of which in quantity of two are used in the top design;

VI. Amplitude Feedforward Loop

A. Error Signal Calculation

An error signal is defined as tank amplitude (A_tank) difference between its average value during beam loading condition and average value on flat-top area. The averaging is performed during *FF_gate* and *FTA_gate* in first and second cases respectively (see Appendix B for timing details). The result then gets scaled by *wght* coefficient from 0 to 1.0 of its original value. The schematic of discussed circuitry is shown in *Fig. 14*.

This part doesn't add up any latency to the data path due to the fact that error is calculated during N-th cycle and applied immediately when it's required on N+1-th cycle, which defines the main feature of FEEDFORWARD technique.

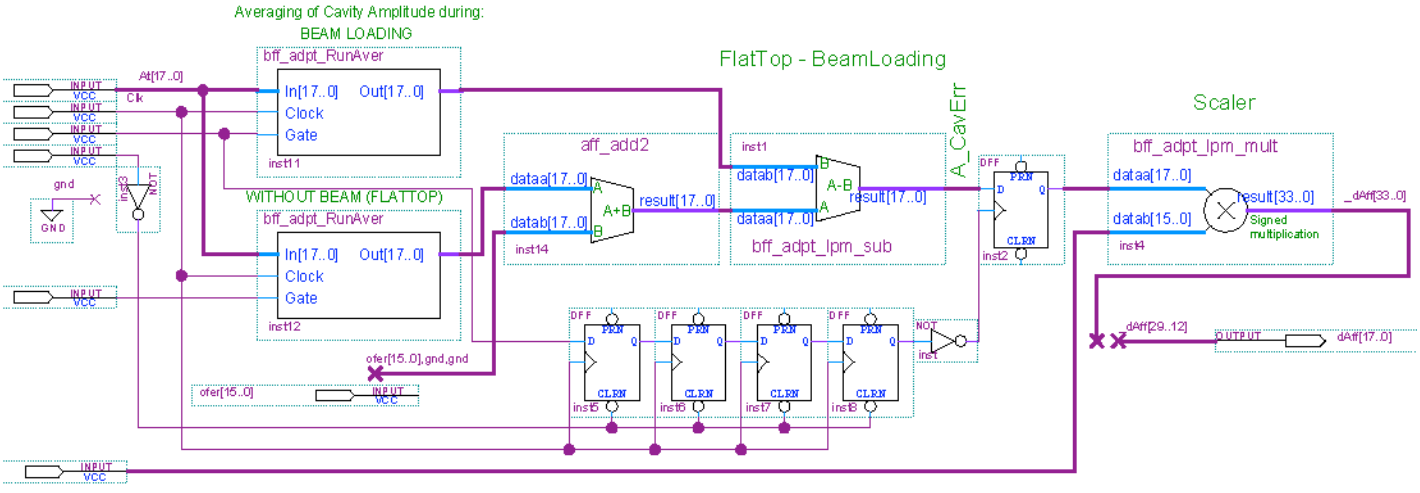


Fig. 14. Feedforward Error Calculator (AFF.bdf function);

B. Manual and Adaptive Modes

The system has two operation modes: adaptive and manual. *ADPT* input to FFS1.bdf function determines which one is in effect at a moment (*Fig. 15*).

Feedforward pulse in any mode is composed by major six input parameters: fife of which *Slope_Gate*, *BL_Gate, Stas_Gate, smag* and *tlop* are common for both modes. The sixth parameter depends on mode and determines the pulse magnitude. In manual mode the magnitude parameter is simply equal to *wght* value, of course, after applying two left shifts, justifying for bit-resolution difference. In adaptive mode it is a product of *wght*Aer* multiplication plus spike magnitude *smag* (see Appendix C).

C. Composing the Feedforward Pulse

The composite pulse is shown in *Fig. 16*. As it can be seen, the *Slope_Gate* specifies a ramp stage of the pulse. *Stas_Gate* forms a spike pulse. *BL_Gate* is responsible for the rest of the pulse, at which the magnitude inclines by degree specified by *tlop* word. While the ramp prevents modulator from sparking on sharp front (the typical slope time is approximately 1 μs), the spike eliminates beam loading transient and the *tlop* compensates for pulse tail rise in adaptive mode. This value is yet to be found for each station.

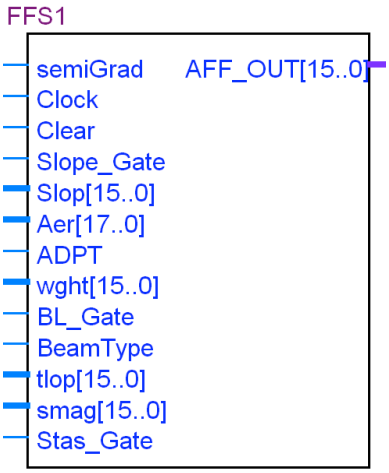


Fig. 15. The Wrapper of Feedforward Pulse Composer;

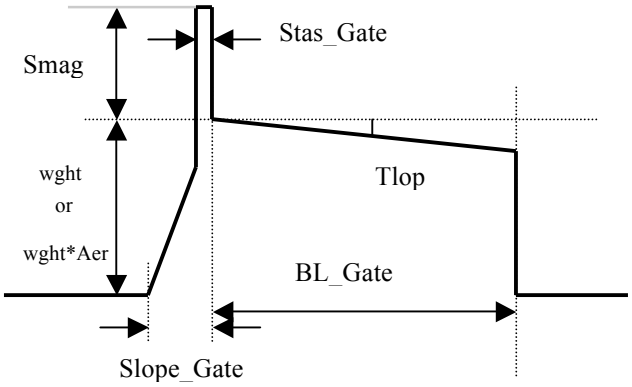


Fig. 16. Feedforward Composite Pulse;

D. Beam Type Manipulation

There are three beam types used in Linac. They differ mostly by intensity and beam loading timing. A feedforward beam qualifier pulse accompanying each RF-cycle at beam presence allows the system to decode what type is going to be present during the cycle. The decoder residing within timing unit passes the information through *Beam_Type[1..0]* bus to the stand alone two-to-four decoder. Outputs of that decoder are used as selectors on *BeamType* inputs on three independent FFS1.bdf functions, - separate for each beam type. Then, the composite pulse outputs from these threes are directed to multiplexer (see Appendix C for details), controlled by *Beam_Type[1..0]* bus as well. The multiplexer finally routes the selected compensation pulse to the DAC.

E. Clamp

The clamper implemented in FFS1.bdf prevents output from become negative and clamps the maximum value.

F. Adaptive Feedforward Algorithm

The implemented adaptive algorithm calculates tank amplitude error as a difference between amplitudes on flattop and during beam loading. Then it scales the error and adds the result to current beam feedforward pulse. The obtained value will be used as beam feedforward during next beam loading conditions. See staged representation in *Fig. 17*.

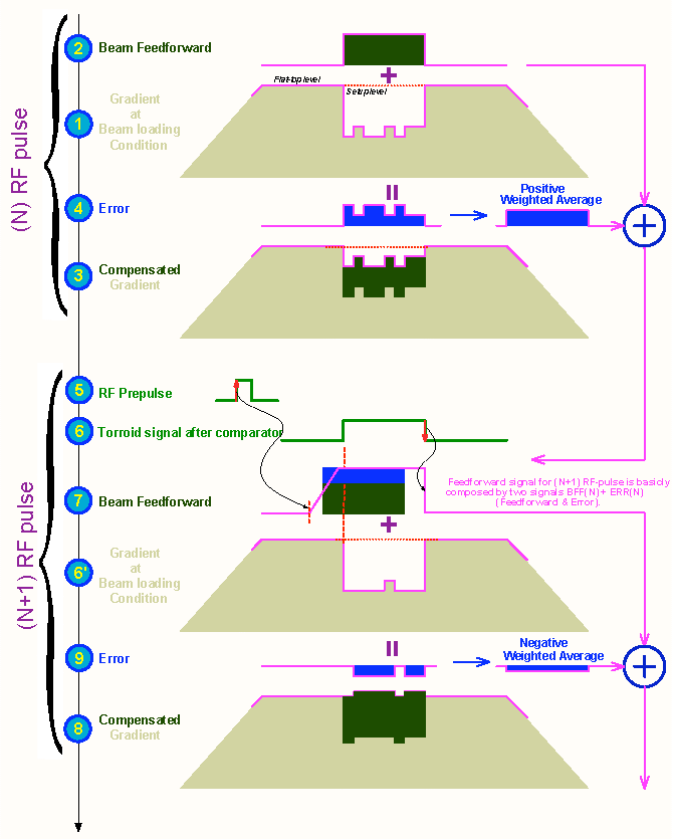


Fig. 17. The Feedforward Algorithm Implementation;

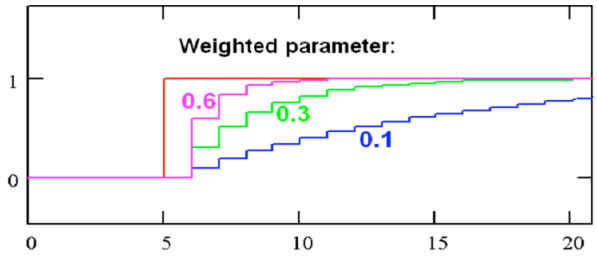


Fig. 18. Asymptotic convergences of feedforward step function versus wght parameter;

VII. Phase Feedback Loop

The phase feedback is second major loop in the system. Partially in Chapter IV, we have already started discussing how tank gradient's quadrature components get converted in phase with use of *atan*-polynomial function. We'll continue description of the loop from this point and on.

A. Loop Gain

The phase loop gain is controlled by *Gain* register through multiplier (see *Fig. 20*). Multiplied *K*Theta* signal is used for both a monitoring and for a Polar-to-Quadrature Conversion (PQC).

B. Polar to Quadrature Converter

The PQC conversion is performed with normalized amplitude, i.e. there is only a phase component presented in *_Ie* and *_Qe* outputs. The conversion is based on formulas:

$$\_I_e = A_{NORM} \cdot \cos(Gain \cdot \theta_{ER})$$

$$\_Q_e = A_{NORM} \cdot \sin(Gain \cdot \theta_{ER}), \text{ and}$$

realized in *Thet_2_IQ.bdf* with use of 5th order polynomials:

$$\_I_e = 1 - \frac{(Gain \cdot \theta_{ER})^2}{2!} + \frac{(Gain \cdot \theta_{ER})^4}{4!}$$

$$\_Q_e = \frac{Gain \cdot \theta_{ER}}{1!} - \frac{(Gain \cdot \theta_{ER})^3}{3!} + \frac{(Gain \cdot \theta_{ER})^5}{5!},$$

where $A_{NORM} = 1$. To figure out a difference between analytical and polynomial *Sine & Cosine* functions take a look at *Fig. 19*. It is seen that practically there is no difference between them on ±30° interval. For applications where it is required bigger range, the CORDIC algorithm [5] can be used.

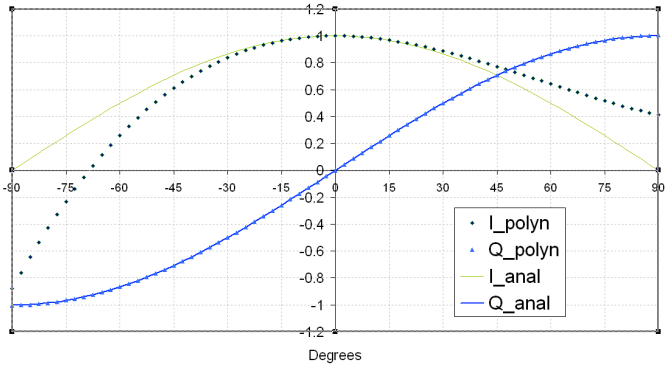


Fig. 19. Comparison of analytical and polynomial sine & cosine functions;

C. Output I & Q Adjustment

_Ie and *_Qe* are mixed with user defined *I_adj* and *Q_adj* terms in *_IQ_out_Adj.bdf* function, providing output phase tuning (fan-out phase adjustment) by formulas:

$$I_{\text{mod}} = A_{adj} \cdot \cos(Gain \cdot \theta_{ER} + \theta_{adj})$$

$$Q_{\text{mod}} = A_{adj} \cdot \sin(Gain \cdot \theta_{ER} + \theta_{adj})$$

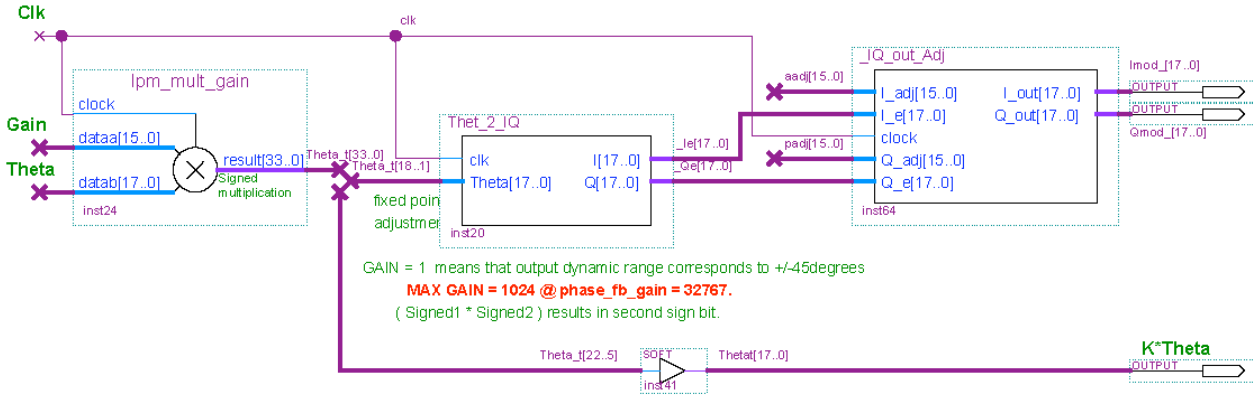


Fig. 20. The part of phase feedback loop in Phase_FB.bdf file after atan polynomial;

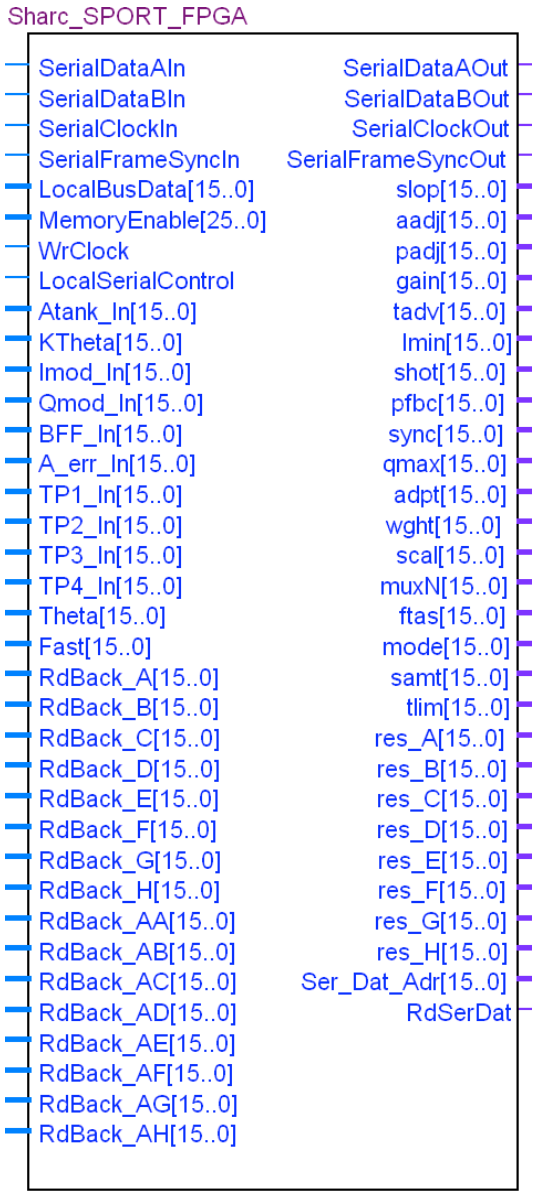


Fig. 21. Sharc-FPGA interface through SPORT;

VIII. Tuner Control Loop

To keep cavity at resonance the slow tuner control loop is used. The forward power IF frequency (IF_{FWD}) of 12.58 MHz comes to control loop input from RF-module. The downconverter in the loop doesn't have NCO but uses simplified scheme instead. In this approach IF_{FWD} frequency is mixed on multiplier with programmatically delayed IF_{TANK} (see System Block Diagram in Appendix C). The *Delay* register can be programmed to one of fife possible values (0^o, 72^o, 144^o, 216^o, 288^o), providing the better resolution for downconverted baseband signal. Then the result is filtered on low-pass filter (LPF) and scaled by *SCAL* value on multiplier. The averaged value during beam loading then latched and kept on DAC output before next rf-pulse.

IX. Control and DAQ Utilities

A. Serial SPORT and Local Bus HOST Interface

This is a very simple and first implemented serial interface that we commissioned and started using for communication between Sharc DSP and Cyclone II FPGA, while a parallel local bus interface had been worked on. Since we have not moved all of FPGA input/output registers to parallel bus yet it is left here for consistency (see *Fig. 21*).

1) Data Receive through Serial SPORT

FPGA receives data from Sharc DSP via SPORT with use of *SerialDataAIn* and *SerialDataBIn* inputs. The reception takes place only if serial register address is even. Address and data are clocked in by *SerialClockIn* and *SerialFrameSyncIn* signals the way it is specified in [6]. Received data updates one of the corresponding 16-bit set parameter on the output of Sharc_SPORT_FPGA.bdf function: *slop, aadj, padj, gain, tadv, Imin, shot, pfbc, stnc, qmax, adpt, wght, scal, muxN, ftas, mode, samt, tlim, resA through resH*. For detail parameter addresses see Appendix D.

2) Data Transmit through Serial SPORT

If the address of serial register (see previous paragraph) is odd, then the current transaction is decoded as transmission from FPGA. The output data is latched in transmitter and clocked out from FPGA to Sharc through *SerialDataBOut* output by *SerialClockOut* and *SerialFrameSyncOut* signals.

For example: for set parameters described above the transmission to Sharc can occur if addresses of parameters incremented by one. The other parameters (read only):

- waveforms: *Atank_In, KTheta, Imod_In, Qmod_IN, BFF_In, A_err_In, TP1_IN, TP2_In, TP3_In, TP4_in and Theta*;
- readbacks: *RdBack_A through RdBack_G and RdBack_AA through RdBack_AH*;

are transmitted to Sharc as well by issuing address select and read strobe on *Ser_Dat_Adr* and *RdSerDat*. For address map see Appendix D.

3) Interchangeability of Serial and Host Interfaces

Historically we started working with system when it was just an evaluation board prototype with the only control option through serial interface. When new VXI card become available, the firmware was ported on it. As the system getting mature, the more parameters are available for control through Parallel Local Bus. So far we have to keep both the Serial and Parallel Interfaces available for communication between Sharc DSP and Cyclone FPGA. Sharc_SPORT_FPGA.bdf provides the interchangeability by switching control to one or other bus with use of *LocalSerialControl* qualifier on input (0/1 stands for Serial/Parallel). By default on power-up the control is granted to Serial bus, but at anytime the write from Local Bus to *ControlBit* (see Appendix D) register can switch the control to Parallel bus. It is done internally on multiplexers, which select parallel bus data, address and write strobe from *LocalBusData, MemoryEnable* and *WrClock* inputs.

B. *Waveform Capture Controller and Buffers*

Eleven waveforms *A_tank, Theta, Imod, Qmod, ff_out, A_er, tra, tpb, tpc, tpd and Pure_Theta* are captured during each rf-cycle, triggered by *StartSnapshort* pulse generated within TimingUnit. The start position of the pulse is specified by *shot* setting in respect to *LLRF_Trigger* signal (see Appendix C,D).

StartSnapshort pulse and NCO clock divided by 12 waveform controller inputs shown in *Fig. 22* are used for generation all necessary signals for capturing: *Snap_Clk, Snap_Enable, DiagnosticMemoryAddress*. The signals then go to *Write, WriteEnable* and *WrAddr* inputs of Waveforms.bdf function (*Fig. 23*) where they used for simultaneous writing into eleven 16bit-wide 4kWord-deep buffers. Four test point buffers called *tra, tpb, tpc* and *tpd* use *muxN*-multiplexed signals as it was shown in Table 1.

The stored buffers are available by the end of each rf-cycle. Slot-0 controller is informed about the event through *Trig2* backplane line. The buffers are decoded through *MemoryEnable* lines and the values from the *RdAddr* are read back to VXI controller by *Read* pulses.

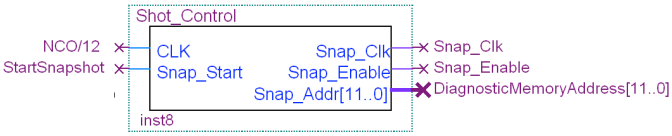


Fig. 22. Waveform Controller;

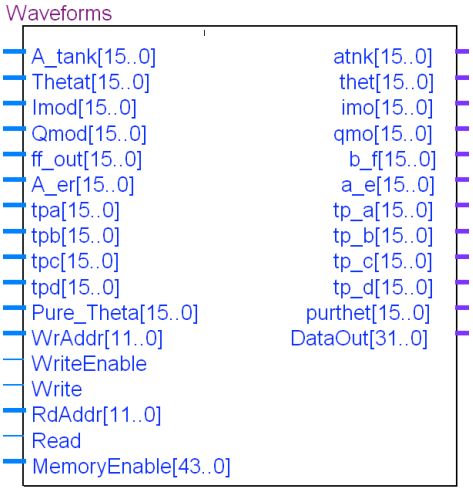


Fig. 23. Waveform buffers;

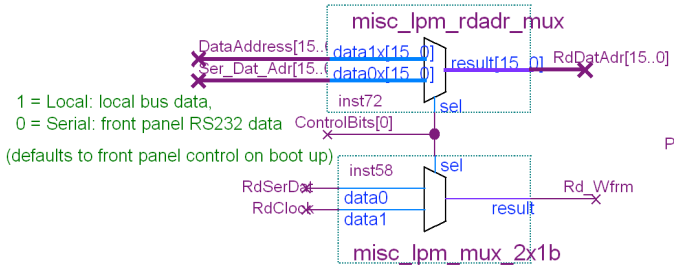


Fig. 24. Waveform buffers readback selector;

X. Locking 201.25MHz phase Quadrant

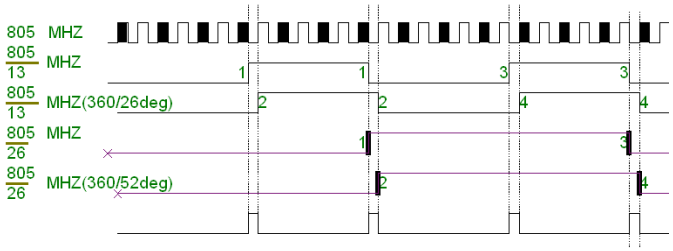


Fig. 25. Locking the Phase to proper quadrant;

To provide the synchronous rf phase in all fife Low Energy Linac cavities their all respective LLRF systems has to be locked to the same quadrant of 805MHz reference line. In real life it is much easy to do in respect to 201.25MHz or even low reference frequency and that has been done.

201.25MHz reference is generated at the same source where 805MHz line is available at each station.

The idea how the phase lock can be made is shown in *Fig. 25*. Black-solid color represents a "virtual" search quadrant on 805MHz reference line. The "virtuality" becomes more observable if we would take a look at 201.25MHz reference line, but we'll skip it and start with 805/13 MHz line, which is available in FPGA.

If we will switch a glance

XI. DSP Firmware

A. *Software Structure*

XII. Power-up Sequence

A. *Initialization Section*

B. *Libraries*

C. *Interrupts*

D. *Debug Options*

E. *Main function*

XIII. Discussion

XIV. Conclusion

1) To reduce the latency of trigonometric function calculations (either by CORDIC or polynomial) it is proposed to use the multiple of the sampling frequency within conversion block. The frequency and its phase are to be programmed in FPGA's PLL unit.

2) Current version (7 March 2008) Sharc firmware mostly centered around serial SPORT transmission between SHARC and FPGA. This is baud-rate limited approach requires making a modification to Sharc code. That in turn would allow to

APPENDIX A: FIXED POINT ARITHMETIC BASICS

Though Altera's Cyclone II supports floating-point calculations, but this is not as great an approach for signal processing as it is for general-purpose computing because the range of values needed in FPGA is fairly small. For this reason, *fixed-point* arithmetic is used. It provides such crucial for any feedback systems advantages as a minimal latency possible along with reduced number of utilized resources. Though, it comes on expense of code developer, who has to be very concentrated during programming, tracking the point position after virtually each mathematical operation.

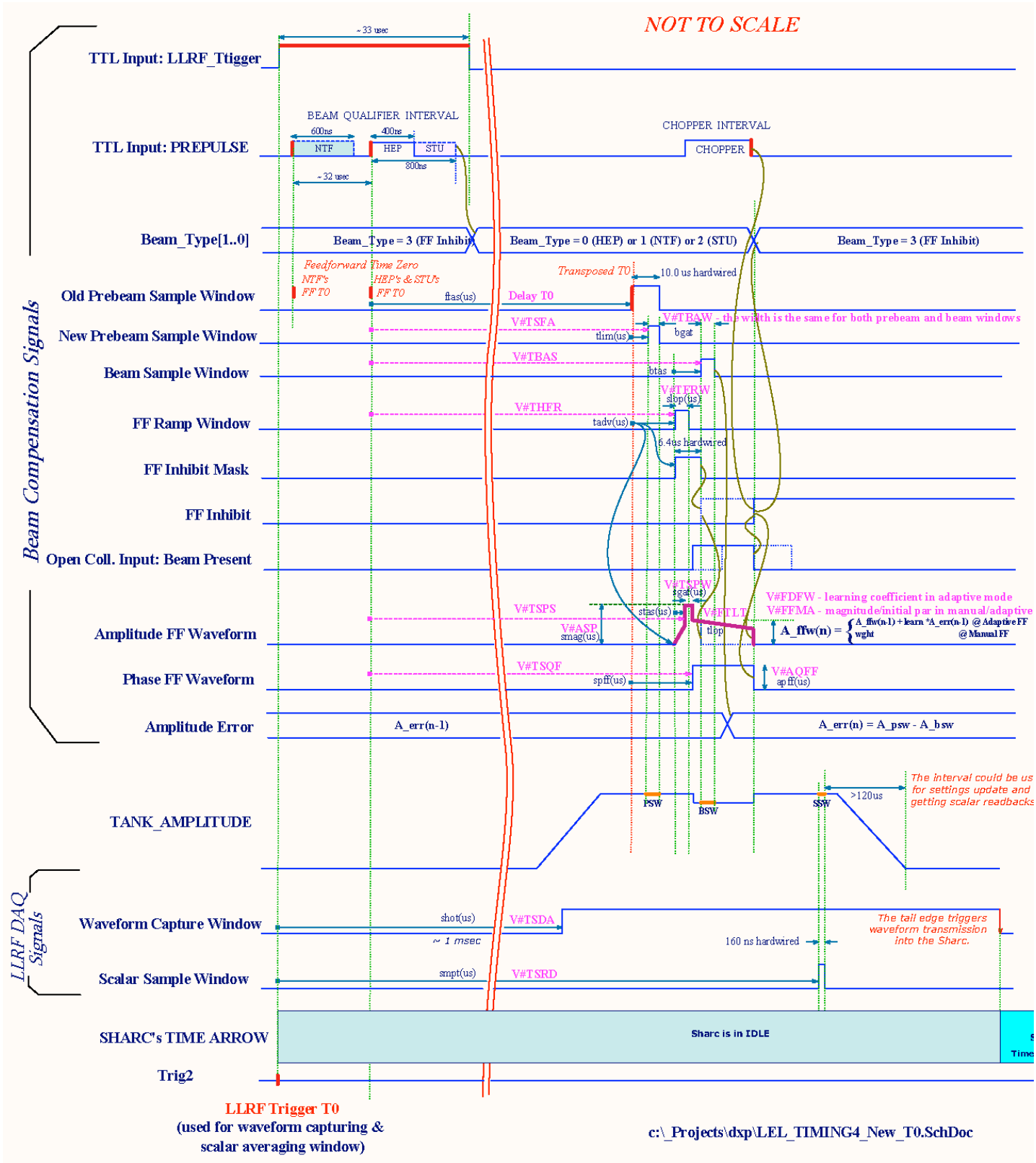
A. Examples of fixed point number representations

- Unsigned number U(2,3) has 2 + 3 = 5 bits and the range is from 0 to $2^2 - 2^{-3} = 3.875$. The value 0x1A (1,1010b) is $11.010_b = 1\bullet2^1 + 1\bullet2^0 + 0\bullet2^{-1} + 1\bullet2^{-2} + 0\bullet2^{-3} = 3.25$;
- Unsigned number U(-3,11) has -3 + 11 = 8 bits and the range is from 0 to $2^{-3} - 2^{11} = 0.12451171875$. The value 0x4b or $0100,1011_b = 0\bullet2^{-3} + \underline{1\bullet2^{-4}} + 0\bullet2^{-5} + 0\bullet2^{-6} + \underline{1\bullet2^{-7}} + +0\bullet2^{-8} + 1\bullet2^{-9} + 1\bullet2^{-10} = 0.0732421875$;
- Signed numbers are presented in two-s complement notation and have one extra sign bit in respect to unsigned numbers.

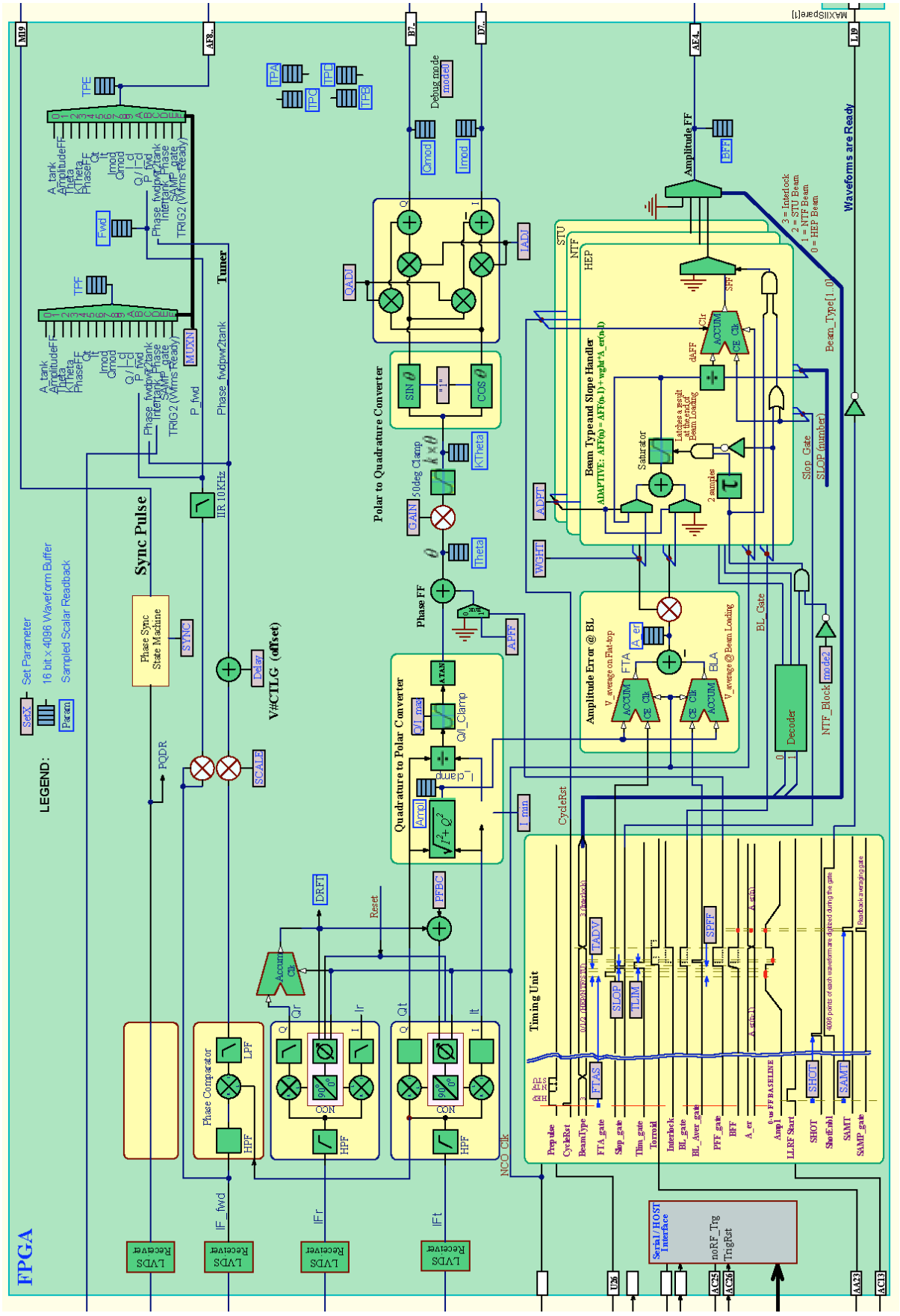
B. Fixed Point Arithmetic Rules

- Unsigned Wordlength: U(a,b) = a + b;
- Signed Wordlength: U(a,b) = a + b + 1;
- Unsigned Range: $0 \le x \le 2^a - 2^b$;
- Signed Range: $-2^a \le x \le 2^a - 2^b$;
- Addition operand: X(a,b) + Y(c,d) is only valid if a=c and b=d;
- Addition result: X(a,b) + Y(a,b) = Z(a+1,b);
- Unsigned multiplication: X(a,b)•Y(c,d) = Z(a+c, b+d);
- Signed multiplication: X(a,b)•Y(c,d) = Z(a+c+1, b+d);
- Unsigned division: $X(a,b)/Y(c,d)=Z(a+c,\ \log_2(2^{c+b} - 2^{b-d}))$;
- Signed division: X(a,b)/Y(c,d)=Z(a+d+1, c+b);

APPENDIX B: THE SYSTEM TIMING DIAGRAM



APPENDIX C: THE SYSTEM BLOCK DIAGRAM



APPENDIX D: FPGA PARAMETERS MEMORY MAP

FPGA Parameters, Serial & ACNET Memory Maps

V.Tupikov

	Param Group	FPGA Param Name	Alies	ADDRESSES via sPORT (HEX) Set	Read	Through	ADDRESSES via Local Bus (HEX) Set	Read	Through	ENGINEERING UNITS Range	Clamp	Default	INTEGER UNITS Range	Clamp	Default	Terminal Commands Wr	Rd	ACNET Commands Wr	Rd	NOTES
1		slop		0	1		0	1		0 .. 1023us		1 us			0x3d	slop	rad6 0			
2		aadj	(iadj)*	2	3		400	401		0..1		0.8			?	aadj	rad6 1			
3		padj	(qadj)*	4	5		800	801		[+/-180] deg		0			?	padj	rad6 2			
4		gain		6	7		C00	C01		0..1023		0			0	gain	rad6 3			
5		tadv		8	9		1000	1001		0..1023us		30us			0x741	tadv	rad6 4			
6		imin		a	b		1400	1401		0..1		0.1			0x1999	imin	rad6 5			
7		shot		c	d		1800	1801		0..4095us		1020us			0xf6b9	shot	rad6 6			
8		pfbc		e	f		1C00	1C01		[+/-180]deg		0 deg			0	pfbc	rad6 7			
9		sync		10	11		2000	2001		0..1		0			0	sync	rad6 8			
10	SET	qmax		12	13		2400	2401		0..2		0.3			0x266	qmax	rad6 9			
11		adpt		14	15		2800	2801		0..1		0			0	adpt	rad6 10			
12		wght		16	17		2C00	2C01		0..1		0.2			0x1999	wght	rad6 11			
13		scal		18	19		3000	3001		0..1		1			0x7fff	scal	rad6 12			
14		muxN		1a	1b		3400	3401		0..7		7			0x3aa8	muxN	rad6 13			
15		ffas		1c	1d		3800	3801		0..4095us		970us			0x	ffas	rad6 14			
16		mode**		1e	1f		3C00	3C01		0..65535		0			0	mode**	rad6 15			
17		samt		20	21		10400	10401		0..4095us		1100us			0x4284	samt	rad6 16			
18		tlim		22	23		10800	10801		0..1023us	5us	5 us		?	0x135	tlim	rad6 17			
19		RES_A	(Delay)	44	45		12C00	12C01									rad6 26			BFF flattop incline
20		RES_B	(Itop)	46	47		13000	13001		-32768	0 cnts/us	32767			0x0000	itop	rad6 27			BL aver start
21		RES_C	btas	48	49		13400	13401		0..1023us		5 us			0x135	btas	rad6 28			BL aver winsow
22		RES_D	bgat	4a	4b		13800	13801		0..1023us		5 us			0x135	bgat	rad6 29			Spike start
23		RES_E	stas	4c	4d		13C00	13C01		0..1023us		5 us			0x135		rad6 30			Spike window
24		RES_F	sgat	4e	4f		14000	14001		0..1023us		5 us			0x135		rad6 31			Spike magnetide
25		RES_G	smag	50	51		14400	14401		0..1										
26		RES_H		52	53		14800	14801												
27		ControlBits***					4000	4001												Serial/Local tougle
28		smp0	(Atank)	24	25		10C00		20fff	0..1							smp0			
29		smp1	(Theta)	26	27		11000		21fff	[+/-180] deg							smp1			
30		smp2	(A_en)	28	29		11400		22fff	0..1							smp2			
31		smp3	(Drift)	2a	2b		11800		23fff	[+/-180] deg							smp3			
32		smp4	(Imod)	2c	2d		11C00		24fff	0..1							smp4			
33		smp5	(Qmod)	2e	2f		12000		25fff	0..1							smp5			
34		smp6	(BFr)	30	31		12400		26fff	0..1							smp6			All readbacks
35		smp7	(Pfwd)	32	33		12800		27fff	0..65536		2			0x0002		smp7			are averaged over
36	READBACK	smp8	FaultCnt	34	35		14C00										smp8			256 points
37		smp9		36	37												smp9			
38		smp10		38	39												smp10			
39		smp11		3a	3b												smp11			
40		smp12		3c	3d												smp12			
41		smp13		3e	3f												smp13			
42		smp14		40	41												smp14			
43		smp15		42	43												smp15			
44		A_tank			1000	1fff		20000	20fff								ara0 dec#			
45		KTheta			2000	2fff		21000	21fff								ara1 dec#			Send out 4096 point
46		Imod			3000	3fff		22000	22fff								ara2 dec#			buffer with applied
47		Qmod			4000	4fff		23000	23fff								ara3 dec#			dicametion.
48	WAVEFORM	FF_Out			5000	5fff		24000	24fff								ara4 dec#			
49		A_er			6000	6fff		25000	25fff								ara5 dec#			
50		tpa			7000	7fff		26000	26fff								ara6 dec#			Signals are
51		tpb			8000	8fff		27000	27fff								ara7 dec#			selected by
52		tpc			9000	9fff		28000	28fff								ara8 dec#			muxN
53		tpd			a000	afff		29000	29fff								ara9 dec#			
54		Theta			b000	bfff		2a000	2afff								araa dec#			
55		fast			c000	cfff		2b000	2bfff								arab dec#			
56																				
57		wadN														wad1		wad1		write to ADC1
58																wad2		wad2		write to ADC2
59	SPI****				N/A			N/A								wad3		wad3		write to ADC3
60																wad4		wad4		write to ADC4
61																wad5		wad5		write to Clock_IC
62		radN															rad1			read from ADC1
63																	rad2			read from ADC2
64																	rad3			read from ADC3
65																	rad4			read from ADC4
																	rad5			read from Clock_IC

muxN	TPA	TPB	TPA	TPB
0	Ifr	Ift	SINr	COSr
1	Qr	Ir	Qt	It
2	Prepulse	HEP	NTF	STU
3	FTA_gate	Slop_gate	Toroid	Ampl
4	Tlim_gate	BL_Aver	Samt	Interlock
5	A_er	Ktheta	Samp_gate	*ShotEnbl
6	CycleRst	HEPout	NTFout	STUout
7	Qmod	Imod	Tuner	BFF

*) The command known to Terminal is aadj & padj for update in engineering units;
**) bit0=MUX_2_SCOPE, bit1=FAST_SAMPLING, bit2=NTF_BLOCK
***) Bus supervision is available only through VXI.
****) Though SPI devices don't relate to the table purpose, they are listed here for consistency

#) dec field represents decimation number 2^dec:
0 - no decimation
1 - decimate by 2
2 - decimate by 4
3 - decimate by 8
4 - decimate by 16
5 - decimate by 32
6 - ZOOM excempt: by this qualifier the DSP will send very first 256 points from a buffer

ACKNOWLEDGMENT

The preferred spelling of the word “acknowledgment” in American English is without an “e” after the “g.” Use the singular heading even if you have many acknowledgments. Avoid expressions such as “One of us (S.B.A.) would like to thank ... .” Instead, write “F. A. Author thanks ... .” **Sponsor and financial support acknowledgments are placed in the unnumbered footnote on the first page, not here.**

REFERENCES

[1] L. Allen, “2005 Proton Plan Review,” Available: www-accel-proj.fnal.gov/Proton_Plan/index.html
[2] T. Butler, L.Allen, E.Cullerton, V.Tupikov, *Proposed LLRF Improvements for Fermilab 201.25MHz Linac,* Proc. Of LINAC 2006, Knoxville, TN, USA;.
[3] P.Varghese, B.Chase, B.Barnes, J.Branlard, P.W.Joireman, D.Klepec, U.Mavric, V.Tupikov, *Multichannel Vector Field Control Module For LLRF Control Of Superconducting Cavities,* Proc. Of PAC07, Albuquerque, New Mexico,. USA;.
[4] U.Rohde, J.Whitaker, *Communication Receivers - DSP, Software Radios, and Design*, 3d ed., McGRAW-HILL, 2001
[5] J.Stein, *Digital Signal Processing - Computer Science Perspective*, 2nd Ed, 2000;
[6] ADSP-21368 *Sharc Processor Hardware Reference*, Analog Devices Inc, September 2006;
[7] Ed’s LLRF Trigger, 2008;
[8] Paul’s Linac LLRF Timing Signals, 2008;

XV.

List of changes applied to new LLRF System on Low Energy Linac (May-June 2009)

1. Feedforward list of parameters was triplicated allowing now to make fine tuning in manual and adaptive mode separately for each beam type: HEP, NTF and STU. The following properties can be separately adjusted:
    a. Feedforward ramp start;
    b. Feedforward ramp width;
    c. Feedforward learning coefficient;
    d. Feedforward Manual amplitude / Adaptive amplitude initial value;
    e. Feedforward tilt;
    f. Beam loading sample start (the width parameter is common for all beam type samples);
    g. Spike start;
    h. Spike width;
    i. Spike amplitude
2. Adaptive mode for HEP, NTF & STU can be set separately.
3. Waveform capturing trigger was updated allowing now to get snapshots by following events:
    a. LLRF Trigger;
    b. HEP beam;
    c. NTF beam;
    d. STU beam;
    e. Any combination of b, c and d;
4. Extra sampling window was created, which works together with beam loading sample allowing gradient slope determination during beam loading. The result is available as scalar readback parameter and can be used for closing external by applying calculated correction to tilt parameter.
5. Adaptive mode update event was fixed. The amplitude is now updated only in a case when torroid signal was detected.
6. One of front-panel serial connector (top one) was assigned as a debug port for four signals (one of which is multiplex-able by new DEBX command which available from rs232 terminal) for in field system verification.

Adaptive mode for HEP, NTF & STU can be set separately

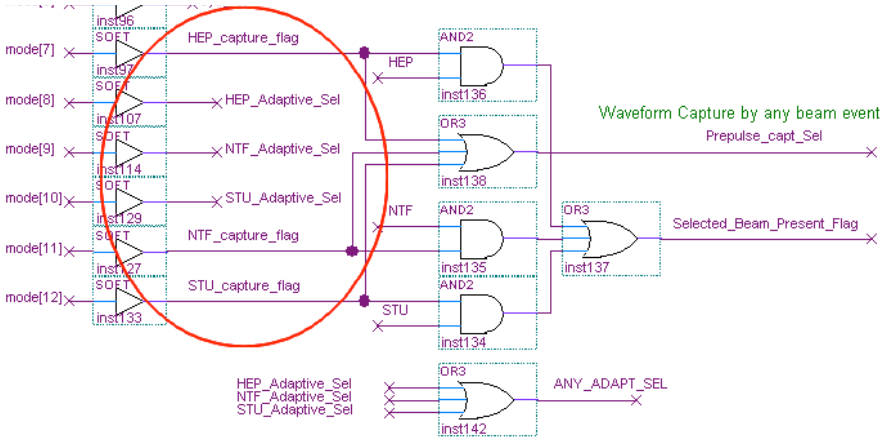


Fig. 1 Adaptive mode for HEP, NTF & STU can be set separately (see mode[8], mode[9] & mode[10] bits of binary-switch control word).

TABLE MODE Parameter Bits

FPGA Name	Bit #	Description
Mux2scope	0	Not used
Sampl16ns	1	Switches between 192 ns and 16 ns sampling rate
Ntf_inhibit	2	FF will be prohibited during NTF beam
Rst_flt_cntr	3	Clear fault counter
ZeroPDrift	4	Reset Phase Drift Value
Freeze_wfms	5	Freeze captured waveform, prohibit update its buffer
Sys_rst	6	System reset
HEP_capture_flag	7	Enable waveform capturing triggered by HEP beam
HEP_Adaptive_Sel	8	Adaptive mode ON for HEP beam
NTF_Adaptive_Sel	9	Adaptive mode ON for NTF beam
STU_Adaptive_Sel	10	Adaptive mode ON for STU beam
NTF_capture_flag	11	Enable waveform capturing triggered by NTF beam
STU_capture_flag	12	Enable waveform capturing triggered by STU beam
	13	-- not used --
	14	-- not used --
	15	-- not used --

Waveform Capture Trigger by Chosen Beam Type

The feature was added during LE Linac LLRF code update in May 2009. It allows to the operator by setting Acnet page binary switches called “HEP_Capture_Flag”, “STU_Capture_Flag” and “NTF_Capture_Flag” to form a waveform capturing event OR-mask from chosen beam types. For example if we set two switches “HEP_Capture_Flag” and “NTF_Capture_Flag” the waveforms will be taken for either HEP or NTF beams only.

IMPLEMENTATION

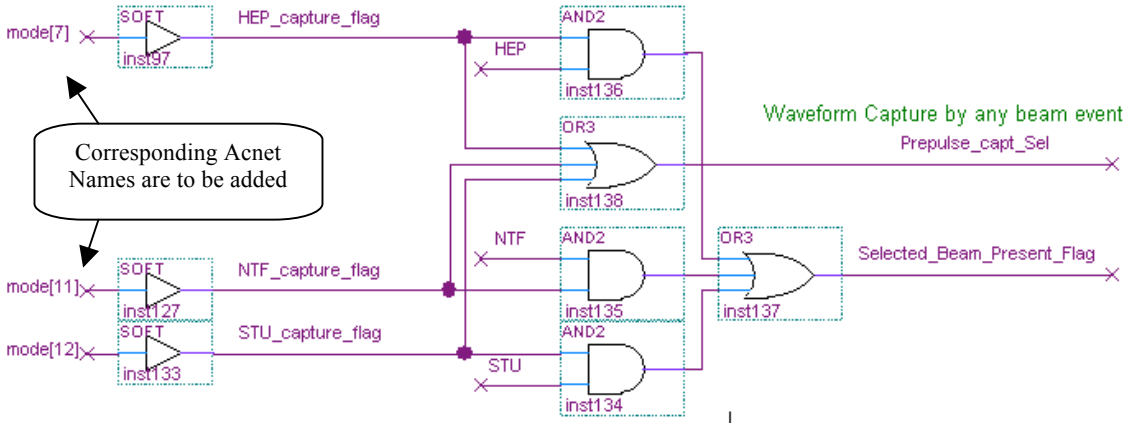


Fig. 2 Beam Type Event Masking

How the masking is done can be seen in Fig.2. Mode[7], mode[11] and mode[12] bits of binary switch register MODE are used in two networks. In the first one they are ORed together resulting in “Prepulse_capt_Sel” signal. This is basically the counterpart to the old firmware signal with the only difference that it is generated by three (not one) bits of MODE register. In further module called Start_Tim this “Prepulse_capt_Sel” signal multiplexes capturing trigger from LLRF_Start to Prepulse (0/1). In second network the same flag bits mask a real beam presence event during RF-cycle allowing HEP, NTF or STU beam pass through the masked AND gate. The result of three gates output is then ORed on output OR3-gate. The product called “Selected_Beam_Present_Flag” is then go to trigger event postprocessing after LLRF_StartTim module as it is shown in Fig.3.

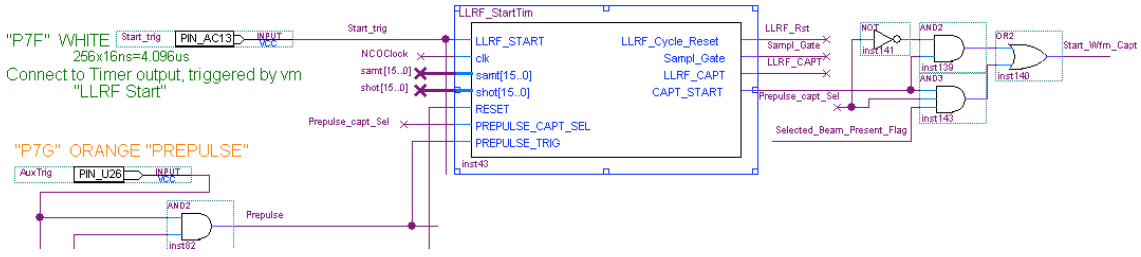


Fig. 3 Beam Type Event Processing during waveform capturing (DAQ).

Second Sampling Window during Beam Loading

The feature was added during LE Linac LLRF code update in May 2009. Placed along with the first beam loading sampling window but at the end of the beam pulse it allows to determine the gradient incline during the compensation. The incline is supposed to be used as a slow feedback input at adaptive mode to eliminate gradient tilt during compensation. The measured incline could be inverted, scaled and used as a setting for a tilt (the incline on top portion of the ff pulse).

IMPLEMENTATION

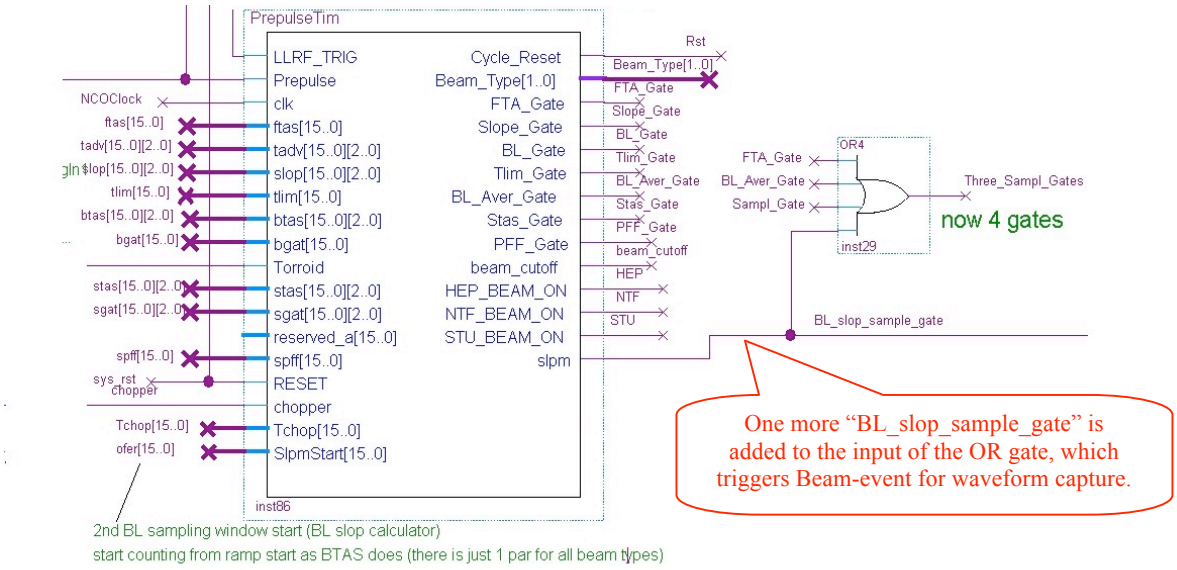


Fig. 4 PrepulseTim Module generates sampling windows during beam loading.

As PrepulseTim module parameter *btas[15..0]* (Acnet name *TBAS*) defines the beginning of gradient sampling window during beam loading, the new parameter ***ofer[15..0]*** (FFOF) does the same for second sampling window placed at the end of beam loading. The output signals from PrepulseTim module are always Prepulse-driven, which means that they are activated just at beam present RF-cycle only. In previous LLRF system there was just one sampling window *BL_aver_Gate*. In updated system this window is accompanied by *BL_slop_sample_gate.* The both signals go to AFF module where they are used as samples for gradient calculations. The sample width is defined by *bgat[15..0]* (Acnet name TBAW) for both samples. The difference of two samples leaves AFF module as *dBLslop[17..0]* bus and becomes available to Acnet as a scalar readback (early known as *fault_count[15..0]*). The feedback loop should be closed outside FPGA in either Slot0 or DSP.

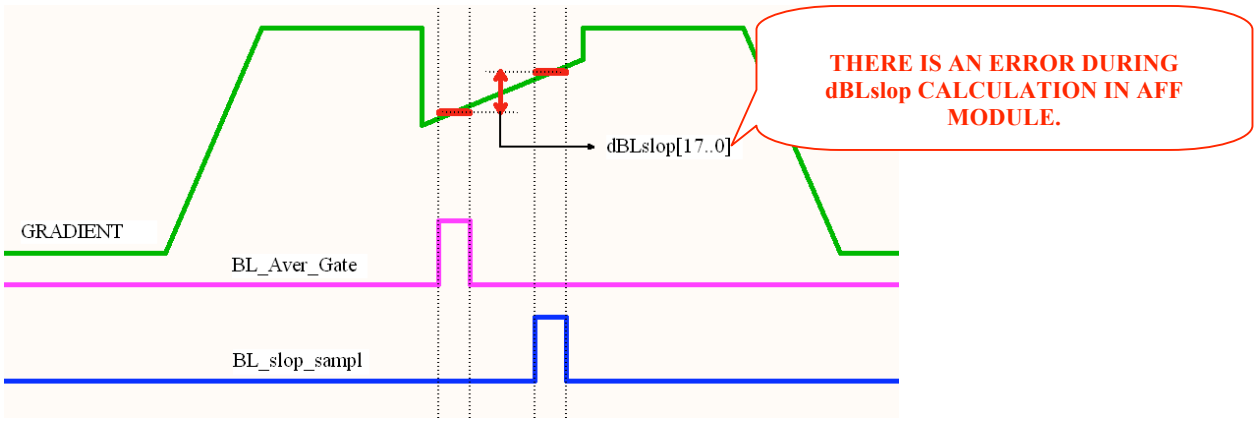


Fig. 5 Exaggerated diagram shows incline calculation.

Beam Feed-forward Pulse Fix

The following fix was applied to the adaptive mode algorithm. The update of FF table is occurred only in a case of torroid signal presence. Torroid is connected to the Adapt_FF_Latch input, where it is latched on flip-flop by front edge of *BL_aver_Gate*. The flip-flop output called *Upd_Enable* serves as a gate for module's output *aff_latch*, used for feed-forward table update at the end of RF-cycle.

Extended List of Parameters for Amplitude Feed-forward Compensation

The list was extended during LE Linac LLRF code update in May 2009. After new LLRF system on Low Energy Linac was commissioned it became evident that having separate set of feed-forward pulse controlling parameters for each beam type would be beneficial. To minimize changes to FPGA code it was decided to create just one extra index parameter, instead of triplication the set of nine-s shown in Table "Parameter Names selectable by BTIX index-parameter" below. The index could be thought as a register, similar to Intel's x86-series CPU index-register, allowing addressing to the memory pages. In our case we don't use the whole page space, so the index would rather resemble us a multiplexer control word. On firmware level it means that if DSP is instructed to make a change to one of beam-dependant parameter, it should update BTIX indexer parameter first and then program the parameter itself. The valid values for BTIX are:

0 – corresponds to HEP beam;
1 – corresponds to NTF beam;
2 – corresponds to STU beam;
3 – corresponds to NO beam (NOT USED);

In FPGA all beam-dependant parameters where combined in arrays of three elements, one element for each beam type. Feedforward pulse is generated in FSS1 module depending on the value of Beam_Type[1..0] word, which comes from Beam_Decoder module. Since the beam decoding is done approximately 1 ms in advance to compensation, there are a plenty of time for Beam_Type[1..0] bus to get stabilized.

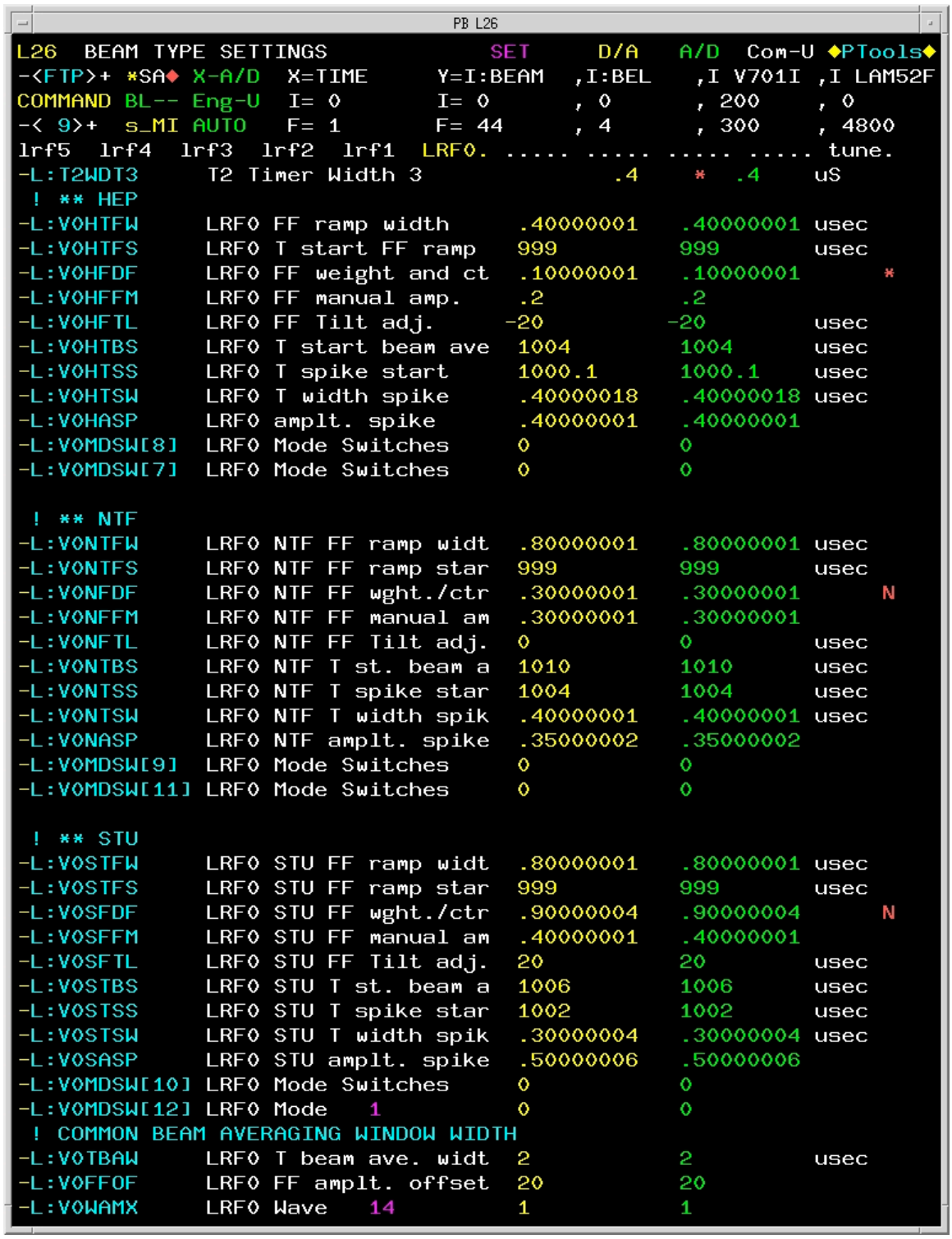


Fig. 6 The Acnet page featuring the combined extended list of parameters for separate tuning of feedforward pulses at different beam types.

TABLE Parameter Names selectable by BTIX index-parameter

Fpga name	Acnet name	Ser Bus Addr (Hex)	Description
Slop	**TFRW**	**0**	FF ramp width
Tadv	**THFR**	**8**	FF ramp start
Adpt	**FDFW**	**14**	FF learning coef
Wght	**FFMA**	**16**	FF manual ampl.
Tlop	**FTLT**	**46**	FF tilt
Btas	**TBAS**	**48**	BL aver start
Stas	**TSPS**	**4C**	FF spike start
Sgat	**TSPW**	**4E**	FF spike width
Smag	**ASP**	**50**	FF spike ampl.

NOTE: BTIX = 0 /1 /2 for HEP / NTF / SRU beam types respectively

Debug Port

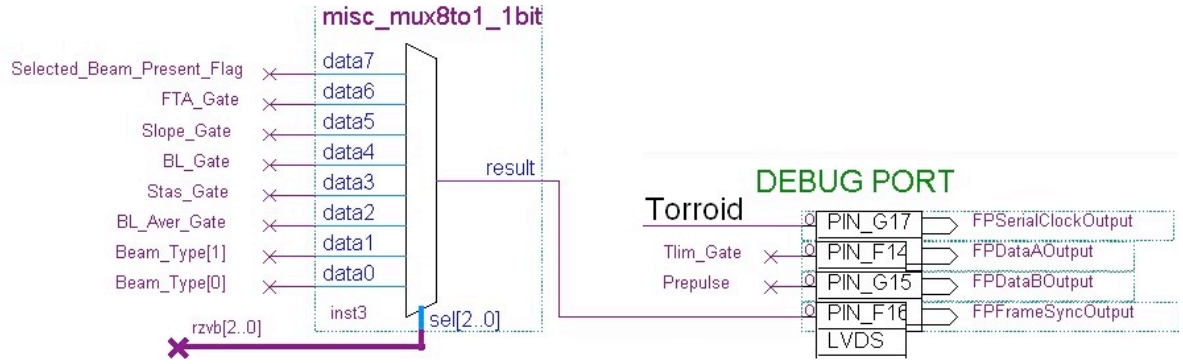


Fig. 7 Front panel serial Port connector with set of debug signals including the multiplexed ones.

TABLE FPGA Parameters

FPGA Name	Acnet Name	Serial Address	Description
Slop	**H**TFW, **N**TFW, **S**TFW	0	FF ramp width
Aadj	ARF	2	Output RF ampl.
Padj	QFOC	4	Fan-out phase
Gain	QGN	6	Phase loop gain
Tadv	**H**TFS, **N**TFS, **S**TFS	8	FF ramp start
Imin	QMNI	a	I_tank limiter
Shot	TSDA	c	Waveform capture start
Pfbc	QFB	e	Fan-back phase (startup script)
sync	QSYN	10	Phase sync
qmax	QMQI	12	Q/I clamper
Adpt	**H**FDF, **N**FDF, **S**FDF	14	FF learning coef
Wght	**H**FFM, **N**FFM, **S**FFM	16	FF manual ampl.
Scal	CTLG	18	Tuner control loop gain
muxN	WAMX	1a	Waveform multiplexer A
Ftas	TSFA	1c	Transposed T0
mode	MODE/MDSW	1e	Binary switches (see table below)
Samt	TSRD	20	Scalar sample window start
Tlim	**????**	22	Flat-top sample window start
Tank_scal	SGRD	24	Tank readback
Theta_scal	SQT	26	Theta readback
It_scal	STKI	28	I_tank readback
Phase_Drift_scal	SRPE	2a	--- Not used ---
Imod_scal	SMI	2c	I_mod readback
Qmod_scal	SMQ	2e	Q_mod readback
FF_out_scal	SAFF	30	FF amplitude (useless)
Fwd_Pwr_scal	SFWD	32	Forward Power (unitless)
Slop_er_scal	**????**	34	Slop error (former Fault counter)
Phase_Quadr_scal	SQUA	36	Phase quadrant (V)
Qt_scal	STKQ	38	Q_tank readback
Reserved_scal	--- Not used ---	3a	--- Not used ---
Reserved_scal	--- Not used ---	3c	--- Not used ---
Reserved_scal	--- Not used ---	3e	--- Not used ---
Reserved_scal	--- Not used ---	40	--- Not used ---
Reserved_scal	--- Not used ---	42	--- Not used ---
Delay	COFF	44	Tuner control phase offset
Tlop	**H**FTL, **N**FTL, **S**FTL	46	FF top tilt (in counts/sample)
Btas	**H**TBS, **N**TBS, **S**TBS	48	Beam loading sample window start
Bgat	**????**	4a	Beam loading sample window width
Stas	**H**TSS, **N**TSS, **S**TSS	4c	FF spike start
Sgat	**H**TSW, **N**TSW, **S**TSW	4e	FF spike width
Smag	**H**ASP, **N**ASP, **S**ASP	50	FF spike amplitude
Ofer	**FFOF**	52	2nd BL sample window start
Spff	TSQF	54	Phase FF start
Apff	AQFF	56	Phase FF amplitude
Btrq	None (internal)	58	Buffer transfer request
mxxN	WBMX	5a	Waveform multiplexer B
Tchop	TCHP	5c	Chopper shutoff time
btix	**None**	5e	Beam Type Index (dsp internal)
debx	**None**	60	Debug multiplexer

NOTE: tripled Acnet name means that the command from dsp to fpga is preceded by BTIX command (which = 0/1/2 for HEP/NTF/STU).

M_O_D_U_L_E "LLRF_StartTim"

The module generates pulses for DAQ and waveform triggering.

Inputs:

- LLRF_START – LLRF start trigger coming from external Timer. The portion of Prepulse trigger overlapped by LLRF_START serves as beam qualifier. The width of the trigger is equal to flat-top area of gradient.
- CLK – 65 MHz nco clock.
- SAMT[] – start time for scalar sampling window.
- SHOT[] – start time of waveform capture.
- RESET – system reset from (mode[6]).
- PREPULSE_CAPT_TRIG – waveform capture trigger select (LLRF/BEAM at 0/1).
- PREPULSE_TRIG – Prepulse trigger coming from external source. It is used during waveform capturing at beam present event (PREPULSE_CAPT_TRIG=1).

Outputs:

- LLRF_Cycle_Reset – one clock-wide pulse that starts 65.5 us away from front edge of LLRF_STRAT. It is used for initializing an external Shot_Control module only.
- Sampl_Gate – is used for scalar averaging.
- LLRF_CAPT – Restarts a state machine in the external Shot_Control module only.
- CAPT_START – initiates waveform capturing by either LLRF_START or Prepulse trigger depending on PREPULSE_CAPT_SEL value.

M_O_D_U_L_E "PrepulseTim"

The module generates time intervals and control signals for beam feed-forward pulse shaping. The output signals get generated by Prepulse trigger, i.e. when beam present only

Inputs:

- LLRF_TRIG – start trigger coming from external Timer. The portion of Prepulse trigger overlapped by LLRF_START serves as beam qualifier. The width of the trigger is equal to flat-top area of gradient.
- Prepulse – Prepulse trigger coming from external source. It is used during waveform capturing at beam present event (PREPULSE_CAPT_TRIG=1).
- CLK – 65 MHz nco clock.
- FTAS[] – defines both a moved new T0 point and start time of gradient flat-top averaging gate.
- TADV[] – defines offset from T0 to ramp start.
- SLOP[] – defines the width of ramp portion of BFF pulse.

- TLIM [] – defines the width of inhibit mask.
- BTAS[]– defines offset from ramp start to gradient flat-top averaging window.
- BGAT[] – defines the width of FTA, BLA and SLM gates.
- Torroid – beam present signal taken from torroid.
- STAS[] – ofset from ramp start to spike start pulse.
- SGAT[] – defines the width of spike pulse.
- SPFF[] – defines phase FF start time (from T0).
- RESET – system reset from (mode[6]).
- Chopper – decoded chopper portion of the Prepulse line.
- Tchop[] – defines a delay from chopper front edge to cutoff_beam front edge.
- SlmpStart[] – defines offset from ramp start to slop defining averaging window.

Outputs:

- Cycle_Reset – 3 clock-wide cycle starts with beam qualifier portion of Prepulse. It is used for AFF and FSS1 modules clearing.
- Beam_Type[] – HEP=0, NTF=1, STU=2, NONE=3.
- FTA_Gate – gradient flat top averaging gate, the front edge of which is used as new T0 point.
- Slope_Gate – ramp portion of the BFF gate.
- BL_Gate – BFF gate without ramp portion.
- Tlim_Gate – FF inhibit mask gate.
- BL_Aver_Gate – gradient beam loading averaging gate.
- Stas_Gate – spike portion gate of the BFF pulse.
- PFF_Gate – phase FF gate.
- Beam_cutoff – beam cutoff front.
- HEP – HEP beam.
- NTF – NTF beam.
- STU – STU beam.
- Slpm – Slop defining averaging gate.

M_O_D_U_L_E "Adapt_FF_Latch"

The module generates latch pulse when system is in adaptive mode. The pulse is used for updating learning coefficient during current rf-pulse, when beam really present in the cavity.

Inputs:

- Delayed_T0 –. defines the moved new T0 point.
- CLK – 65 MHz nco clock.
- Torroid – beam present signal taken from torroid.
- BL_Aver_Gate – gradient beam loading averaging gate.

Outputs:

- Aff_latch – two clock-wide latch 522-us delayed from Delayed_T0 signal. The signal is issued only when the beam is really had being present during the current rf-cycle. The condition is determined by Torroid signal.

M_O_D_U_L_E "AFF"

The module calculates the gradient difference between beam loading and flat-top conditions. The slop during beam loading is also calculated.

Inputs:

- A_tank[] –
- Clock – 65 MHz nco clock.
- FF_gate –
- Clear –
- FTA_Gate –
- LearnCoef[][] –
- Slpm –
- Beam_Type[] –

Outputs:

- A_err[] –
- dBLslop[] –

M_O_D_U_L_E "FSS1"

The module forms amplitude feed-forward pulse.

Inputs:

- Clock – 65 MHz nco clock.
- Clear – asynchronously clears ramp and tilt generating accumulators.
- Slope_gate – enables clock in ramp generating accumulator. . It is used in forming of output multiplexer control signal.
- Slop[] – number of steps necessary for ramp generation.
- A_err[] – the value defining FF amplitude change in adaptive mode.
- ADPT_SEL – adaptive mode selector generated by mode[8], mode[9] or mode[10] bits for HEP, NTF and STU beams respectively.
- Aman[] – FF pulse amplitude in manual mode or aiming parameter in adaptive mode.
- BL_gate – enables clock in tilt generating accumulator. It is used in forming of output multiplexer control signal.
- BeamType – beam type select.
- Tlop[] – tilt incline in counts per sample.

- Smag[] – spike amplitude.
- Stas_Gate – spike gate width in samples.
- AFF_LATCH – adaptive mode learning coefficient update pulse.

Outputs:

- AFF_OUT[] – Amplitude FF pulse output.

FSS1 FIX:

1) Spike magnitude coming to aer_mux in FSS1 module must be multiplexed by ADPT_SEL line. It has to be done to provide correct spike magnitude in manual and adaptive mode.

TOP SHEET FIX:

1) To allow the Adapt_FF_latch module issuing AFF_LATCH in manual mode at least once apply VCC to Torroid input. Otherwise BiPolSaturator' Latch in FSS1 will be low, preventing correct amplitude handling.

MFC BOARD CHECK LIST FOR LOW ENERGY LINAC

V.Tupikov 03/25/2009

On Acnet page D27 select two waveforms L:V#WDGA & L:V#WDGB;

1. Power-up conditions (ADC P10G)

Check RF phase lock by either front panel LED or ACNET parameter:

- LED is located at bottom left side of MFC front panel;
- L:V#SQUA value must be 3.0 +/- 0.2 V (this device represents digitized result of phase comparison between old and new 201 MHz reference lines generated in RF module). The value matches front panel BNC readback.

2. Gradient, Reference Line and Forward Power IFs (ADCs P10C,A,D):

TANK GRADIENT IF

- set waveform multiplexers L:V#WAMX = 5 & L:V#WBMX = 6 to verify tank in-phase and quadrature signals Q_t = 0 and I_t = +MAX (~0.9);

REFERENCE LINE IF

- set waveform multiplexers L:V#WAMX = 15 & L:V#WBMX = 15 to verify reference in-phase and quadrature signals I_r = -0.45 and Q_r=0;

FORWARD POWER IF

- set waveform multiplexers L:V#WAMX = 11 to verify forward power P_fwd;

3. Tuner Controller

- set waveform multiplexers L:V#WAMX = 12 to verify phase between forward power IF and tank IF;
- check actual signals (old and new systems) on the scopes;

4. Intertank Phase or Auxiliary Input (ADC P10G)

- set waveform multiplexers L:V#WAMX = 13 to verify any arbitrary signal (could be an intertank phase though) applied to P10G connector;

5. Tank Gradient AMPLITUDE

Tank gradient waveform (amplitude) is available two ways:

- by setting waveform multiplexers L:V#WAMX = 0 and monitor waveform on L:V#WDGA;
- by selecting directly gradient waveform L:V#WDGA without multiplexing;

6. LLRF and Beam Qualifier TRIGGERS

- set mode switch L:V#MDSW[7] = 0 to verify that waveform is updated with a rate up to 15 Hz. The waveform update indicates that LLRF_TRIGGER comes to the system.

- set mode switch L:V#MDSW[7] = 1 to verify that waveform is updated only when there is a beam in the cavity. You should see a beam loading dip on gradient waveform all time. The waveform update indicates that Beam Qualifier trigger comes to the system.

OTHER USEFUL BINARY SWITCHES

- L:V#MDSW[1] = 0/1 selects 200ns / 16 ns sampling rate. This means that we can capture waveform with better resolution. NOTE: at 16ns sampling rate the waveform points are separated o the plot by 64 ns (4 x 16 ns) time interval, where 4 represents number of tabs in used boxcar filter.

- L:V#MDSW[5] = 1 freezes waveform.

7. Amplitude FEEDFORWARD

SAMPLING INTERVALS

Set waveform multiplexer L:V#WAMX = 14 to visualize sampling markers on gradient waveform;

- use L:V#TBAW to adjust window widths for pre-beam average and beam loading time average (the 3d marker has hardwired width = 10 us);

- use L:V#TSFA to adjust pre-beam window start time;

- use L:V#TBAS to adjust beam window start time;

NOTE: the marker height can be used for beam type decoding:

- HEP beam has shortest height ~ 0.007;
- NTF beam has middle height ~ 0.015;
- STU beam has middle high height ~ 0.030;
- INTERLOCK beam has highest height ~ 0.060;

MANUAL MODE FF

Set waveform multiplexer L:V#WAMX = 1 to visualize feedforward pulse;

- use L:V#THFR to adjust ff ramp start time;
- set waveform multiplexer L:V#WAMX = 1 to verify amplitude feedforward signals;
- use L:V#TFRW to adjust ramp width;
- use L:V#TSPW to adjust spike width;
- use L:V#ASP to adjust spike amplitude;
- use L:V#FFMA to adjust manual amplitude;
- use L:V#FTLT to adjust tilt;
- use L:V#TCHP to adjust chopper shutoff time;

ADAPTIVE MODE FF

- use L:V#FDFW to adjust ff learning coefficient (=0.1);
- set L:V#MDSW[8]=1 to switch to adaptive mode;

To simplify observation, use L:V#MDSW[7]=1, which allows triggering by beam event.

8. Phase FEEDBACK

MANUAL MODE FF

THETA waveform (amplitude) is available two ways:

- by setting waveform multiplexers L:V#WAMX = 2 and monitor waveform on L:V#WDGA;
- by selecting directly THETA waveform L:V#WQ without multiplexing;

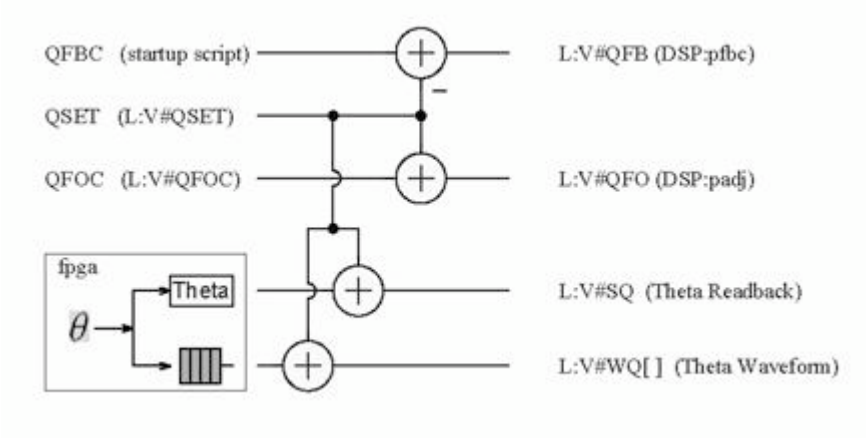


Fig.1 Calibrating cavity phase applying an offset.

- check that flat-top area is close to synch angle (~ -32.0deg). If it is not, - then change QFBC value in startup script to satisfy the requirement;

- set waveform multiplexer L:V#WAMX = 3 (KTHETA);

- set L:V#QGN=-1 to close phase loop. You have to observe a flipped THETA waveform;

- use L:V#TSQF to adjust phase ff-pulse start time;

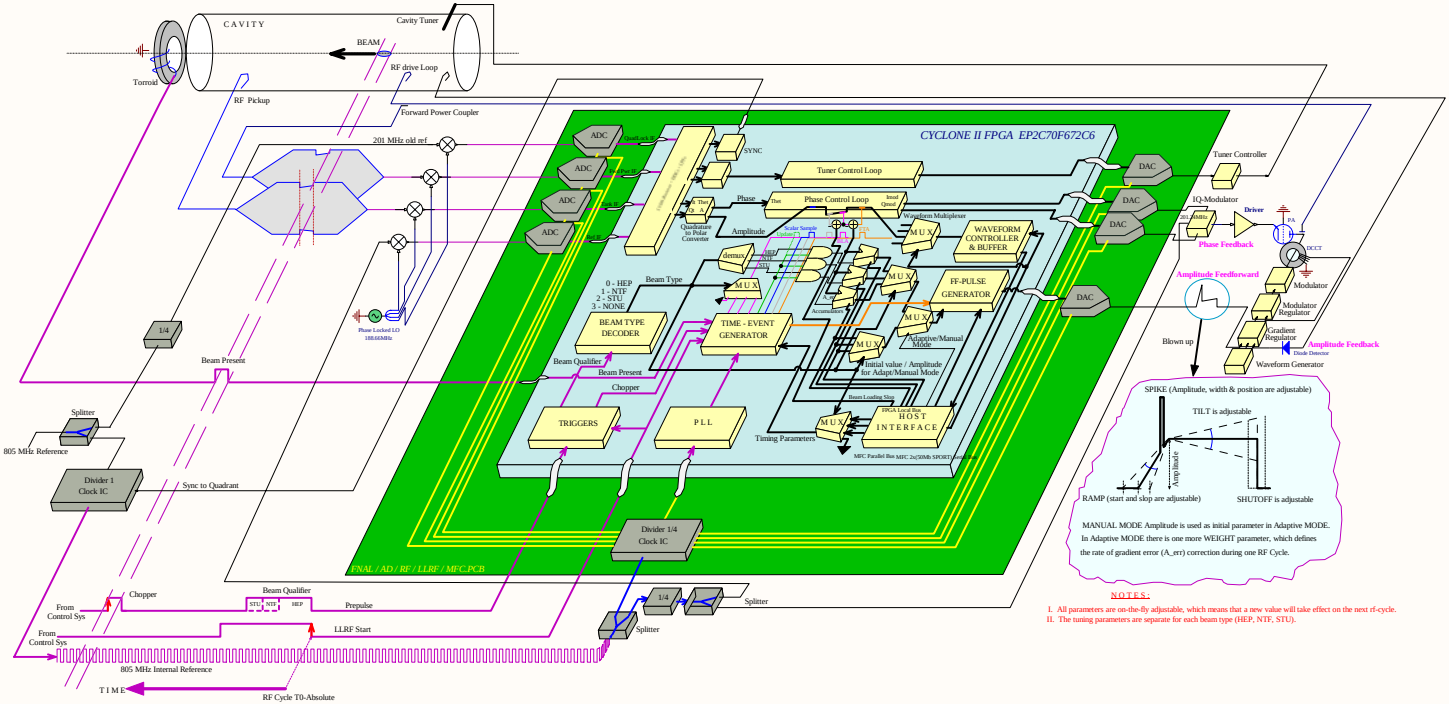
- use L:V#AQFF to adjust phase ff-pulse amplitude;

ADAPTIVE MODE FF

NOT IMPLEMENTED YET !!!!

MODULATOR LINEARITY CHECK

FERMILAB LOW ENERGY LINAC FPGA-BASED MULTIBEAM ADAPTIVE FEEDFORWARD DESIGN



NOTES:

I. All parameters are on-the-fly adjustable, which means that a new value will take effect on the next rf-cycle.

II. The tuning parameters are separate for each beam type (HEP, NTF, STU).

Linac Low Level Upgrade Controls/Front-end Interface

Paul W. Joireman, Vitali Tupikov, Ed Cullerton and Brian Chase

Contents

Tables

Figures

Overview

Each low energy LINAC station is tied to a single VXI crate which exists as a front-end node on the accelerator controls network. Each VXI crate contains a crate controller, often called the slot 0 controller, an analog RF module and a multi-channel field control (MFC) module. The phase loop and feed-forward calculations are implemented in firmware on the MFC board. The connection to ACNET is managed by on the slot 0 controller using a MOOC/AN software layer that passes data between the control system and the front-end software. ACNET requests to set or read front-end devices, which may occur at any time during the LINAC pulse, write or read data in CPU local memory on the crate controller. Once the beam pulse has traversed the LINAC station and a digitization cycle is completed, the MFC card signals the crate controller, at which point new settings are transferred to the MFC hardware and readings are retrieved. This prevents inadvertent change to the control algorithm during the beam pulse.

Devices

A set of ACNET devices exists for each low energy LINAC station. Settable devices are used to control the feed forward compensation algorithm implemented in the MFC board firmware. Each front-end contains a set of read-only ACNET devices which give averaged values of acquired waveform data for diagnostics. We adopt a standard naming convention for the ACNET devices of all LINAC nodes in order to promote training and system knowledge. All ACNET device names have the following form:

`L:VNxxxx`

Where `L:V` is a standard prefix (`L` = LINAC and `V`=VXI) and `N` represents the station number 1-5. The remaining characters `xxxx` identify the actual device, setting or read-only. Values for `xxxx` are given in Table 2 and Table 5 for the setting and read-only devices, respectively. We use the following conventions for the first character after the station number N:

Letter	Description
Q	Phase devices
T	Timing devices
S	Scalar devices (read-only)
C	Cavity tuning devices
W	Waveform Devices

Table 1: Conventions used first letter after station number in devices names

Setting devices

The settable devices are shown in Tables 2 through 5. The devices may be loosely grouped into devices which control the phase loop, devices which control the timing of the feed-forward correction pulse, feed-forward amplitude loop control and some miscellaneous control devices.

Mnemonic	Range	Default	Units	Description
QSET	0-360		degrees	Phase setting
QFOC	0-360		degrees	Phase fan out calibration
QSYN (QSYNC)	0-1	0	[binary]	Phase quadrant synchronization trigger
QGN (GAIN)	-18-18	0		Phase loop gain
TSDA (SHOT)	500-1500	800	μsec	Start time for data acquisition
ARF (AADJ)	0-1.0	1.0		RF level to drive amplifier
CTLG (SCAL)	0-1.0	1.0		Cavity Tuner loop gain
QMNI (IMIN)	0-1.0	0.1		Phase loop minimum I
QMQI (QMAX)	0-2.0	0.3		Phase loop maximum Q over I
WDGB (MUXN)	0-15	7		Diagnostic waveform B multiplexor
MODE	0-511	0		Bit vector for control (see Table 4)
TSRD (SAMT)	800-1500	1070	μsec	Sampling time for scalar read-back devices
TSFA (FTAS)	900-1000	970	μsec	Time start for Flat-top averaging
CDLY (PFWD)	0-100.0	2.0	μsec	Cavity tuner delay adjustment
FTLT (TLOP)	±32767	0		Feed-forward tilt adjustment
TBAW (BGAT)	0-20.0	10.0	μsec	Width of beam averaging window/gate
ASP (SMAG)	0-1.0	0.0		Amplitude of "spike"
FFOF (OFER)	0.0-50.0	0	μsec	FF slope sample time
TSQF (SPFF)	0-100.0	30.0	μsec	Start time of phase feed-forward
AQFF (APFF)	-10.0-10.0	0.0		Amplitude of phase feed-forward loop
WDGA (WVMX)	0-15	0		Diagnostic waveform A multiplexor
TCHP	0-100	1	μsec	Chopper shutoff time

Table 2: Setting device names, ranges and a brief description for each device

A subset of devices for each front-end system, RF station, has been defined uniquely for each beam type used in the LINAC, HEP, NTF and Study. A list of these devices is shown in

HEP Beam	NTF Beam	Study Beam	Range	Default	Description
HTFW	NTFW	STFW	0-4	1	Time feed-forward ramp width, in μsec
HTFS	NTFS	STFS	0-100	30	Feed-forward slope start time, in μsec
HFDF	NFDF	SFDF	0-1.0	0.1	Adaptive feed-forward weight and control
HFFM	NFFM	SFFM	0-1.0	0	Feed forward manual amplitude
HFTL	NFTL	SFTL	±32767	0	Feed-forward tilt adjustment
HTBS	NTBS	STBS	0-15.0	5.0	Time feed-forward ramp width, in μsec
HTSS	NTSS	STSS	1.0-4.0	1.0	Start time of "spike" , in μsec
HTSW	NTSW	STSW	0-1.0	0.25	Width of "spike" window/gate, in μsec
HASP	NASP	SASP	0-1.0	0.0	Amplitude of "spike"

Table 3: Setting device names for beam specific devices.

Phase Devices

The MFC board accepts inputs from the analog RF board containing a reference RF signal and tank RF signal at an IF frequency of 12.5 MHz. Each RF signal is passed through a I/Q demodulator, the resulting I and Q signals being processed under the influence of the settable devices. Six parameters are used to control a phase feedback loop within the firmware, QFO, ARF, QFB, QMNI, QMQI and QGN. The value of QFB is a calibration adjustment for the station specific fanback phase. QFB is added to the reference IF phase prior to I/Q demodulation of the IF Tank signal. After I/Q demodulation of the tank signal, the amplitude ($\mathtt{sqrt(I^2+Q^2)}$) and phase ($\mathtt{tan^{-1}(Q/I)}$) of the tank signal are calculated. The QMNI setting is used to clamp the I value prior to Q/I division to avoid NaN results and QMQI clamps the output Q/I prior to taking the arc tangent. The phase is then multiplied by the QGN setting and converted to an I/Q signal. This signal is then phase shifted using the values of QFO in degrees along with ARF, an amplitude adjustment. The resulting I/Q output of the MFC board are sent to the RF module for up-conversion to 201 MHz. The I/Q modulated RF is then send to the HLRF power amplifier driver. The QSYN parameter is used to re-synchronize the phase to the appropriate quadrant relative to the 805 MHz reference.

Timing Devices

Six devices are used to specify timing values used by the algorithm, *TFW, TSFA, *TFS, TSDA, TFFI and TSRD; where * stands for H, N, or S depending on the beam time and all are specified in units of microseconds. A timing diagram showing these devices relative to a nominal cavity RF waveform is given in Figure 1. Three devices are defined relative to the global LLRF start trigger. TSDA specifies the "snapshot" start time for digitization of all acquired waveforms; each waveform contains 4096 data points sampled at ~61.9 MHz for a total sampling time interval of ~66 μsec. TSFA, an acronym for Flat Top Averaging Start, specifies start time for an averaging gate. Within this 10 msec gate, the RF signal is accumulated and averaged for later use. TSRD, sampling time, specifies the start time for a 160 ns gate in which the read-only devices are calculated by averaging the acquired samples.

The remaining timing parameters are not defined with respect to the LLRF start trigger. The *TFS device specifies the start of the beam feed-forward slope relative to the TSFA setting. The beam feed-forward pulse is linearly increased to its maximum value prior to the beam loading time in order to minimize sparking in the modulator. The *TFW, device specifies the ramp time for the start of the beam feed-forward pulse. TFFI, time limit, specifies a time, relative to TADV. This time is used to disable the beam feed-forward if no beam is observed. (It seems that TFFI is no longer necessary if we have a beam qualifier, unless there is to be a case where no beam is observed when the beam qualifier is active.)

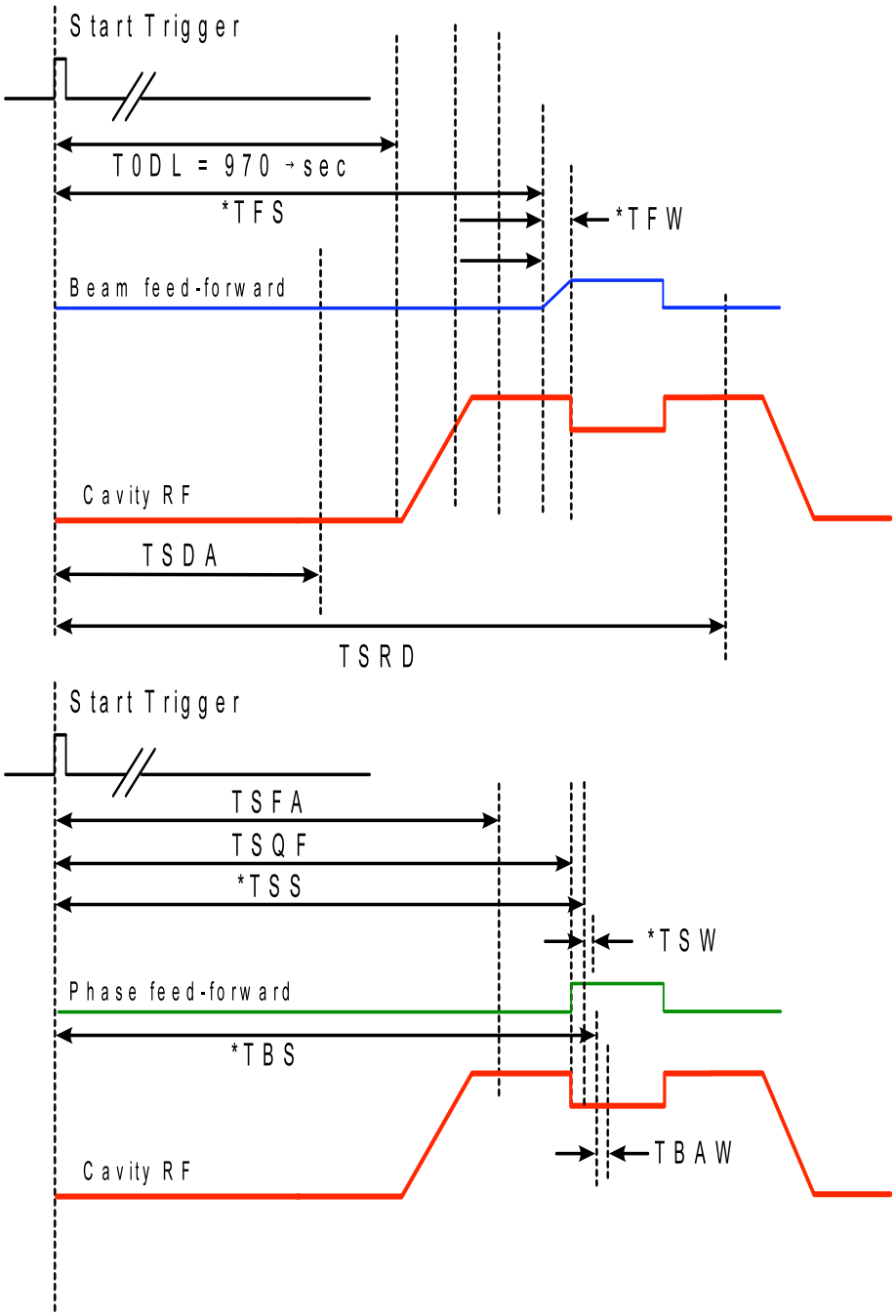


Figure 1: Timing diagram for settable devices, where * stands for H, N, or S depending on the beam time

A more detailed timing diagram is also available from the LLRF web-site.

Feed-Forward Amplitude Control Devices

The devices for feed-forward amplitude control are *FMA and *FDF, where * stands for H, N, or S depending on the beam time. The *FDF setting is related to the adaptive feed forward correction. *FDF responds to SETDAT requests from Acnet and basic control requests. A

SETDAT request or value setting, sets the value of *FDF and this is used in the code as the value of the adaptive feed forward coefficient. Basic control settings to *FDF enable or disable the adaptive feed-forward algorithm. Basic status on an Acnet parameter page displays . if the adaptive feed-forward is enabled and * if adaptive feed-forward is disabled.

If the adaptive feed-forward algorithm is disabled, feed-forward amplitude can still be applied to the system by changing the value of *FMA. This mode is called *manual mode* and the device *FMA stands for feed-forward manual amplitude.

Adaptive feed-forward enable/disable functionality is implemented at a lower level as an additional bit in the MODE device bit vector, bit 8. The operation of the MDSW and MODE devices is described below.

The amplitude error is calculated as the difference between the no-beam RF flat-top amplitude and the beam-loaded flat-top amplitude. The error signal multiplied by the c = *FDF or *FMA setting, is then added to the feed forward correction signal using the equation:

$$\text{output}(n) = \text{output}(n-1) + c * \text{error}(n-1)$$

where n is the sample number. The output is then passes through a saturation block and is used to compute a beam feed-forward signal which is added to the gradient waveform by the external gradient regulator hardware.

Miscellaneous Setting Devices

Additional devices used by the front-end are CTLG, CTOF, WAMX, WBMX. The CTLG device is a multiplicative scaling factor applied to the RF forward power to tank phase signal. The WAMX and WBMX settings are used to select one of 16 possible waveforms to. The possible waveforms that can be routed to either test point are given in Table 7.

MODE/MDSW Setting Devices

The MODE device represents a bit vector of 9 bits which functionally act as DIP switches to set control values in the FPGA. This device exists as part of the original set of devices for downloading to the FPGA. Rather than remove this device from the original list, which would require significant modifications to DSP and Slot 0 code, I chose to implement a secondary array device MDSW each element of the array device representing 1 bit in the mode vector. The bit position definitions can be seen in Table 4.

Bit Position	Description
0	Waveform output: (0) buffer, (1) scope
1	SAMPLE rate: (0) 200 ns , (1) 16 ns
2	NTF beam switch: (0) unblocked, (1) blocked
3	FAULT COUNTER RESET: (0) disabled, (1) enabled
4	Zero phase drift: (0) disabled, (1) enabled
5	Freeze waveform capture: (0) disabled, (1) enabled
6	System reset : (0) disabled, (1) enabled
7	HEP Waveform capture trigger: (0) LLRF_START, (1)

	PREPULSE_START
8	HEP Feed-forward learning enable (1) disable (0)
9	NTF Feed-forward learning enable (1) disable (0)
10	Study Feed-forward learning enable (1) disable (0)
11	NTF Waveform capture trigger: (0) LLRF_START, (1)
12	Study Waveform capture trigger: (0) LLRF_START, (1)

Table 4: Bit position descriptions for the MODE register, set by the MDSW array device.

When one of the array elements of MDSW is set the callback function `LelMODEAnCllbk()` is executed. The function sets or clears the appropriate bit in the MODE value. The callback function uses the current value of the MODE device along with the selected bit from MDSW in order to maintain the state of the MODE register. In other words, only the selected bit is touched an all others are left in their previous state before the setting was received.

Read-only devices

The scalar read-only devices are shown in Table 5. These devices represent average values calculated from the samples acquired during a 160 ns gate or window starting at `TSRD` μsec after the start trigger. The average tank phase `()SQT` is calculated on the front-end using an atan2() function and using the I and Q arguments from the `()STKI` and `()STKQ` values.

Mnemonic	Range	Units	Description
QFBC	0-360	degrees	Fan back phase correction
QFO (PADJ)	±180	degrees	Fan out phase adjustment
QFB (PFBC)	±180	degrees	Phase fanback adjustment
T0DL	900-1000	μsec	Phase fanback adjustment
SGRD	0-1		Average tank amplitude
SQ	±180	degrees	Average applied phase angle
STKI	0-1		Average tank I
STKQ	0-1		Average tank Q
SRPE	±180	degrees	Average phase drift
SMI	0-1		Average I modulator value
SMQ	0-1		Average Q modulator value
SAFF	0-1		Average beam feed-forward amplitude
SFWD	0-1		Average forward power
SFLT			Fault counter
SQUA	0-1		Average phase quadrant
SQT	±180	degrees	Average tank phase

Table 5: Read-only device names, ranges and brief descriptions

Expert Settings

The low energy linac front-ends have several "expert" settings that can only be changed through a command line interface on the front-end. These settings represent station specific data that we do not want to expose as ACNet devices in order to prevent random settings to their values. The values of the expert settings are typically determined empirically or by a policy for each station and set from the startup script. A separate document describing the procedure for changing these values is available.

Waveforms

The firmware acquires a common set of 15 waveforms which represent acquired or processed signals within the firmware and are updated every 15 Hz pulse. Any of these waveforms may be routed to the front-end through diagnostic waveforms. At present the front-end uploads a subset of four waveforms, RF amplitude, phase, and two diagnostic waveforms A and B. ACNET array devices have been created for all waveforms and the device names are given in Table 6.

Waveforms may be viewed using either D27, an array plotting application in ACNET, or a LabVIEW client program (LELPP_Wave) connecting to a BackDoor server on the front-end. The LabVIEW program is displaying the data on a BackDoor event that is announced each time the 15 Hz task runs so the data display is synchronized to the front-end hardware timing. The LINAC Wave viewer is at `Y:\Projects\LLRF\Software\LabVIEW\VIs\LINAC`. In order it you must install a LabVIEW runtime engine for version 8.5 on your machine. To modify the code you must have a LabVIEW development environment. The source code is at `Y:\Projects\LLRF\Software\LabVIEW\Source Code\LINAC`.

The setting of the WAMX and WBMX devices control which waveforms are sent to the diagnostic waveforms WDGA and WDGB respectively. Table 7 provides a mapping between the WAMX or WBMX setting and the waveforms routed to the test point waveforms.

Device	Description
WGRD	RF amplitude/gradient
WQ	Phase error
WDGA	Diagnostic waveform A
WDGB	Diagnostic waveform B

Table 6: Waveform device names and descriptions

Index	Waveform
0	Amplitude of Tank
1	Amplitude Feed-Forward
2	Tank phase (Theta)
3	Phase Controller Output (KTheta)
4	Phase Feed-Forward

5	Q_tank
6	I_tank
7	I_mod
8	Q_mod
9	I_clamp
10	Q/I_clamp
11	Forward Power
12	Forward Power to Tank Phase
13	Intertank Phase
14	Gradient with sampling markers
15	I_ref (WDGA) or Q_ref (WDGB)

Table 7: WDGA or WDGB setting map to test point waveform output.

Front-end software

Overview

Software running on the front-end provides the interface between the control system and the LLRF hardware which is responsible for acquiring RF signals from the tank and transforming them into RF drive signals to send to the external HLRF amplification system. The software is composed of a number of separate software libraries. The focus of this document will be on the "front-end" software library developed by the LLRF group. The front-end library, hereafter front-end software, maintains the ACNET devices described in the previous section and is responsible for forwarding new setting values to the LLRF hardware to control the LLRF phase loop and feed-forward amplitude loop. In normal operation, the front-end software waits for a backplane trigger to begin its work; the trigger occurs at a nominal rate of 15 Hz, the LINAC repetition rate. Once the trigger is received, the front-end software writes any settings that have changed since the previous trigger to the hardware. It then reads both reading values and waveform data from the hardware and stores these in local CPU memory. Once this work is complete, the front-end software waits for the next trigger.

Implementation Details

The front-end software is written in C/C++ and compiled to run on a PowerPC based single board computer architecture under the VxWorks RTOS. The source code is stored on nova and managed by CVS version control system.

Class structure

The front-end software consists of a series of interacting classes shown in Figure 2. The main class is the Station class which is a container class for all other classes within the system. The Station class is implemented as a Singleton pattern, one Station per front-end. When the front-end reboots, the Station class constructor is executed. The station class constructor constructs all its member objects. The Station class contains member objects that manage the

connection to ACnet using the AN library, these are constructed and attached to ACnet using `MoocNew()`. After the Station class is constructed an AutoDownload of the last devices settings is requested and once completed, the 15 Hz task tied to the FPGA data acquisition is started on the front-end and the last settings received from AutoDowload are then written to the FPGA. A trigger handler is then attached to the line which indicates that data acquisition is complete and trigger sensing on the controller is enabled. After hooking functions to a reboot request, the system is marked as ready by pulling the VXI trigger line 2 low to activate it. This is equivalent to an RF drive enable signal used in the Main Injector.

The status of the reboot can be monitored by watching the device `()V5STAT`. This devices takes on different values during the reboot process to indicate the progress of the reboot sequence. The values of `()V5STAT` and their descriptions are given in **Error! Reference source not found.**.

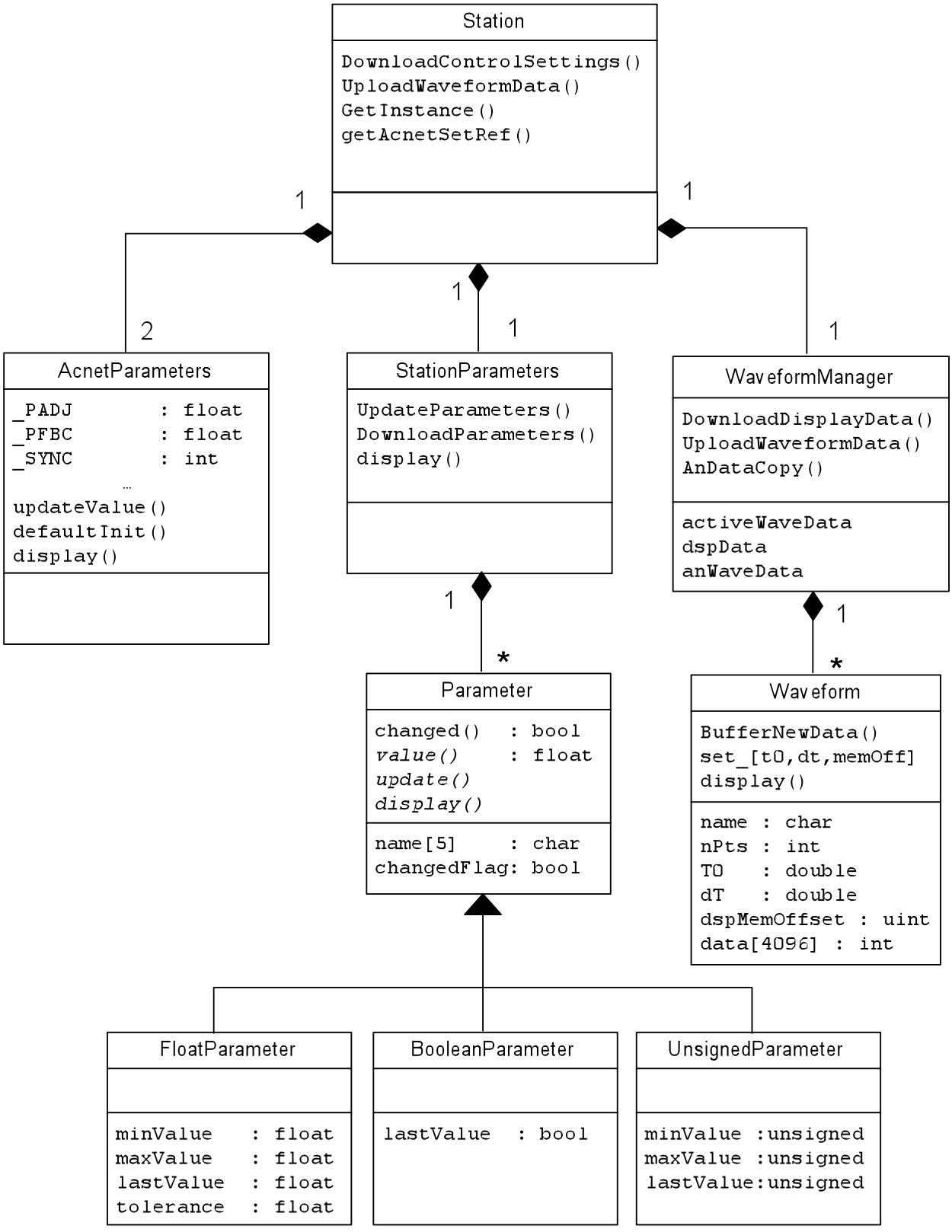


Figure 2: Front-end software class structure

The Station class constructor in turn invokes the constructors for all other classes shown in Figure 2 and places the station in a safe default state. Default values shown in Table 2 are used to update the values of the member data of the AcnetParameters setting object. Local values of the device values are stored by the StationParameters class in an array of Parameter objects. The Parameter class is an abstract base class providing a common interface for the concrete classes, FloatParameter, BooleanParameter and UnsignedParameter. The behavior of each concrete class depends upon the underlying parameter it represents. The Station container class also contains a WaveformManager object which maintains an array of Waveform objects, one representing each waveform that is acquired by the MFC board.

The front-end software also contains a number of "free" functions not tied to any particular class which provide member functions for displaying information about the front-end, either data `LinacDump()` or interface `LinacHelp()`, and other auxiliary functions, for example `LinacMain()`, the main entry point for the front-end, similar to a C `main()` function, as well as `LinacTask()` which is started as a background task in VxWorks and is used to perform the 15 Hz work, tied to the LINAC cycle, on the front-end.

Control flow

From a reboot the front-end software first loads the VxWorks OS. Commands contained in a startup script are then executed to load all the software libraries used in the low energy LINAC application. These libraries fall into three classes, ACNET support libraries, software support libraries, and front-end software. I will focus on the front-end software control flow, the front-end software is the primary "application" on the front-end and it uses the services of the ACNET and other support libraries to accomplish its work.

Once the libraries are loaded, the VXI resource manager software runs to determine the modules installed in the VXI crate. Some error logging objects are created initially for later use by the front-end software and then ACNET is started using `MoocNew()`. After ACNET is running the startup script issues a call to `LinacMain()`, which issues a call to the Station constructor. Once the station object is constructed along with all its member classes, a MOOC AutoDownload of the last settings for station parameters is requested. The list of parameters AutoDownloaded is set using the D121 PA. After auto-download is complete, the integrity of the communication with the MFC board is verified and the auto-downloaded parameters are sent to the board. A background task for parameter download to the MFC is started at a low priority of 135 and the `LelDaqDoneTrigHandler()` handler is attached to backplane trigger line 2. This trigger line will be pulsed by the MFC hardware after every data acquisition cycle has completed. The trigger handler executes at interrupt context and gives the semaphore allowing the parameter download task to begin its execution. `LinacMain()` then waits for the first trigger to be received and then pulls backplane trigger line 3 low to indicate the station is ready for operation. During the operation of `LinacMain()` the system status `()STAT` is upgraded from 0 to 4 to indicate the passage of specific steps in starting the station. The enumerated meanings for the `()STAT` value are shown in Table 8.

()STAT System Status Value	Meaning
0	Station has been rebooted and front-end code has yet to run
1	The Station object constructor has run without error
2	The requested MOOC Auto-Download operation has completed
3	The data acquisition complete trigger handler has been attached
4	Triggers are being received by the slot 0 controller

Table 8: System Status values and descriptions

At the end of `LinacMain()`, a heartbeat task is started that checks that the number of triggers received, incremented in `LelDaqDoneTrigHandler()`, is actually changing. If this task detects that the number of received triggers is not changing the system status `()STAT` is downgraded to level 3.

During normal operation, the parameter download task is idle waiting for the semaphore to be given, once this occurs in response to the data acquisition complete trigger the following actions are taken. The `update()` member function is invoked for each element the StationParameters object Parameter arrays with an input value obtained from the AcnetParameters setting object. The `update()` member function compares the value received to the stored value, if the value is new this parameter is marked as changed. All changed parameters are then written to the MFC hardware. Only changed parameters are written in order to minimize accesses to the hardware. Following the write of all changed parameters, the FPGA memory is read to obtain the values for all the read-only devices.

After writing the new control settings to the MFC board, the scalar read-only variables are read from the MFC board, scaled to floating point values and placed in memory accessible to the database devices defined for these variables.

Communication with MFC board

Writing settings

Settings are received from ACNET through setting a database device owned by the front-end and routed to the callback function `LELSetParamCllbk()`. The callback function extracts the setting value and forwards the setting and a device index to the `AcnetParameters::updateValue()` member function. This function checks the incoming setting against the ranges defined in Table 2. If the value is out of range an error is reported and the setting is not forwarded. As a result, the ACNET reading for this device will be inconsistent with the setting value. If the value is within an acceptable range, the setting value is used to update the value stored by the `AcnetParameters` class.

On the next 15 Hz cycle, control settings are written to the MFC board from the `Station::DownloadControlSettings()` function routed through the `StationParameters::DonwloadParameters()`. At the lowest level the generic MFC library function `MfcWriteLelParam()` writes floating point values to the SHARC DSP.

`MfcWriteLelParam()` takes three parameters, a reference to the MFC software, an index for the variable and a floating point value to write to that variable. This function then writes the value into a buffer on the DSP and marks the parameter as changed in the DSP code. When all settings have been written an interrupt is sourced on the DSP which scales all changed parameters to fixed point values and writes these values to the FPGA using the DSP SPORT.

Accessing readbacks

Values for the read-only devices are read from the DSP FTP variable memory using the MFC library function `MfcReadFParam()`. The function takes a reference to the MFC software object as its first parameter, an index to the FTP data variable and a pointer to a variable to return the data. The returned data value is scaled to a floating point value by the DSP prior to being read by the slot 0 controller.

Acquiring waveforms

Waveform acquisition is initiated from the LinacTask() 15 Hz work task by calling the `Station::UploadWaveformData()` member function which determines the start time for the acquired waveforms by reading the value of the TSDA device and retrieving the acquired waveform sample rate with `StationParameters::tSample()`. Next the `UploadWaveformData()` member function of the `WaveformManager` class is called and this uploads waveform data for only those four waveforms marked as active. The `WaveformManager` member function loops through all the waveforms, selects the active waveforms, sets the sample rate, start time and DSP memory offset for the waveform and then initiates an upload of the waveform data by invoking the `Waveform::BufferNewData()` member function. This method simply forwards its calls to the `MfcReadPackedDSPWave()` function from the MFC shared library. This method takes a mfc software object, DSP memory offset, number of points to read and a pointer to a data array and extracts data packed as two 16 bit integers into a 32 bit address.

Supporting Libraries

LLRF

The low energy LINAC LLRF front-end depends upon a number of supporting libraries that are developed and maintained by the AD-LLRF group at Fermilab. These libraries, their order of loading and a brief description of their purpose are shown in Table 9.

Library	Purpose
ksvxi	Kinetic Systems VXI API library
ksresman	Kinetic Systems VXI Resource Manager implementation library
misc	Provides timing delay functions MiscWaitMs() and MiscWaitUs()
drror	Provides error reporting vad_err macro
addrbook	Provides global address book functionality
log	Provides error logging facilities, LogMsg() function

an++	Provides ACNET support at a higher level than MOOC
Anmemarr	Provides support for ACNET reading of front-end arrays
mfc	Software interface to MFC board

Table 9: LLRF shared libraries used by the LINAC project

VxWorks

The nodes currently use vxWorks version 5.5 with a kernel built for the MVME5500 by Steve Foulkes of the Computing Division. This kernel resides at:

/fecode-bd/vxworks_boot/fe/rfies/lib/VW_55/MVME5500/kernel/vxWorks-vxi

and it is loaded at boot time prior to the execution of the startup script. Each node in the low energy linac exists as a node on the network with the name, IP address and ACNET trunk and node assignments given in Table 10.

Node	Acnet Name	IP	Trunk and Node
node0740	lip740	131.225.131.142	0x09BE
node0741	lip741	131.225.131.143	0x09C1
node0742	lip742	131.225.131.144	0x09C2
node0743	lip743	131.225.131.145	0x09C3
node0744	lip744	131.225.131.146	0x09C4
node0745	lip745	131.225.131.147	0x09C5

Table 10:Table of node names, IP addresses, trunk and node names.

MULTICHANNEL VECTOR FIELD CONTROL MODULE FOR LLRF CONTROL OF SUPERCONDUCTING CAVITIES*

P.Varghese#, B. Chase, B. Barnes, J. Branlard, P.W. Joireman, D. Klepec, U. Mavric, V. Tupikov
Fermilab, Batavia, IL 60510, U.S.A.

Abstract

The field control of multiple superconducting RF cavities with a single Klystron, such as the proposed RF scheme for the ILC, requires high density (number of RF channels) signal processing hardware so that vector control may be implemented with minimum group delay. The MFC (Multichannel Field Control) module is a 33-channel, FPGA based down-conversion and signal processing board in a single VXI slot, with 4 channels of high speed DAC outputs. A 32-bit, 400MHz floating point DSP provides additional computational and control capability for calibration and implementation of more complex control algorithms. Multiple high speed serial transceivers on the front panel and the backplane bus allow a flexible architecture for inter-module real time data exchanges. An interface CPLD supports the VXI bus protocol for communication to a Slot0 CPU, with Ethernet connections for remote in system programming of the FPGA and DSP as well as data acquisition.

Introduction

The MFC board is an FPGA based 33 channel down-conversion and signal processing module designed for vector control of multiple cavities with a single Klystron such as the proposed RF scheme for the ILC[1]. An overview of the main components of the board is shown in Fig. 1. There are 4 DAC channels for RF outputs and multiple high speed serial transceivers on the front panel and the backplane bus to allow a flexible architecture for inter-module real time data exchanges. A floating point DSP provides additional computational capability for calibration and implementation of more complex control algorithms. Both the FPGA and DSP have external SDRAM memory for waveform and diagnostic data storage. Nonvolatile Flash memory is used for DSP program and FPGA configuration storage.

The interface CPLD supports the VXI bus protocol for communication to a Slot 0 CPU, with Ethernet connections for the control system interface, remote in system programming of the FPGA and DSP as well as for data acquisition and diagnostics.

Signal I/O and Clock Distribution

Thirty channels of IF inputs are transformer coupled to 4 , 8-ch, 12-bit, 65 MHz ADCs with a voltage gain of 2, through an impedance matching filter network[2]. Two channels are DC coupled through differential ADC drivers with fixed gain. The 33'rd RF input is connected to a 14-bit, 105MHz parallel ADC through a transformer coupled impedance matching network. This channel can be used in a fast Klystron feedback loop to improve the cavity field vector control performance. An LO input of up to 1.6 GHz can be divided down to provide 8 clock signals through a clock distribution chip. The 4 8-channel ADC's and the clock distribution chip have digital calibration ports that are connected to the DSP's SPI bus which can be used for adjusting clock ratios, clock skew, generating test patterns and for powering down the ADC's between pulses. The digitized data from each ADC is transferred to the FPGA on 8 serial LVDS lines clocked at 6x times the sample rate with double data transfers in each clock period. Two dual channel 14-bit, 260 MHz DACs provide a total of 4 RF output channels. An optional auxillary clock input can be used to drive 2 DAC channels and two external trigger inputs are available to synchronize processing on the board. There are 8 additional backplane triggers that can be used for synchronizing clocks and other processes between boards.

Signal Processing FPGA

The primary signal processing functions of this board are performed in a Cyclone II FPGA (EP2C70F672C8). The device has 300 9-bit multipliers, 1.5 Mbits of Ram, 4 PLL's , 672 pins with 472 user I/O and 68000 LE with a max clock speed of over 400 MHz. Fig.2 shows the signal processing functions implemented in the FPGA.

Serial Data from the 32 RF channels is converted to 12 bit parallel in the Serial-Parallel latch. Downconversion is performed by multiplying the data with an 18 bit scaled and offset Cosine/Sine table to provide a composite gain plus rotation. The tables are 256 deep and they can be written to by the DSP or the slot0 CPU through the bus interface. The 24 I,Q pairs corresponding to the cavity fields are summed for vector processing. A pair of CIC filters completes the signal processing before the feedback error is computed. A gain and klystron linearizer multiplier table provides loop gain. A feedforward input is added and the output from the fast klystron loop is summed in before upconversion to the IF frequency. The I and Q signals are outputted to the external modulator through a dual channel 14-bit DAC. Reference signals for the beam and cavity phase (one for each cryomodule with 8 cavities each) are processed in 4 of the auxiliary channels. Phase computation is done in the DSP and the corresponding I and Q setpoint tables are updated. Extensive diagnostics are available at various points along the signal chain.

*Work supported by Fermi Research Alliance LLC. Under DE-AC02-07CH11359 with the U.S. DOE
#varghese@fnal.gov

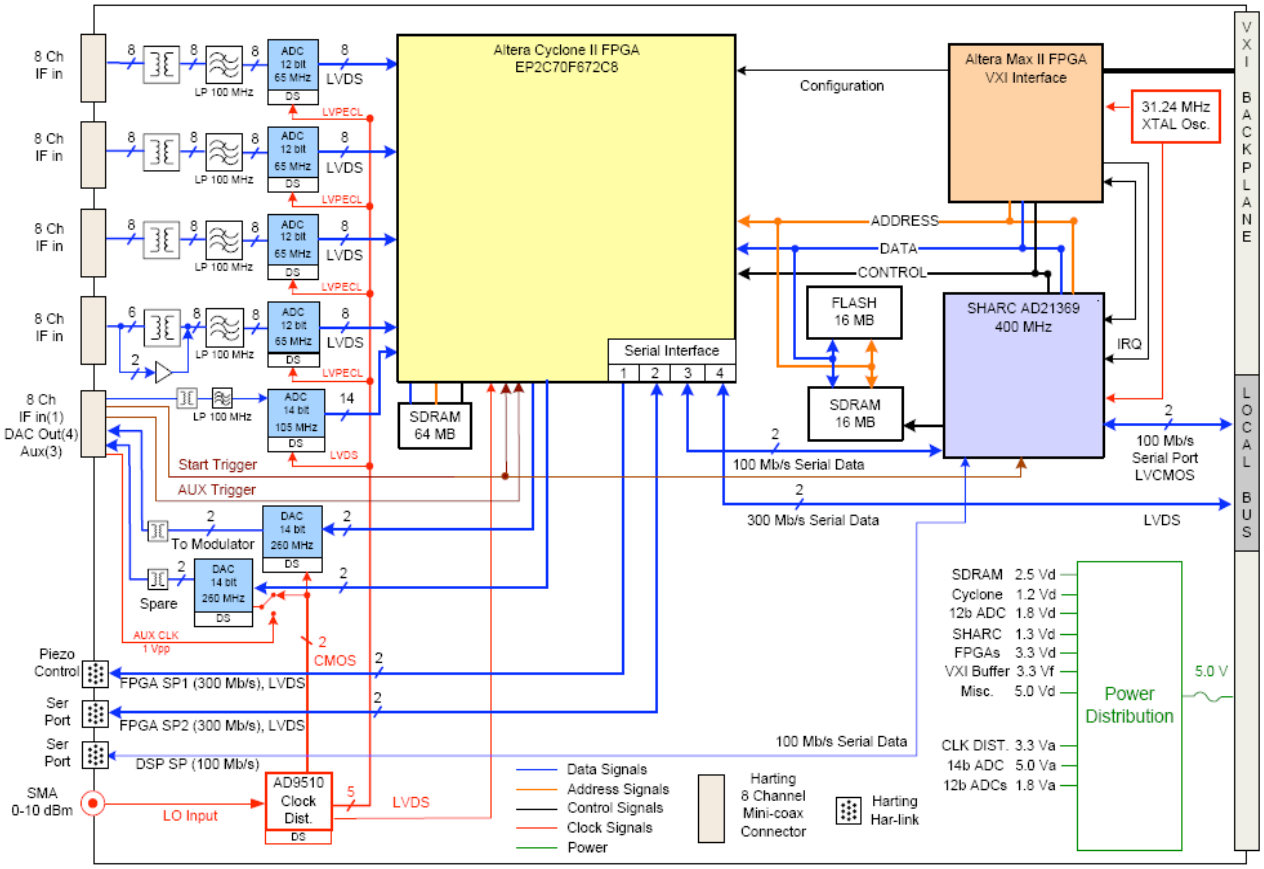


Figure 1: MFC Board Overview

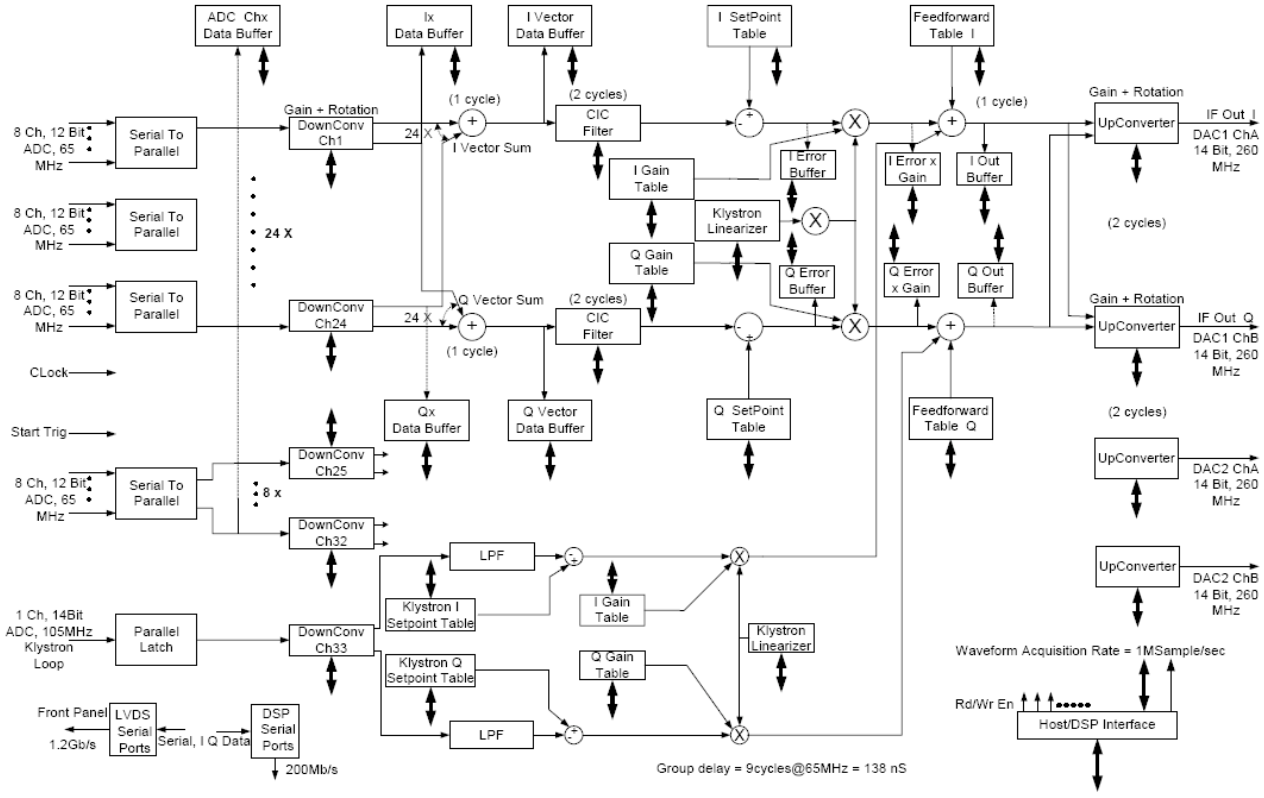


Figure2: FPGA Signal Processing

The tables and diagnostic buffers in the FPGA may be written to or read by the DSP or the slot0 CPU through the host and bus interfaces. Data can be transferred to the larger external memory through the SDRAM interface or may be read directly by the slot0 CPU.

Test Results

The results from testing of the prototype board are presented here. Figure 4 shows data acquired with the 1-channel 14-bit ADC. The measured noise floor is 110dB and it is close to the theoretical limit. The sampling frequency was 65 MHz and the input IF signal is 13MHz at 4 dBm power input.

Figure 5a shows the comparison between the FFT of one channel and the FFT of a vector sum of 6 channels. In the first case a signal with 10MHz IF frequency and +4dBm power is fed into the MFC board. Both graphs are displayed using the Hanning window. It can be calculated that the noise floor is approximately 70dB at bandwidth that equals half the sampling frequency. The red spectrum was obtained by feeding a signal with 13MHz IF frequency and +4dBm power into 6 channels. The spectrum represents the vector sum of these 6 channels. As can be seen there is a decrease of approximately 8dB in the noise floor. The sampling frequency in both cases equals to 1313MHz/21 and we acquired 16k of data.

Figure 5b shows the cross-talk between adjacent channels on the MFC board. For this test we fed a signal at 13MHz IF and +4dBm into one channel on the MFC board and observed the 13MHz component in adjacent channels. We acquired 16k of data at sampling frequency 1313/21MHz. According to the board layout (see board layout) the first figure represents measurements between two channels that are located on the top layer of the board. The second graph on the right is the same measurement but for three adjacent channels on the bottom layer of the MFC board. In all cases the signal was fed into the middle channel.

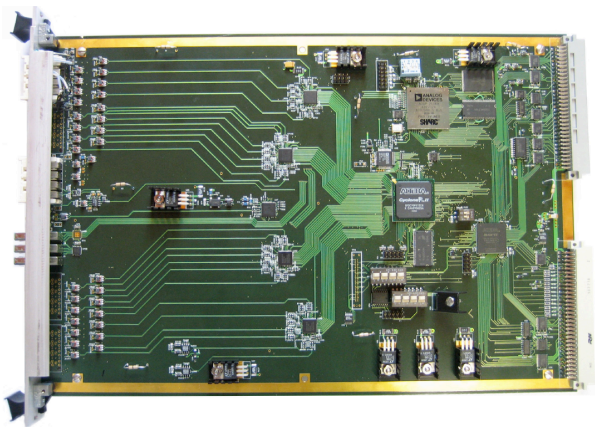


Figure 3: MFC board

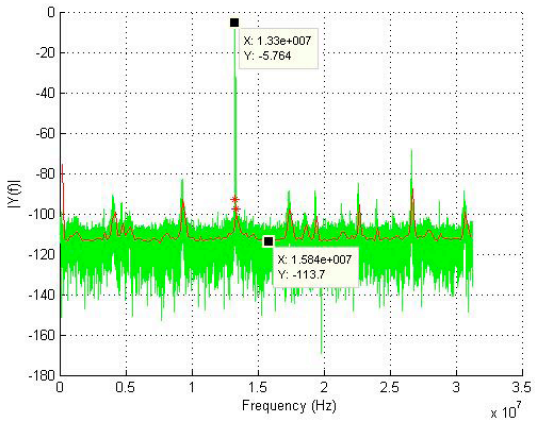


Figure 3: MFC board

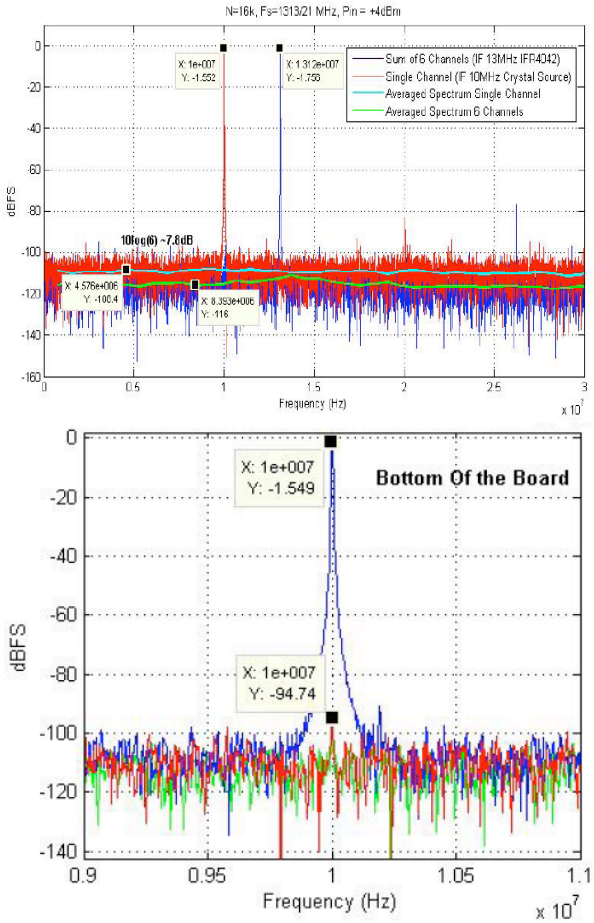


Figure 5: 8 ch ADC - Process gain and Crosstalk

References

[1] J. Branlard, "Survey of LLRF Development for the ILC", PAC'07, Alburquerque, June 2007, FROA06.

[2] U. Mavric et al., "A 96 Channel Receiver for the ILCTA LLRF System at Fermilab", PAC'07, Alburquerque, June 2007, WEPMN102.

LLRF TIMING FOR FERMILAB 201.25 MHZ LINAC

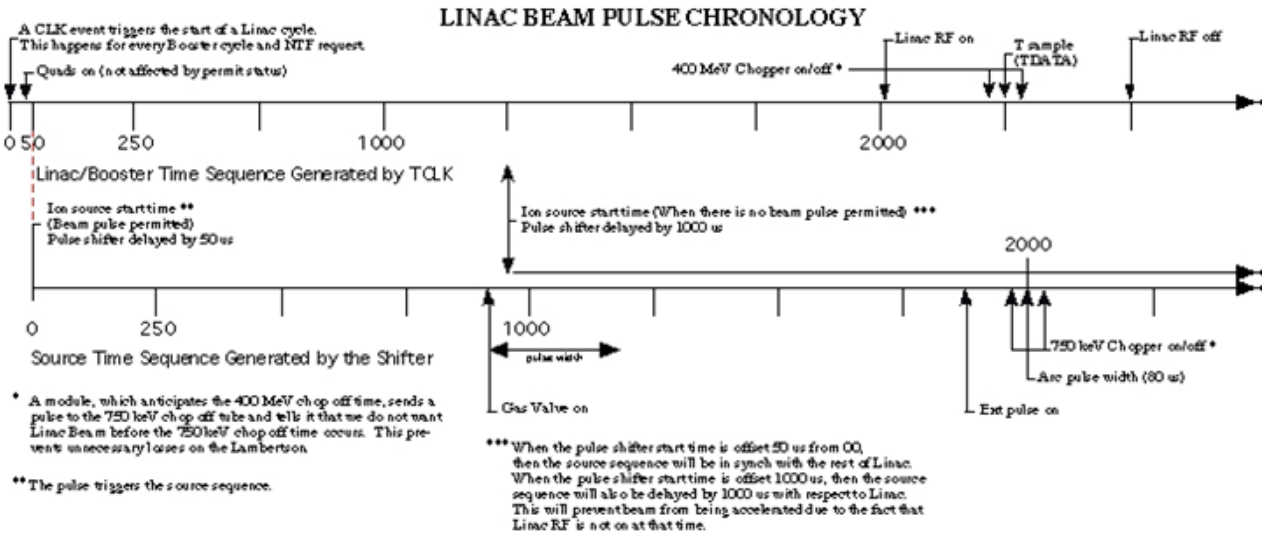
T. Butler, FNAL, Batavia, IL 60510, USA

INTRODUCTION

To synchronize the LLRF system to the entire LINAC, it is important to understand how the LINAC is timed to the rest of the accelerator complex. This synchronization is critical to ensure that the LLRF amplitude feedforward pulse is timed to the correct type of beam at the right time. The 201.25 MHz Pulsed RF system runs at 15 Hz repetition rate with a duty factor of 0.06%. Even though high energy physics (HEP) beam never runs at 15 Hz, due to present RF limitations in the Booster, the Linac RF systems run at 15Hz in order to keep the temperature of the cavities at constant temperature. If the systems were only pulsed during beam requests, the temperature, and therefore performance, of the Linac RF system would suffer. At this time, only NTF can run a constant 15 Hz priority beam during patient treatment.

LINAC TIMELINE

To insure that the proton beam is injected into the Booster at its minimum bending radius, the Linac RF systems are synced via the Booster reset event, which is delayed (1/15) [Hz^{-1}] – 2 [ms] = 64.66 [ms] from the previous Booster magnet current minimum (BMIN). This enables the Linac to inject beam at minimum magnetic field in Booster, keeping the 15 Hz synced at all times by generating a timing reference point 2 ms before injection, giving RF systems time to start up. The Booster reset then generates a $AF event on the timeline which is 2 ms before beam. This reset is sent regardless of Booster GMPS being on or off. This is accomplished by switching over from TCLK to LCLK during GMPS outages. This switch over is done automatically by the Linac control system. Although beam can start and stop at different time for different types of beam, all beam types except NTF require that beam be present at 2000 μs after $AF. This time, defined at TDATA, is described in the timing diagram below. Since it is desired always sample the Linac waveforms during beam in order to observe loading effects, the Linac sample & holds use TDATA to sample the read back waveforms. During NTF events, NTDATA, which is set at 1974 μs, is used to since NTF bean starts ~ 26 μs before beam



LINAC LLRF TIMING

Each LLRF system requires timing information in order run the RF system. These systems include the driver, the modulator high voltage (HV) pulse, and the LLRF amplitude feed forward system. The RF ON pulse, L:TRFON, as seen below in the list of general Linac timing parameters, is generated in the Pre-Acc area node 602, and is sent to each RF station via the pulse repeater system. The same repeater system is also used to send the quadruple timing pulse to all of the power supplies.

General Linac Timing Parameters							
Device	Descriptive Text	Value	Units	Device	Descriptive Text	Value	Units
L:TRFON	Low Energy RF ON	1764.4	US	L:TRFONW	L:TRFON Pulse Width	1	US
L:TSHIFT	Shifter Start Pulse	49.5	US	L:TSHFTW	L:TSHIFT Pulse Width	1	US
L:TTDATA	TDATA Pulse Backup	2000	US	L:TDATAW	L:TTDATA Pulse Width	1	US
L:TZERO	TZERO Timing Channel	0.8	US	L:TZEROW	L:TZERO Pulse Width	1	US

This RF ON pulse is used by both the interlock module and the waveform generator to create all of the driver and RF pulses respectively. The Driver pulse, created by in Linac interlock system, is sent to the LLRF system and is used as a gate for the LLRF RF output on the new system. This was done to insure that the LLRF system only applies power to the Linac driver after interlocks are all made up, thereby removing interlock responsibility from the new LLRF system. The RF ON pulse was originally thought to also be a good option for giving the LLRF systems a start trigger (LLRF ON) for waveform capture, but a new timing parameter, L:V#LLTR, was created for each RF station and is feed to each station from each stations respective node (611-615). For example, in RF Station #5, L:V5LLTR is created in node 615 and is sent from the relay rack behind the main console to LLRF5 using RG58 cable. The full list of these parameters is seen the following chart. By using this new pulse for the LLRF trigger, instead of the RF ON pulse, we are able to generate a pulse approximately 1 ms before beam, which gives ample head room for the LLRF system to do waveform capture before the RF driver pulse is turned on.

Linac LLRF Timing Parameters (Outside VXI System)

Device	Descriptive Text	Value	Units	Device	Descriptive Text	Value	Units
L:V1LLTR	RF5 LLRF Trigger	920	uSec	L:V1LLTW	RF5 LLRF Trigger Width	180	uSec
L:V2LLTR	RF5 LLRF Trigger	920	uSec	L:V2LLTW	RF5 LLRF Trigger Width	180	uSec
L:V3LLTR	RF5 LLRF Trigger	920	uSec	L:V3LLTW	RF5 LLRF Trigger Width	180	uSec
L:V4LLTR	RF5 LLRF Trigger	920	uSec	L:V4LLTW	RF5 LLRF Trigger Width	180	uSec
L:V5LLTR	RF5 LLRF Trigger	920	uSec	L:V5LLTW	RF5 LLRF Trigger Width	180	uSec

Along with the LLRF ON pulse, each LLRF system requires a secondary trigger signal to implement beam feed forward. This signal needs to carry two types of information, the beam type (HEP, NTF, MTA, Studies, etc.) and beam timing. After discussions of different topologies, it was decided the best way to convey this beam information to each LLRF system simultaneously would be to send a pulse from controls node 602, located in the pre-acc control room, to each station. Instead of using a pulse repeater to send the signal, a 5 way signal splitter and buffer module (Beam Sync TTL Pulse Distribution #EB-180651) was installed to send this signal to all 5 LLRF systems. These signals are transmitted using the same Heliax cable used to feed RF signals to the new LLRF system. Three hardwired timers, listed below and located in node 602, are presently used to create and send timing information to the 750keV beam chopper. The standby pulse is listed, but since that beam in not accelerated in the Linac, it is not important as far as the LLRF system is concerned. All three timers are triggered by different clock events, so one, and only one, of them fires on a given 15 Hz cycle.

Linac LLRF Chopper Timing Parameters (Outside VXI System)							
Device	Descriptive Text	Value	Units	Device	Descriptive Text	Value	Units
L:TCHHON	HEP Chopper ON	1983.3	US	L:TCHHOW	L:TCHHON Pulse Width	1	US
L:TCHHOF	HEP Chopper OFF	2035.3	US	L:TCHHFW	L:TCHHOF Pulse Width	1	US
L:TCHNON	NTF Chopper ON	1957	US	L:TCHNOW	L:TCHNON Pulse Width	1	US
L:TCHNOF	NTF Chopper OFF	2019	US	L:TCHNFW	L:TCHNOF Pulse Width	1	US
L:TCHTON	Tuneup Chopper ON	1983.3	US	L:TCHTOW	L:TCHTON Pulse Width	1	US
L:TCHTOF	Tuneup Chopper OFF	2003.3	US	L:TCHTFW	L:TCHTOF Pulse Width	1	US
L:TCHSON	Standby Chopper ON	1987.5	US	L:TCHSOW	L:TCHSON Pulse Width	1	US
L:TCHSOF	Standby Chopper OFF	1997.5	US	L:TCHSFW	L:TCHSOF Pulse Width	1	US

Using the type of beam and start time of beam, a local application, called LLLT, running on node 602, creates a pulse that is calibrated to be approximately 1 ms before beam and whose pulse width is predefined to be one of 4 different beam lengths to account for the 4 current types of beam. This is done automatically whenever changes are made to the above parameters. The following table lists the different types of beam that are, and will be, accelerated in Linac which use the LLLT local application to create the appropriate beam “pre-pulse” signal. Note that the difference between LRFTH and LRFTN is (986.1 μsec – 959.8 μsec = 26.3 μsec), which should be equal to the time difference between TDATA and NTDATA discussed earlier (2000 μsec – 1974 μsec = 26 μsec). Also note that the implement of the MTA beam trigger and beam event will be added at a future date.

Linac LLRF Timing Parameters (Outside VXI System)							
Device	Descriptive Text	Value	Units	Device	Descriptive Text	Value	Units
L:LRFTH	LE LLRF HEP Beam Trigger	986.1	uSec	L:LRFTHW	L:LRFTH Pulse Width	0.4	uSec
L:LRFTN	LE LLRF NTF Beam Trigger	959.8	uSec	L:LRFTNW	L:LRFTN Pulse Width	0.6	uSec

L:LRFTT	LE LLRF Tuneup Beam Trigger	986.1	uSec	L:LRFTTW	L:LRFTT Pulse Width	0.8	uSec
L:LRFTM	LE LLRF MTA Beam Trigger	986.1	uSec	L:LRFTMW	L:LRFTM Pulse Width	1.0	uSec

This beam pulse, also referred to as a pre-pulse, is only sent during beam events and is not necessarily a 15 Hz event. Only when NTF runs priority beam does this event occur at a 15 Hz repetition rate. Since this event does not send out anything during standby pulses, which do not need LLRF feed forward since they are out of sync with the RF, it was decided to time this event to occur during the LLRF ON pulse. The LLRF On pulse is then 'or'd with the pre-pulse inside the new LLRF system to insure that this pulse is decoded properly as the beam pre-pulse. The LLRF ON pulse width is set a 180 μsec to insure that variations in timing for future beam types in Linac will still fall within the LLRF ON pulse length. This pulse adds the secondary benefit of reduction susceptibility to noise, which is very prevalent in the Linac gallery, on the pre-pulse transmission line.

The third benefit is that, in order to decode the stop time, i.e. width, of the beam being accelerated in Linac, it was decided to 'or' the chopper gate pulse alongside the pre-pulse. This 'or' module is also located above node 602 and was installed before the LLRF upgrade was started. This chopper pulse is used for 2 purposes. First, the LLRF system uses it to shutdown the LLRF feed forward pulse if somehow the chopper fires but beam current never makes it down the Linac, due to an interlock glitch, failure in the chopper power supply, or any other downstream machine failure. This shutoff feature prevents the LLRF from over powering the accelerating cavities when failures occur, which should prevent any sparking in the accelerating cavity. There is a fixed 6 μsec delay from the start of start of the LLRF feed forward pulse to insure that the rising edge of the chopper pulse as occurred. If the chopper pulse does not appear before 6 μsec passes, the feed forward pulse is halted. Along with the rising edge, the falling edge of the chopper pulse is also used by the new LLRF system to shutdown the LLRF feed forward pulse. This is also done to prevent over powering the accelerating cavities after the beam has left the cavities. This can occur since the response time of the modulator can be as large as 10 μsec . This is a parameter L:V#TCHP is used set to delay from the end of chopper pulse to the end of the feed forward pulse. It is currently set at all stations to 0 μsec since the time delay in the modulator is greater than the time of flight of beam from the chopper to each respective station. Future studies will involve delaying the chopper pulse increase the time advance of the chopper pulse seen at each station with respect to beam toroid signal.

In order to test out future code changes, a LLRF test station was set up in the Linac gallery room #LIG-139 on a spare relay rack. An 805 MHz signal was pulled from the reference line in the tunnel, exiting behind LRF5, to be used as the frequency standard. A RG-58 coaxial cable was also pulled from behind LRF4 to send a Linac Clock signal to the test area. This signal is feed from a pulse repeater that was installed by controls group to send the clock signal throughout the Linac. In order to replicate the entire pre-pulse, LLRF trigger, chopper, and beam toroid pulse, spare timing channels from node 612 were also pulled. These parameters are listed below. Working together, these signals are used to create a working test station called LRF0. Unlike a working RF station LLRF system, the test station LLRF system can only be used to check the amplitude feed forward feature in manual mode and does not perform any phase feedback or feed forward.

Linac LLRF Test Station Timing Parameters							
Device	Descriptive Text	Value	Units	Device	Descriptive Text	Value	Units
L:T2DLY2	T2 Timer Delay 2 (LLRF Trigger)	920	uSec	L:T2WDT2	T2 Timer Width 2	180	uSec
L:T2DLY3	T2 Timer Delay 3 (Pre-Pulse)	986	uSec	L:T2WDT3	T2 Timer Width 3	0.4	uSec
L:T2DLY4	T2 Timer Delay 4 (Chopper)	1983	uSec	L:T2WDT4	T2 Timer Width 4	52	uSec
L:T2DLY5	T2 Timer Delay 5 (Beam Toroid)	1990	uSec	L:T2WDT5	T2 Timer Width 5	52	uSec
L:T2DLY6	T2 Timer Delay 6 (Spare)	2000	uSec	L:T2WDT6	T2 Timer Width 6	1	uSec

LLRF Linac Engineering Note LEN-0002

Beam Feed-forward Compensation

V.Tupikov (x6322) 23 February 2009

There are two modes of beam feed-forward compensation on Linac: manual and adaptive. Both of them are triggered by front edge of "Prepulse" signal, but just adaptive mode uses the width information of the same trigger for beam type decoding. Please refer to timing diagram for details.

System triggers

LLRF Start (RFON)

This is the main trigger presented on each RF-cycle. It affects the following processes:

- Feed-forward compensation;
- Waveform capturing;
- Data acquisition;

Ppepulse

As it was said above – this is the trigger responsible for feed-forward compensation. It performs two main functions:

- Defines start time of 1-ms interval before beam comes to the cavity;
- Encodes the beam type by its width:
    - the width = [370..430]ns corresponds to HEP beam;
    - the width = [570..630]ns corresponds to NTF beam;
    - the width = [770..830]ns corresponds to STU beam;
- Waveform capturing by beam present event (when armed);

Manual Mode Compensation

Manual mode doesn't require a beam type decoding and works the same way for all beam types. The feed-forward pulse is shaped with use of Acnet parameters to provide the best beam compensation. The tuning parameters are:

- Ramp start time V#THFR;
- Ramp width V#TFRW (rise time of the pulse to its maximum);
- Spike start time V#TSPS;
- Spike width V#TSPW;
- Spike magnitude V#ASP;
- Pulse magnitude V#FFMA;
- Pulse tilt V#FTLT (magnitude change expressed in 16-b word count change per 16ns);
- Beam chopper advanced shutoff V#TCHP;

NOTE: all timings relate to time-0, which is a front edge of "LLRF-Start' trigger.

To arm the feed-forward during the current RF-cycle, "Prepulse" trigger must be completely overlapped by "LLRF-Start" trigger. The feed-forward pulse will be issued in approximately 1-ms after the front edge of the "Prepulse".

Adaptive Mode Compensation

In addition to the Acnet parameters listed in Manual Mode paragraph, there are few mores for Adaptive Mode:

- binary mode switch V#MDSW[8] (toggling between modes);
- weight coefficient for adjusting gradient error during compensation cycle V#FDFW;

The shape of compensation pulse is formed mostly the same way as it is done in manual mode. The differences are:

- Pulse magnitude becomes dynamic variable, depending on its history (last value) and last cycle weighted error: Aff(n) = Aff(n-1) + weight*Aerr(n-1). It is worth to note here that spike magnitude is not affected;

- The magnitude correction is done for each beam type individually. Feed-forward pulse magnitudes (Aff-s) are stored in different registers for HEP, NTF and STU beams;

Again, the front edge of "Prepulse" arms feed-forward compensation circuit. Then a trigger width decoder determines the beam type and issues two types of beam identifiers:

- 2-bit beam type bus (00b-HEP, 01b-NTF, 10b-STU, 11b-NO BEAM);

- three beam select lines per each beam type: HEP_SEL, NTF_SEL, STU_SEL;

The bus identifier selects the channel of output multiplexer connected with corresponding beam type magnitude register. Each single line identifier goes to the modules where selected beam type gets processed.

Beam Type Diagnostic

New LLRF system provides built-in graphical beam type diagnostic. Bring up a plot window L:V#WDGA from D27 and make plot multiplexer L:V#WAMX = 14, - you'll see gradient waveform with either one sampling window marker (spike) @ no beam present or 3 sampling window markers @ HEP, NTF or STU beam. The height of each of these markers identifies beam type as follows:

- A_spike = 0.06 corresponds to NO BEAM;

- A_spike = 0.03 corresponds to STU BEAM;

- A_spike = 0.015 corresponds to NTF BEAM;

- A_spike = 0.0075 corresponds to HEP BEAM;

In a case when Beam Type versus Time plot is required it can be done as followings. Waveform #14 has to be selected for array L:V#WDGA, the way it was described earlier in this section. Locate the array index, which corresponds to the area of scalar sampling window (usually on rear top of the gradient). The index should be somewhere around 447 as in my example (see below). Now the value of L:V#WDGA[447] can be either used as readback, which is not convenient, or observed on Time plot. To be able create a time plot with array element you'll need to use Utility window's FTP Plot instead of Parameter page's FTP.