# 

Figure 1 Top level of FEB firmware

# General

Figure 1 show the top level block diagram of the FEB firmware. The firmware source is written in VHDL. Xilinx IP cores instantiated in the firmware are generated using Xilinx Coregen. The HDL is synthesized using Mentor Graphics Precision, producing an EDIF netlist. Xilinx ISE is used to place and route the synthesized netlist, check timing, and generate a bitfile. Mentor Graphics Modelsim is used for simulations. The test bench is written in VHDL and TCL. Matlab is used for all DAQ testing and data analysis for both simulation and hardware testing.

There are two versions of FEB firmware, one for FEB4 and one for FEB5. The firmware has many components that are shared, in which case, these components come from the same source file. The VHDL file, top\_common\_struct.vhd, is used as the top level for both versions. The VHDL package file, register\_address\_map.vhd, contains the parameters used to specify how the VHDL is synthesized. These parameters will specify how to generate and synthesize a design as FEB4 or FEB5. The package file also defines the address map and command definitions.

The FEB utilizes registers in a 16 bit address and data space to control its functionality. One of these registers is used as a command register and the rest are used to specify details of how the FEB operates. Detailed information on all of the FEB Registers and commands can be found in NOVA Document 4662-v20 “NOvA FEB Programming Manual”

# Data Path

The detector data path through the FPGA, shown in blue in Figure 1 , begins with the digitized ASIC data input from the ADC. Each FEB will process a serial data stream from 32 APD channels. The processing of each channel will time share the same FPGA resources, but remain functionally independent. The data will first be deserialized, time-stamped, and then sent to the Data Processing block where it is sparsified and buffered. The buffered data is formed into packets in the DCM interface and transmitted to the DCM.

## ADC Deserialization

The main difference between the FEB4 and FEB5 is APD data sampling rate. In the case of FEB4, each APD channel is sampled at a rate of 2MHz whereas in FEB 5 each channel is sampled at a rate of 8MHz

### FEB4 quad ADC

The FEB4 uses the CERN Chipidea AD41240 ADC for APD data digitization. This is a quad 12-bit ADC with two 12-bit parallel outputs operating at double data rate. In the FPGA, the ADC data from all 32 APD channels are merged into a single data stream and processed using a single instance of time multiplexed FPGA resources.

### FEB5 octal ADC

The FEB5 uses two Analog Devices AD9222ABCPZ-40 ADCs for APD data digitization. This is an octal 12-bit ADC with 8 serial outputs. The 16 serial data streams from both ADCs are source synchronous and include one frame bit from each ADC for data alignment. The frame signal is deserialized in the same manner as the 16 data streams requiring a total of 18 deserializers.

In the FPGA the deserializers include a phase detector and board deskew circuit that will automatically adjust the data stream input timing such that the serial data will always be sampled midway between data edges. Each serial data stream is input into two independent selectable tap delay lines providing two locations to sample a single data bit. One delay is considered to be the master and is used to deserialize the data. The other delay is the slave and is used to scan the data looking for the data transition point. The master and slave delay are adjusted together with an offset until the master detects a change and the slave has determined it has found an edge. The slave uses the information from past delay settings to find the transitions. The algorithm only advances with transitions on the data line. This method can be used for serial communication up to 1Gbs using a Xilinx Spartan 6 device. Detailed information can be found in Xilinx XAPP 1064 “Source-Synchronous Serialization and Deserialization”

Since the channel sample rate is 4X greater than the sample rate of FEB4, the deserialized APD data from 32 channels is split into four data streams and processed in parallel using four instances of time multiplexed FPGA resources to process the data.

## Timing

The FEB uses an internal time counter to timestamp the APD data. The time is synchronized across different FEBs using the Sync signal on the FEB-DCM interface link. When the FEB receives the Sync signal it will set the current FEB time to a preset value. The current time can be started and stopped using the corresponding FEB commands.

Each ADC on the FEB uses a 16 MHz sample rate. The 16MHz ADC clock is generated by the PLL based Clock Generator using the DCM interface link 32MHz clock as a reference. This clock generation introduces a phase ambiguity on the 16 MHz ADC clock. The DCM interface link sync signal is expected to have a fixed phase relationship with the 16MHz clock and is used by the FEB to adjust for this timing ambiguity.

## Data Sparsification and Triggering

The FEB data sparsification uses a magnitude over-threshold scheme where a FIR filtered version of the channel’s data is compared to a preset threshold. If the magnitude exceeds the threshold, an event will be triggered and the corresponding data from the triggered channel will be buffered until it can be sent to the DCM. The trigger will look for a maximum value over the threshold during a single event rather than the first value over threshold. The FIR filter coefficients used are [1 0 0 -1]. This is referred to as Dual Correlated Sampling (DCS). The characteristics of this filter will be dependent on FEB version as the sampling rates are different. In the case of the FEB4.X, the filter will be the difference of samples that are separated by 1.5us. In the case of FEB5.X, the filter will be the difference of samples that are separated by 375ns. The triggering threshold can be uniquely set for each channel.

When an event is triggered, the FEB will collect a number of continuous samples, before and after, the sample that triggered the event. The number of samples preceding the trigger and the number of samples collected after is a settable parameter. This format is referred to as Multipoint readout. This will allow the samples surrounding an event to be used offline to reduce the signal noise. In Multipoint readout, the preset number of continuous samples of an event will be sent in a single data packet to the DCM. Multipoint is the readout mode currently used, however, it was not the original specification for transmitting detector data to the DCM. The original specification called for a triggered event to send a single 12-bit data point, the maximum DCS value over threshold. This readout mode is no longer used but was re-written and included alongside Multipoint as an option to ease the transition to the current Multipoint mode. The original readout mode referred to as Legacy mode. In both readout modes, each triggered event generates one data packet sent to the DCM.

The FEB will include a diagnostics mode where the sparsification trigger can be turned off, referred to as “oscilloscope mode”. In this mode, the FEB will operate similar to a digital storage oscilloscope. The digitized APD data from the enabled channels will be buffered in one continuous time slice. Since the buffer memory is fixed and shared between all channels, the length of the time slice is determined by the number of channels that are enabled. Oscilloscope mode is used in conjunction with the FEB's pulse generator functionality. When the FEB is enabled and ready to take data, the data buffer must be 'triggered' using the internal pulser signal. If the Event buffer becomes full, it will signal the FEB to stop taking data and put itself in idle mode. The number of contiguous samples returned is determined by the FEB to DCM throughput, data buffer depth, and number of channels that are enabled. The FEB can be set to limit the number of samples that are returned.

## DCM Interface Link

The DCM interface will coordinate the data that is sent between the DCM and FEB. Communication between the FEB and DCM uses an 8b10b protocol. Comma characters are used for alignment.

The DCM interface link command signal is used to send register read and write requests from the DCM to the FEB. The data signal is used to transmit data from FEB including triggered APD data, timing data, status data and register read data. Information on this protocol can be found in NOVA-doc-814,” Link Interface Specification for the Nova Front End Board to Data Concentrator Module Connection”.

# Control

The control block will coordinate the operation of the FEB and serve as an interface to components on the FEB that are external to the FPGA. It utilizes a combination of commands and registers to control the operation of the FEB. Registers are used to set FEB parameters and commands are used to change FEB modes of operation or initiate an event.

## ASIC Programming

The ASIC contains several configuration registers. The Control block will provide an interface to the ASIC configuration registers and allows it to be configured using the standard DCM interface register read and write commands. The FEB has a register for each settable ASIC parameter. Once all of the ASIC registers have been set, the FEB Set ASIC command will transfer the configuration to the ASIC. This will also transfer the previous ASIC configuration back into the corresponding FEB registers which are available to be read-back. This ASIC configuration loop provides a verification that the configuration is properly being received by the ASIC.

## Pulse Generator

The FEB contains a configurable pulse generator, only used in testing and debugging, that provides a way to synchronize the digitization of APD data with external stimulus such as ASIC charge injection. A single pulse can be selected as well as pulse train with a configurable period and pulse length. The pulse generator signal can be sent to any of three locations, the data readout trigger, the ASIC charge injection, and a FEB external connector that can trigger external instruments. The pulser signal that is sent to the read out trigger is advanced by a fixed amount to allow samples prior to an external event to be also included in the data. This synchronization is required to catch the data that corresponds with an external stimulus. The Pulser Enable bits determine what part of the FEB will see the pulse. A common test preformed is to start the FEB in oscilloscope mode and use the pulser to trigger the buffering of APD data. After a fixed delay, an external pulse is then used to trigger a calibrated injection of charge. The APD data including the injected charge is then read out and analyzed. The pulser can also be set to periodically inject charge into the ASIC using the APD data itself to trigger events. This would generate a constant stream of events that are processed as they are actual events.

## SPI Interface

There are several components on the FEB that use the SPI bus communication protocol. The Control block provides a SPI bus interface to these components using the standard DCM interface register read and write commands. These components include a temperature sensor, 128 byte EEPROM used to store the board serial number, and an ADC and DAC used to interface with the TECC and high-voltage regulator.

### FEB Temperature

The FEB Temperature sensor is read by the FEB using the SPI interface. This readout is controlled using the corresponding FEB command. The FEB provides access to this readout through the DCM interface link.

### Serial Number

A unique FEB Serial number is stored as ASCII values in nonvolatile memory on the FEB. The memory contains 128 characters and the first 12 characters represent the FEB's serial number. The first 6 characters of the serial number is the PCB version and the last 5 is the unique board identifier. e.g. FEB4.0-00016. The entire character string can be read using the Serial Number Register. Each time this register is read, the FEB returns the registers current value to the DCM and fetches the next character from the memory. The character pointers are controlled using the corresponding FEB commands.

### TECC Control

The TECC is controlled by providing an analog “Setpoint” along with Enable signal. The TECC is monitored by reading the values of two analog signals, “DriveMonitor” and “TempMonitor”. The ADC and DAC used in this interface are controlled by the FEB using the SPI interface. The FEB will provide access to the TECC through the DCM interface

The FEB will automatically monitor the temperature and drive signals reported from the TECC and look for potential problems in the APD cooling system that could physically damage the detector. The FEB will automatically disable the TECC and flag an error condition if the temperature monitor is too high or the drive monitor is too high for a configurable amount of time. The threshold levels that generate errors are fixed values and the time threshold is programmable and measured in minutes.

### High Voltage

The ADC used in the high voltage regulator is controlled by the FEB using the SPI interface. The FEB provides access to the regulator settings through the DCM interface.

# Clock Generator

The Clock Generator contains a PLL to generate all of the clocks required by the FEB using a 32 MHz reference clock provide on the DCM Interface link. The logic primarily uses the 64 MHz and 128 MHz clock. The FEB also uses 4MHz, 16MHz, and 32MHz for APD data digitization and timing synchronization.

# Soft Power-up

FEB V5.x consumes 1.1 amps of current on the Low Voltage supply where 50% of this current is required by the 2 ADCs on the board. When 64 FEBs, which are attached to one Power Distribution Box, are synchronously programmed with firmware or un-programmed with the sync signal, they produce a 50 amp current step. This step, in conjunction with the power supply sense cable feedback-loop, produce a voltage ringing on the power supply which has amplitude which falls outside the preset limits. To eliminate this problem, each FEB will delay the powering of the ADCs by a different amount; distributing a single large step into smaller steps. Once programmed with firmware, the FEB will read its serial number which is stored in external memory. The last 2 digits of this serial number will dinnertime the amount to delay the power-up of the ADC. The ADC power-up will happen automatically when the FEB is programmed with firmware. If the FEB is going to be un-programmed, as happens when a currently programmed FEB is about to be re-programmed, a command will need to instruct the FEB to power-down the ADCs in the same manner. The length of the ramp will range from 0 to 10ms The power state of the ADC will be reported in the status register. This feature is only functional in the Version 5. Any related commands sent to version 4 will safely be ignored.