# WIB Firmware and Software Requirements

Josh Klein

CE Review

March 10, 2020

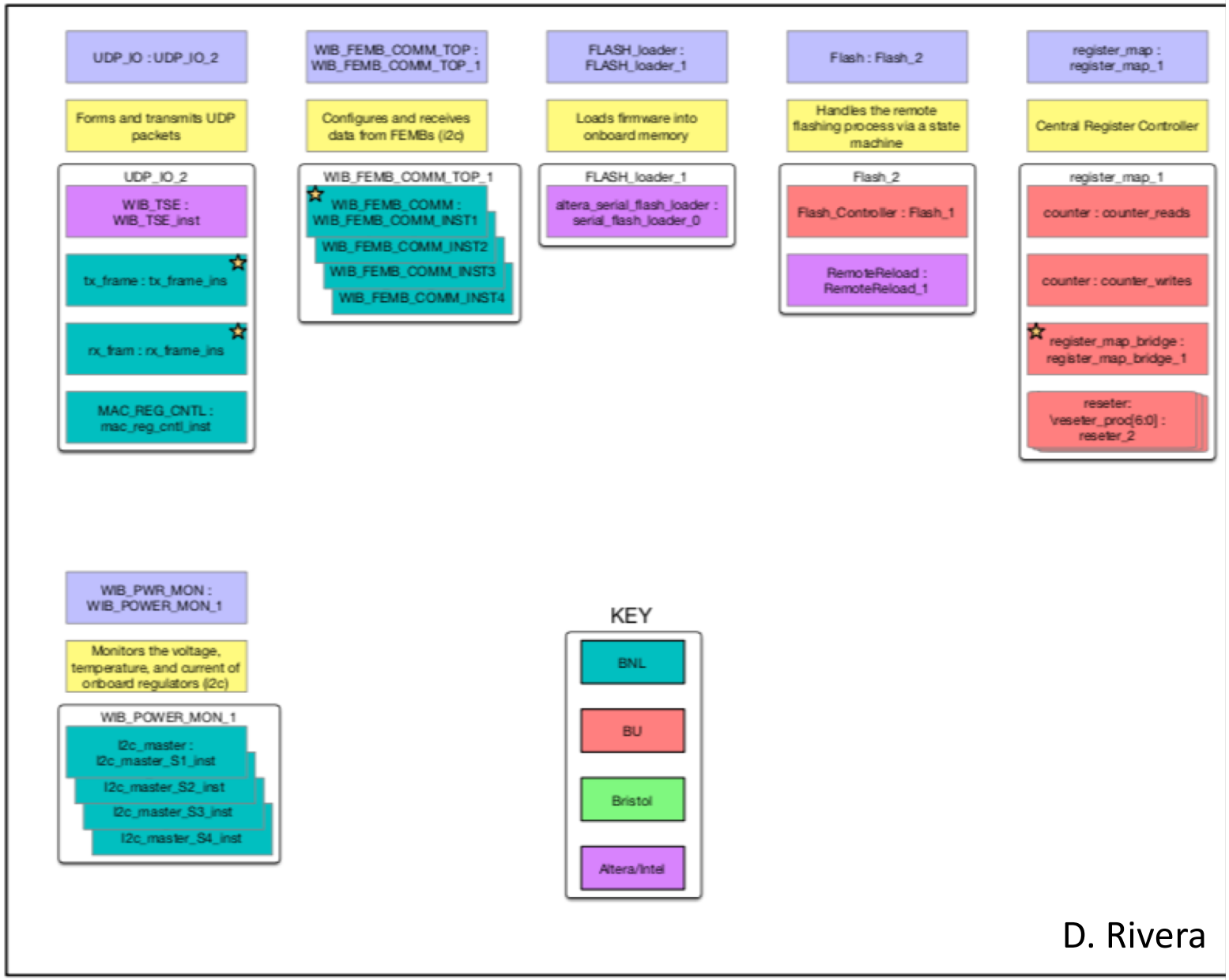
Penn
University of Pennsylvania

DUNE

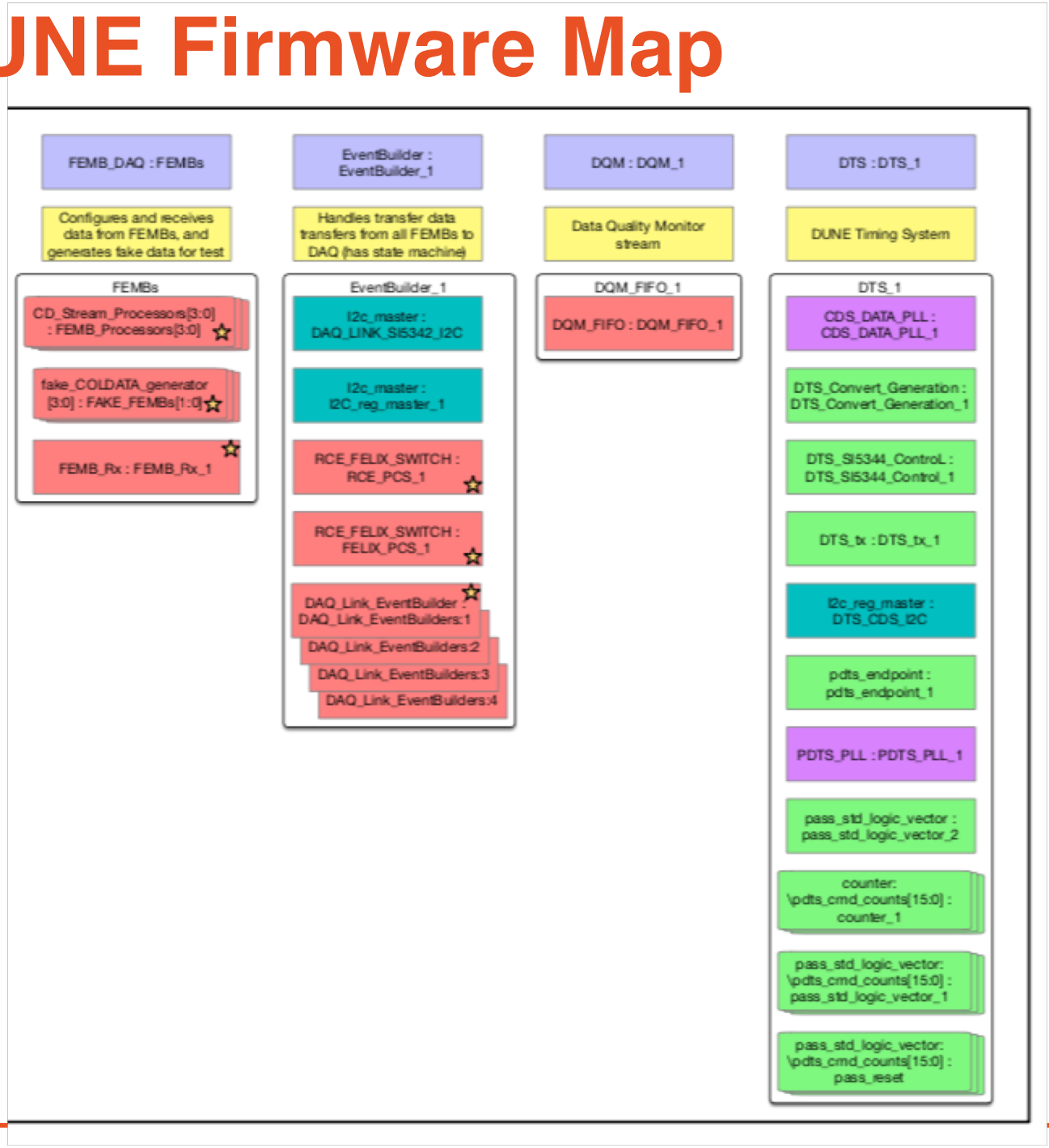
# Background

- ProtoDUNE WIB operated (and continuing) successfully
- FPGA was an Altera Aria V
- All-firmware design, included different firmware sets for different functions (e.g.. Running vs. debugging)

# ProtoDUNE Firmware Map

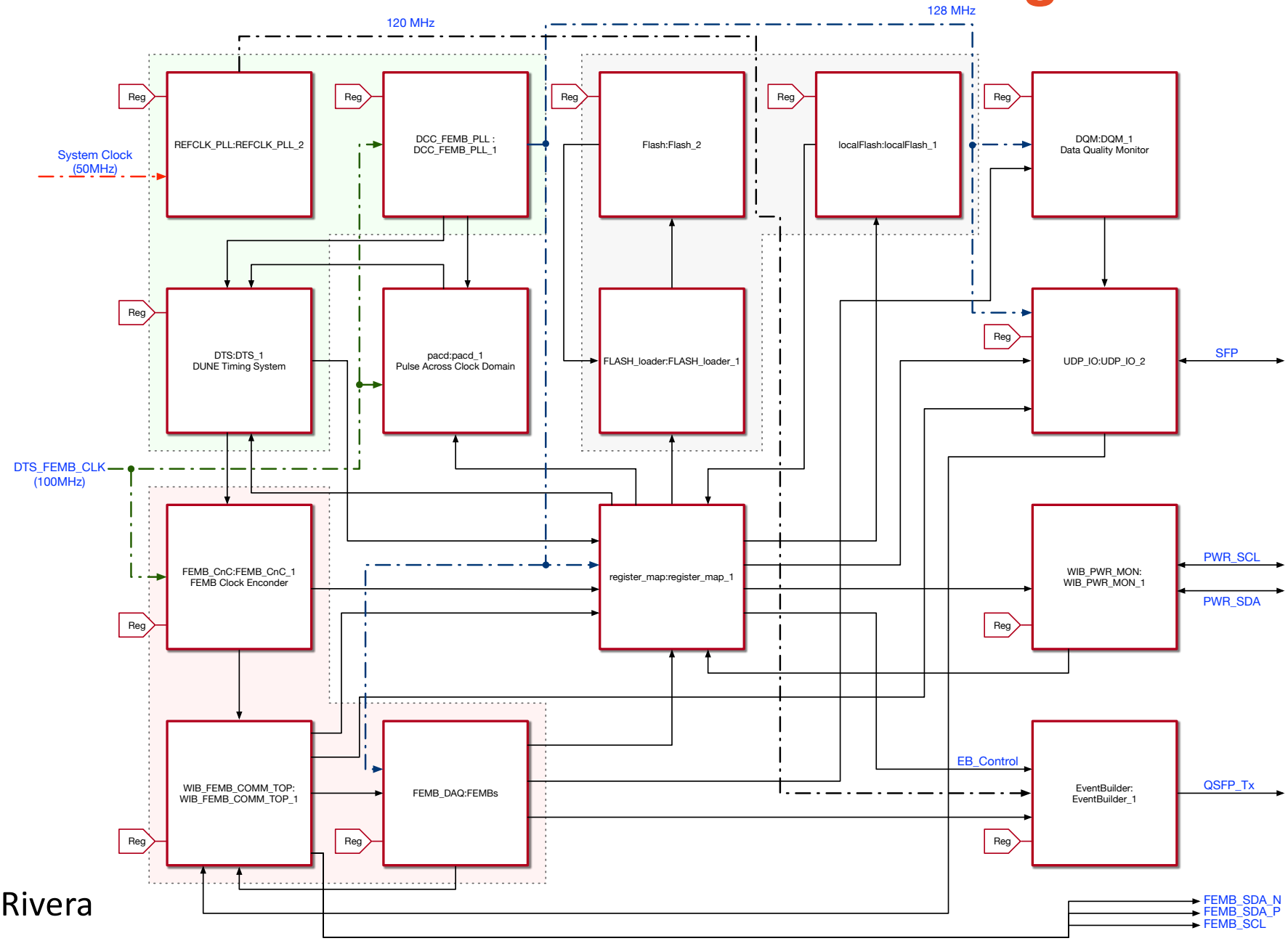


# ProtoDUNE Firmware Map



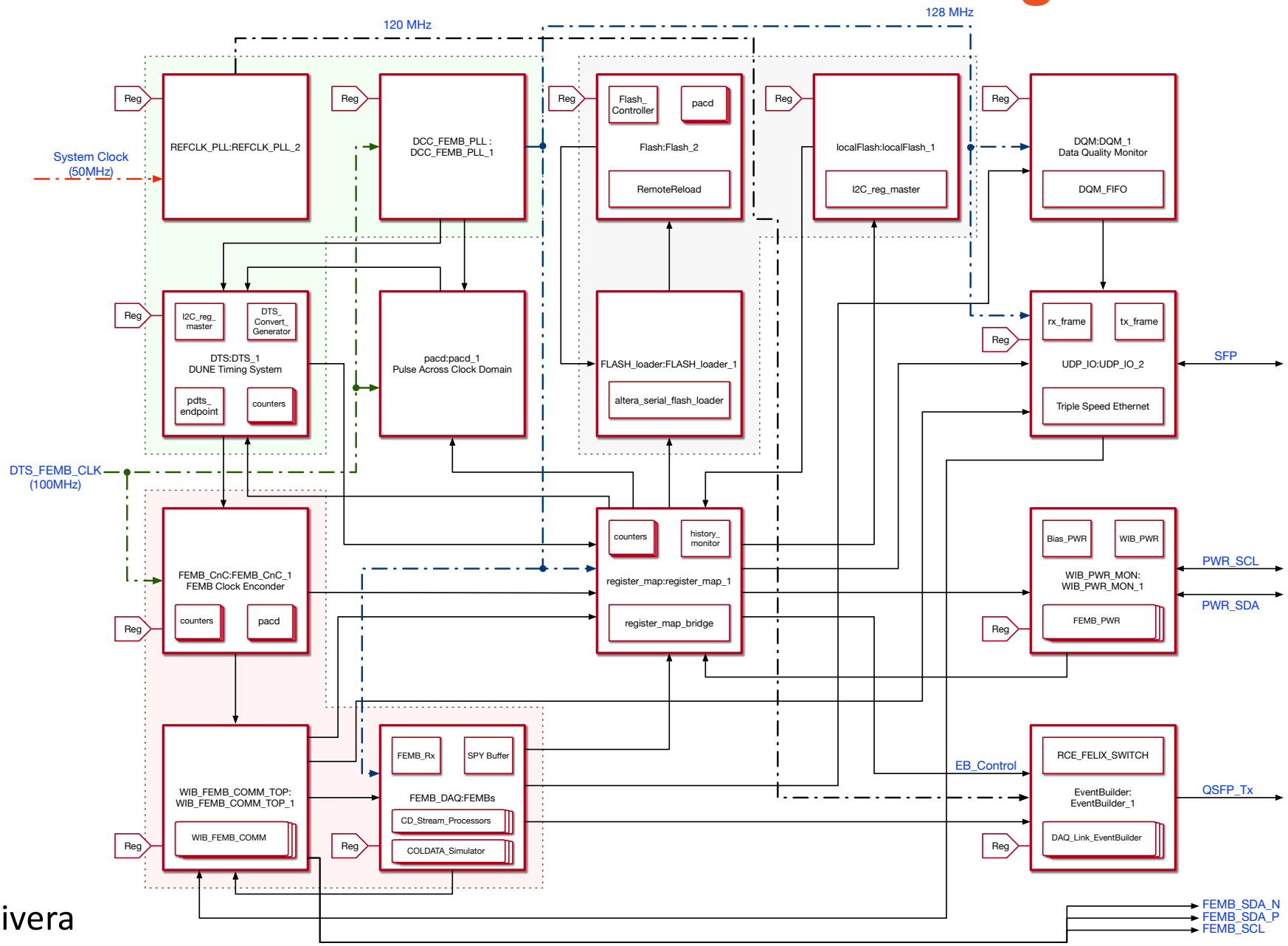

D. Rivera

# ProtoDUNE Firmware Interface Diagram




D. Rivera

# ProtoDUNE Firmware Interface Diagram

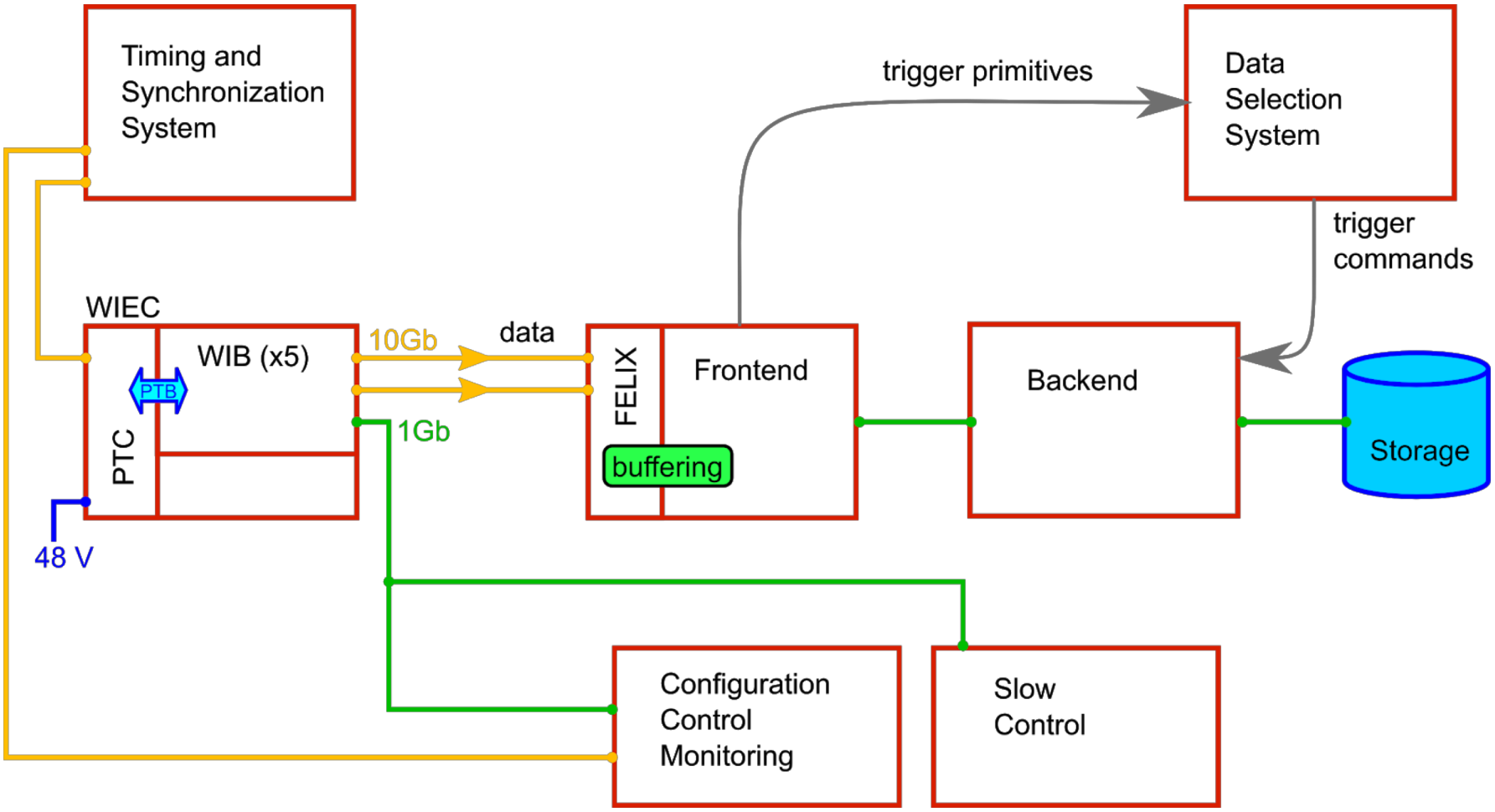



D. Rivera

# Major Changes for DUNE

- Scale: 750 WIBs/SP Module for 20 years with < 1% integral downtime
  →100-1000 times the exposure of ProtoDUNE
- Either COLDATA or CRYO on FEMB; no FPGA
- Upstream DAQ will be entirely FELIX
- Sampling clock generated on FEMB---62.5 MHz from WIB
- ZYNQ Ultrascale+ allows additional flexibility: calibrations, debugging, commissioning, etc.
- WIB will communicate with PTC via I2C bus, transmit interlocks from PTC to WIB, and send WIB hardware monitoring information to PTC.

# WIB in DAQ Context




V. Tishchenko

# WIB High-Level Firmware/Software Requirements

1. Act as DAQ Timing Endpoint,  keeping WIB and FEMB timestamps synchronized

2. Receive high-speed data FEMB(s), re-format to match CE/DAQ Interface specs

3. Provide data tags reflecting WIB and FEMB state(s)

4. Send uncorrupted data to DAQ, with data integrity checks

5. Respond to commands from the DUNE CCM and Slow Controls

# WIB High-Level Firmware/Software Requirements

6. Provide monitoring data to both DUNE CCM and Slow Controls

7. Pass the reformatted data continuously to the DAQ over the high-speed links

8. Provide real-time diagnostic data without impacting data transmission to the DAQ.

9. Provide commands to the FEMBs, including synchronization to the WIB clock, configurations, and calibration commands.

10. Perform FE electronic charge injection calibrations, upon command from DUNE DAQ, minimizing duration

# WIB High-Level Firmware/Software Requirements

11. Provide extracted ADC calibration constants for logging and diagnostic purposes

12. Support debugging and diagnostic modes, such as the generation and transmission of fake data.

13. Handle hardware interlocks and provide monitoring via PTC

14. All of the above can be done and adjusted parametrically via the DAQ CCM, without the need for distinct firmware or software to be loaded into the WIB.

# WIB Logical Interfaces

WIB sits between two systems:

## DAQ

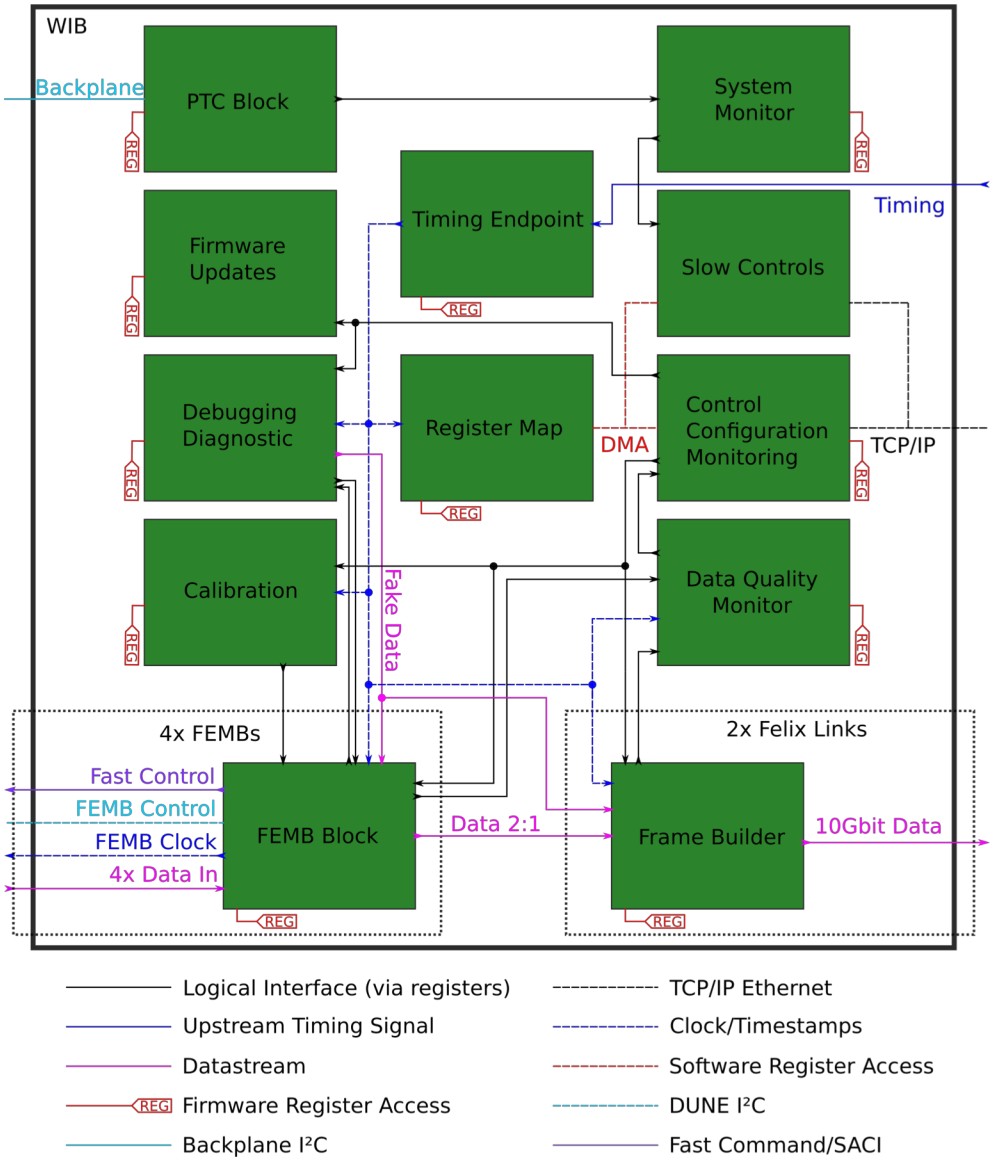
- CCM
- Slow Control
- Upstream DAQ
- Timing system
- Interlocks via PTC

## Cold Electronics

- FEMB commands
  - “Fast commands” + I2C-like for COLDATA
  - SACI interface for CRYO
- Clock
- FEMB data

# WIB Logical Blocks

Division is not just logical but also intended to be developmental




- Communication between Process System (PS) and Programmable Logic (PL) done via memory map, called “Register Map” here following ProtoDUNE (and CTB)
- Have indicated logical connections on this diagram even if they flow through register map
- Explicit register access indicated separately
- Some of these project highly onto ProtoDUNE version, some are linear combinations, some are all new

B. Land

# WIB Firmware/Software Division

- ZYNQ chip allows possibility for some blocks to be all firmware, some all software, some a hybrid
- We have not made any final decisions on the line between these for each block, but some are more naturally one than the other
- Timing critical pieces are all firmware
- High-level interfaces or processes are in software
- Possibility of hybrid interfaces leads to “developmental” as well as logical division

# Firmware/Software Divides

Anticipated firmware/software block mapping:

- Frame Builder
- Timing Block
- System Monitor
- PTC Interface
- FEMB Block
- Slow Controls
- Firmware updates
- Control, Configuration, and Monitoring
- Debugging/Diagnostic
- Data Quality Monitor (DQM)
- Calibrations

Green=all firmware
Orange=firmware+software
Blue=all software

# FEMB Block

FEMB Block functionality:

Interface Blocks

- Clock Transmission (62.5 MHz) to FEMBs
- Reception of data from 4 FEMBs including decoding, stripping and merging
- Command interface(s) to COLDATA or CRYO
- Transmission of merged data downstream to Frame Builder
- Front-end of data integrity checks and error detection
- Transmission of data integrity metrics to WIB DQM

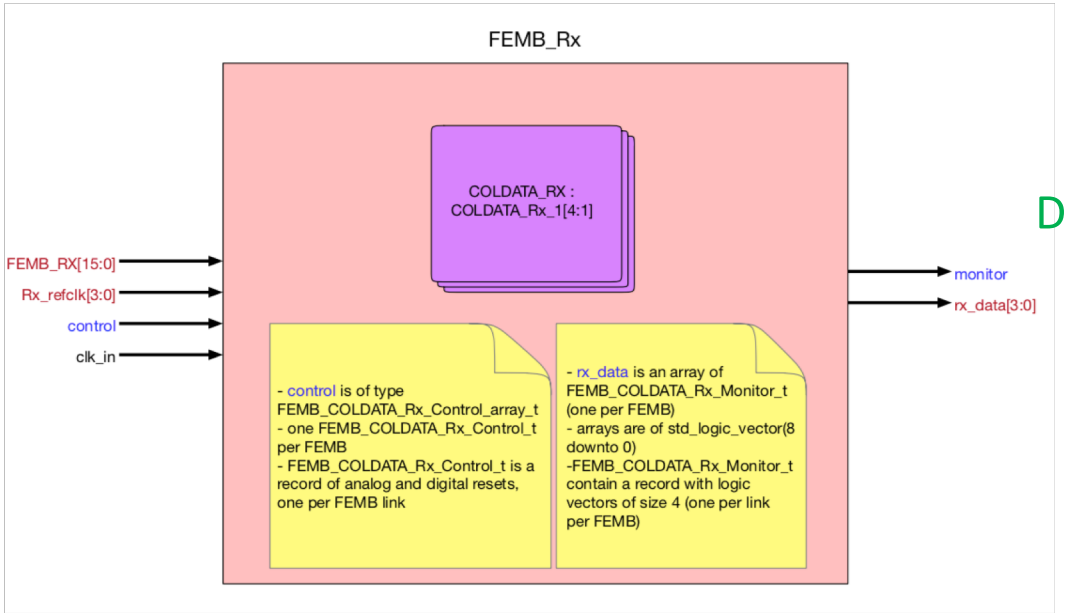
“Process Blocks”

- Self-correction for known and tractable error states
- Reporting of corrections to WIB DQM (ultimately to DAQ CCM)
- Execution of initialization and configuration of FEMBs
- Execution of configuration and start for calibrations
- Readback of calibration constants (for ADCs)
- Execution of synchronization and re-synchronization process

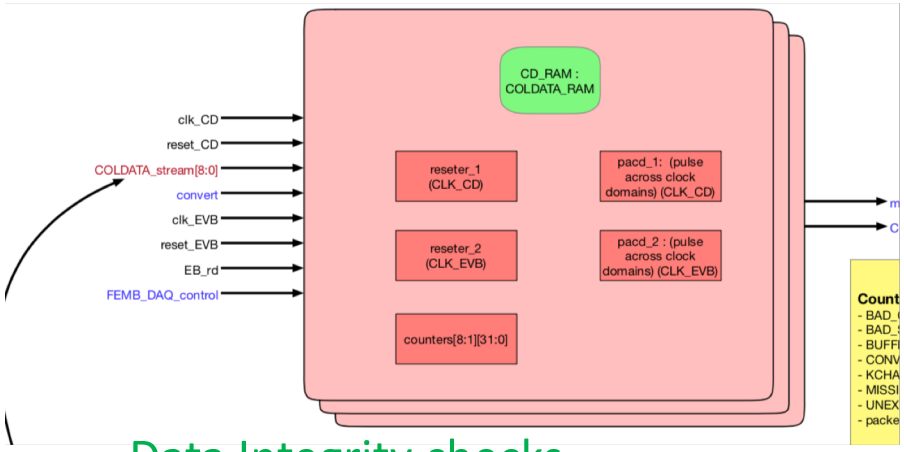
Green=all firmware
Orange=firmware+software
Blue=all software

ProtoDUNE equivalents

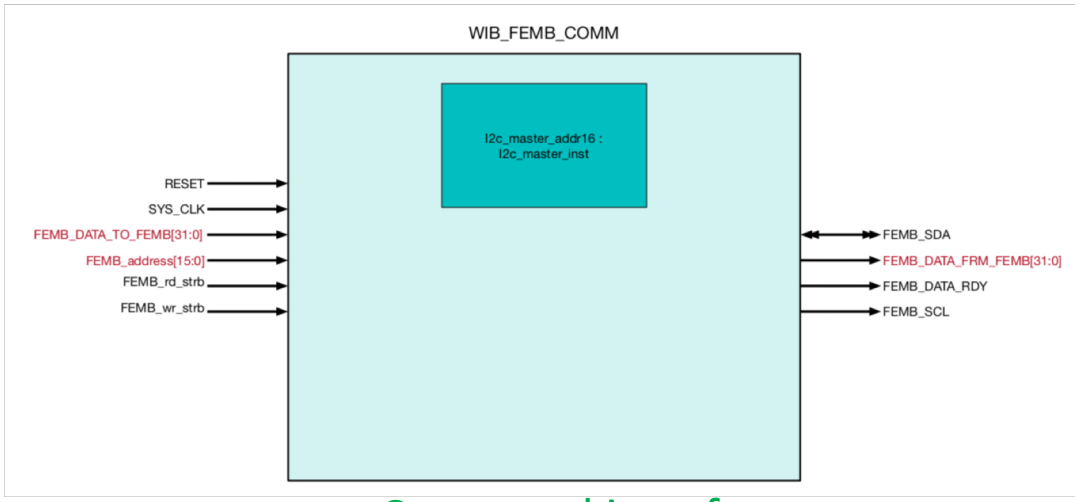
# FEMB Block




Data Reception




Data Integrity checks

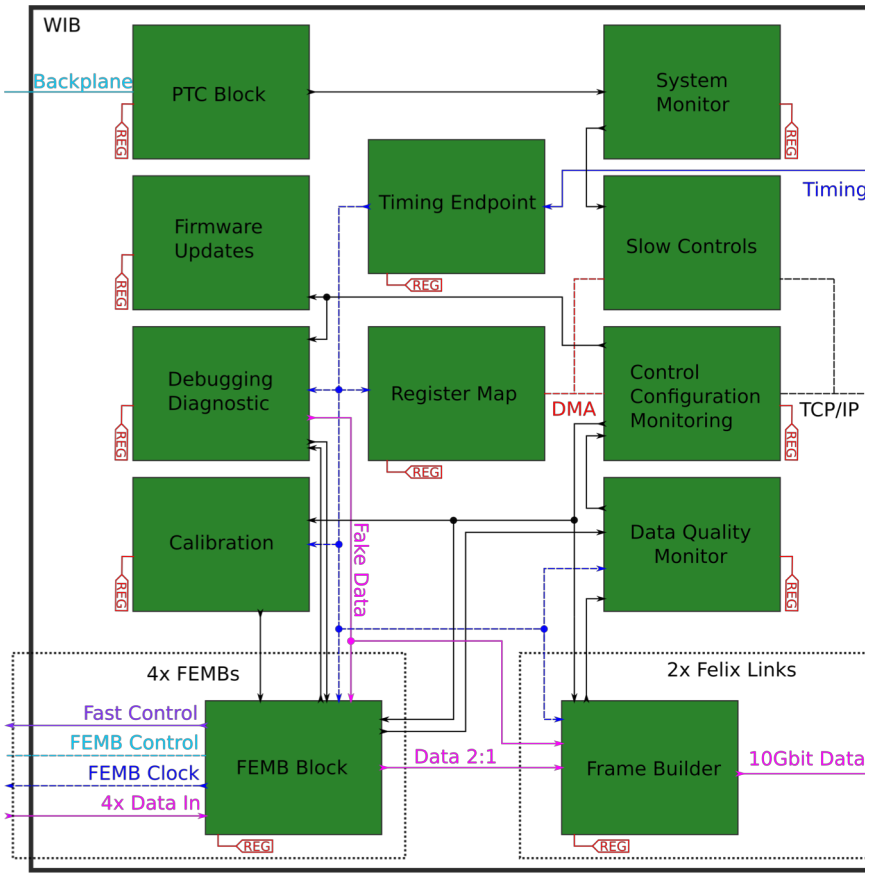


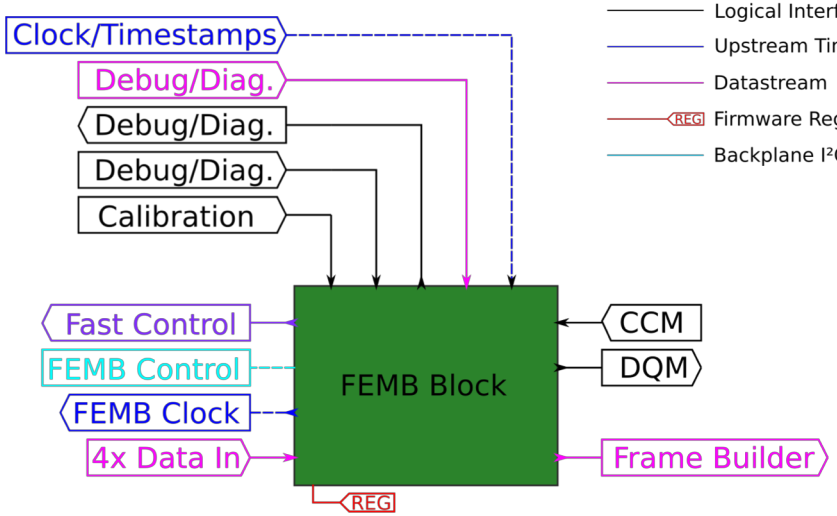

Command interface

# FEMB Block

Interfaces:

- Cold Electronics
  - Control (COLDATA,CRYO) -- bi
  - Clock (62.5 MHz) -- out
  - Data (x4 FEMBs) -- in
- Timing
  - Clock and 16 bit timestamp – in
- CCM
  - Configuration(s) – in
  - Commands – in
- DQM
  - Error conditions – out
  - Metrics – out
- Frame Builder
  - Merged data – out
- Debugging/Diagnostic
  - Streamed data – out
  - Fake data – in
- Calibrations
  - Configurations – in
  - Commands – in



# CCM Block

Responsible for DAQ-CCM interaction and on-board “CCM”

CCM Block functionality:

- Interface with DAQ CCM
- Initiates WIB and FEMB configurations
- Initiates DAQ-sourced resynchronizations
- Initiates DAQ-sourced error corrections
- Reports error states to DAQ CCM
- Provides status access to DAQ CCM
- Initiates self-corrections for simple and known failure modes
- Include a command buffer that allows complex sets of commands to be executed

This is the most complex interface with DAQ

Move into PS means this will be all-new code

Green=all firmware

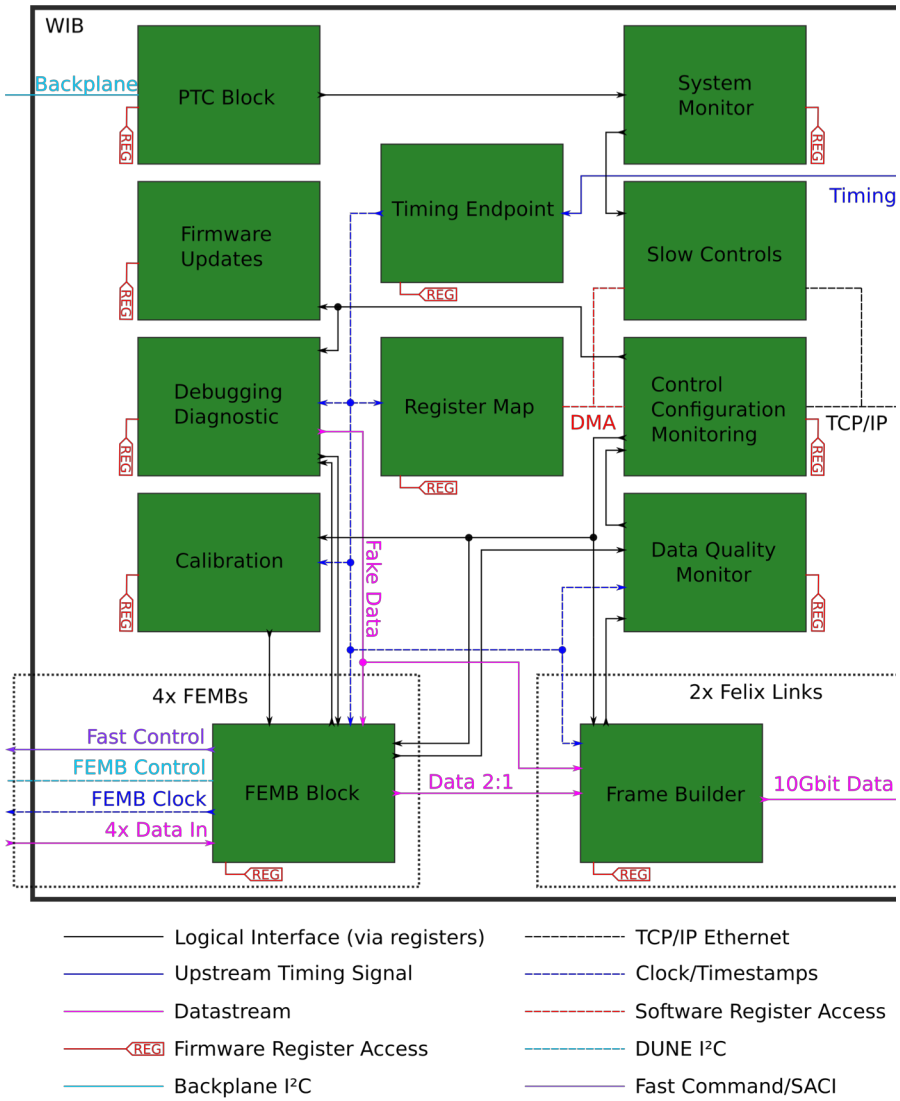
Orange=firmware+software

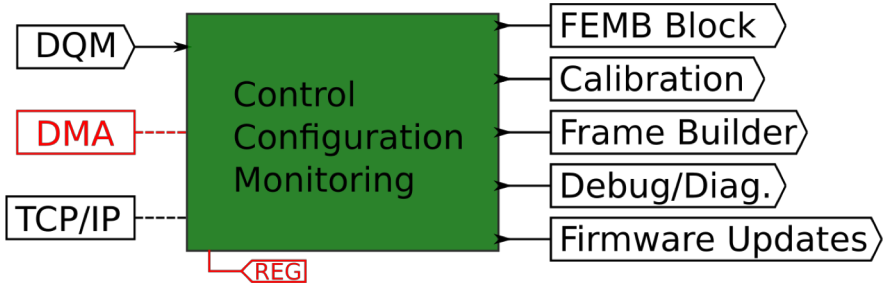
Blue=all software

# CCM Block

Interfaces:

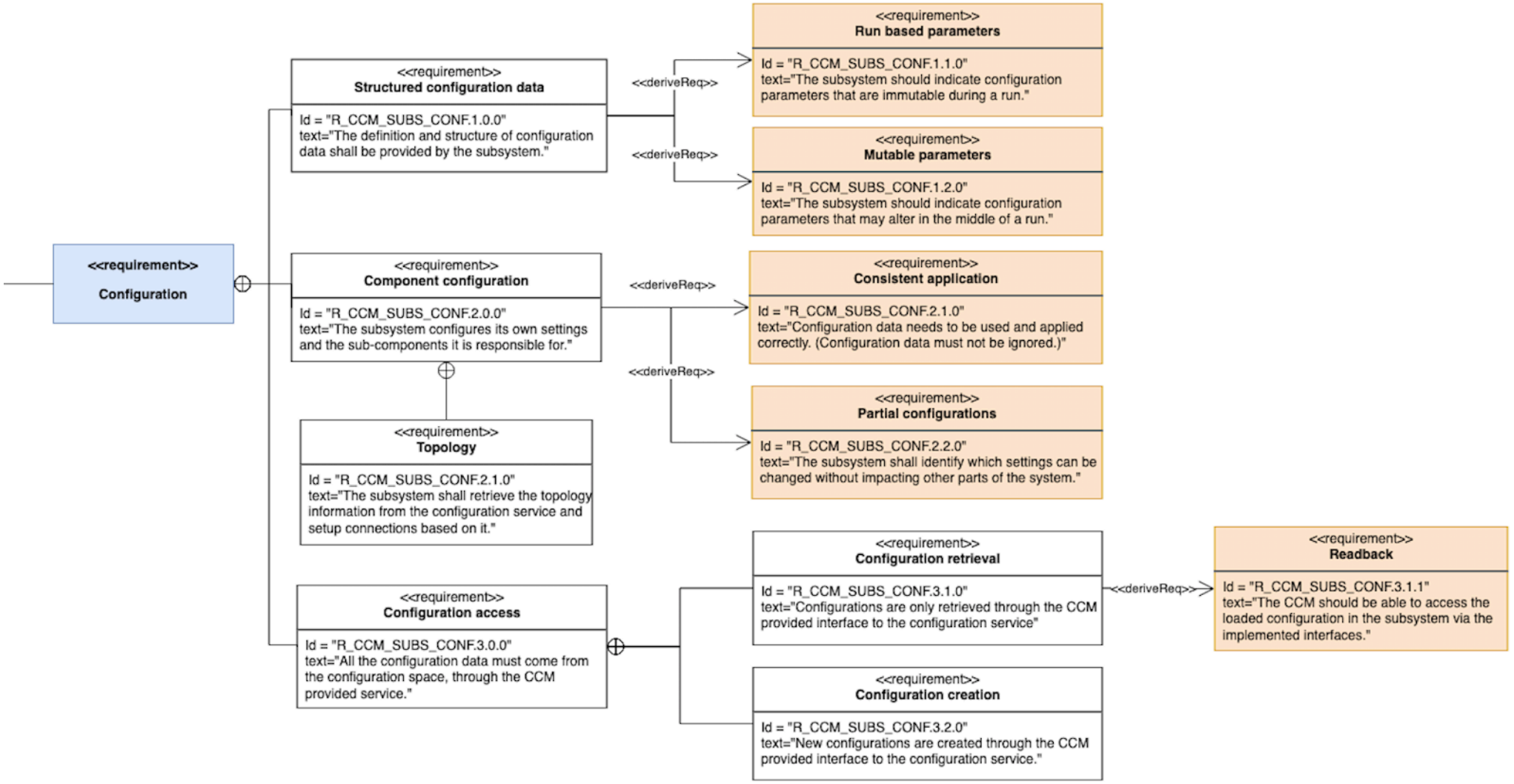
- FEMB
  - Control (COLDATA,CRYO) – bi
  - Status registers – in
  - Configuration params – out
  - Error correction commands – out
- Calibration
  - Configuration params – out
  - Timestamp start – out
  - Handshake start – in
  - Status – in
- Debugging/Diagnostic
  - Mode – out
  - Status – in
- DQM
  - Error states – in
- Frame Builder
  - Enables – out
- Firmware Updates



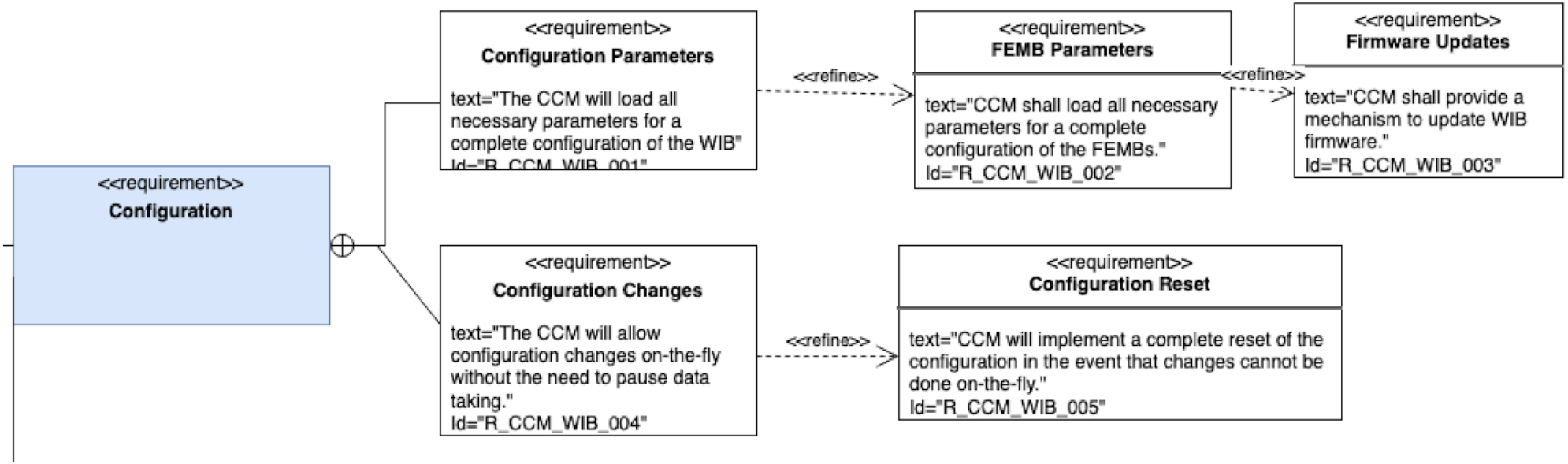
# CCM Block

Requirements diagram from DAQ for Configuration: CCM to WIB



# CCM Block

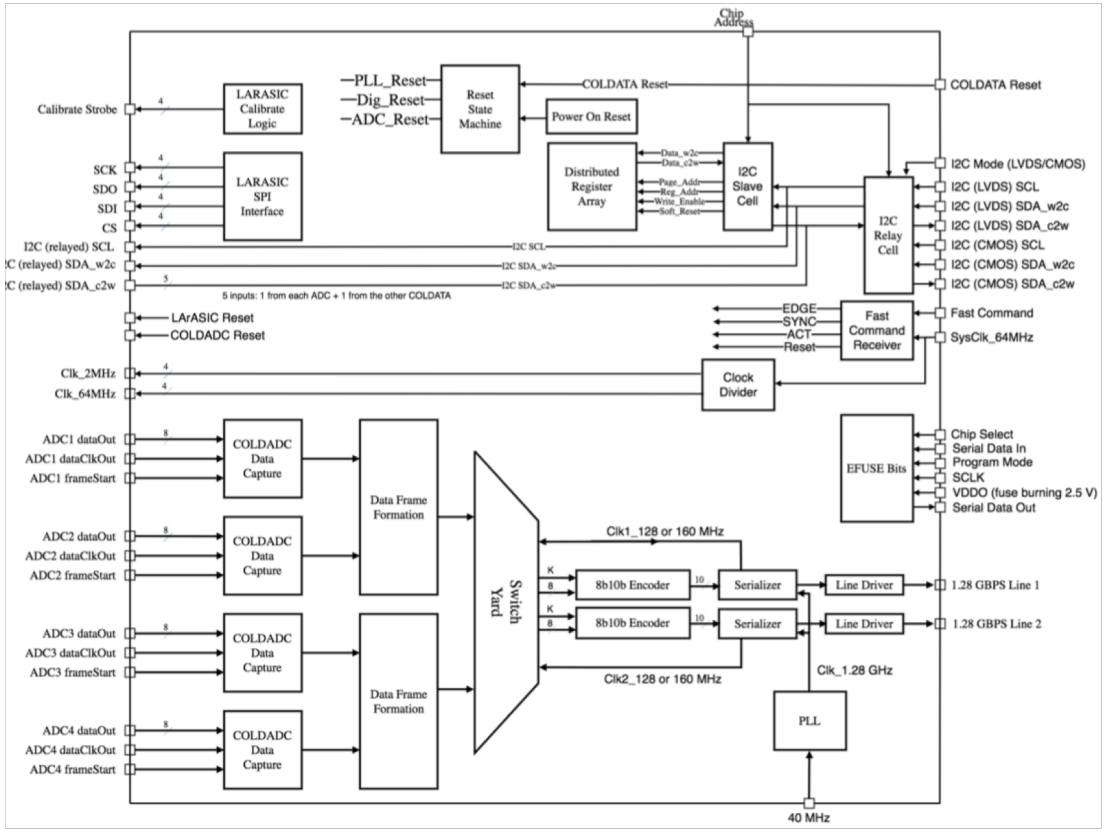
Requirements diagram from CE for Configuration: WIB to CCM



# CCM Block

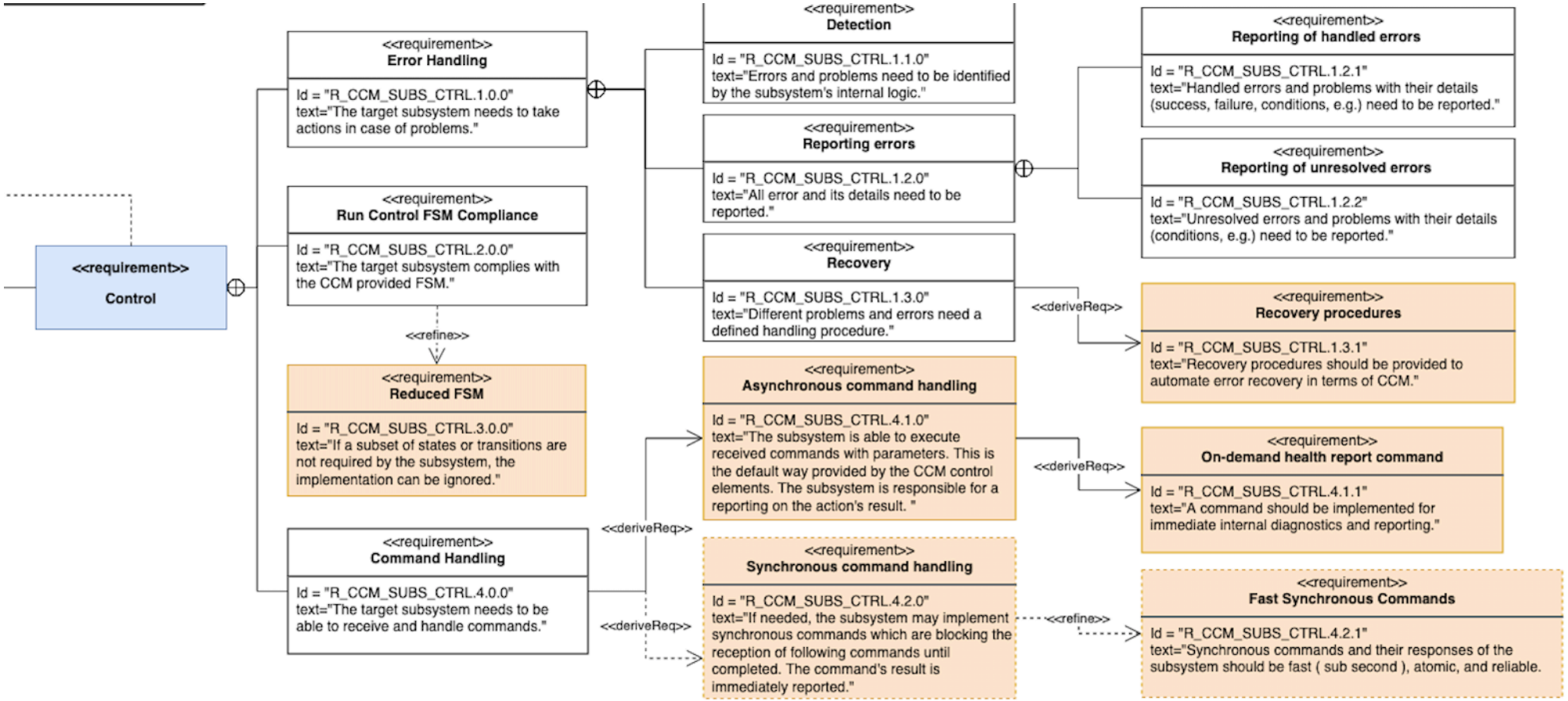
WIB configuration parameters:

- Map of enabled channels/ASICs/FEMBs – Can change during runs
- Gain and shaper settings on FEMB – "immutable" during a run
- Ordinal number of 62.5 MHz clock tick for FEMB sampling clock alignment (COLDATA)
- Data start
- Many other COLDATA/LArASIC parameters
- Calibration parameters – Can change during runs
  - Number of DAC settings
  - DAC intervals
  - Number of pulses/setting
  - Start time



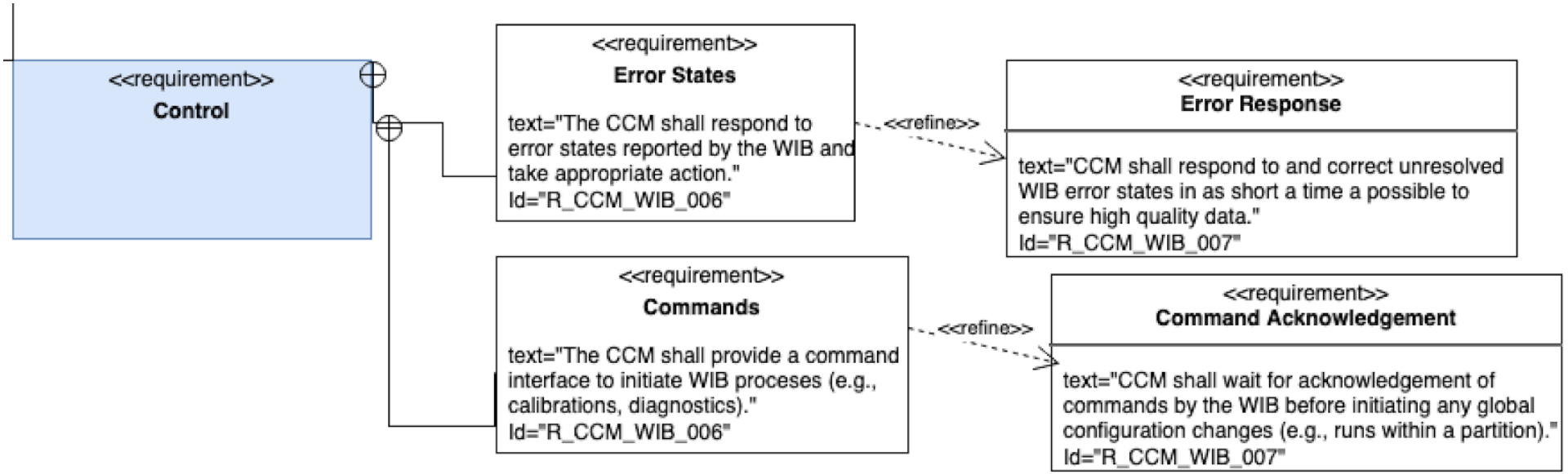
# CCM Block

Requirements diagram from DAQ for Control:  CCM to WIB



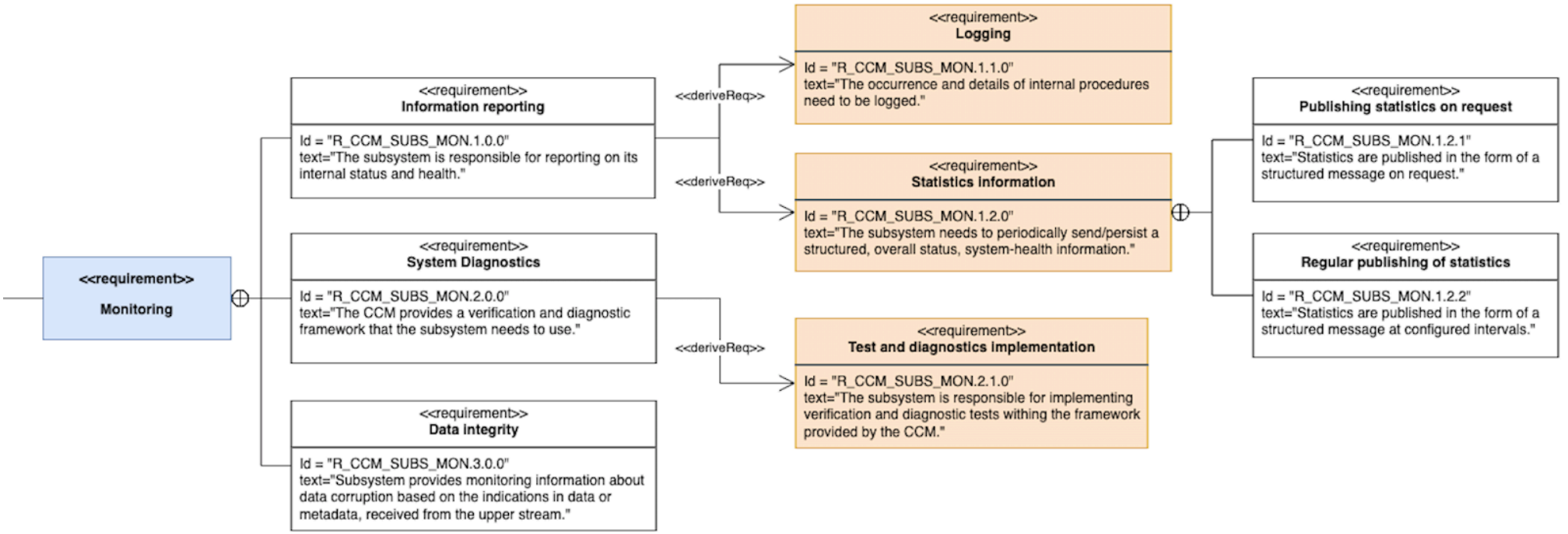
# CCM Block

Requirements diagram from CE for  Configuration:  WIB to CCM



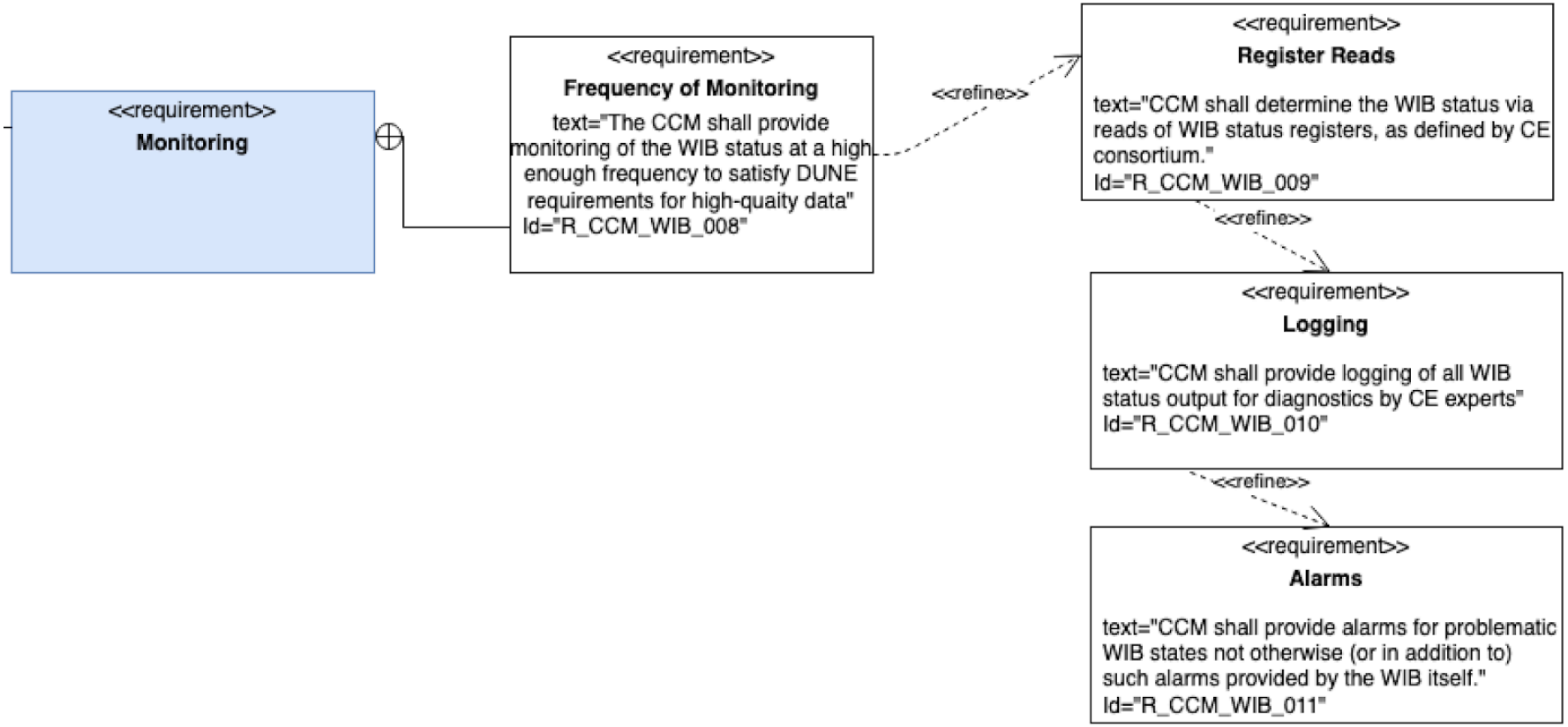
# CCM Block

Requirements diagram from DAQ for Monitoring:  CCM to WIB



# CCM Block

Requirements diagram from CE for Configuration: WIB to CCM




On-board monitoring is done through WIB DQM, errors reported to CCM

Environmental monitoring done through system monitor read by SC

Interlocks come through PTC interface

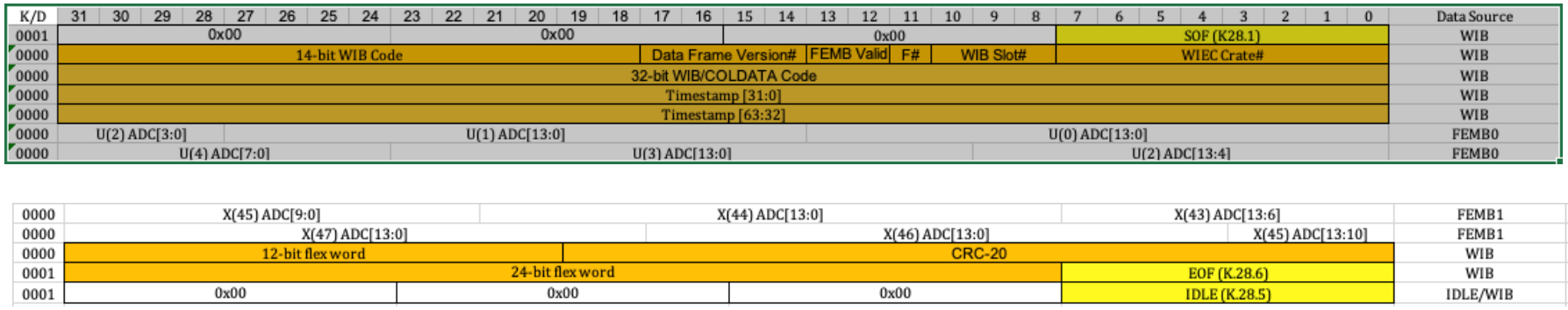
# Frame Builder Block

Responsible for Packaging of each frame to be sent to FELIX---not dramatically different from ProtoDUNE “Event Builder”

Green=all firmware
Orange=firmware+software
Blue=all software

Frame Builder Block functionality:

- Reception of data from FEMB block
- Addition of status and error bits to frame
- Addition of geographic bits to frame (e.g., WIEC number)
- Expansion of timestamps to 64 bits
- Addition of formatting characters for FELIX (Interface document)
- Additional data integrity checks
- Transmission of data to FELIX



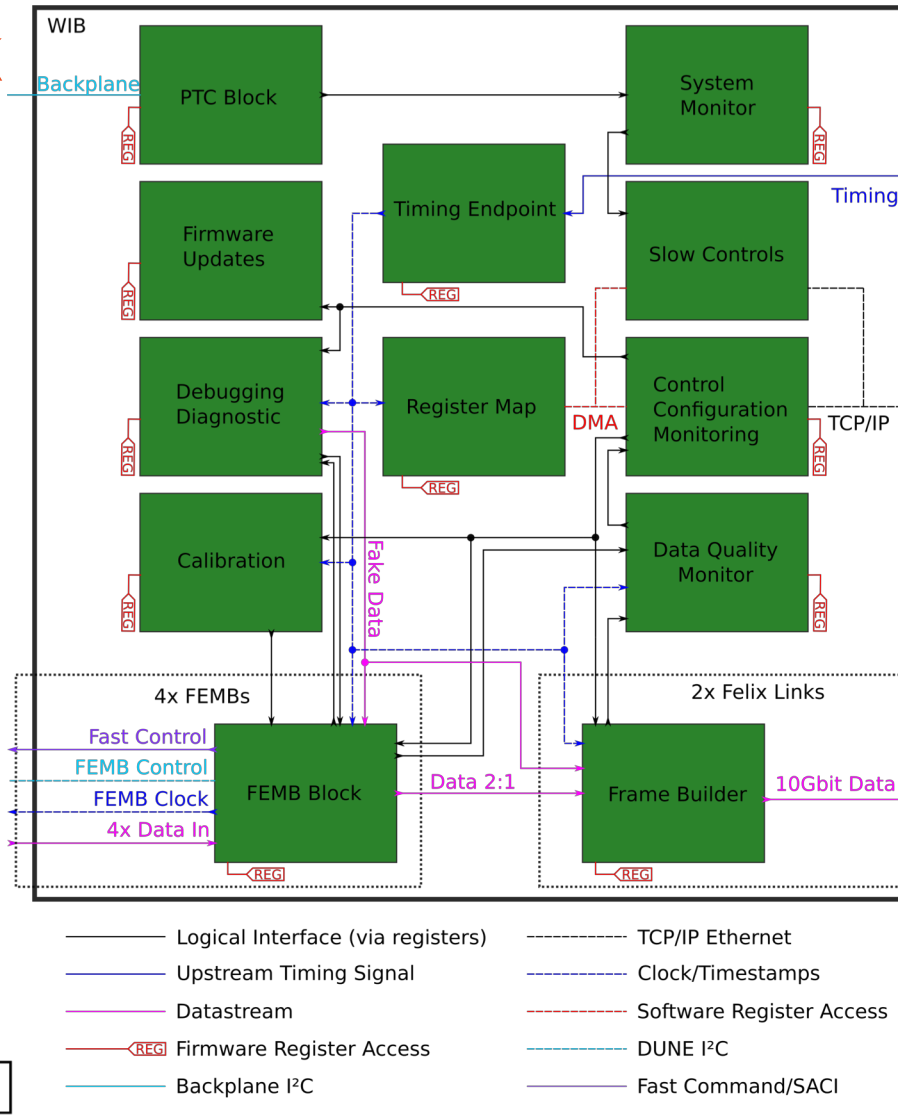
| K/D | 31 30 29 28 27 26 25 24 | 23 22 21 20 19 18 17 16 | 15 14 13 12 11 10 9 8 | 7 6 5 4 3 2 1 0 | Data Source |
|---|---|---|---|---|---|
| 0001 | 0x00 | 0x00 | 0x00 | SOF (K28.1) | WIB |
| 0000 | 14-bit WIB Code | Data Frame Version# | FEMB Valid, F#, WIB Slot# | WIEC Crate# | WIB |
| 0000 | 32-bit WIB/COLDATA Code | | | | WIB |
| 0000 | Timestamp [31:0] | | | | WIB |
| 0000 | Timestamp [63:32] | | | | WIB |
| 0000 | U(2) ADC[3:0] | U(1) ADC[13:0] | U(0) ADC[13:0] | | FEMB0 |
| 0000 | U(4) ADC[7:0] | U(3) ADC[13:0] | U(2) ADC[13:4] | | FEMB0 |

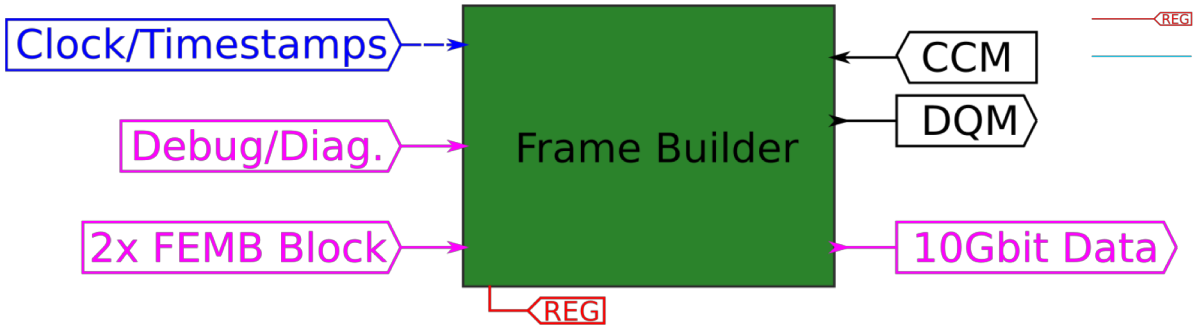
| | | | | |
|---|---|---|---|---|
| 0000 | X(45) ADC[9:0] | X(44) ADC[13:0] | X(43) ADC[13:6] | FEMB1 |
| 0000 | X(47) ADC[13:0] | X(46) ADC[13:0] | X(45) ADC[13:10] | FEMB1 |
| 0000 | 12-bit flex word | CRC-20 | | WIB |
| 0001 | 24-bit flex word | | EOF (K28.6) | WIB |
| 0001 | 0x00 | 0x00 | 0x00 | IDLE (K28.5) | IDLE/WIB |

# Frame Builder Block

Interfaces:

- FEMB
  - Data – in
- Debugging/Diagnostic
  - Fake data – in
- DQM
  - Data integrity metrics – out
- CCM
  - Enables – in
- Timing
  - 64-bit timestamps – in
- Register Map
  - Status and error bits -- in



# Frame Builder Block

Examples of error bits, addresses, tags:

- Map of enabled channels = 9 bits
- WIEC Crate Number = 8 bits
- Calibration flags
  - Which FEMB(s) are sending calibration data (4 bits)
  - Which DAC setting frame corresponds to  (4 bits ?)
- Diagnostic flags (4 bits)
  - FEMBs which DAQ should stream (and ignore for hit-finding purposes)
- Error flags
  - 64-bit timing error (1 bit)
  - 16-bit timing error (8 bits)
  - ADC error bits

As we gain experience with system, these will expand---92 available and some are “flex” bits and context dependent

# Data Quality Monitor (DQM) Block

Responsible for high-level analysis of data integrity and generating errors

DQM Block functionality:

- Reception of metrics from FEMB block
- Reception of metrics from Frame Builder block
- "Analysis" of metrics and determination of error conditions
- Error condition transmission to CCM for on-board correction or DAQ correction
- Transmission of high-level diagnostics to CCM for logging and analysis

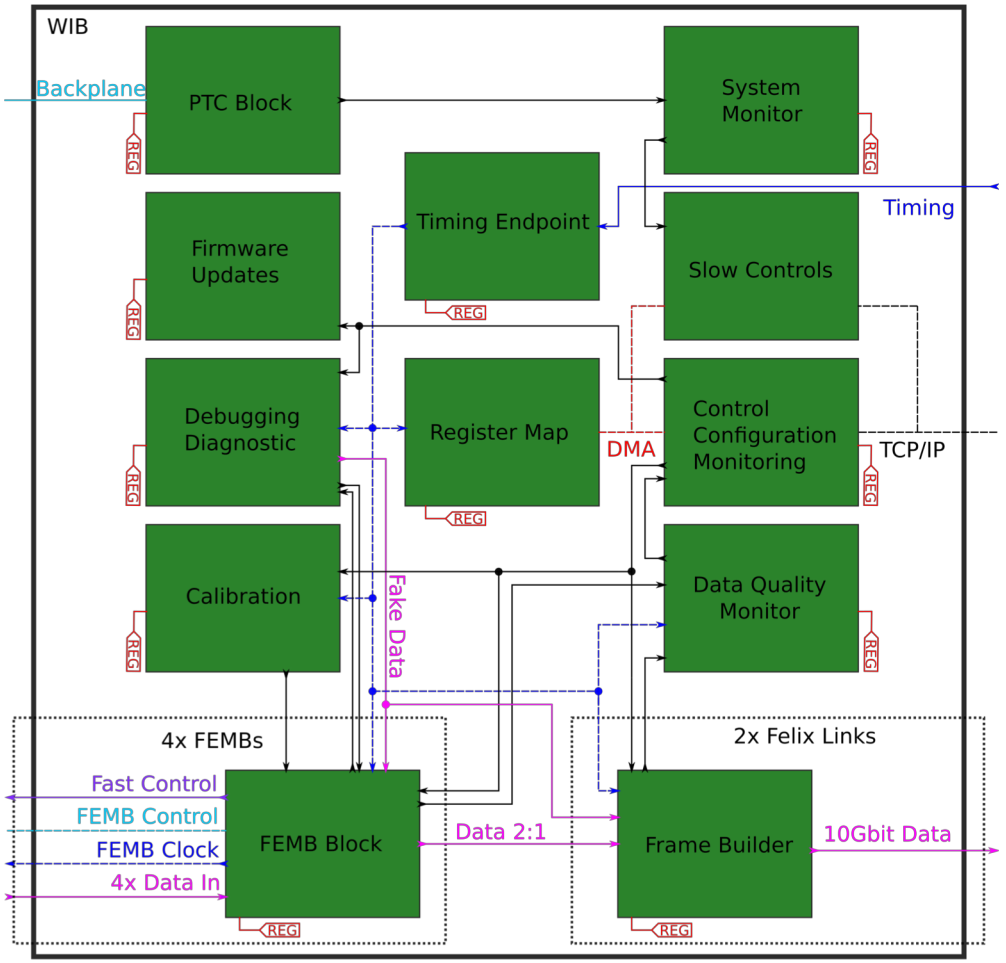
Green=all firmware
Orange=firmware+software
Blue=all software

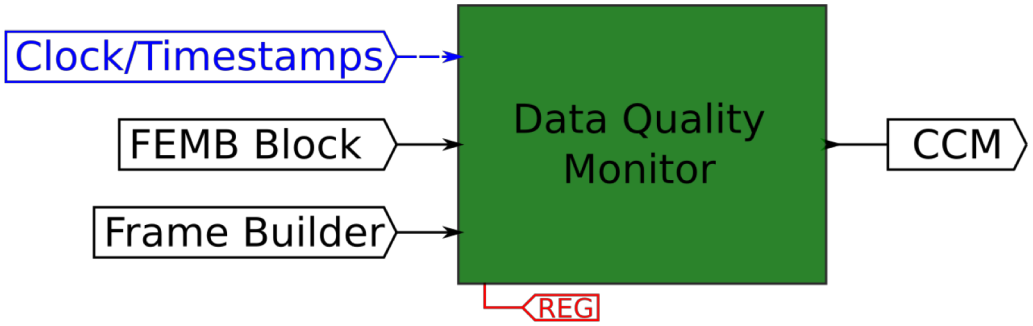
DQM takes no action itself; it informs CCM which may take action
This expands some of the ProtoDUNE "CD Stream Processor" functionality

# Data Quality Monitor (DQM)

Interfaces:

- FEMB
  - Metrics – in
  - Errors – in
- Debugging/Diagnostic
  - Streamed data – in
- Frame Builder
  - Data integrity metrics – in
  - Errors – in
- CCM
  - Errors – out



# DQM Block

Input Metrics and Errors:

- FEMB block timestamp error flags
- Channel/FEMB ADC running sums
- Channel/FEMB ADC rms
- Frame builder timestamp errors

Outputs:

- 16-bit FEMB timestamp corruption
- Sampling clock error
- ADC flatlined
- ADC baseline out of spec
- Diagnostic histograms

As we gain experience with system, these will very likely expand

# Calibration Block

Responsible for running FE calibrations

Calibration Block functionality:

- Initiation of FE ADC calibrations
- Initiation of FE pulser calibrations
- Generates calibration flags to be included by Frame Builder
- Provides acknowledgment/handshake to CCM on when calibrations have ended

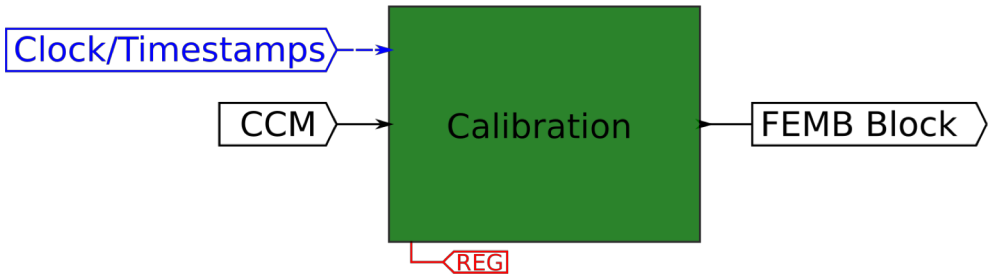
This replaces functionality that used to be handled by DAQ

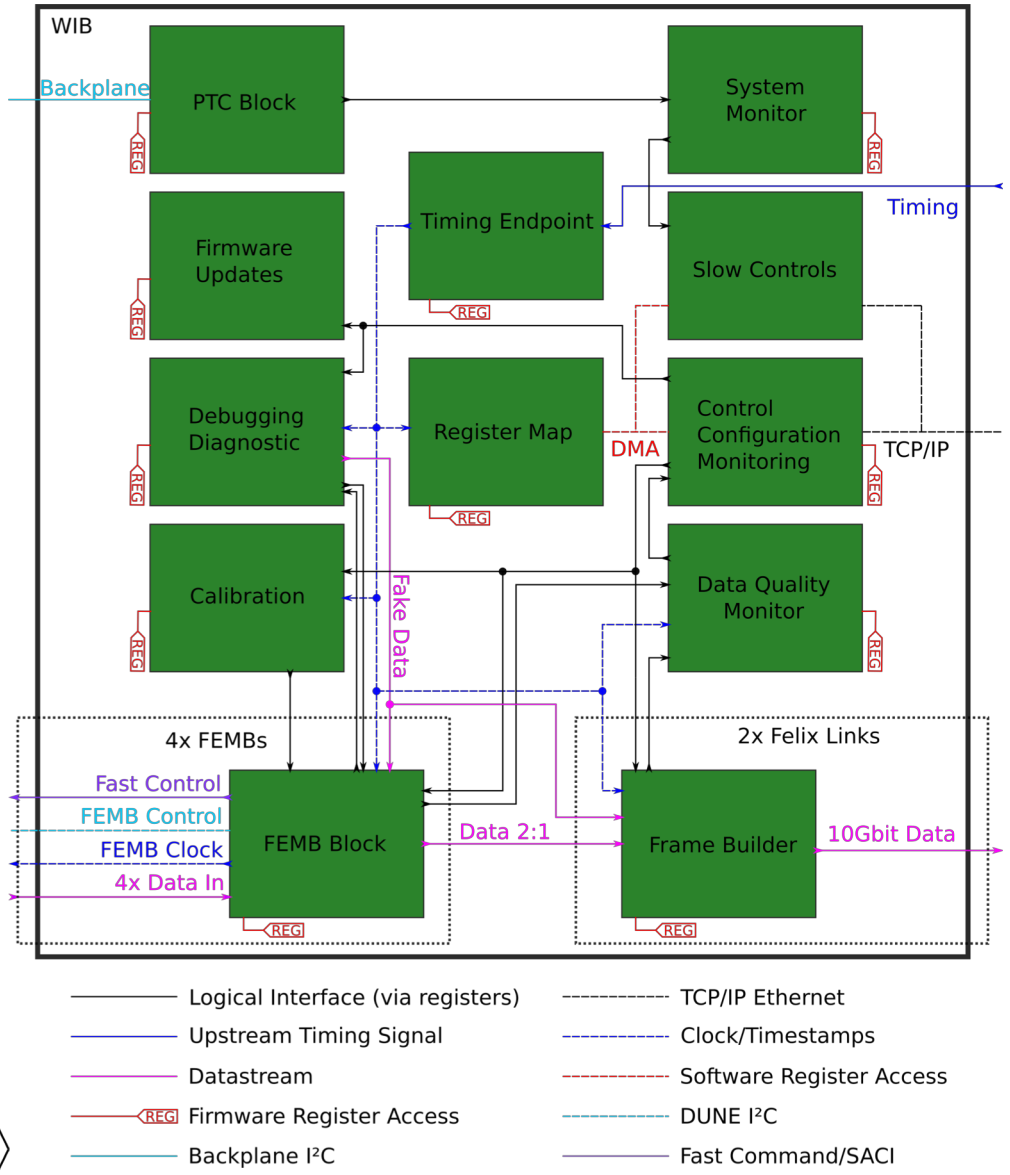
Green=all firmware
Orange=firmware+software
Blue=all software

# Calibration Block Interfaces

Interfaces:

- FEMB
  - Configuration – out
  - Pulser start command – out
  - DAC setting change  -- out
  - ADC self-calibration start – out
- CCM
  - Configuration parameters – in
  - Start time – in
  - Calibration end – out
  - Calibration ongoing – out
- Frame Builder (via registers)
  - Calibration status – out
- Timing
  - 64-bit timestamp -- in



# Debugging/Diagnostic Block

Responsible for providing experts with ways of understanding problems

Debugging/Diagnostic Block functionality:

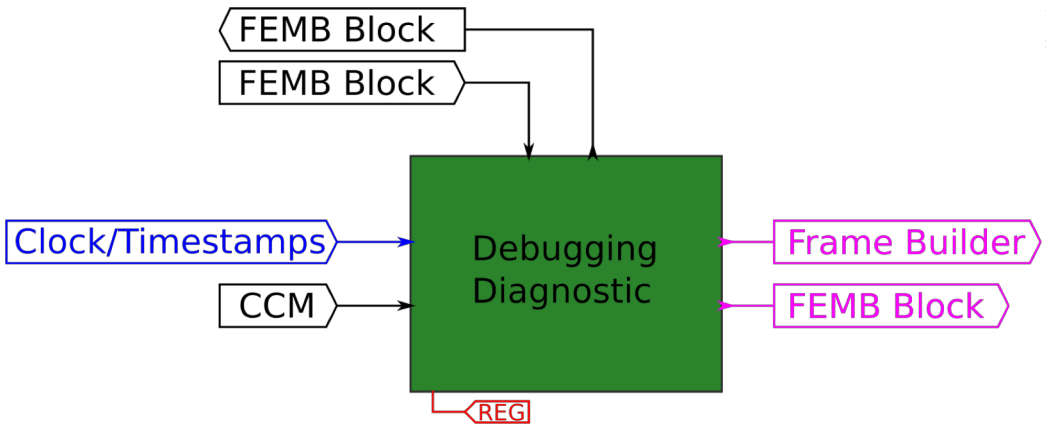
- Generates "fake data" for FEMB and Frame Builder blocks
- Handles local "streaming" mode into memory for testing
- Allows direct register reads/writes on WIB and FEMBs

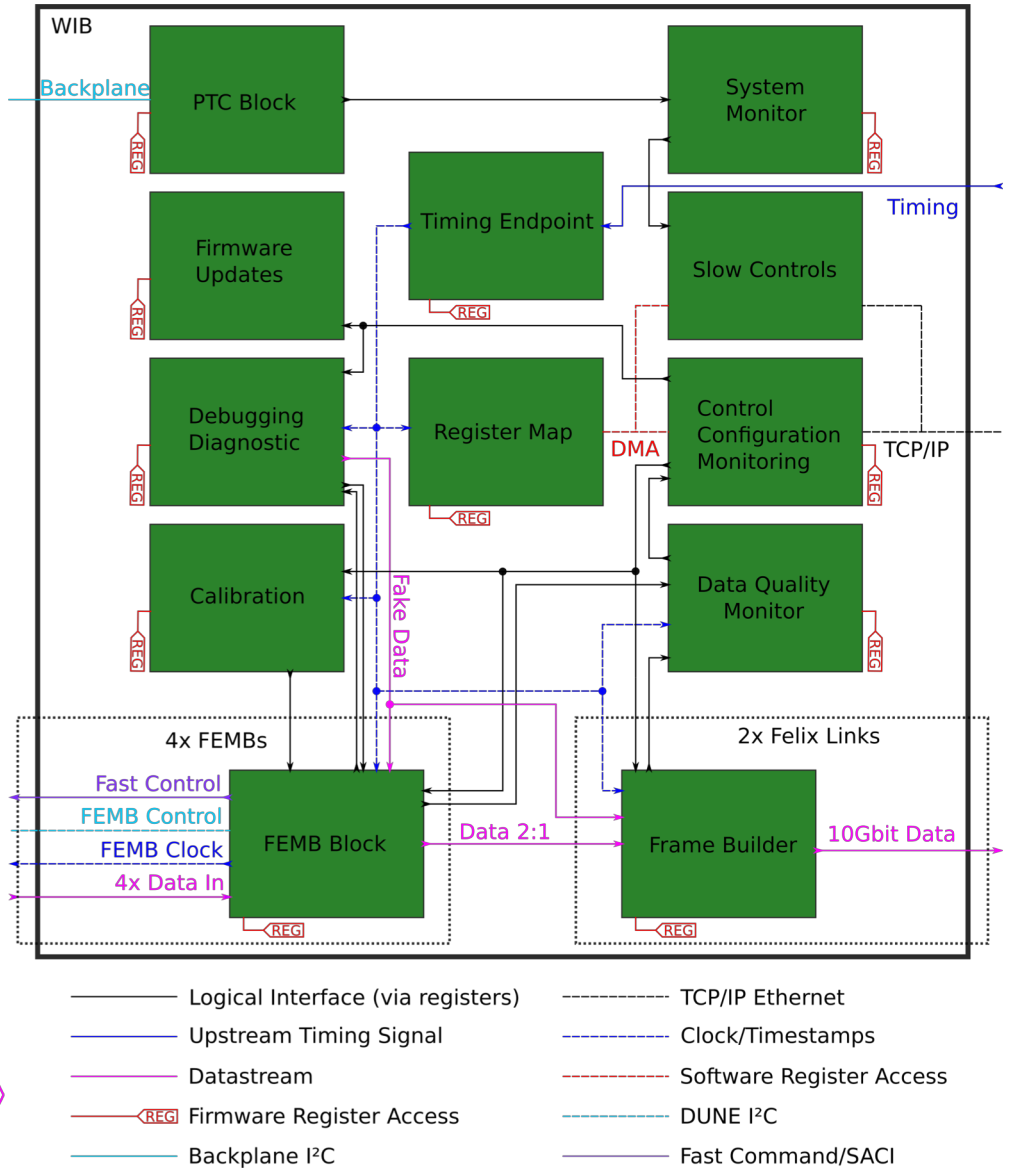
Green=all firmware
Orange=firmware+software
Blue=all software

# Diagnostic/Debugging Block Interfaces

Interfaces:

- FEMB
  - Streamed data – in
  - Streamed data enables – out
  - Fake data – out
- Frame Builder
  - FELIX “echo” – bi
  - Fake data – out
- Timing
  - 64-bit timestamps for fake data – in



# Slow Controls Block

Responsible for interface with DAQ/SC

Slow Controls Block functionality:

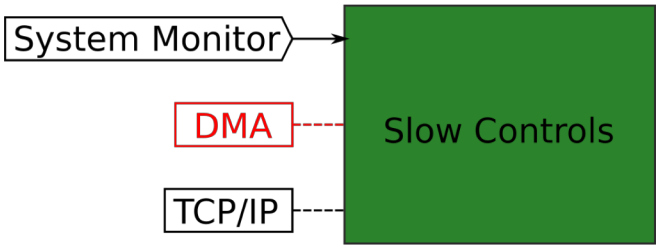
- Soft resets of system that runs on ZYNQ
- Power cycles for WIB independently from other WIBs
- Non-interlocked environmental monitoring (voltages, temperatures, currents)
- Transmission of monitoring results to DAQ

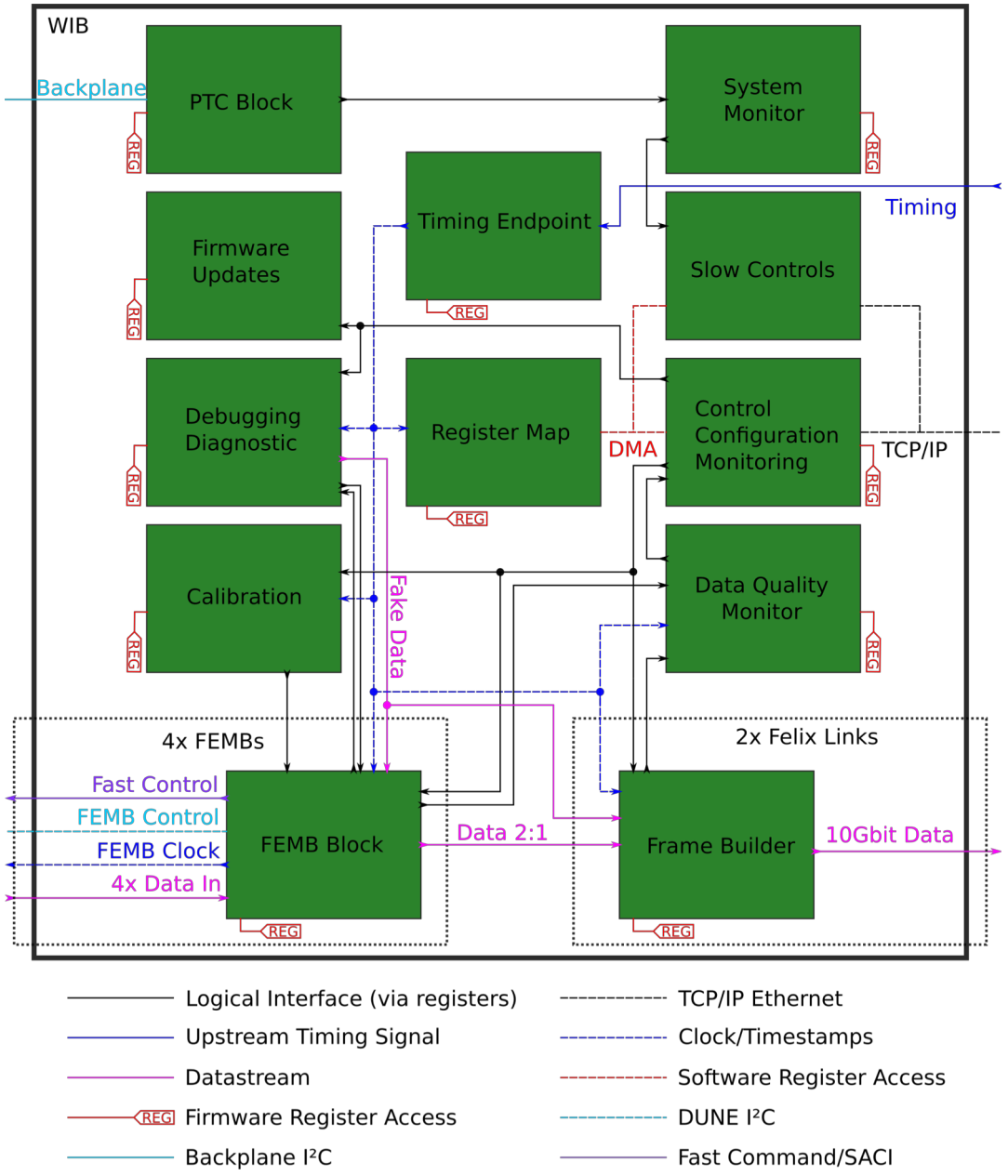
Green=all firmware
Orange=firmware+software
Blue=all software

# Slow Controls Block Interfaces

Interfaces:

- System Monitor/Control
  - Hardware values – in
  - Power control – out



# Timing Block

Responsible for interface with DAQ Timing System

Timing Block functionality:

- Provides 62.5 MHz clock for distribution to FEMBs
- Provides 64-bit DUNE timestamp, synchronized globally

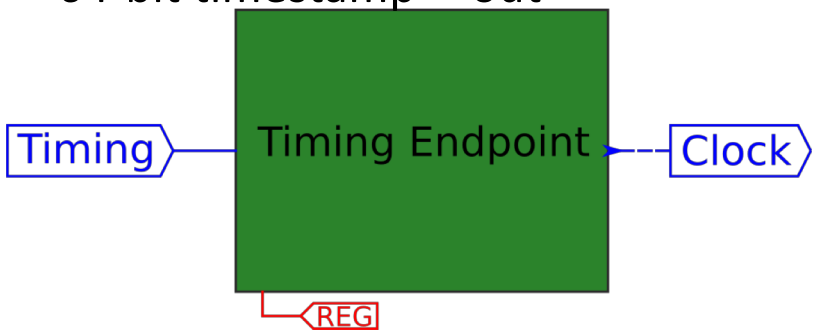
Green=all firmware
Orange=firmware+software
Blue=all software

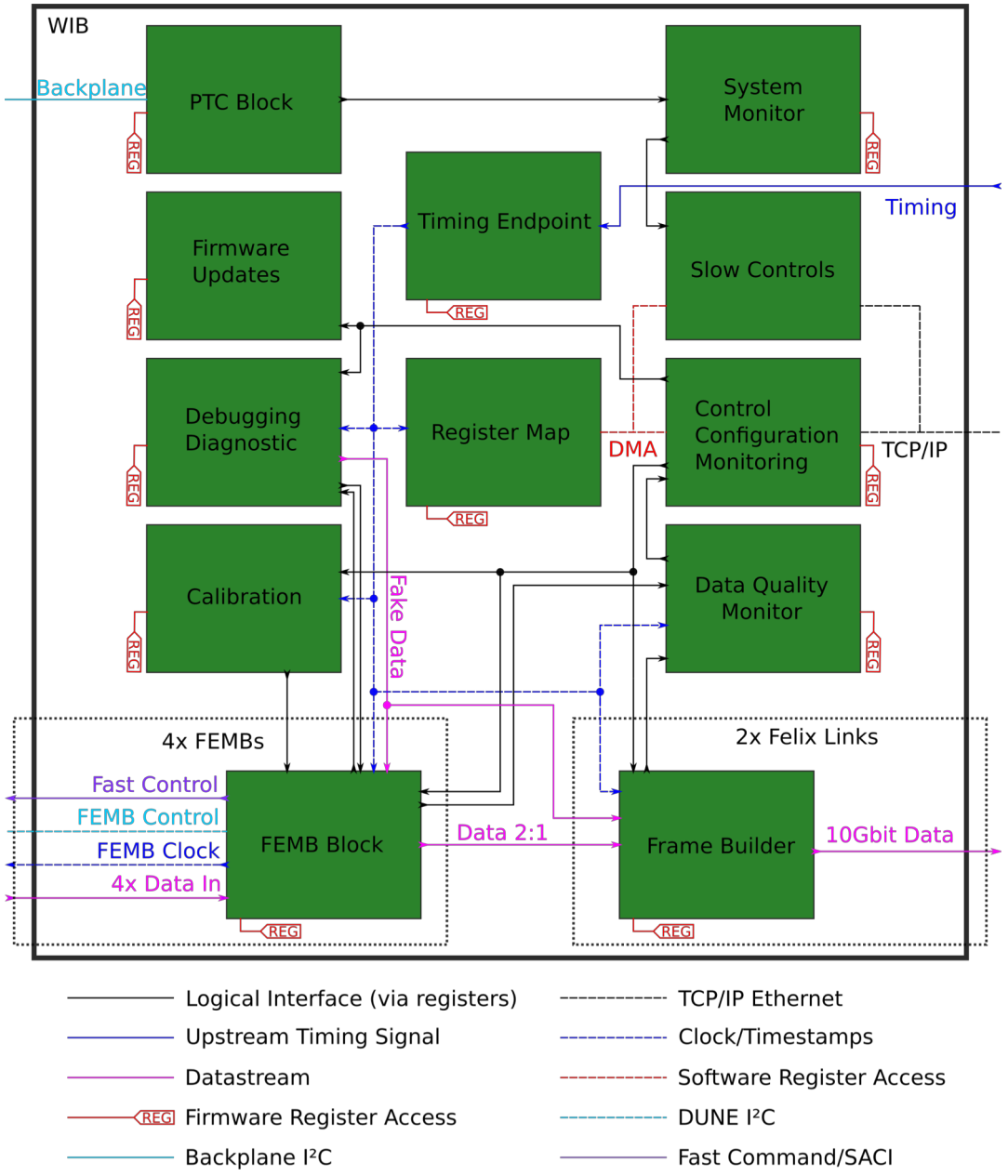
The IP block for this is expected to come from DAQ

# Timing Block Interfaces

Interfaces:

- DUNE timing system
  - Clock – in
  - Timing codes – in
  - Echo for delay measure – out
  - Error codes – out
- FEMB Block
  - 62.5 MHz clock – out
  - 16-bit timestamp – out
- Frame Builder
  - 64-bit timestamp – out
- Debugging/Diagnostic
  - 16/64-bit timestamps – out
- Calibration
  - 64-bit timestamp -- out



# Firmware Updates Block

Responsible for handling changes to firmware when needed

Firmware Updates functionality:

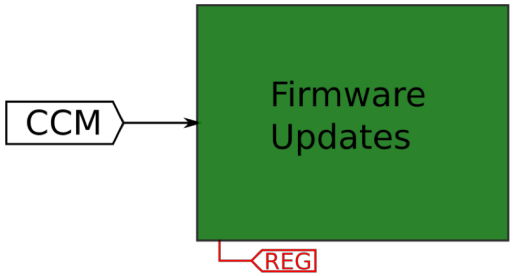
- Via flash memory
- Via RAM image

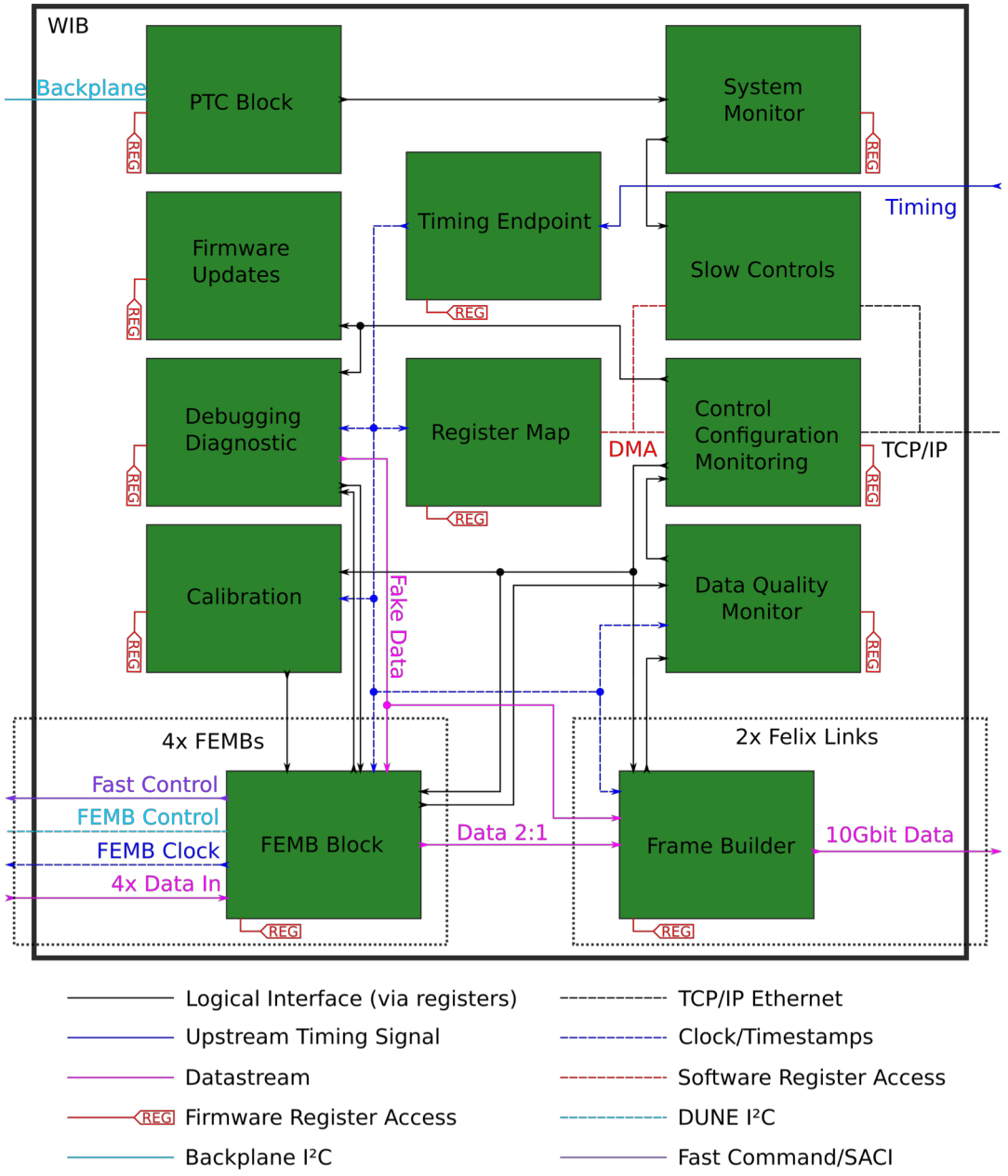
Green=all firmware
Orange=firmware+software
Blue=all software

# Firmware Updates Block Interfaces

Interfaces:

- CCM
  - New firmware – in



# PTC Interface Block

Responsible for I2C interface with PTC

PTC functionality:

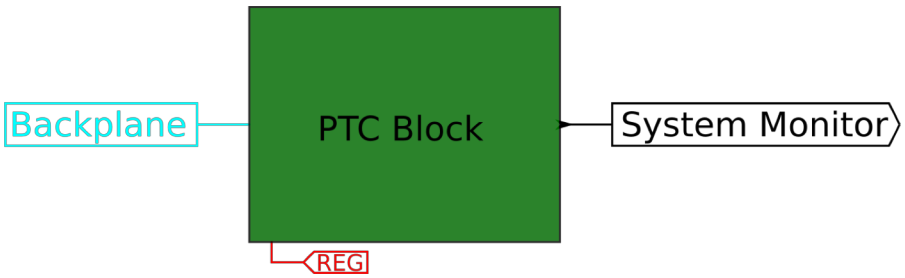
- Handling of hardware interlocks

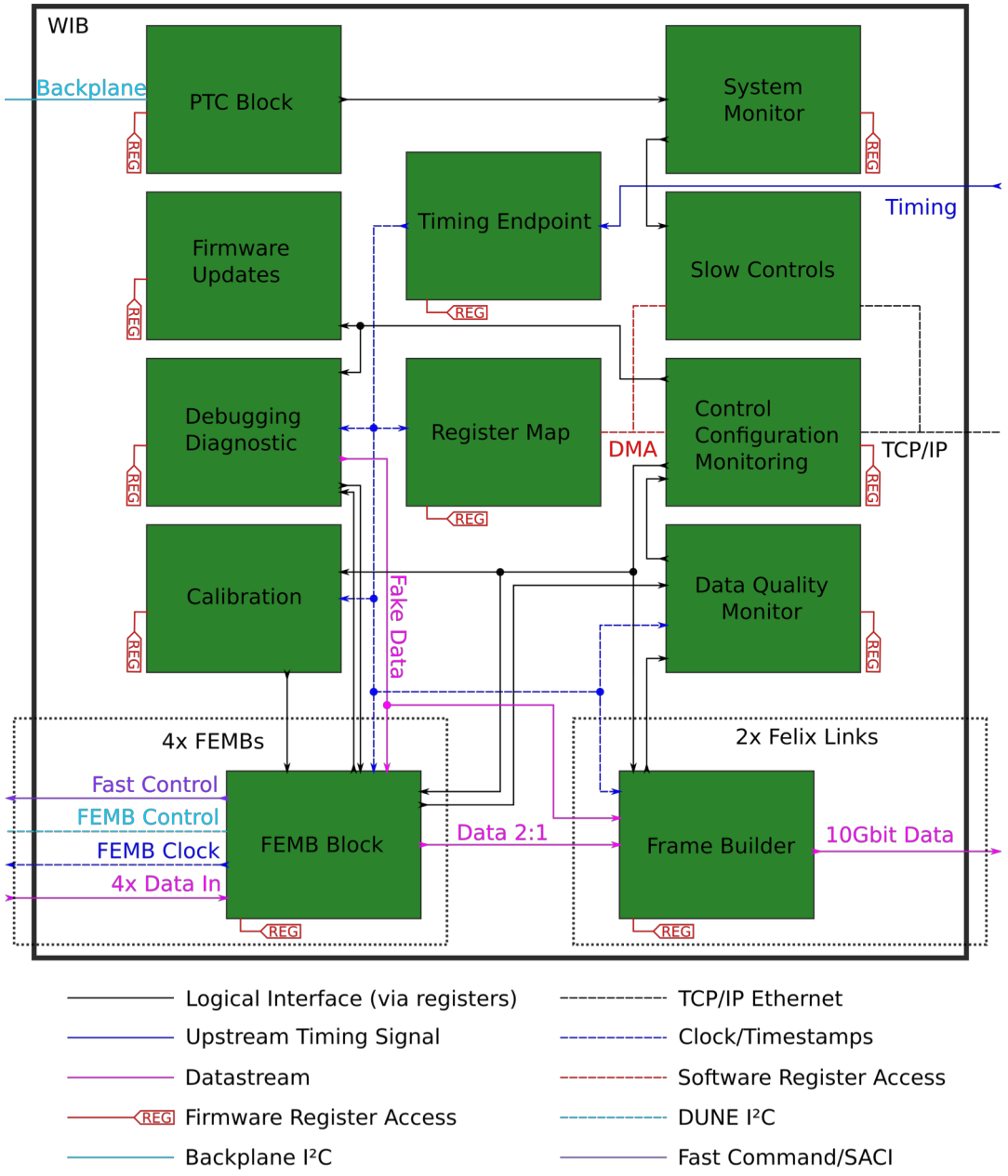
Green=all firmware
Orange=firmware+software
Blue=all software

# PTC Block Interfaces

Interfaces:

- Backplane/PTC
  - Interlock messages – in
- System Monitor
  - On-board environment – in
  - Interlock control -- out



Pulser Calibrations/COLDATA

# Use Case

- CCM writes calibration configuration parameters to Calibration block (or default)
  - Number of DAC settings
  - DAC intervals
  - Number of pulses/setting
  - Rising edge time
  - Time between rising edges
  - Which FEMB(s) will be calibrated
- CCM sends timestamp (> 1 s in the future) for start of calibrations

While (iDAC_intervals < nDAC_intervals)

1. Calibration block writes configuration parameters via I2C-like interface in FEMB block to FE
2. At timestamp start, Calibration block writes ACT signal via Fast Command in FEMB block, putting COLDATA in calibration mode
3. Calibration block sets calibration bit for affected FEMBs
4. Frame Builder adds calibration bits to output frames
5. Calibration block tells CCM that calibrations have started

While (timenow-timestamp_start < cal_pulse_period*nPulses) wait;

- Calibration block sends next calibrate command taking COLDATA out of calibration mode;

Return(to 1);

- Calibration block tells CCM it is done
- Calibration block unsets status bit for affected FEMBs

# Summary

- High-level requirements for WIB firmware/software have been written
- Design is evolving rapidly---primary blocks have been identified
- Major interfaces have been identified
- A subset of ProtoDUNE firmware can be used
- Still need decisions on line between firmware and software on WIB