ARGONNE NATIONAL LABORATORY
HEAVY ION DISCUSSION GROUP

Argonne NATIONAL LABORATORY

GAMMASPHERE UPGRADE PROJECT OVERVIEW

SIGNAL PROCESSING AND CONTROLS

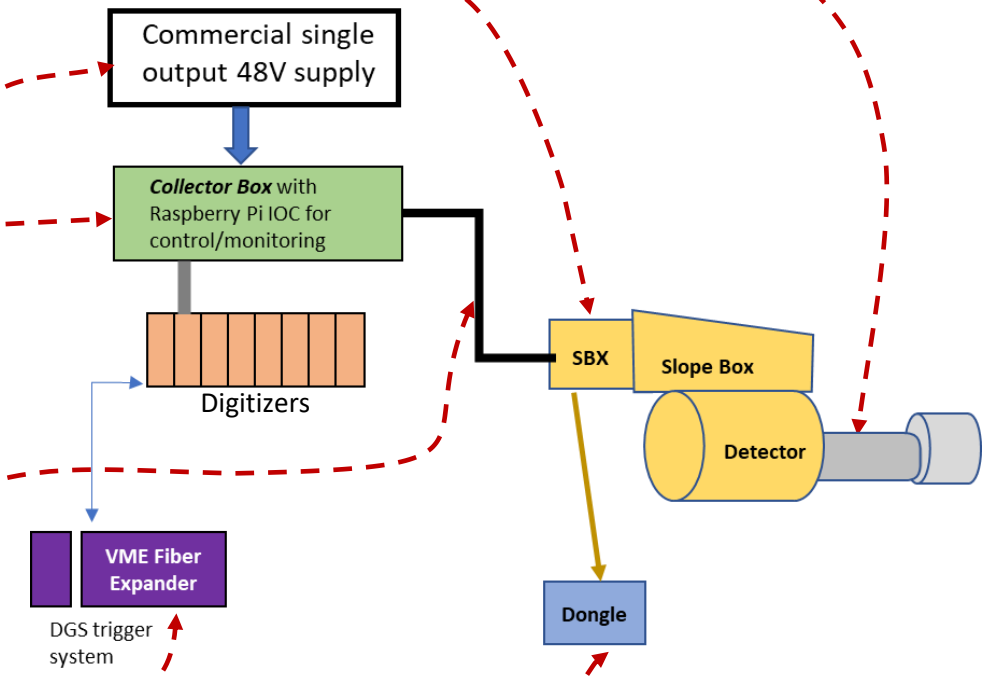
JOHN T. ANDERSON
Physics Division
Argonne National Laboratory

- ***Agenda***
- New signal processing hardware & firmware
- New Data Acquisition Software
- New Control and Monitoring system
- Tools for maintaining large control systems

December 2, 2022

Lots of new hardware – understanding the block diagram

- ***ANL Preamp***
  - replaces LBNL preamp
  - Improved operation, added features, software-controllable.
- ***Power Board of Slope Box Extension (SBX)***
  - Generates ***all*** detector voltages from single floating 48V input
  - Monitors temperature and controls cooling fan of SBX
- ***Pickoff Card of SBX***
  - Communications hub between the **Collector Box** and the detector subsystem.
  - Does all analog signal processing between detector and digitizer.
- ***Collector Box and new bulk power supply***
  - Provides 48V power distribution to up to 30 detectors
  - Provides control and monitoring fanout/fan-in for up to 30 detectors
  - Maps and reroutes analog signals from SBX pickoffs to digitizers
  - Implements power monitoring and ground fault detection
- ***DVI cables***
  - Commercial product replaces the **gray cables** of original Gammasphere.
- ***Fiber Expander***
  - Optical interface to allow distribution of trigger hardware into crates mounted around the array.
- ***"Dongle"***
  - Provides EEPROM for BGO shell data and reads back the GS# (array position) – allows Collector Box to ***dynamically generate EPICS databases*** in response to detector installation or removal.

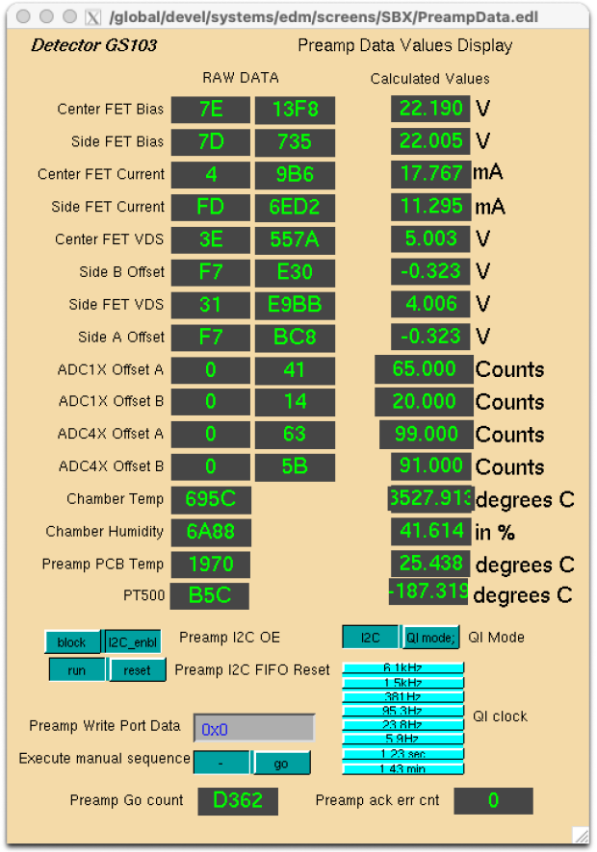


The Pickoff and Power Board are both inside the Slope Box Extension (**SBX**).

The new ANL Preamp

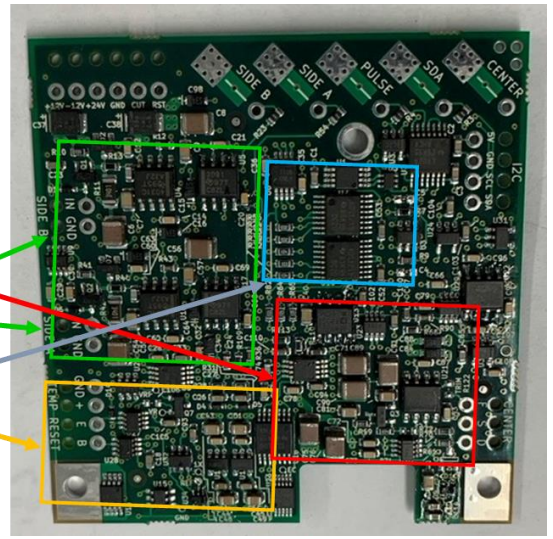
With cables as installed into detector

Control & monitoring view seen by user



- **Software-accessible ADCs** measure
  - GeCenter FET bias
  - GeCenter FET drain voltage
  - Side FET drain voltage
  - Side FET bias
  - Side FET total current draw
  - Side A vs. Side B offsets
  - PT500 temperature
  - Preamp PCB board temperature
  - Preamp chamber humidity
- ***Interfaced to EPICS through SBX pickoff card's serial communication to/from the collector box.***
  - works both in ***array*** and ***standalone*** configurations
  - *simplifies operation by removing manual updating of screens, scripts and databases.*

In-situ FET monitoring identifies failing or failed FETs without removing the detector from the array



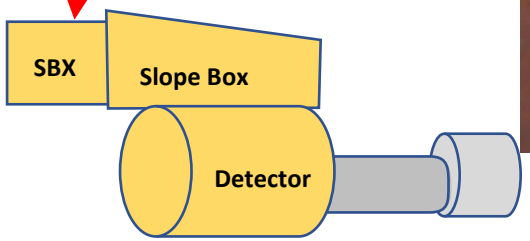
- Improvement over and replacement for the original LBNL preamp
- Completely new design based upon modern ICs
- Digital potentiometers provide **software-controlled** rise time, FET bias voltage and preamp reset speed

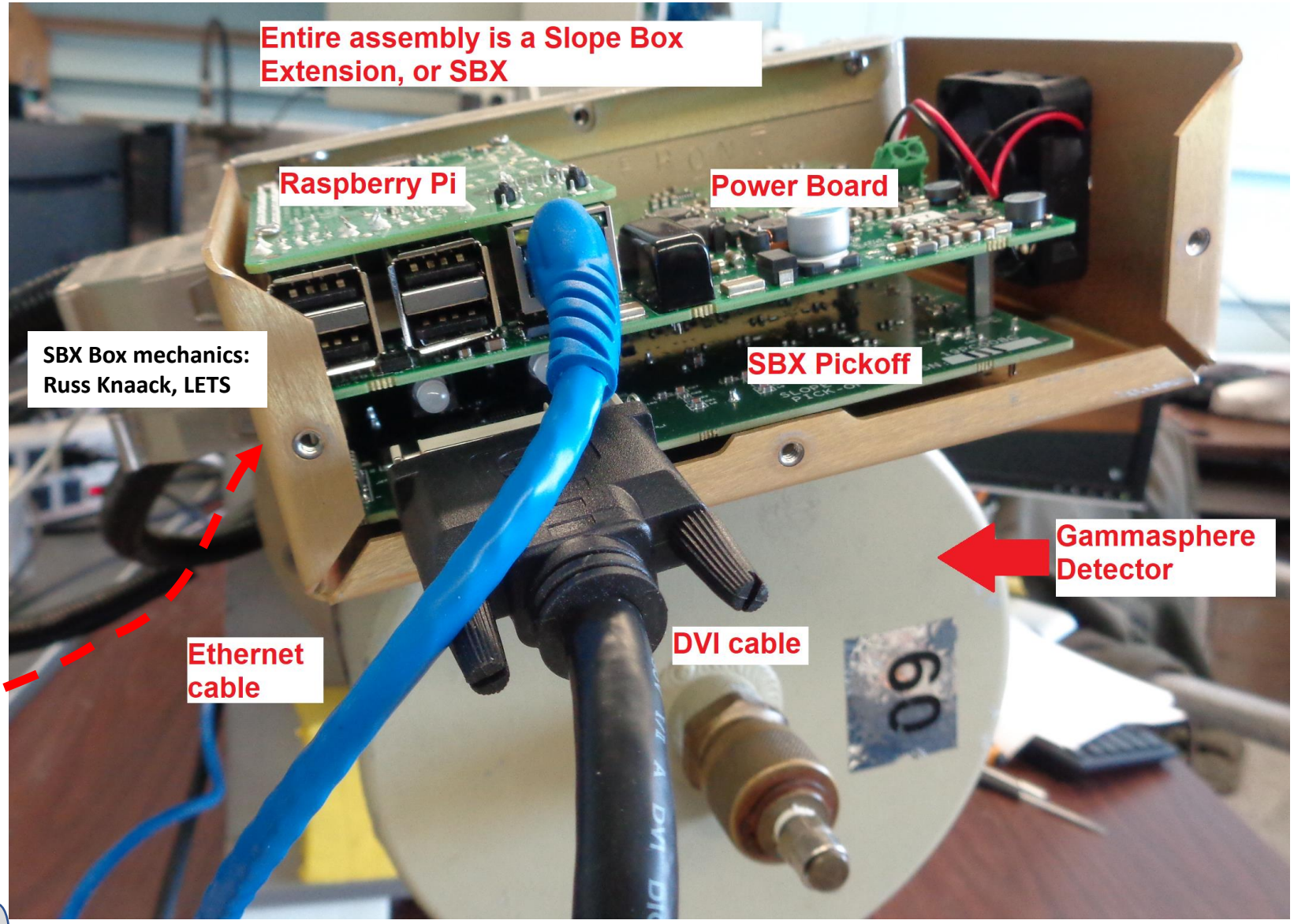
- **Low Noise Test pulser** provides software-controllable amplitude test signals at rate controlled by FPGA of SBX pickoff.
- **EEPROM** stores detector test results and operational set points, allowing for ***dynamic configuration*** of alarms, limits and setup scripts in response to adding or removing detectors.

The Slope Box Extension (SBX)

An assembly of circuit boards that plugs directly into the Slope Box, completely replacing the VXI crate, VXI power and gray cables.

- The ***SBX Pickoff*** is the control interface and handles all the signal processing.
- The ***Power Board*** provides all needed voltages from a single 48VDC source.
- The ***Raspberry Pi*** is an EPICS IOC mounted on the detector when it is used in the *standalone* configuration.
- When used in the *array,* the Raspberry Pi is replaced by the ***dongle interface card*** and the Raspberry Pi of the ***collector box*** is the IOC.

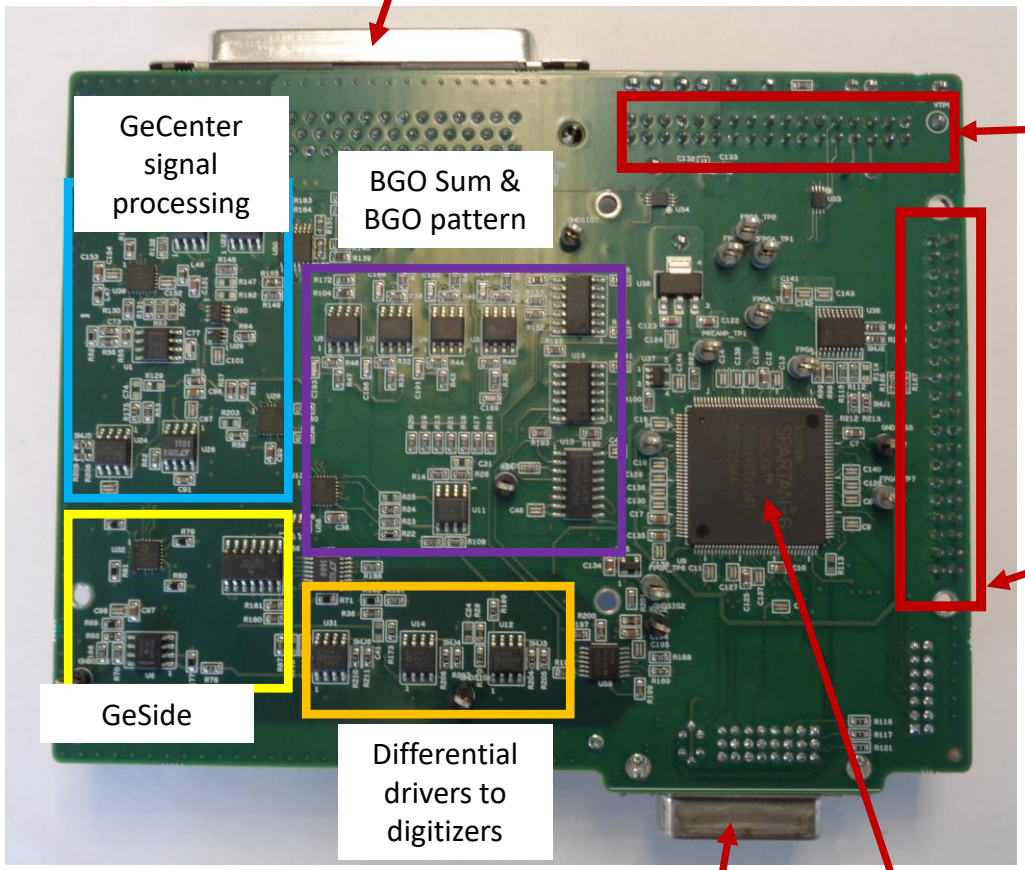




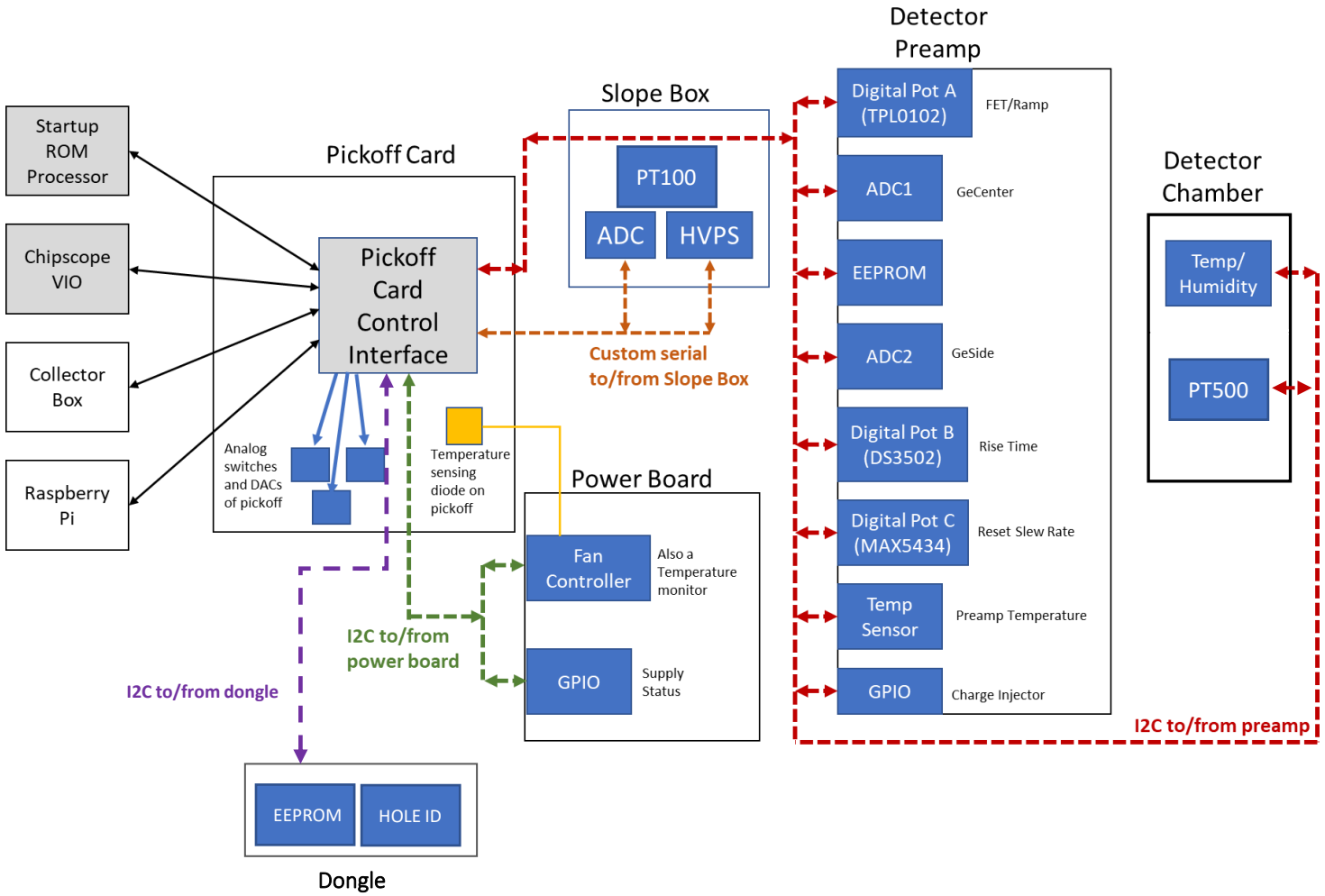
The terms ***"standalone"*** and ***"array"*** will be explored in depth as we proceed.

The SBX Pickoff card

- ***Processes*** the analog signals from the detector through four signal chains and drives differential analog signals to digitizers – replaces original VXI pickoff card.
- ***FPGA-based*** design that provides communication hub interfacing the **preamp, power board, dongle** and **slope box** to EPICS through serial interface
- ***Analog signal paths*** are completely software controlled
  - GeCenter time constant & gain
  - BGOsum gain
  - GeSide signal selection & mode
  - BGOpattern signal selection & mode
  - DC offsets of differential signals
  - BGO discriminator and preamp reset thresholds
  - ***Many*** new signal combination modes enable new measurements – please contact us for details.
- ***Automatically scans and collects*** data from Slope Box, preamp, power board and dongle into dual-port memory so entire detector appears as one interface to the IOC
- ***Implements BGO discriminators*** with fast serial interface to **Collector box** so FPGA of collector can form the *electric honeycomb* suppression data
- Implements *data logging/histogramming memory* for monitoring of various system parameters (preamp reset rate, detector voltages, etc.)



The SBX Pickoff FPGA is a multi-protocol communications hub with many sub-processors



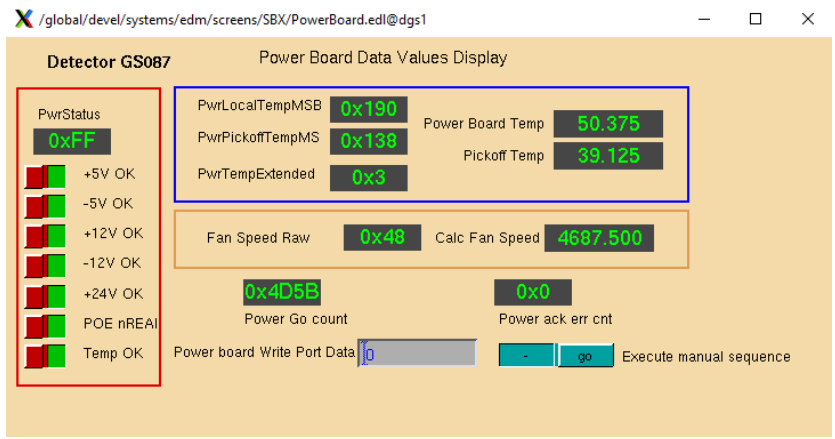
All access to pickoff registers and the dual-port RAM is through ONE serial port, either the local Raspberry Pi (standalone) or the DVI cable from the Collector (array). The FPGA senses which port is present automatically.

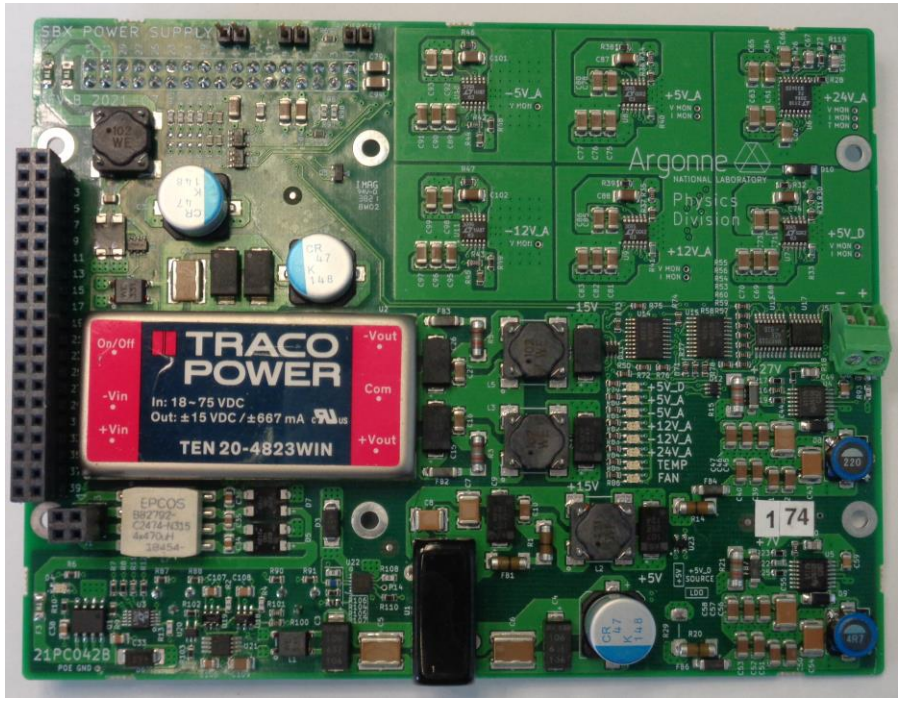
The FPGA of the SBX pickoff performs automated scanning of the **Preamp,** the **Power Board** and the **Slope Box** continuously in the background using *scan programs* stored in ROM. At startup, the EEPROMS of both the **Preamp** and the **Dongle** are also read.

Within the pickoff firmware, *scanner machines* run these scan programs with the help of *transactor machines* that perform individual reads/writes tailored to the specific communications protocols of each device (speed, signal types, etc.). All *scanners* can be paused to allow EPICS to manually insert commands to the *transactors* to set any detector device to new values.

The status information collected by all these *scanners* is then written into a *dual-port RAM* by an *arbiter* machine. Values in the dual-port RAM are read by EPICS just like any register of the pickoff FPGA, hiding all device-specific timing from EPICS.

The Power Board of the SBX

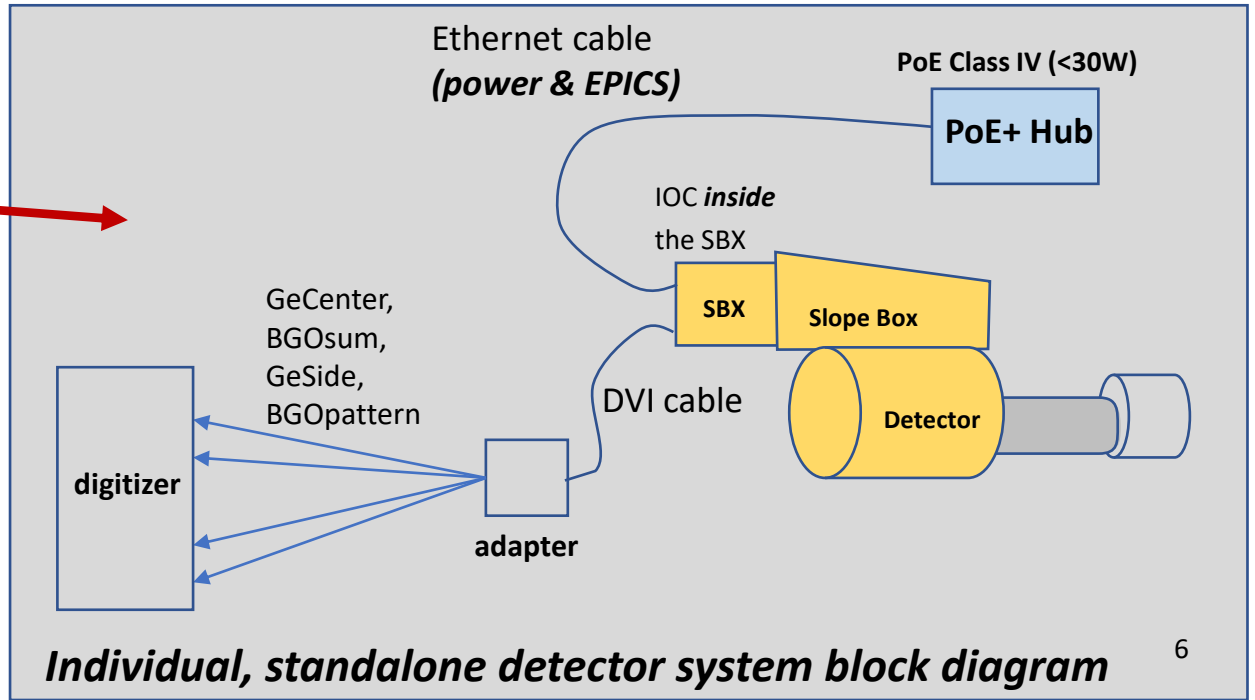




All-in-one, point-of-load power system for detectors plus SBX.

- Powers entire system from single 48VDC input
- Generates all voltages for slope box, preamp, pickoff board and Raspberry Pi
- ***Power Over Ethernet*** compliant – allows a single detector plus SBX to be run *entirely from one PoE port,* ***enabling standalone detector operation***

Details of the Power Board design, functions and features in next talk by M. Oberling

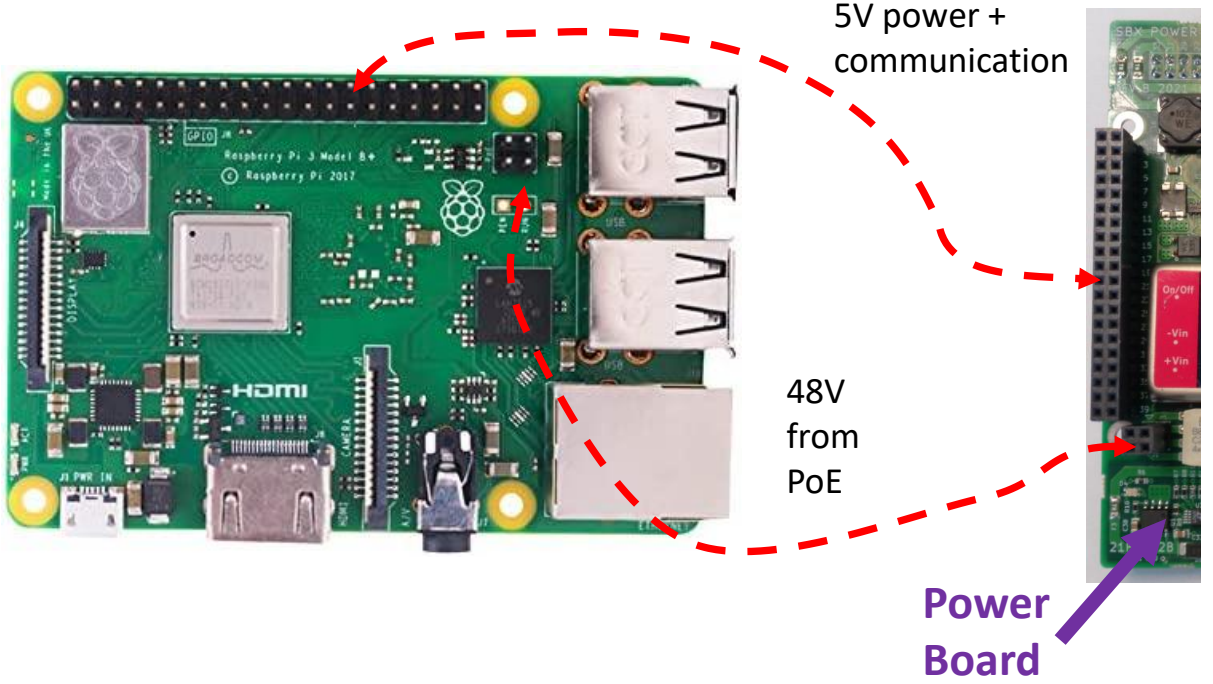


Individual, standalone detector system block diagram

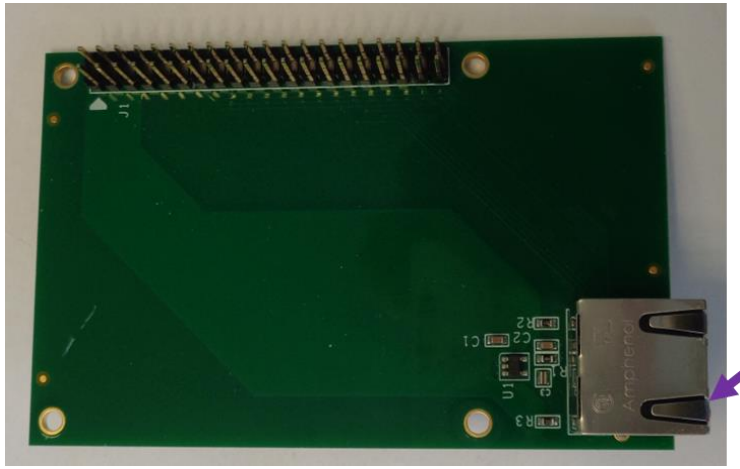
Raspberry Pi and/or dongle interface

When operating in ***standalone*** mode, the Raspberry Pi plugs into the stacked connector of the **Power Board** to both receive power and communicate with the **SBX Pickoff.** The Raspberry Pi 3B+ was specifically chosen for the 48V breakout of PoE power. EPICS on the 3B+ only uses 5% of the CPU. The Pi 4 confers no benefit and uses more power.

GPIO pins of the Pi are connected to the FPGAs of the SBX pickoff and the Collector Box for use as needed.



When operating in ***array*** mode, the Raspberry Pi is removed, and the ***dongle interface*** card takes its place in the SBX. IOC functionality moves to the **collector box.** The ***dongle interface*** connects the **SBX Pickoff** to the **dongle** that provides array position and BGO shell data from its EEPROM.



The **dongle** mounts to the **array frame** and never moves, providing a non-volatile memory of information specific to the unique array position.

The Collector Box

The ***Collector Box*** replaces ***all*** the functionality of the VXI crates for up to 30 detectors

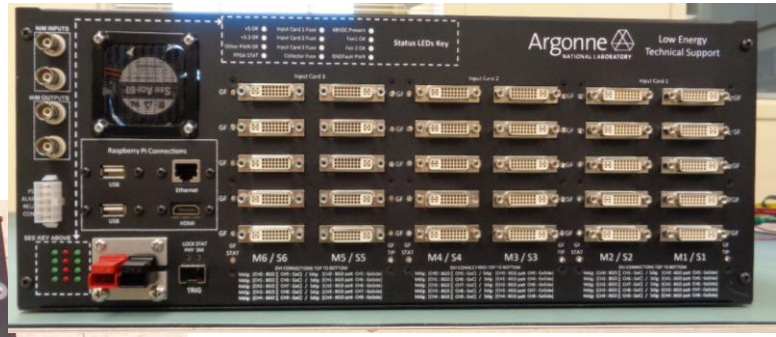
- control & monitoring (EPICS IOC)
- power distribution
- analog signal routing

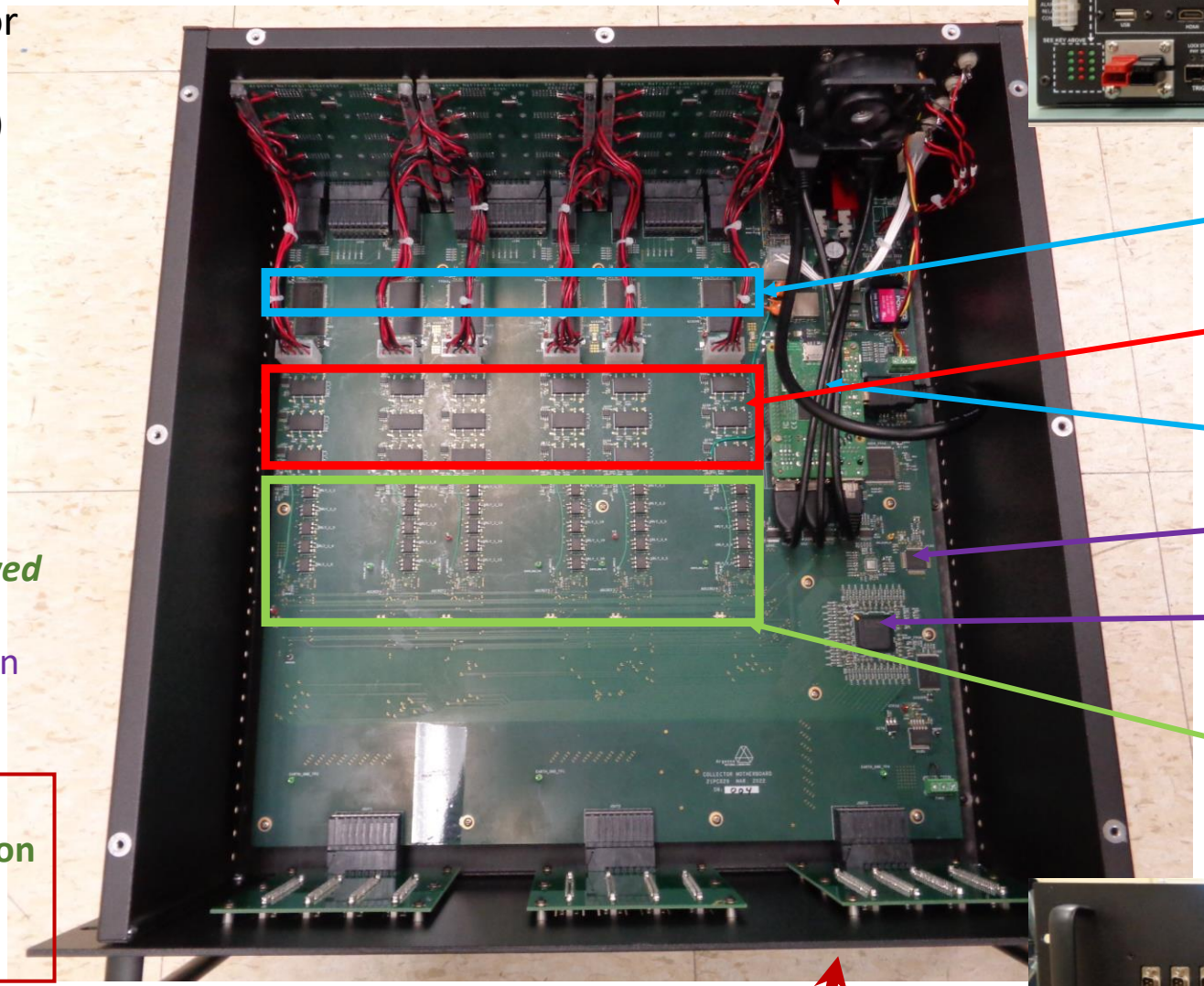
The ***Collector Box*** adds ***new*** features and capabilities to the system:

- ***dynamic reconfiguration of all EPICS databases in response to detectors being added or removed***
- distribution of trigger's clock for array-wide pulser synchronization
- interface to trigger for electric honeycomb
- **single-point earth ground**
- **automated ground fault detection**
- **per-detector power control and monitoring**

Details in next talk by M. Oberling

DVI Inputs at rear





Control FPGAs

Power distribution system (M. Oberling talk)

Raspberry Pi (EPICS IOC) - replaces VXI IOC

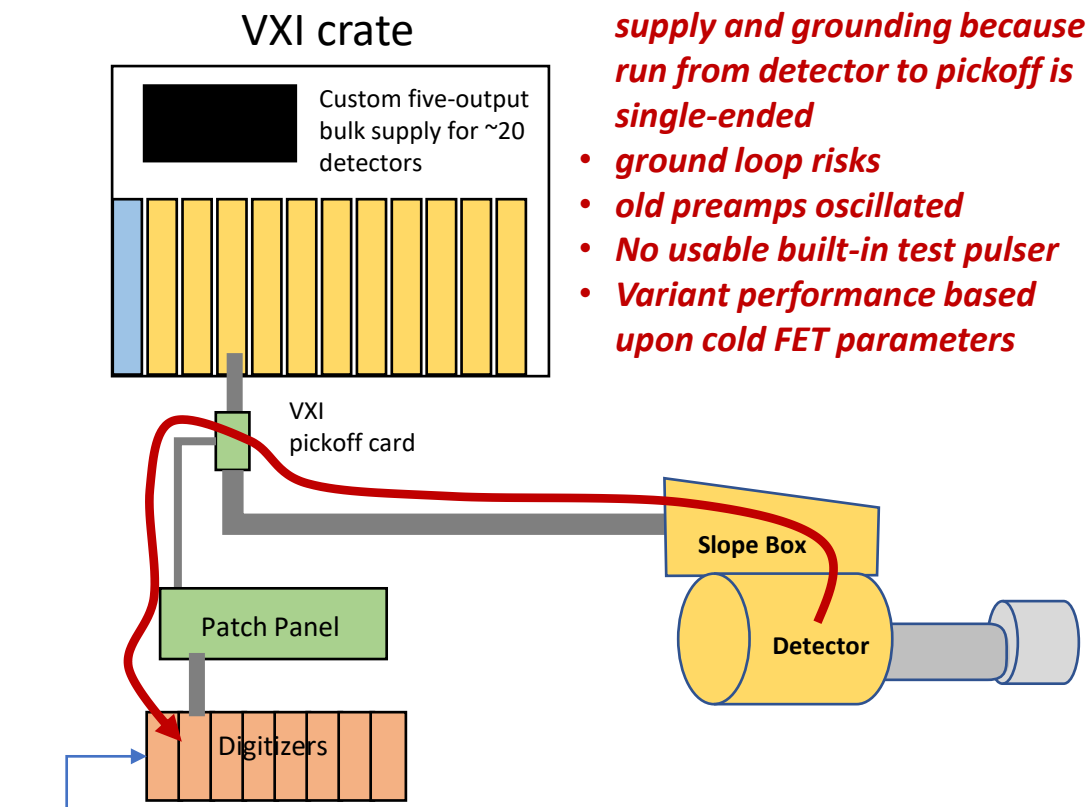
Trigger system interface

Electric Honeycomb FPGA

Ground fault detection and system monitoring (M. Oberling talk)

Digitizer outputs at front

Summary of Upgrades and Improvements to the Signal path

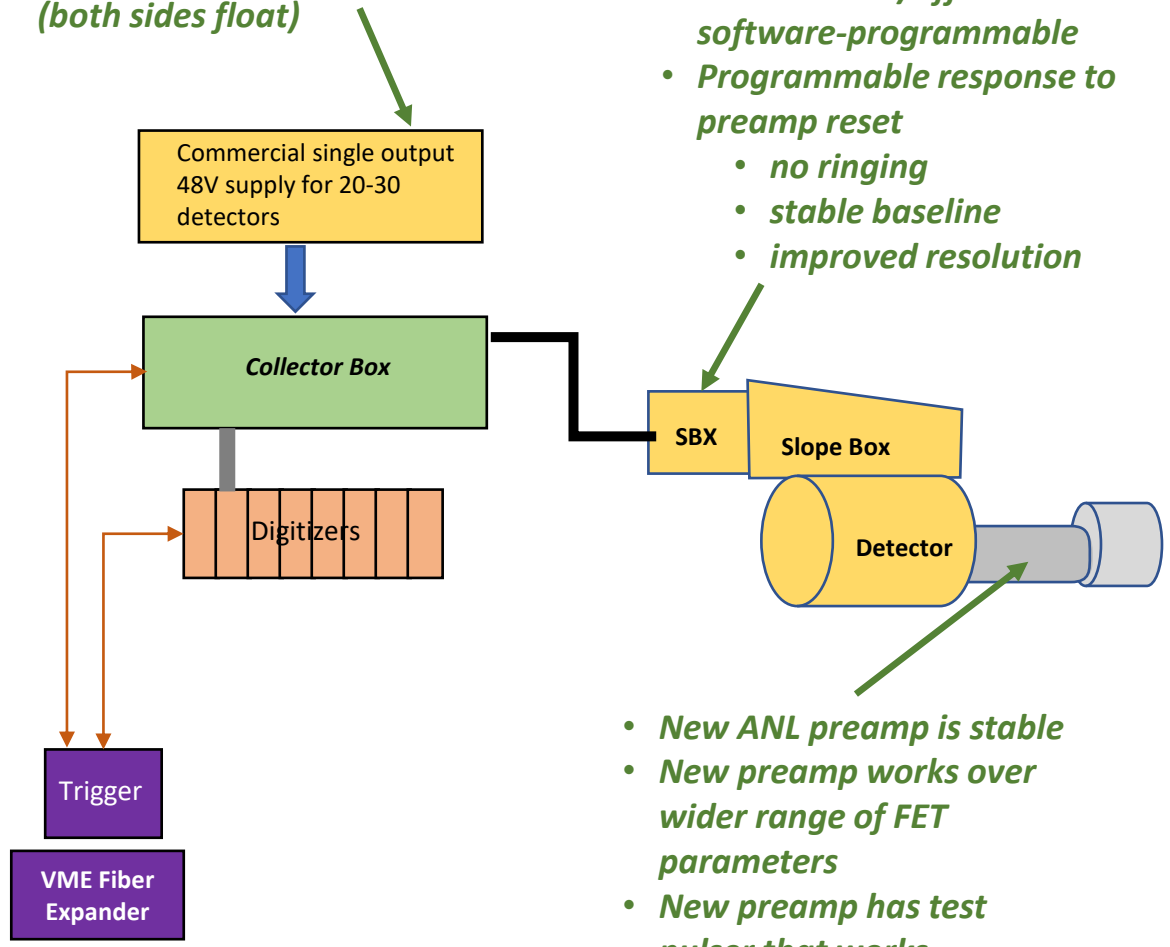


- *Noise introduced from power supply and grounding because run from detector to pickoff is single-ended*
- *ground loop risks*
- *old preamps oscillated*
- *No usable built-in test pulser*
- *Variant performance based upon cold FET parameters*

- *Some VXI pickoffs 'ring' after preamp reset*
- *No BGOpattern signals*
- *Can only see one of the side channels*
- *Pain of pots & jumpers*
- *DC offset is fixed*

Before

- *Bulk power supply has its own path to ground; chassis is isolated from detector frame and collector box*
- *Outputs of bulk supply are isolated (both sides float)*



- *SBX pickoff is fully differential*
- *Full support of BGOpattern*
- *Can see both side channels*
- *All selections/offsets software-programmable*
- *Programmable response to preamp reset*
  - *no ringing*
  - *stable baseline*
  - *improved resolution*

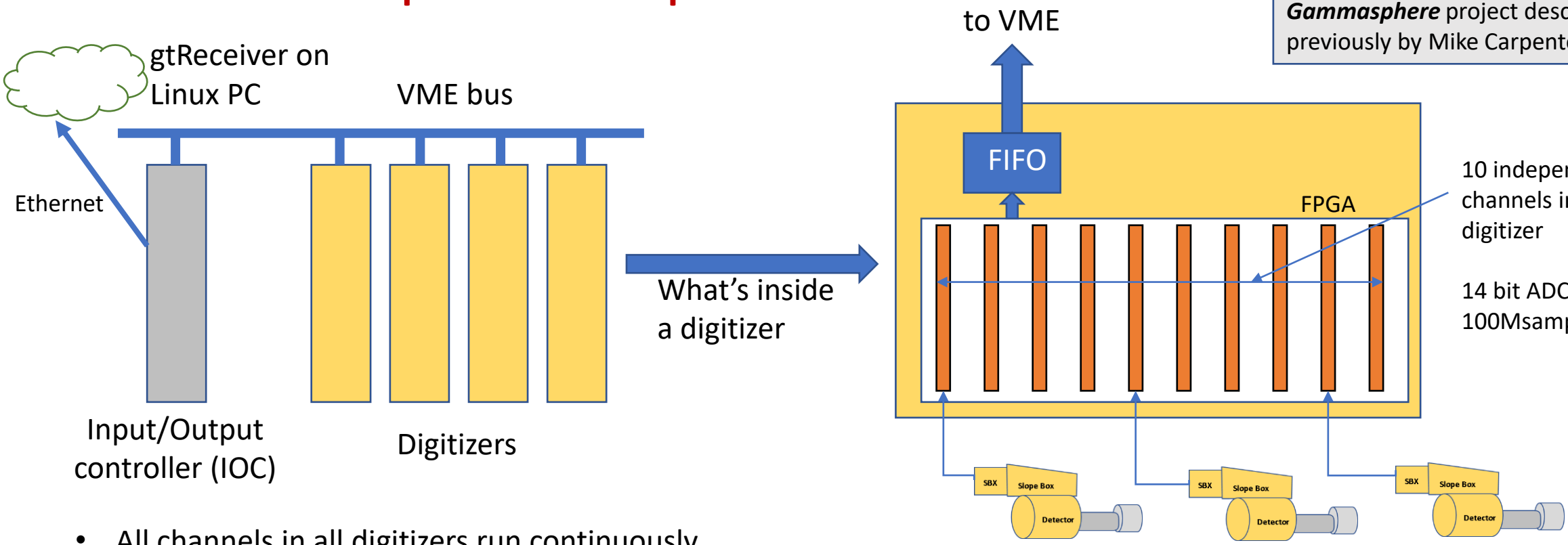
- *New ANL preamp is stable*
- *New preamp works over wider range of FET parameters*
- *New preamp has test pulser that works*

After

NEW DATA ACQUISITION SOFTWARE

Overview of Gammasphere Data Acquisition

This was implemented in the ***Digital Gammasphere*** project described previously by Mike Carpenter



- All channels in all digitizers run continuously.
- When discriminator logic marks leading edges of gamma-ray signals, energy sums, timing and other data are stored in a ***header*** identifying the event.
- If the event is selected for readout by the **trigger system,** the *header* and a programmable amount of *waveform* is transferred from the channel to the board-wide FIFO.
- The IOC scans the FIFOs to see if there is data to read out. If so, the IOC reads it out into *buffers.*
- The **gtReceiver** program sends messages to each IOC when the receiver is ready for more data. The IOC then breaks apart *buffers* into UDP packets and sends them to **gtReceiver.**
- **gtReceiver** then routes the packets to files that may be organized by-digitizer or by-channel.

Upgrades and Improvements to the Data Acquisition Software

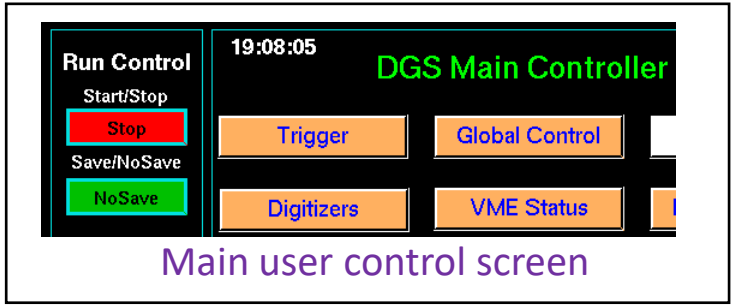
- ***Known DAQ problems at start of project:***
  - Issues with readout failing in setups where data rate digitizer-to-digitizer varies widely (“buffer drain”); also exhibited in sort/merge as memory overflow when sorting by timestamps
  - Issues with readout failing to collect all data at end of run (data left in digitizers)
  - stop of run unclean (script sending Ctrl-C to receiver processes)
  - No ability to read data from trigger modules (list of triggers by timestamp, TDC for TAC-II)
  - No way to embed status/error messages in data stream to mark any potentially corrupted events
  - Data flow statistics unreliable
  - IOC processor excessively busy, readout rate not as high as it should/could be

Before

- ***GtReceiver6 rewrite:***
  - now runs with more output files to eliminate sort/merge problems.
- ***‘InLoop” rewrite:***
  - now guarantees all digitizers flushed
  - sends embedded “readout done” message in data after flush; gtReceiver can now shut down in response, thread by thread.
  - now can read from triggers
  - sends embedded status/error messages with timestamps.
- ***‘OutLoop” rewrite:***
  - calculates detailed and correct data flow stats.
- ***Queue/buffer system redesign/rewrite***
  - Can run full-out, crashes prevented by flow control
  - Impossible to run out of buffers
  - Dynamic thread priority adjustment

After

Inside the IOC



Main user control screen

Detectors fill digitizers with data.

VME crate with digitizers

Pile of buffers.

qFree
list of pointers to free buffers

qWritten
list of pointers to buffers with data

qSender
list of pointers to checked buffers

gtReceiver on Linux PC

inLoop

outLoop

minisender

inLoop takes a pointer from qFree, fills the buffer pointed to with VME data, then moves the pointer to qWritten. Status buffers are generated for empty digitizers and error conditions.

outLoop takes a pointer from qWritten, checks the data for sanity, and if ok, moves the pointer to qSender.

minisender waits for gtReceiver to ask for data. When this occurs, minisender takes a pointer from qSender, formats the buffer into packets and sends it to gtReceiver. When done the pointer is returned to qFree.

Run/Stop from **EPICS**

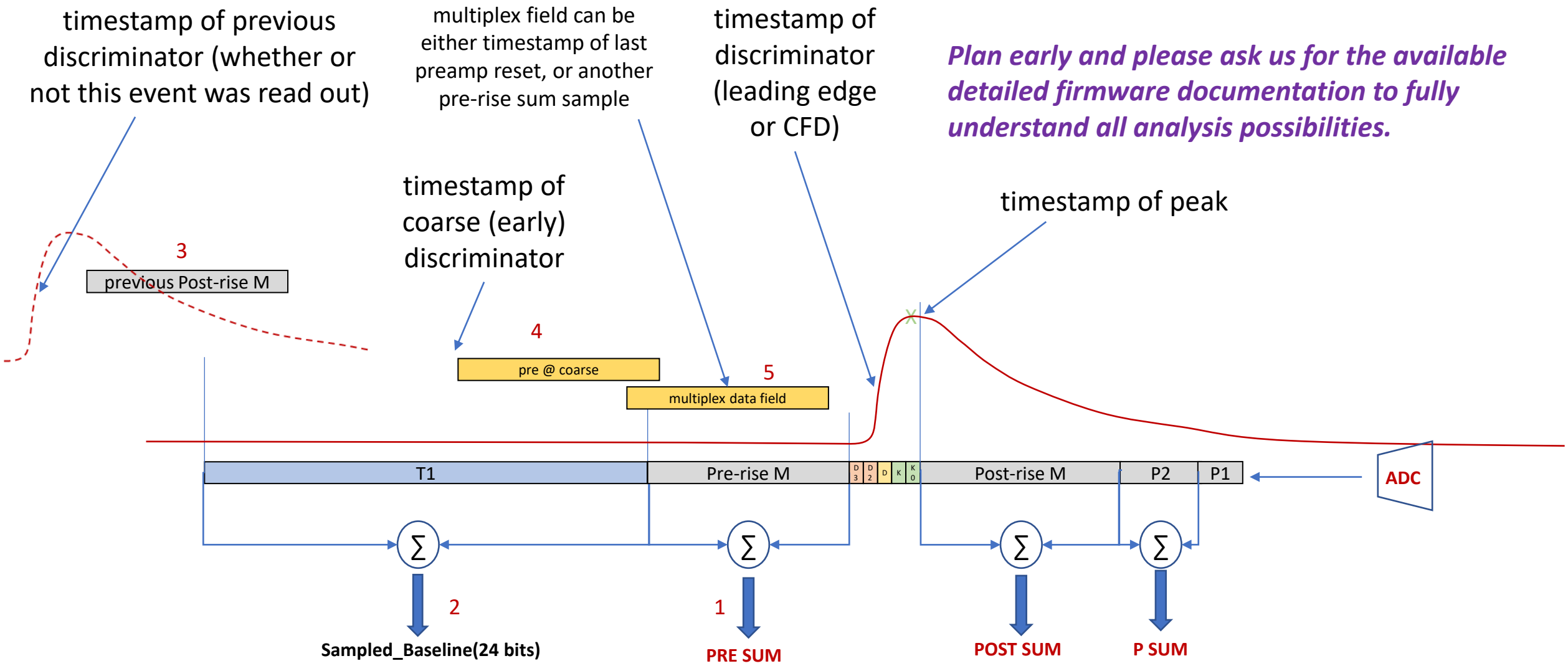
Statistics to **EPICS**

Save/NoSave from **EPICS**

Contents of the digitizer header of each event have changed

gtReceiver saves all this for every event; the user must ensure that all values are properly analyzed for best experiment results.

Plan early and please ask us for the available detailed firmware documentation to fully understand all analysis possibilities.



Each event can have up to ***five*** energy sums from before the leading edge of the event for baseline and pole-zero correction.

Split post-rise sums are available to allow for energy extraction at extreme rates or analysis based upon decay time

Digitizers provide critical information <u>besides</u> just timestamps and energy sums

triggering

pileup

errors

hints for proper analysis

important CFD flags

FLAG NAME	MEANING	USAGE
TTS	G_TS_MODE. if 0, the TRIGGER_TIMESTAMP_DATA field is the timestamp when the trigger message ***arrived.*** If 1, it is the timestamp of the messa	
PBYP	If 0, the digitizer is in TTCL mode. If 1, the PEQ is bypassed and the digitizer is in Internal-Accept-All mode.	
PF	Pileup Flag; event was piled up.	
PO	Pileup Waveform Only. User has only allowed readout of piled-up waveforms.	
GE	General Error. A general internal error has occurred and the firmware should be reloaded.	
SE	Sync Error. The digitizer reports a timestamp synchronization error.	
OF	Offset Flag. This data is an Extended Event whose readout if offset due to readout interference.	
PV	Peak Valid. The peak-sensing logic has found a peak in this event, so the peak timestamp is valid.	
ED	External Discriminator Flag. This event was caused by external discriminator, not internal leading edge discriminator logic.	
VF	Veto Flag. If digitizer were enabled to process router vetoes this event would have been vetoed.	
WF	Write Flags. 0:ADC data is 14 bit with flags. 1: ADC data is 14.2 format, no flag bits.	
PTE	Pileup Time Error flag. User has set illegal combination of holdoff/pileup values Digitizer must be completely reset.	
P2M	P2 Buffer Mode. 0:P2 length set by reg_p_window. 1:Length of (P2+Post) set by M, P2 set by reg_p_window, Post length is 'm'-'p'.	
CPTS	Inidicates mode of the multiplex field. 0: field is 2nd early pre-rise sum. 1: field is timestamp of last preamp reset.	
CEM	CFD Esum Mode. 0: capture pre- and post-rise energy when the CFD fires. 1: Capture energy sums using delayed copy of LED instead.	CFD ONLY
CV	FD Valid Flag. 1: CFD OK. 0: CFD samples are invalid and timestamp is timestamp of pre-arming leading edge discriminator. Always 0 in LED mode	CFD ONLY
TSM	Time Stamp Match. 1: bits 37:30 of previous CFD match bits 47:30 of this event. 0: upper TS bits do not match. Always 0 in LED mode.	CFD ONLY

The header of the digitizer data contains **many** very useful and necessary flag bits, placed there to ensure correct analysis of your data is possible. Analysis software often lags firmware. Failure to process and utilize all the header data can result in difficulties in analysis and/or errors. Please ensure your analysis software decodes ***everything*** in the header and that it ***uses*** all the information provided.

Anyone who saves waveform data is also ***strongly*** encouraged to use the *timing mark* and *downsample* bits embedded within the waveform data. Documentation is available for you to read.

NEW CONTROL AND MONITORING SOFTWARE

EPICS for the Raspberry Pi

A complete set of device support and driver interface routines have been written that allow a Raspberry Pi to communicate with FPGAs in SBXs and the Collector Box using a variant of the **SPI** protocol.

This software implements process variables (PVs) of types ***ai, ao, bi, bo, mbbi*** & ***mbbo*** with some support for ***waveform*** PVs. Multiple ***Data Types (DTYPs)*** are implemented to support all the features of Gammasphere and the new hardware.

- ***Local Serial*** PVs communicate between the software and FPGAs of the various boards to set control values and read status.
  - A "mailbox" system of temporary variables has been implemented to allow PVs associated with the same register to share the data from one read so duplicitous transactions are eliminated, saving limited serial bandwidth.
- ***Soft Control*** PVs are used to modify overall operation and/or modulate the activity of ***Local Serial*** PVs, such as setting "mailbox" values to enable/disable diagnostic printouts as PVs of different types execute.
- ***I2C*** PVs cause multiple serial transactions to occur, communicating with the "transactor" machines of the SBX Pickoff FPGA to send commands to chips in the **Preamp** and **Power Board.**
- ***Calculation*** PVs implement conversion routines more complex than EPICS basic scaling such as PT100/PT500 conversions to both resistance and temperature.
- ***Step*** PVs implement sequential up/down stepping using ranges and rates defined by ***Soft Control*** PVs, to control ramping of high voltage in the detectors.

In addition to the SPI pins of the GPIO connector (SCLK, MOSI, MISO, CS), within the Collector other GPIOs are used to select whether serial transactions are aimed at devices within the collector or to SBXs connected to the Collector. The "device selection code" is asserted before each transaction by the EPICS device support code based upon PV parameters.

Upgrades and Improvements to the Control and Monitoring System

- ***EPICS in VME crates***
  - Databases did not match firmware, updates laborious and error-prone.
  - Databases/screens did not provide access to all control functions.
  - No access to data FIFOs of system trigger or router trigger modules.
- ***EPICS for detector monitoring & control***
  - Need for EPICS to read out new/improved status data from preamps & power boards
  - New hardware (pickoff/preamp/power board) requires EPICS support for all the new features and functions
  - Need for way to run detectors ***outside of the array and to eliminate failing VXI system***

Before

- ***Spreadsheet-based Database Generation System:***
  - Single source (Excel spreadsheet with VBA) generates all EPICS databases, all device support C code and some of the VHDL firmware ***from one button click.***
  - Spreadsheet system also *verifies* which PVs are found in any screen definition file and *generates* a generic screen file of all PVs for deployment.
- ***Rewrite of EPICS device support/driver***
  - Can now read data from trigger modules.
- ***Pickoff Card with FPGA***
  - provides single point access for control & monitoring of Slope Box, preamp and power board
- ***EPICS on the Raspberry Pi***
  - Developed complete device support for SPI serial communication with FPGA of pickoff card; reused in design of collector
  - Hardware/firmware design of pickoff allows for control from a locally plugged in Raspberry Pi or over same DVI cable that carries power and analog signals; allows for completely standalone detector
- ***Power-Over-Ethernet (PoE) Compatibility***
  - SBX design allows powering and operating a detector completely from one PoE network port.

After

TOOLS FOR MAINTAINING LARGE CONTROL SYSTEMS

Spreadsheet-based system for database and code generation

All interfaces to hardware can be abstracted as a list of *registers* at different addresses that may be single values or composed of *bit-groups* (e.g., bits 9-0 are some value, bit 10 is a single bit control, bits 15..12 are a 1-of-16 option select, bit 11 is unused). From this abstract definition of hardware, the electronics engineer develops firmware and test stand software, the programmer develops software and data structures, and the scientist develops screens and scripts. Much duplicitous effort is expended during development. Over time, these end products inevitably become dissociated, and much effort is wasted cleaning up discrepancies when/if they are found.

WHY YOU NEED THIS

A **spreadsheet-based system** (Excel + VBA) has been developed that allows anyone to define the registers and bit-groups in a straight-forward way, and from this code underlying the spreadsheet writes many output products that are all internally consistent with each other with **one button click** that takes only *seconds* to run:

- VHDL implementation of registers for firmware
- C data structures of registers/address/bit-groups for software
- EPICS databases of process variables that map to all registers and bit-groups
- EPICS databases of *global* PVs that allow fan-out of control values across an arbitrary system of multiple IOCs
- Example EDM screen files with a control for every PV in the system

WHAT IT DOES

Outputs are formatted by using templates, so the system can easily adapt to any EPICS implementation. The code also contains checking tools to search existing VHDL, database or EDM files against the names within the spreadsheet. This package is used to generate all Gammasphere databases for digitizers, triggers, SBXs and collector boxes, plus the data structures used to compile the VME IOC EPICS driver and used in the test stand (GammaWare). We also use this system to generate VHDL, C and EPICS for the GRETA trigger that ANL has responsibility for.

HOW POWERFUL IT IS

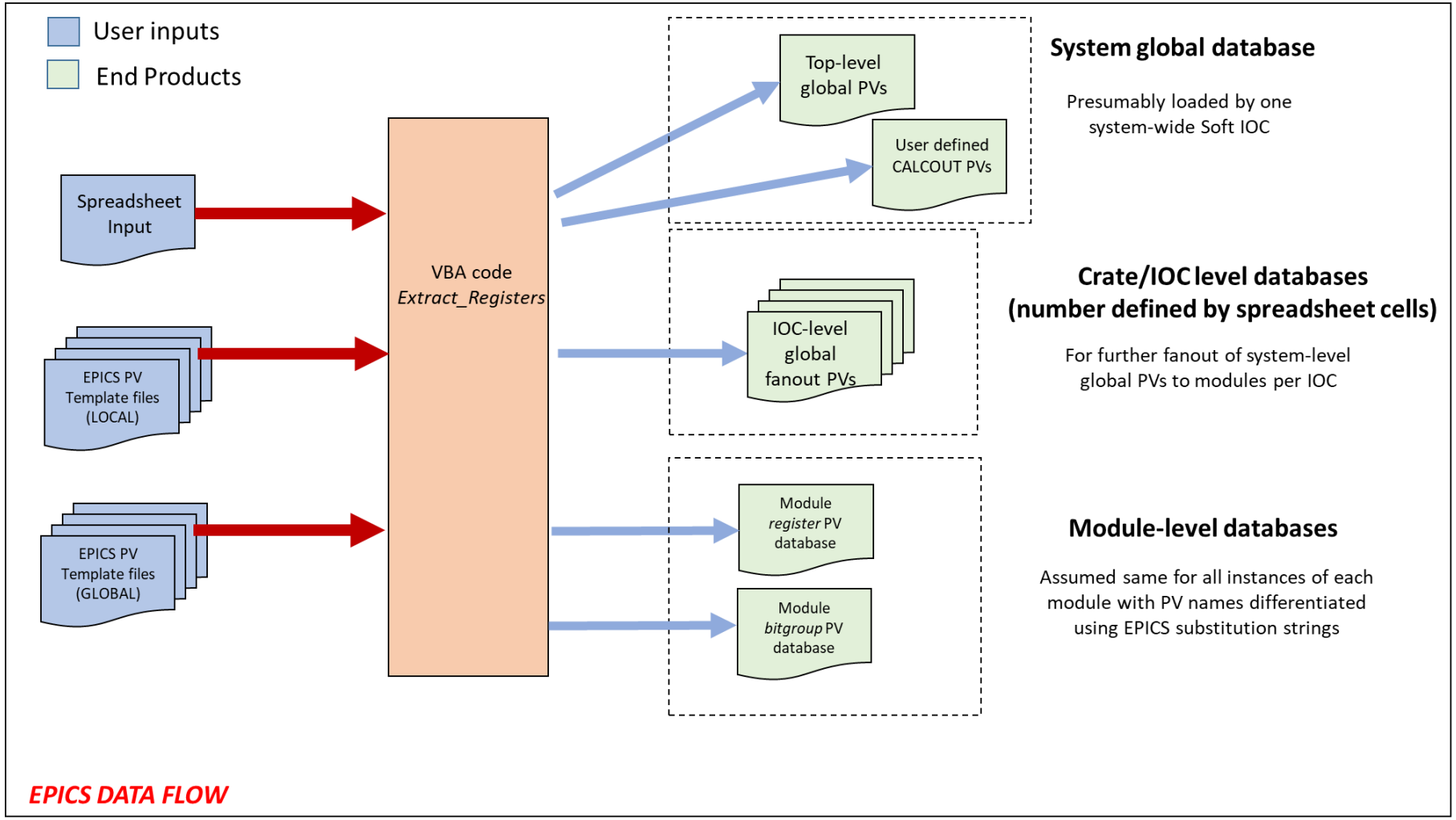
Spreadsheet-based system – EPICS functionality

The **spreadsheet-based system** writes ***the entire EPICS database structure*** for Gammasphere's digitizers with **one button click.** Other spreadsheets for the system trigger and the router triggers generate those databases too, in a few seconds each.

Statistics for generating the digitizer databases of Gammasphere

run time of Extract Registers, seconds	1.5
Lines of spreadsheet processed per second	988
Calls to EscapeCodeProcess	37192
Whole register EPICS template lines processed	4085
Bitgroup EPICS template lines processed	17389

70,688 total lines of fully commented EPICS database, generated in 1.5 seconds



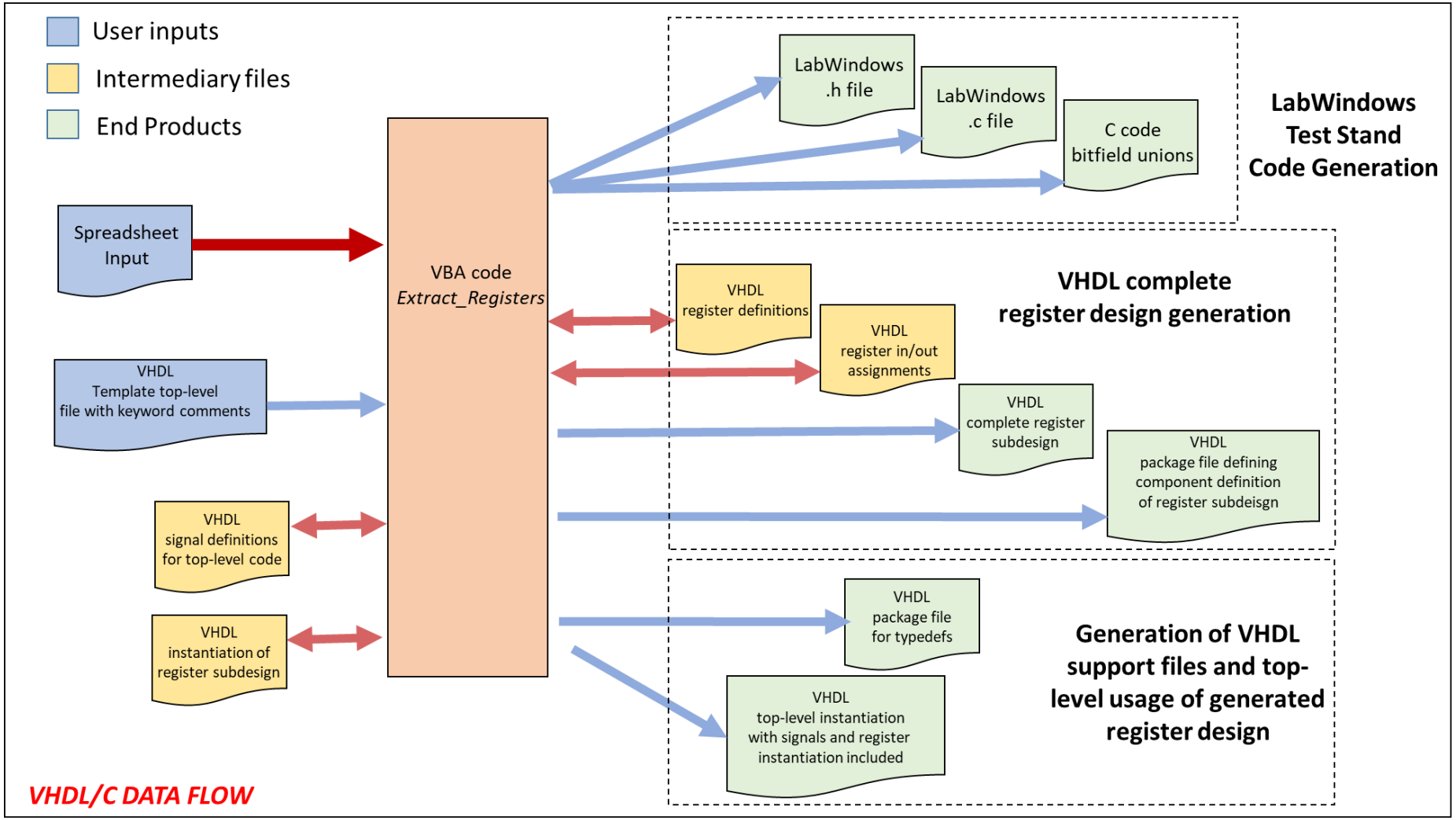
The ***templates*** define the structure of each unique PV type in the system. With different template sets, databases matching different EPICS systems are easily generated.

Substitution keywords in the templates generate the unique PVs for each line processed.

Global PV generation creates all needed fanouts of PVs across many IOCs for large systems.

Spreadsheet-based system for code and firmware generation

The ***SAME*** single button click that generates all the databases throughout Gammasphere ***also*** generates all data structures needed for the test stand software (GammaWare), the EPICS driver for the IOCs of Gammasphere, and parts of the VHDL code source for digitizer, trigger, SBX and collector FPGAs:



Spreadsheet writes C structs and .h typedefs

Spreadsheet writes VHDL code defining registers

Spreadsheet inserts generated VHDL into the structure of the whole firmware design

The ANL team working on the Gammasphere upgrade

LER Staff

Torben Lauritsen
Walter Reviol
Darek Seweryniak
Marco Siciliano
MPC

Term Staff

Pat Copp
Claus Mueller-Gatermann

Low Energy Technical Support (LETS)

John Anderson
Michael Oberling
Ed Boron
Russel Knaack
John Rohrer
Bruce Nardi

PCB Layout

Todd Hayden (HEP)

Outline of Talks

- Introduction – Mike Carpenter
- Gammasphere Upgrade Overview – Pat Copp
- *Intermission*
- Signal Processing and Controls – John Anderson
- Power Distribution, Infrastructure and Mechanics – Michael Oberling

THANK YOU FOR YOUR TIME

Next talk ⟶ Power, Infrastructure & Mechanics : Michael Oberling

Questions, comments, follow-up : jta@anl.gov