FPGA and VHDL Developments for DGS and DSSD

John T. Anderson

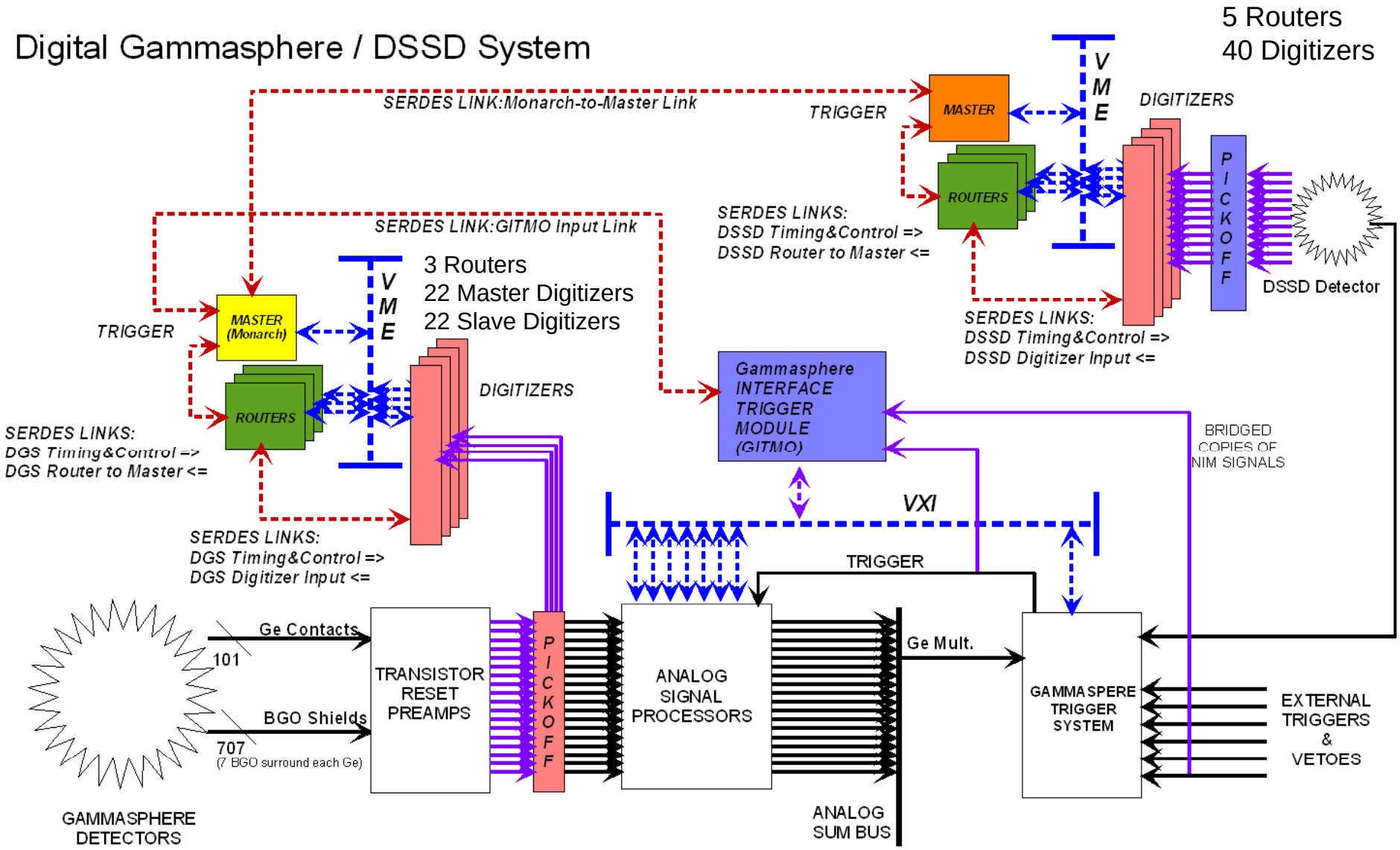
for the ANL group: M. Carpenter, T. Madden, M. Oberling, D. Seweryniak, J. Vander Ploeg, P. Wilt, S. Zhu

September 20, 2012

Overview of Talk

- System architecture (Gammasphere/DGS/DSSD)
- ANL digitizer firmware for DGS and DSSD
  - Digitizer Tester module
- Clock and Timestamp propagation across multiple systems
  - GITMO module
  - Accumulative jitter across larger systems
- Trigger firmware for DGS and DSSD
  - Cross-triggering between multiple systems
- Testability
  - LabWindows + EPICS : partners not competitors

Digital Gammasphere / DSSD System



DGS Digitizers: 4 channels per detector

Master digitizer: Ge Center + BGO Sum Slave digitizer: Ge Side + BGO Pattern

DSSD: 320 independent silicon strip digitizer channels plus 80 PPAC channels

Combined analog/digital Gammasphere plus DSSD

- The DGS and DSSD systems may be run independently or ganged
  - DSSD Master receives clock and command stream from DGS Master, but may run independently.
  - DGS Master receives clock and trigger signals from Gammasphere via GITMO, but may run independently.
  - Total system synchronization to Gammasphere clock demonstrated
  - Triggering can go both ways
    - DGS can be triggered locally, or by GITMO, or by DSSD ***using the SERDES communication links***
    - Design in place for generic trigger propagation rules between systems
- DGS connects to all Gammasphere detectors parasitically
  - Pickoff cards take preamp signal, buffer, and re-drive as differential signal to digitizer
  - Pickoff cards are AC coupled to maximize dynamic range
  - Resolution equivalent to Gammasphere achievable
  - Pole-zero correction done offline; please contact Shaofei for details of algorithm.
  - Systems run concurrently for direct comparison of same events.
- DSSD uses new interface electronics
  - New post-amps manufactured by Phil & Darek
- Event rate improvements
  - Various runs have seen >50kHz event rates occur; 100kHz may be possible
  - In test stand, ANL digitizer + trigger system with readout disabled works to >400kHz
- Unique SERDES output patterns for DGS digitizers vs. DSSD digitizers
  - DGS digitizer performs clean/dirty/module logic prior to transmission (work in progress)
  - DSSD: every channel as they occur, mapped by trigger system into X- and Y-planes
  - No ‘fast’ vs. ‘slow’ data as in GRETINA; all data is ‘fast’
  - No separate line for discriminator bit; all data is over SERDES

Moving to next topic..

- System architecture (Gammasphere/DGS/DSSD)
- ANL digitizer firmware for DGS and DSSD
  - Digitizer Tester module
- Clock and Timestamp propagation across multiple systems
  - GITMO module
  - Accumulative jitter across larger systems
- Trigger firmware for DGS and DSSD
  - Cross-triggering between multiple systems
- Testability
  - LabWindows + EPICS : partners not competitors

ANL Digitizer Firmware Overview

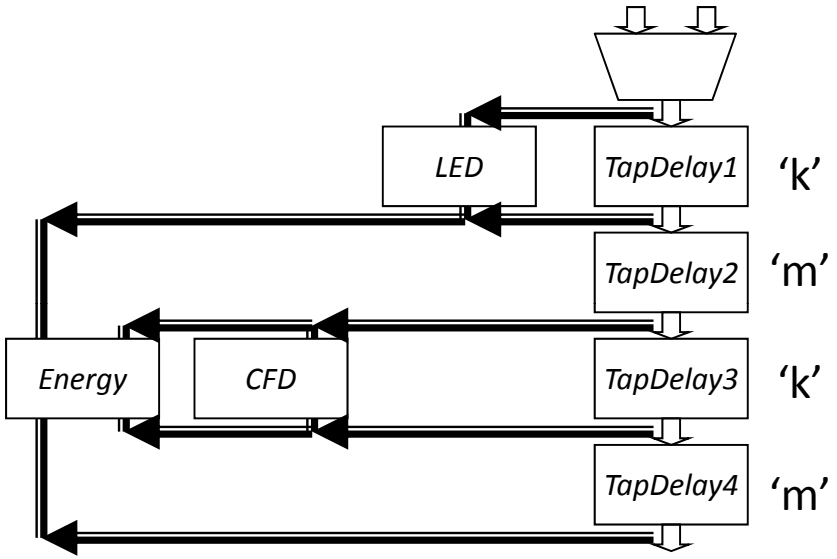
- New channel structure:
  - Leading edge discriminator ONLY (no CFD)
  - Can be set to positive, negative or both edges
  - New delay chain structure
  - Calculates pre- and post-discriminator sums for energy calculation (not trapezoidal algorithm, but similar)
  - Puts timing marks in ADC data for discriminator and peak detect
  - Timestamp of previous discriminator firing given along with TS of current hit
  - Continuously running background baseline monitoring
- Can read out in pileup reject, header only and pileup append modes
  - Reject: piled-up pulses never appear
  - Append: piled-up pulses viewed as one long trace
  - Header only : only headers of all piled-up pulses provided
- Runs in internal, external and TTCL modes
  - In TTCL mode, independent plus/minus windows
  - In TTCL mode, only hit channels read out
- Ability to mark multiple events within single readout buffer
  - Allows for high event rates
  - Throttle logic fully implemented for system flow control
- Unique SERDES output patterns for DGS digitizers vs. DSSD digitizers
  - DGS digitizer performs clean/dirty/module logic prior to transmission (work in progress)
  - DSSD: every channel as they occur
  - No ‘fast’ vs. ‘slow’ data as in GRETINA; all data is ‘fast’
  - No separate line for discriminator bit; all data is over SERDES

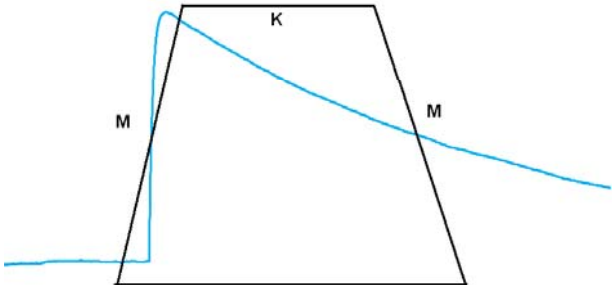
Items still in development

- **DGS-specific Ge/BGO triggering logic (dirty/clean/module)**
  - Structure understood, code written, not integrated into system yet
  - Focusing on DSSD due to ATLAS experiment scheduling
- **Master/Slave digitizers**
  - DGS master/slave design specified, code written and simulated
  - Integration delayed until BGO Pattern signal available in next phase of hardware deployment
- **Ability to change ADC/pipeline frequency**
- **Extension of triggering buffer**
  - Have enough Block Rams ;could do this directly without firmware tricks
  - No time-slice or under-sampling required

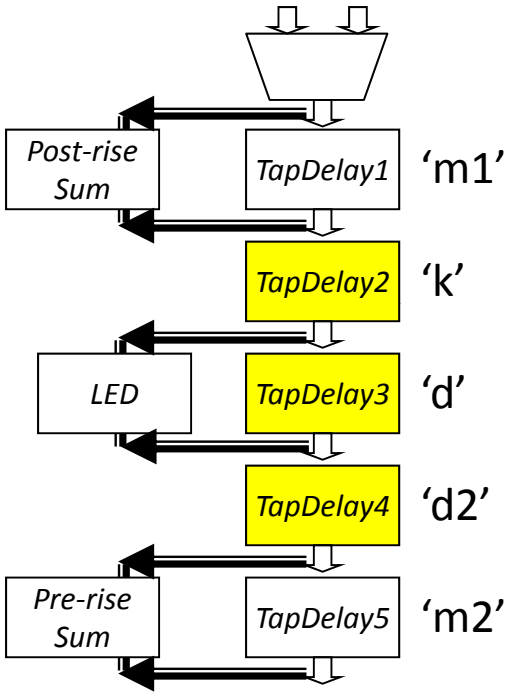
Simplistic Architectural Comparison

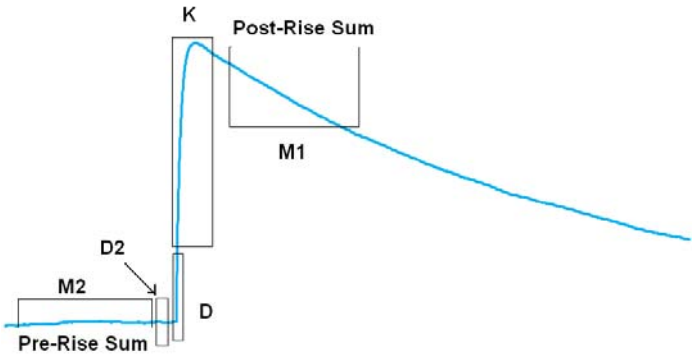
GRETINA CHANNEL ARCHITECTURE



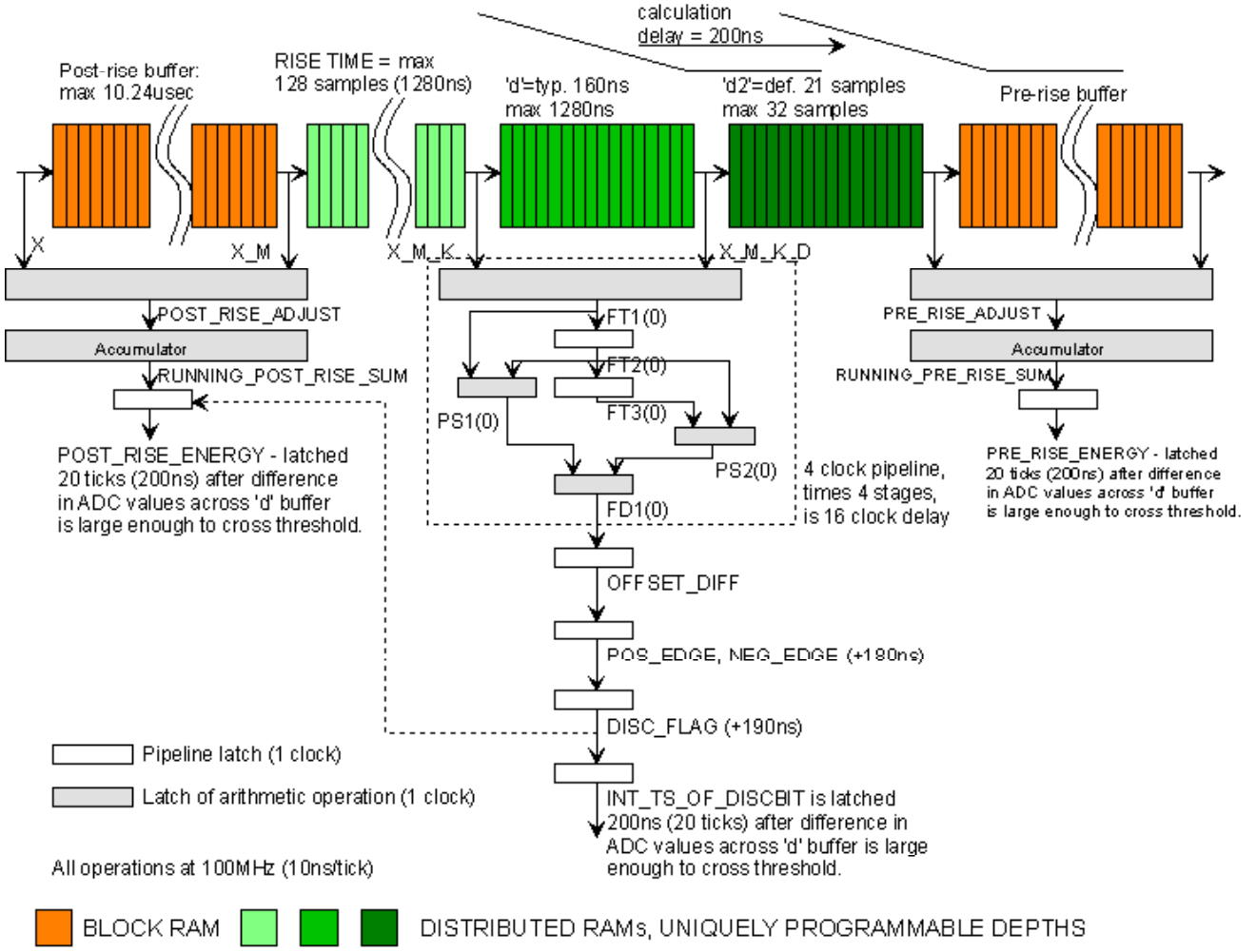


DGS CHANNEL ARCHITECTURE



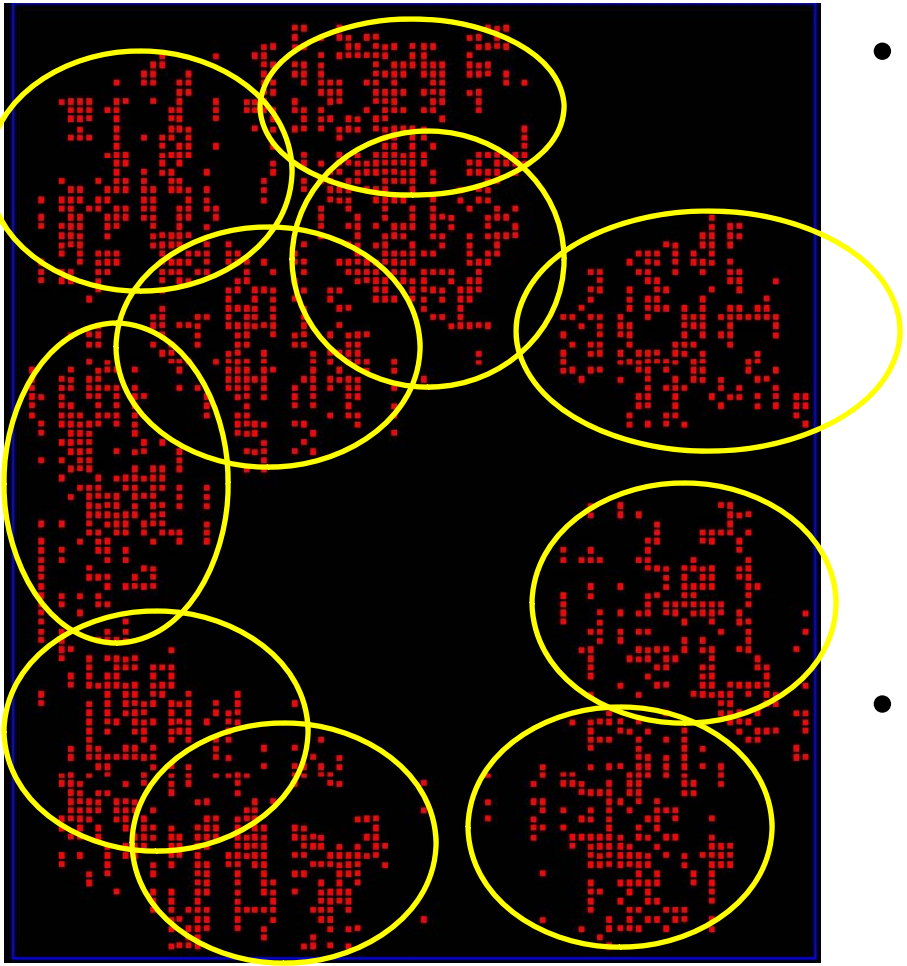


Details of channel pipeline



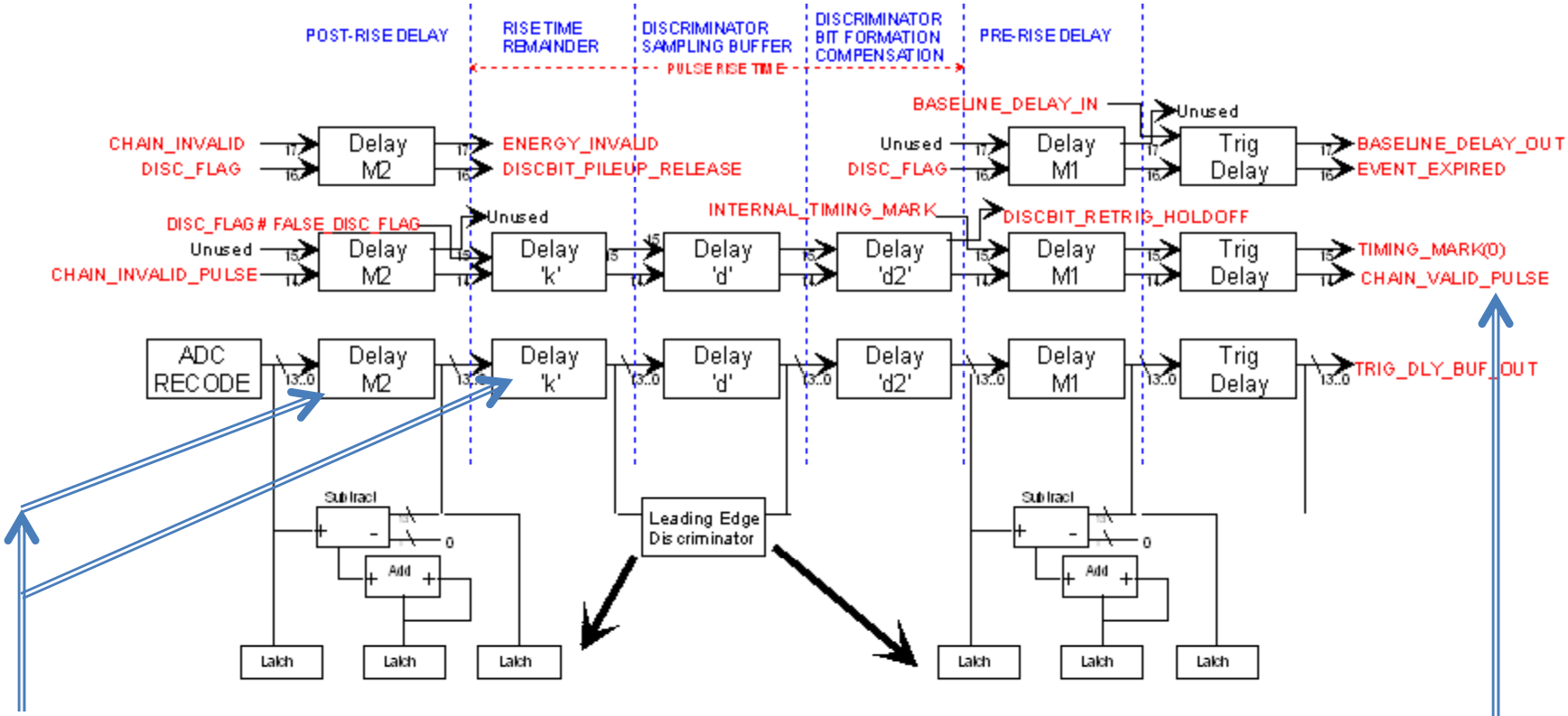
- Everything controlled by the leading edge discriminator
- Pre/post sums, specific samples and time stamp, all saved to latches when discriminator fires
- These latches then read and used to make event header

Distributed Memory



- Each red dot is a “SLICEM”
  - SLICEM is a ***distributed memory cell***
  - Each SLICEM is 16 bits of memory
  - Can be ganged to make larger units
  - Maximum practical size is 128X16 before size/speed fails
    - At larger sizes use Block RAM
- You can roughly make out the 10 channels within the FPGA here by the memory blocks

Some important timing changes



Delay parameters ***automatically adjust*** discriminator operation (pileup/retrigger)

INTERNAL_TIMING_MARK <= DISC_FLAG OR PEAK_FLAG;

FALSE_DISC_FLAG is a signal generated at initial startup when the chain will get a false firing due to the chain being filled with zeroes at reset, but the DC level entering the board is not necessarily at Vmin.

When parameters are changed, channel *automatically* disables itself until the delay chain re-fills with new data

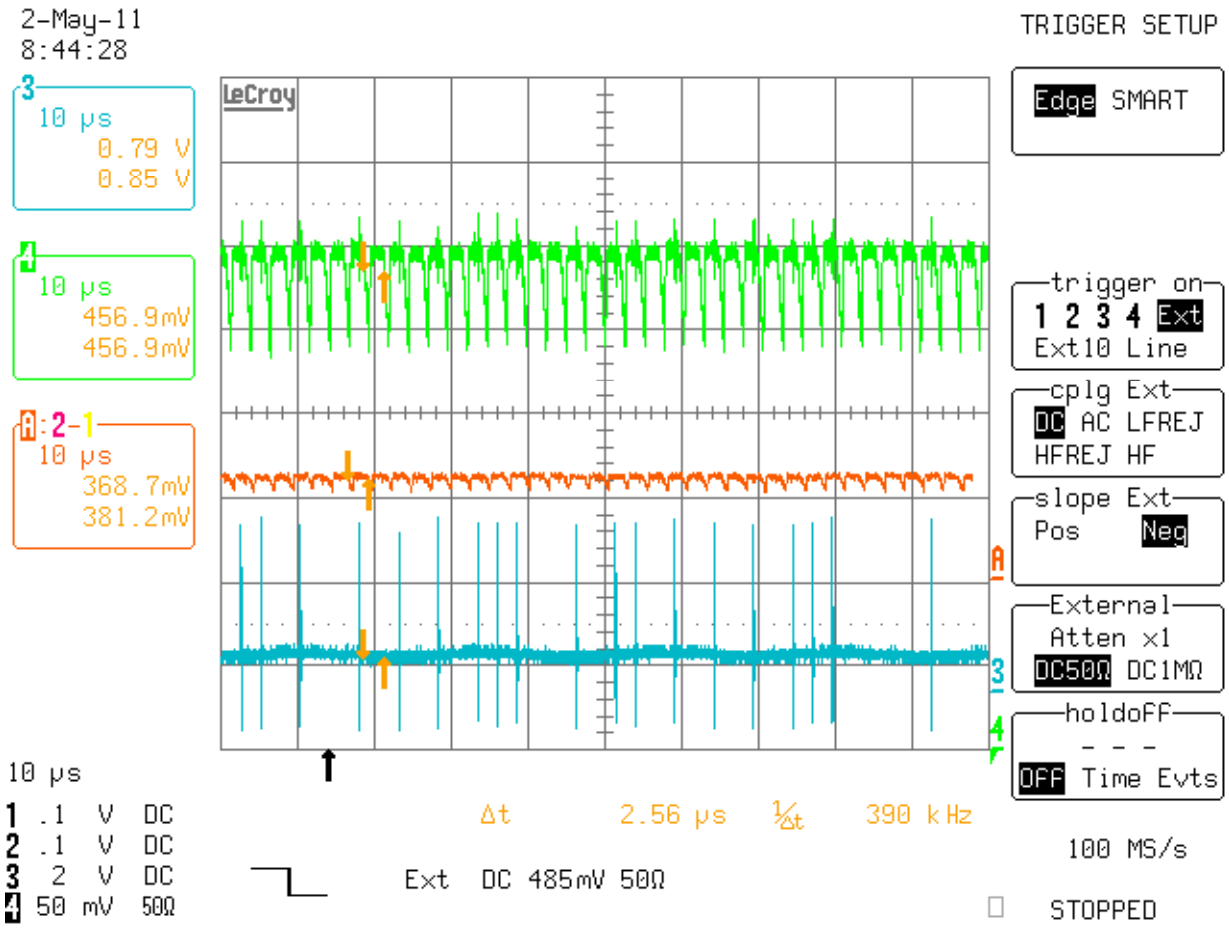
Pileup Logic

- Implemented using discriminator flag and copy of that flag delayed by depth of pre/post summation buffers
  - Counter increments by 1 when discriminator fires
  - Counter decrements by 1 when delayed copy arrives at counter
- If counter > 1, there has been pileup, so no discriminator firings are legitimate; set internal ***pileup_flag*** signal.
- If counter counts 0 1 and then counts down 1 0 without ever having reached 2, that was an undisturbed pulse and can be accepted
  - ***Both pre- and post-event pileup is monitored***
- The filtered discriminator bit is issued <depth-of-summation-buffer> clocks ***after*** the discriminator actually fires, but the ***time stamp*** that is stored is that of the actual firing time
  - This means we know the timestamp of any 'clean' pulse when pileup rejection is on

Pileup Rejection in Action

With pre/post delay at 2.56 us, apply pulse train at 390 kHz (every 2.564 us) from pulse generator. Poor quality pulse generator jitters, so sometimes pulses get rejected for pileup, sometimes they don't.

Frequency was varied. At 387 kHz (2.584 us), get 100% operation. At 392 kHz (2.551 us), get 0% firing rate.



Pending Event Queue

- Small circular buffer of timestamps – ***one per channel***
  - Event added every time discriminator fires
  - Simultaneously an ***Event Expired*** flag is entered into the 10.24us trigger delay buffer along with the data
- As each ***Event Expired*** flag falls out of the trigger delay buffer, read pointer is moved, effectively 'deleting' the expired event
- In Internal Trigger mode, every event is valid.
  - Current timestamp, offset by waveform pretrigger value, is compared against eldest item in Pending Event Queue (read pointer).
  - When match occurs, event readout from trigger delay buffer ensues for programmed length of readout
- In TTCL mode, receipt of trigger decision message initiates a search.
  - Loading and expiry of events is temporarily suspended (typically 100ns)
  - All pending events are compared against timestamp from trigger message with separate window settings + and –
  - All pending events that meet criteria are flagged for readout; then readout occurs as normal.
  - Code written as interrupt-driven to insure no missed events

Moving to next topic..

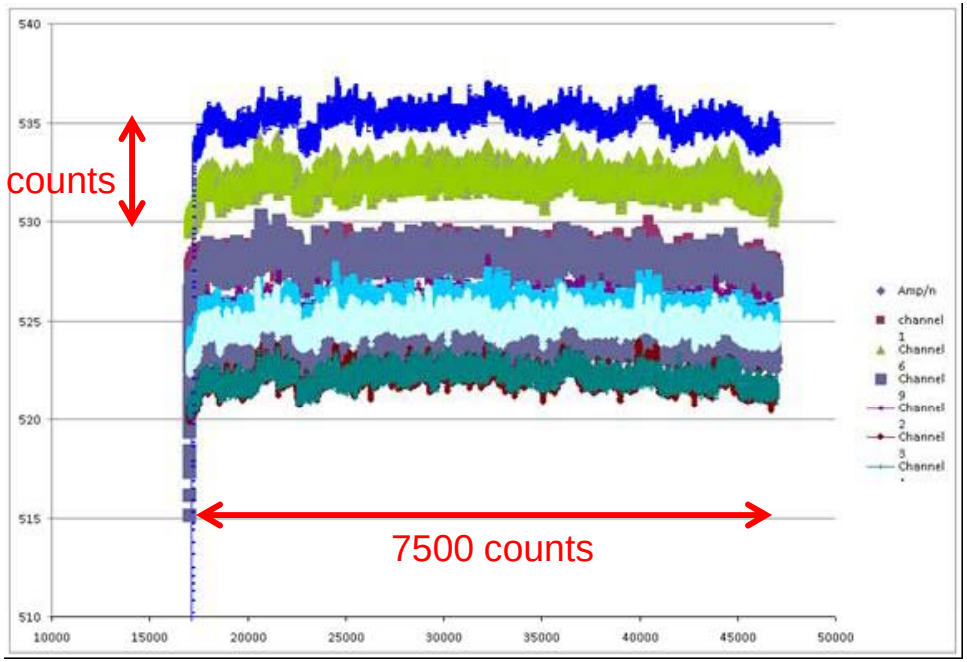
- System architecture (Gammasphere/DGS/DSSD)
- ANL digitizer firmware for DGS and DSSD
  - Digitizer Tester module
- Clock and Timestamp propagation across multiple systems
  - GITMO module
  - Accumulative jitter across larger systems
- Trigger firmware for DGS and DSSD
  - Cross-triggering between multiple systems
- Testability
  - LabWindows + EPICS : partners not competitors

Digitizer Tester Module

- Testing Module designed specifically for use in the digitizer/trigger system
  - Dual 16-bit, 200MHz DAC
  - Arbitrary waveforms generated from memory buffers in FPGA of module
  - Analog buffering and analog multiplexing allows signals to be routed to any channel of digitizer
- Connects to SERDES data stream from trigger system, can generate events under timestamp control or independently
- Used to measure digitizer performance, also to emulate experiment signals to test trigger algorithms.

• Plot of measured amplitude of fixed pulse with swept DC offset
  - Generated by tester
  - Measured by digitizer

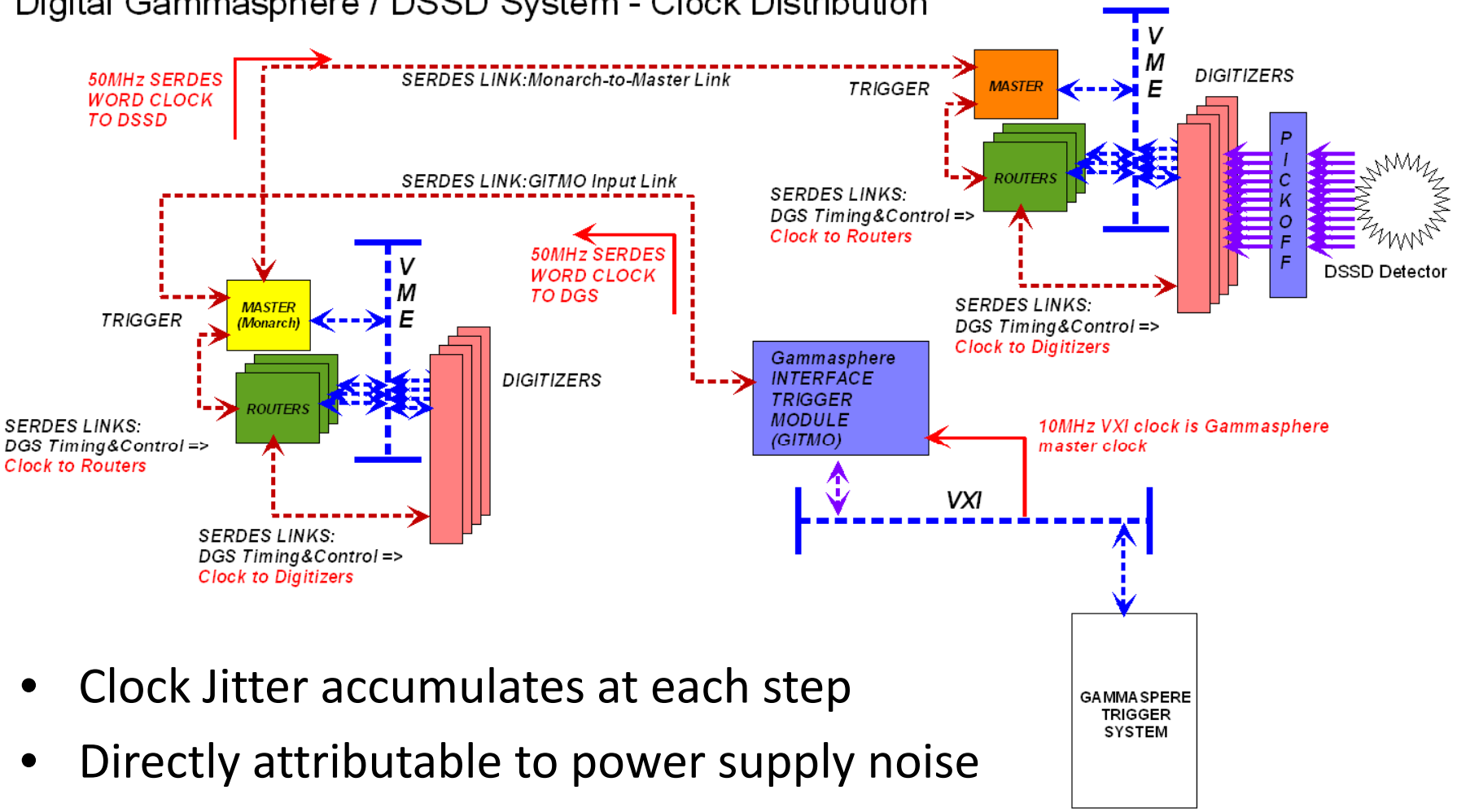
• Tester output noise very low, depends on power supply quality
  - digitizer LSB = 176uV
  - with good analog supplies, tester noise is less than 1LSB of digitizer



Moving to next topic..

- System architecture (Gammasphere/DGS/DSSD)
- ANL digitizer firmware for DGS and DSSD
  - Digitizer Tester module
- Clock and Timestamp propagation across multiple systems
  - GITMO module
  - Accumulative jitter across larger systems
- Trigger firmware for DGS and DSSD
  - Cross-triggering between multiple systems
- Testability
  - LabWindows + EPICS : partners not competitors

Digital Gammasphere / DSSD System - Clock Distribution



- Clock Jitter accumulates at each step
- Directly attributable to power supply noise
- Need filter on clock multiplexer in large systems
  - Triggers as manufactured : approx 110ps per ‘hop’
  - Triggers with added filters: approx 35ps per ‘hop’
- SERDES links drop bits when jitter > approx 300ps

Moving to next topic..

- System architecture (Gammasphere/DGS/DSSD)
- ANL digitizer firmware for DGS and DSSD
  - Digitizer Tester module
- Clock and Timestamp propagation across multiple systems
  - GITMO module
  - Accumulative jitter across larger systems
- Trigger firmware for DGS and DSSD
  - Cross-triggering between multiple systems
- Testability
  - LabWindows + EPICS : partners not competitors

Trigger Firmware in DGS and DSSD

- Overall architecture unchanged
- Routers simpler due to no ‘slow’ vs. ‘fast’ data
- Expansion of command protocol to support larger systems
  - Synchronous System Capture (frame #15)
    - Captures discriminator, trigger and readout statistics throughout system
    - Capture starts based on timestamp, programmable accumulation time
  - Trigger Internal Frames (#12 and #14) for test stand (digitizer tester, data generator)
    - Not expected to be used in experiments, designed for test stand
  - Definition of Frame #17 (for future connections to non-digitizer systems)
    - Have already discussed the “MyRIAD” VME module as a way to integrate additional systems
- Cross Triggering Rules established
  - Reduces each Master to only 5 local algorithms per experiment
  - Trigger decision slots in frames #8, #9, #10 re-purposed to allow propagation of trigger messages from one Master to another
- **Timestamps and trigger messages have controlled propagation**
  - Registers defined to control conditions where one Master is subservient to another and will propagate the “Monarch’s” data instead of its own (e.g. timestamp, trigger types, synchronous capture)

Moving to next topic..

- System architecture (Gammasphere/DGS/DSSD)
- ANL digitizer firmware for DGS and DSSD
  - Digitizer Tester module
- Clock and Timestamp propagation across multiple systems
  - GITMO module
  - Accumulative jitter across larger systems
- Trigger firmware for DGS and DSSD
  - Cross-triggering between multiple systems
- Testability
  - LabWindows + EPICS : partners not competitors

Improving Testability

- Have developed interface between LabWindows and EPICS
- Definition of EPICS PVs for each register allows LabWindows engineering test stand direct access throughout experiment
  - Single #define in the program selects “MXI mode” (local crate) or “EPICS mode” (access thru PVs)
  - Easy to build all the PVs using scripts or Python
- Allows use of every test stand diagnostic feature in the running experiment
- Provides direct comparisons during commissioning and deployment of new firmware ***and*** DAQ software
  - No end of times that one piece of software catches a bug in the other one
  - Reduces total software effort because end use DAQ software is released from having to build in any engineering diagnostics
- Extremely powerful tool!

Thanks for your time and interest!

Backup slides follow…

Current Compile Results

- Timing score : 0!
  - No setup or hold time violations against any clock
- Number of BUFGMUXs (main clock lines)
  - 5 out of 8 62%
- Number of DCMs (clock generation modules)
  - 2 out of 4 50%
- Number of RAMB16s (block RAMs)
  - 67 out of 104 64%
  - 5 BRAM per channel (50 total) plus 1 for board-wide FIFO (readout)
  - ***Includes 16 Block RAMs for built-in "Chipscope" logic analyzer***
- Number of Slices (area utilization)
  - 27155 out of 33280 81%