Experiences in Synchronization and Triggering

John Anderson

Argonne National Laboratory

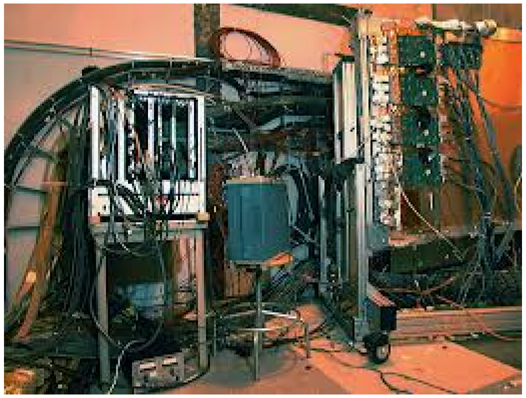
FRIB Data Acquisition Workshop
Argonne National Lab
July 29/30, 2015

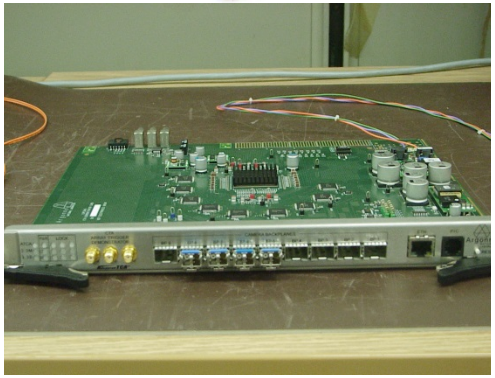
Talk Overview

- **Synchronization and clock distribution**
  - **Current hardware techniques**
  - **Clock-only versus a synchronization data stream**
  - **Empirical results: DGS+DFMA+Other setup**
- **Observations on triggering**
  - **Triggering in multi-detector systems**
  - **“Prompt” versus “delayed” triggers**
  - **Trigger Vetoes and Flow Control**

Hardware methods of synchronization

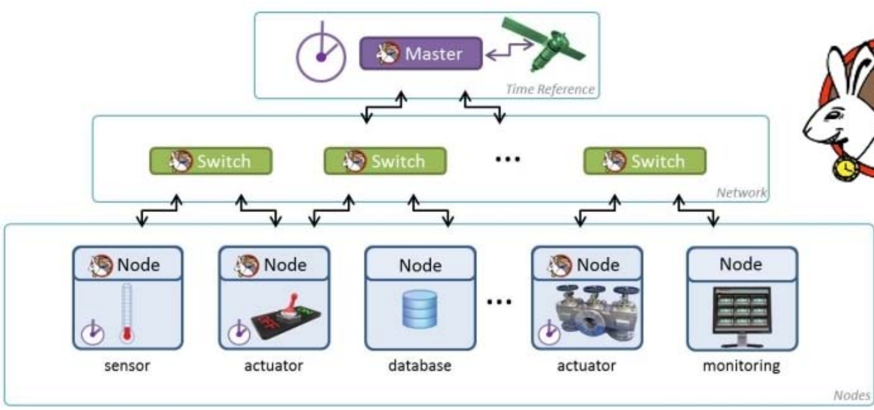
- **"Old school"**
  - One master clock source, use a bunch of fan-out and/or repeater modules.
  - Accumulates latency and jitter with each fan-out or repeater (fiber helps a lot).
  - Quality of clock may depend on how far away you are from the source (fiber helps here too).
- **GPS**
  - No explicit need for fan-outs or repeaters (just buy more receivers); used in astrophysics (CTA, VERITAS) and many other places.
  - Natively not ns-level timing. Needs additional hardware. Provides 1PPS and a "conditioned" oscillator. Cycle-cycle jitter may be large.
    - Addressed by many groups previously – well known issues.

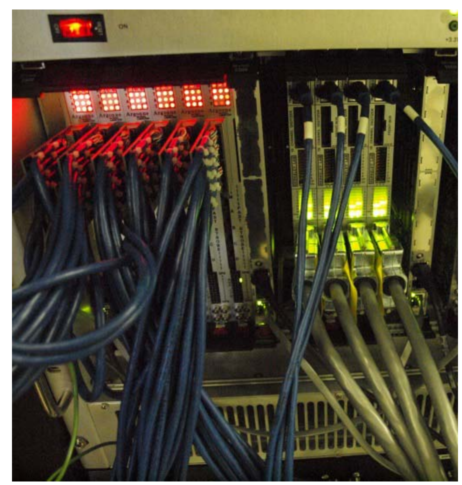


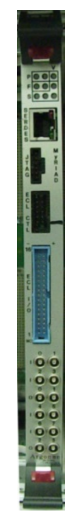


Hardware methods of synchronization

- **High-speed serial links**
  - Two basic types, embedded-clock and Ethernet-based.
  - Both work well , performance similar (<1ns jitter)
  - White Rabbit (http://www.ohwr.org/projects/white-rabbit/wiki/WRUsers) based on IEEE1588 has momentum.
    - Numerous HEP institutions working in this direction(CERN, DESY, GSI)
    - Requires special routers, moderately complex software
  - Modern FPGA SERDES blocks compatible with Ethernet only; require specific firmware for low-jitter, repeatable latency operation.
  - Embedded-clock has existing base in nuclear physics (GRETINA, DGS, DFMA, HELIOS, X-array)







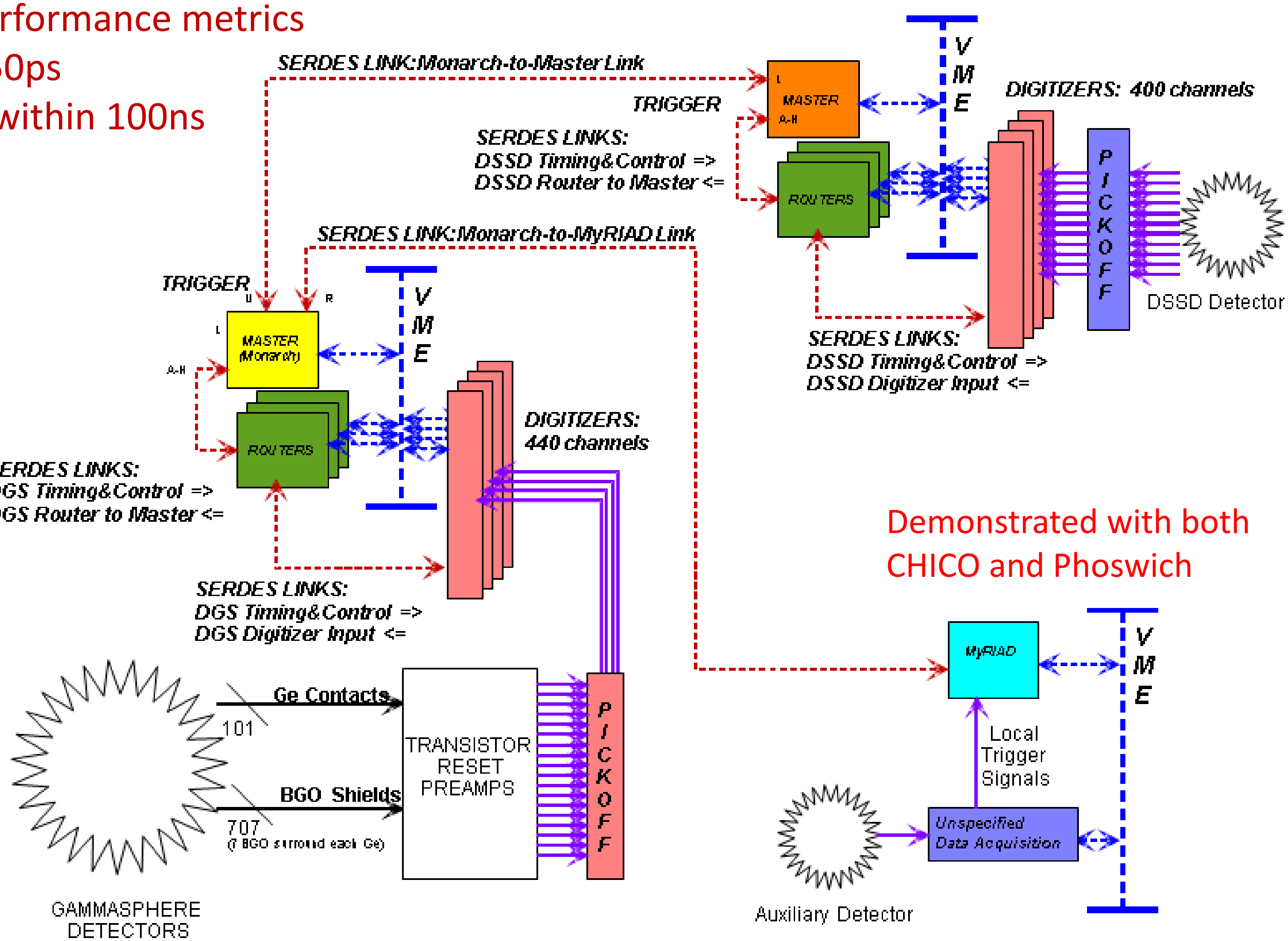
Clock-only versus a synchronization data stream

- All SERDES-based clock distributions allow for synchronous data transmission to all receivers
- Numerous uses for synchronous data in addition to clock
  - Synchronous resets/sync pulses allow for localized time-stamps on all data captured – greatly simplifies event collection & sorting.
  - Multiple trigger decision frames per cycle provides triggering flexibility
  - Synchronous *commands* simplify data quality monitoring by allowing measurement of parameters such as event rates at exactly the same time in all data sources.
  - Synchronous *commands* can also simplify calibration by causing pedestal/baseline measurements in all channels at the same time
- The DGS/Gretina command frame format has evolved to support all these features and is a currently existing, robust solution.
- **All SERDES links are by nature bi-directional, providing a natural technique for error collection, trigger formation or flow control.**

Multiple Detector Setup : DGS/DFMA/Other

Cross-system performance metrics
Clock Jitter: < 150ps
Event selection within 100ns

Demonstrated with both CHICO and Phoswich



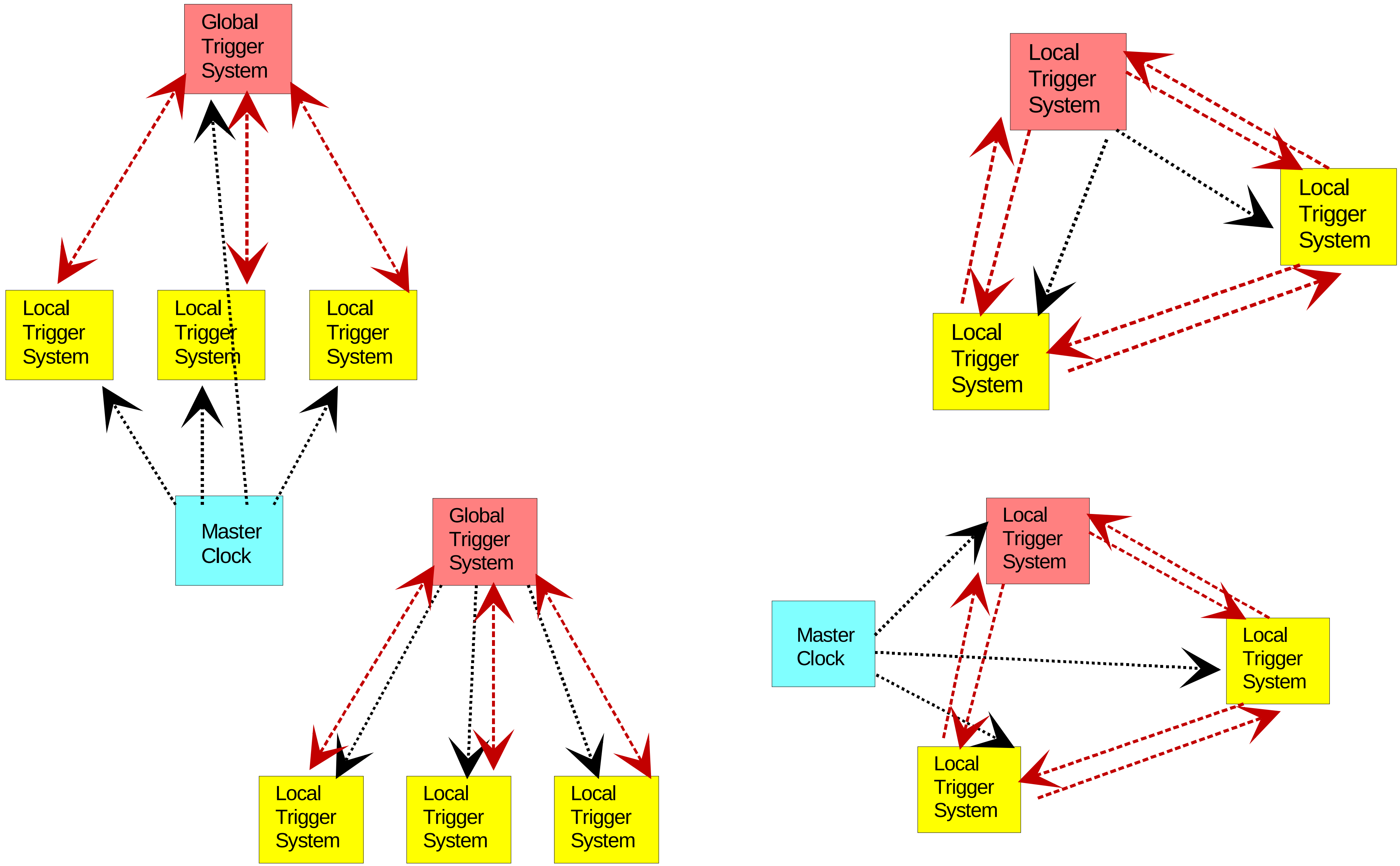
Other clocking considerations

- Clock system may (should?) tie to RF clock to provide correlation between beam and timestamps
- Diagnostic value in being able to run subsystems from either the local clock or the global clock
- Clock should be available in multiple physical formats (e.g. NIM, ECL for both legacy systems and diagnostics)
- As systems become interconnected more heavily, jitter transfer and accumulated jitter become important

Triggering - generalities

- Triggering can be simple logic or complex algorithms
- Multiple algorithms may be active simultaneously
  - Requires trigger type coding
  - Variant latency between multiple active algorithms can be problematic
- Trigger can be hierarchical or "layered"
- Definition of *combination* rules *between* local triggers is as important as the local algorithms themselves, and this is a logical layer boundary.
- Historically clocking system and triggering system are co-mingled, but in multi-detector systems this adds complexity.

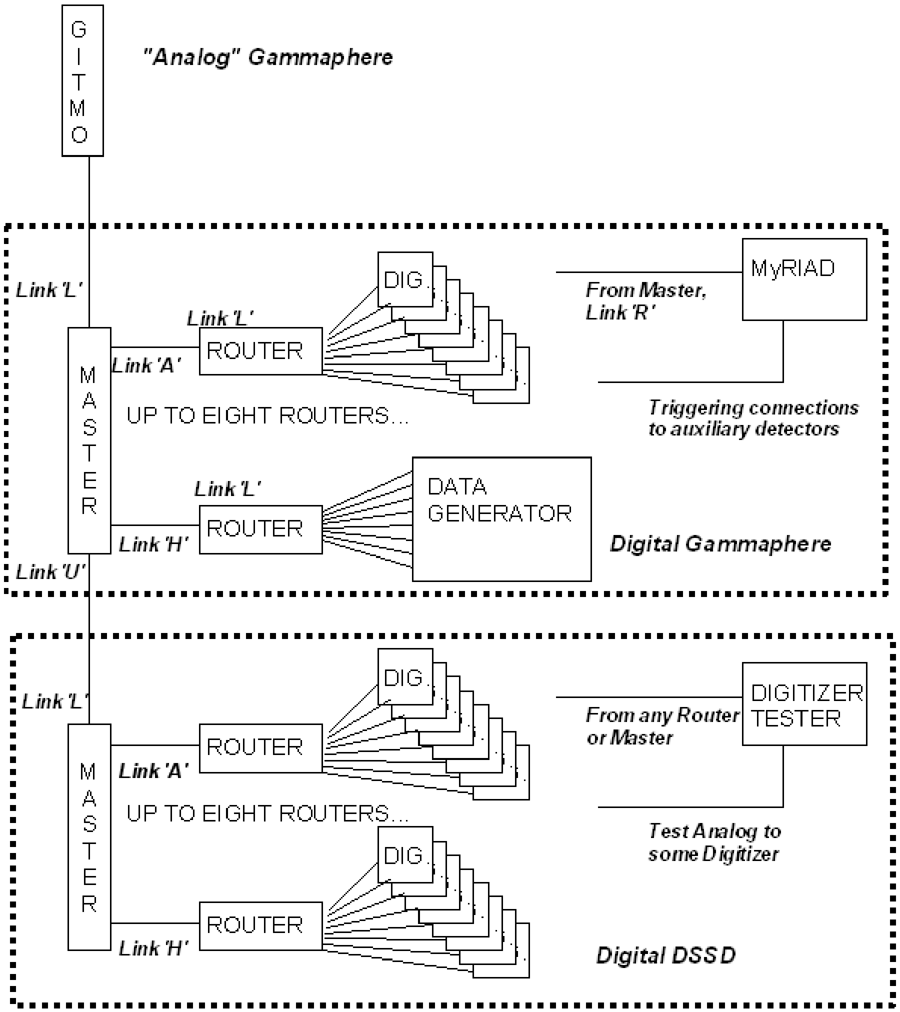
Topologies vary; common ground possible?

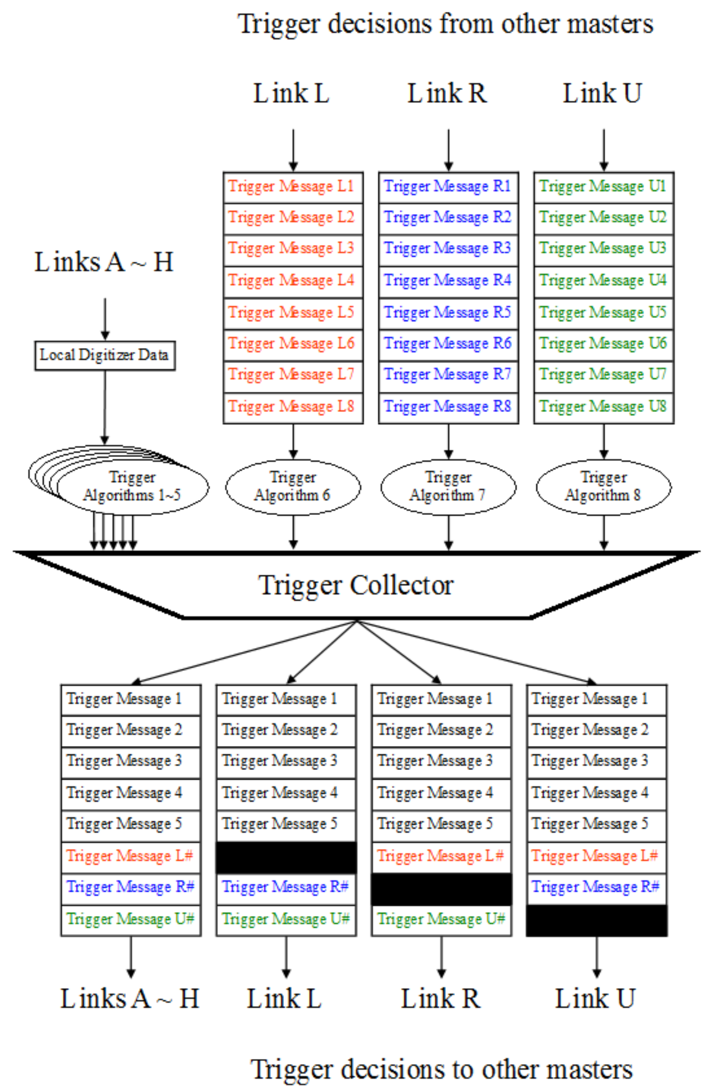


Triggering in multi-detector systems

- The common hierarchical model implies that each detector can generate a *local trigger* based solely upon information in that detector, whereas a *general trigger* may then combine information from all the *local triggers* to make global trigger decisions.
- **General issues:**
  - What must be done when the amount of buffering and/or the latency of trigger formation is mismatched between the detectors?
  - How do disparate detectors understand each others' timestamps for data merging?
  - How do disparate detectors become aware of and respond to flow control (readout cessation) present in one detector but not the other?
  - How do disparate detectors become aware of and respond to trigger veto conditions present in one detector but not the other?
- To address these issues, the *general trigger* must broadcast control information and manage propagation of received data from the *local triggers* into itself and to other *local triggers.*

Example: Clock distribution and cross-triggering in DGS/DFMA





Prompt vs. Delayed triggers

- A prompt trigger is sufficient if the measurement devices do not have deep buffers. If buffers are present the trigger is usually delayed.
  - **Integration and/or pileup logic in buffered digitizers can delay multiplicity information by microseconds after signal arrives at front panel.**
- **Experiences so far:**
  - In CHICO-GRETINA, CHICO-DGS and Phoswich-DGS-DFMA experiments, two general solutions used:
    - Implement a delay of the "prompt" detectors local trigger signal and tune this delay to match the delay of the buffered system.
    - Implement a separate "prompt" pre-trigger in the buffered system, do the coincidence in the unbuffered experiment's trigger, send the coincidence back to the trigger of the buffered system.
- **A valid question is where the line should be drawn between timing (hardware/firmware) and timestamp comparison (firmware/software) in forming cross-detector coincidences.**

Trigger Vetoes and Flow Control

- **Most detectors implement some kind of veto/gating logic to control the local trigger.**
  - As event rates increase, flow control becomes important. If the readout can't keep up, events have to be suppressed or buffers are overrun.
- **How much does one detector care about vetoes or flow control in another detector?**
  - How and when should the *local trigger* of any given detector report that triggers are suppressed (and *why* they are suppressed) to any global trigger system?
  - DGS/DFMA's master trigger-to-master trigger protocol takes the stance that it is better to always report all veto information to the "other detector" and let each *local trigger* determine for itself whether to veto or not when other detectors are vetoed.
- **How much does the event builder need to know about veto?**
  - In experiments with multiple detectors, is there a need for a detector that is in a veto state to report events that might have occurred, but did not because of that veto condition, to a global event builder?

Backup Slides

Details (too many?) about cross-triggering

- **Cross-detector triggering is implemented by having a full bi-directional copy of the entire master trigger command stream sent between the DGS master and the DFMA master. Each of the masters has a "propagation control register" that allows selection of which parts of the command stream from the "other guy" are allowed to propagate *into* a master's local domain.**
  - This concept of propagation control allows frame-by-frame selection of which kinds of trigger messages can be passed. Thus it is possible to set one detector to have multiple *local* triggers but have the other detector only listen to a subset of those triggers.
  - Propagation control is also applied to clock/timestamp in a consistent fashion.
  - Propagation control also applies to system-wide command frames (e.g. internal trigger messages, system synchronous capture, digitizer asynchronous commands).
- **This is fully and independently implemented on links L, R and U of all masters so that a given master could receive and re-propagate triggers and commands from up to three different sources. Each source may send one of two command formats:**
  - Master trigger command format (all 20 frames) or
  - "GITMO" format (single frame repeated every 100ns), for more time-sensitive triggers.
- **The MyRIAD sends messages when the local trigger signal is asserted, so any auxiliary detector can now trigger DGS or DFMA directly through the MyRIAD.**
- **Trigger propagation can hop multiple systems. CHICO could send a trigger to DGS and that could be re-propagated to DFMA. Each system-to-system hop takes about 1.5usec.**
  - Time windows within digitizers are still one per channel, so local triggers may need to be delayed with respect to remote triggers to insure correct selection in TTCL mode.
- **Re-propagated triggers are tagged with a different trigger type code than local triggers, that shows up in the digitizer header.**