

Trigger Task Force

Kickoff Meeting
Scott Oser & Dave Toback
August 1, 2014, 9am PDT

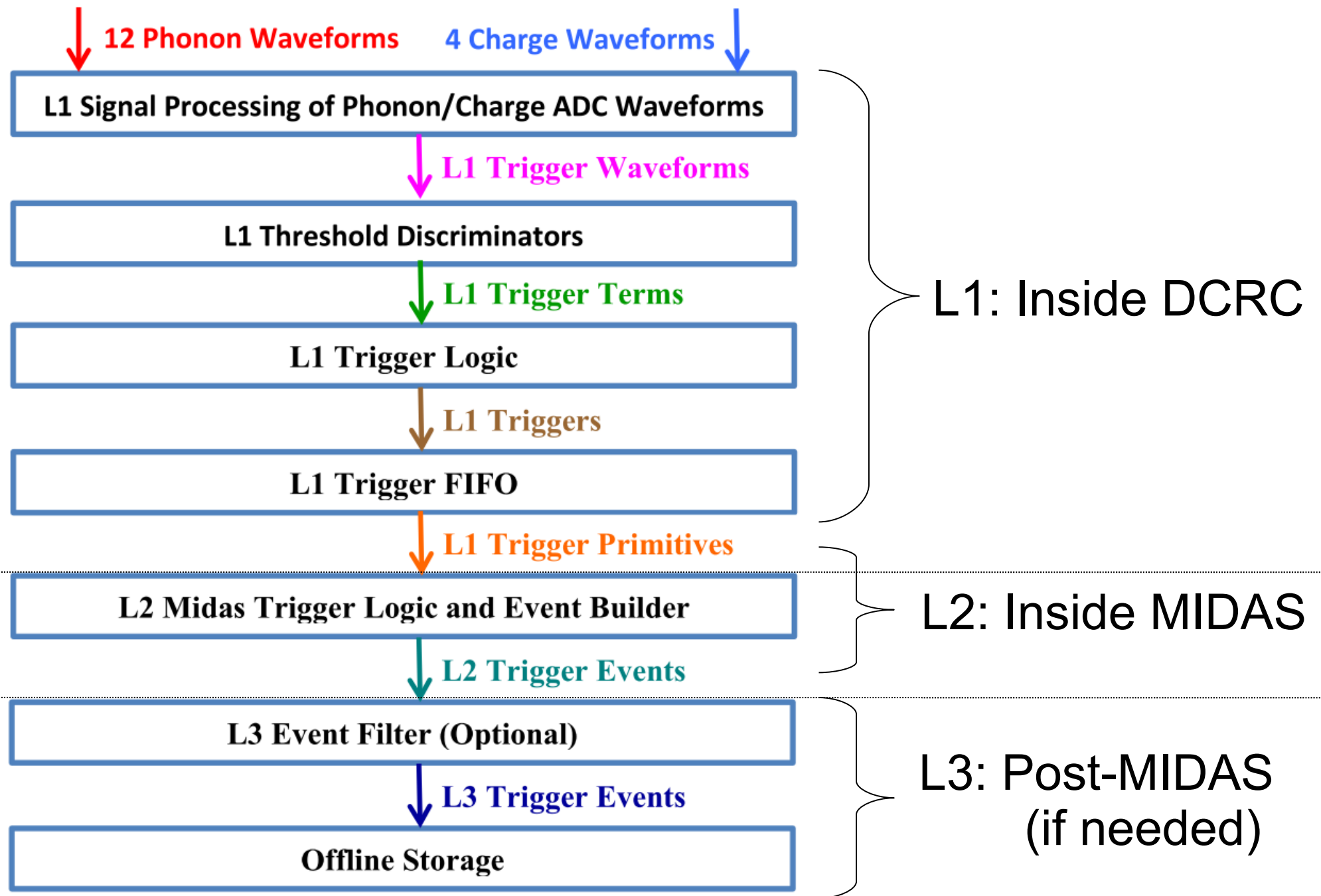
Task Force Members

Co-chairs: Scott Oser, Dave Toback

Members: Anders Borgland, Ray Bunker, Jeter Hall, Lauren Hsu, Ben Loer, Matt Pyle, Jon Wilson

Consultant: Sten Hansen

Ex-officio: Dan Bauer, Richard Partridge



Trigger information available at different levels of the system

- L1: full access to waveforms for a single DCRC. Fast, real-time, limited only by what algorithms can be effectively implemented in an FPGA. Can be run on individual channels or on sums of channels.
- L2: small number ($< \sim 30$) of L1 trigger bits for each trigger from every DCRC, with time stamps. Read once per second by polling DCRCs' trigger FIFOs. No waveform information available.
- L3: full waveforms available for each DCRC, post-readout. Do we need triggering here if we can handle the data volume? Complexity limited only by CPU available for realtime post-DAQ processing.

Task Force Deliverable

Write a document answering a set of specific questions, by October.

What follows is the list of these questions and some initial thoughts/further questions on them.

Names in blue are our guess of who may either be the most interested or essential to a question. If you think your name should be added/deleted to any, just ask!

“Develop a plan for the DCRC firmware that identifies both immediate and long-term capabilities, including the range of functionality to be accommodated in the L1 Signal Processing, L1 Threshold Discriminator, and L1 Trigger Logic blocks.”

Key questions:

- What signal processing algorithms are wanted?
 - Current DCRCs trigger on fast deviations from a baseline average, with a programmable threshold, for individual channels.
 - Some people have strong interest in optimal filtering in FPGA, which will use pulse shape information. Ultimately equivalent to a weighted average of recent samples.
 - Do we trigger on phonons only? On charge as well? Are triggers applied to individual channels or sums to channels?
- Who will write the firmware, especially more complex algorithms?
- What is the testing scheme for the firmware? Buggy trigger algorithms at L1 level seem the most dangerous failure in trigger, due to specialized, non-intuitive nature of firmware. Most of us can't even read the code.
- Do we need a trigger simulation?

Richard, Matt, Sten, Jon, Dave

“Identify key DCRC firmware parameters such as the maximum numbers for the various trigger elements (L1 trigger waveforms, L1 trigger terms, L1 triggers) and the depth of the L1 Trigger FIFO. Based on these key parameters, develop a register map for the DCRC that meets both near-term and long-term requirements.”

Sten, Jon, Jeter, Scott

Some definitions:

- L1 trigger waveforms: the either raw or summed/shaped waveforms used by the FPGA-based trigger algorithms
- L1 trigger terms: sub-trigger produced by algorithm for some threshold, either passed to MIDAS or combined with other trigger terms in logic inside DCRC
- L1 triggers: trigger passed to L2, formed from L1 trigger terms

Trigger FIFO: length determined by polling interval (~1 second) and maximum raw interaction rate in detector. Likely on order of 100,

Strawman register map already exists (Jon Wilson)

“Identify the trigger primitives to be stored for each L1 trigger.”

Key questions:

- How many algorithms will the FPGA run? (We can likely run multiple trigger algorithms simultaneously.)
- How many bits will we pass to L2 trigger for each trigger?
Strawman: one optimal filter trigger on the sum of phonon channels, applied with a normal and super-low threshold (2 bits), as well as a threshold algorithm applied to the sum of phonon channels as well as to individual phonon channels, at two thresholds (26 bits). Total of 28 trigger bits. (Is this doable?)
- Sending more bits increases complexity and decreases speed of L2 trigger. Probably should keep ≤ 32 bits. The bottleneck is probably the speed at which trigger primitives can be passed between the tower front-end nodes and the central trigger processor in MIDAS. (I'm worried about clogging ODB with trigger bit transfers.)

“Develop a preliminary plan for incorporating the neutron veto into the trigger.”

Neutron veto group should make a proposal.

Possibly not a “veto” at all, but rather a tag. Ge trigger may force neutron veto readout, and vice versa.

This must be done at L2 level.

Ben, Scott

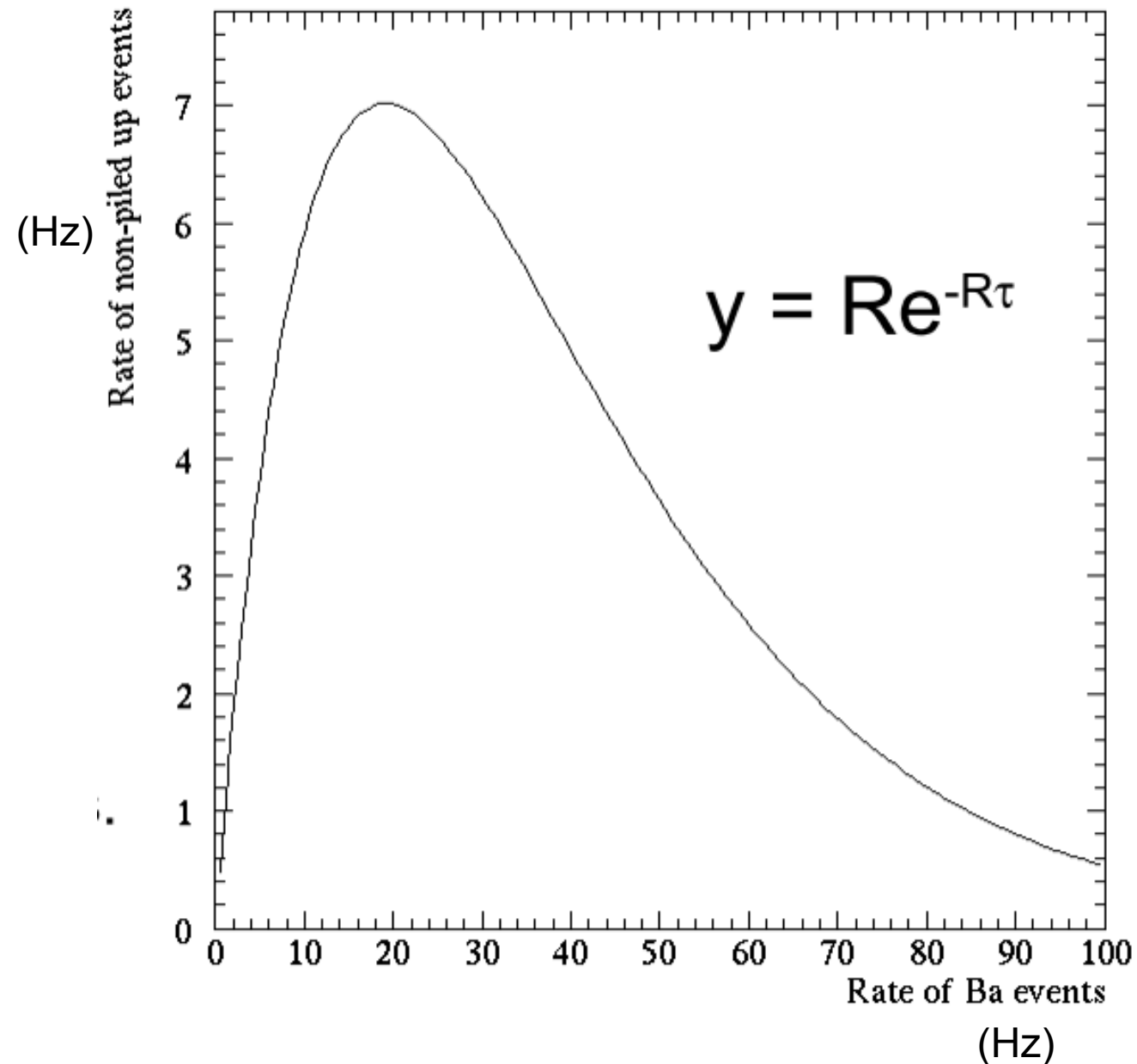
“Define the capabilities required for the L2 Midas Trigger Logic and Event Builder.”

Ben, Scott

Some basic requirements:

- Pileup rejection! (See next slide.)
- Trigger formation time must be “fast enough” that trigger polling interval + L2 formation time + readout time is less than length of DCRC waveform buffer (~3s).
- Trigger timestamping from L1 level is important to make sure we're not fooled by “expired triggers” whose data has already been overwritten
- “Lookback” capacity to form time coincidences between latest and recent triggers.
- Combined triggers between any two detectors, eg. coincidence of nearest neighbors, or readout of “super-low” threshold triggers if another detector has triggered, to look for multiples that are below the standard threshold
- Event builder: merge waveforms into physics events, through time clustering and matching by triggers, and monitoring of “deadtime” (in this context, triggers that could not be read out due to time limitations)

Pileup limitations



We will read out 52ms trace lengths (needed for filtering low-frequency noise). This limits usable total interaction rate to 7Hz per detector, due to pulse overlap. Piled-up events will have different noise properties and so are not useful!

We cannot boost the rate at which you collect special triggers by increasing the source strength and cutting harder in the trigger!

“Develop a plan for low-background triggering, including trigger threshold, rate requirements, and the applicability of selective readout for low-background triggers.”

“Full readout” : read out all detectors whenever one triggers

“Selective readout” : read out only triggering detectors

“Loose readout” : whenever one detector triggers at a normal threshold, read out any detector that triggered above a lower threshold

Key questions:

- Is full readout ever needed (in which we read out all detectors whenever any one triggers)? Would “loose readout” mode provide a useful compromise?
- How do we choose the standard trigger threshold?

Ray, Lauren, Matt, Dave

“Develop a plan for calibration triggering, including trigger threshold and rate requirements, pileup rejection capabilities, and application of selective readout.”

- This mode ultimately determines DAQ throughput (data rate)
- Maximum usable trigger rate of 7Hz (from pileup) sets an upper limit on source strengths and on usable trigger rate.
- Achieving this 7Hz only possible with pileup rejection.
- Selective readout (reading out only triggered detectors) is almost certainly necessary for standard calibrations: is there ANY calibration mode that would read out non-triggering detectors?

Lauren, Ray, Dave

“Develop a plan for random and/or noise triggers.”

What rate?

Interspersed through run, or just at start/end?

Do they need to be truly random, or is periodic sufficient?

Jeter, Ray, Lauren

“Identify the types of 'special' triggers that should be included and how they might be implemented within the trigger framework. Such special triggers may include: nearest-neighbor-double triggers, energy-window triggers, track triggers, and charge/phonon pulser triggers.”

Key questions:

- If trigger rate ultimately limited by pileup, and we can handle the data volume, do we need any fancy triggers? Can we just use the “normal triggers”?
- If we want fancy triggers, who will write them? Can we make it easy for non-experts to code up new triggers?
- Are there special triggers needed for detector tuning/testing?

Lauren, Dan, Scott

“Develop a plan for providing a history buffer that provides a history of triggers prior and subsequent to the triggered event, and specify the pre and post trigger depth.”

It is probably easy (and wise) to write all L1 trigger primitive information to the data stream, so full trigger history should always be available to analysts.

Key questions:

How will this history buffer be used?

At what level of the trigger is it needed?

Jeter, Lauren, Ben, Anders

“Develop a plan for pre-scaling triggers to allow sampling events from a high-rate calibration trigger, sampling noise or neutron veto triggers, and testing a new trigger configuration at low rate.”

Pre-scaling sounds simple in principle ... but does limitation on maximum total rate from pileup to $<7\text{Hz}$ make it useful?

What is the motivation for pre-scaling?

Ray, Lauren, Dave

“Assess the need, functionality, and requirements for the L3 Event Filter, including expected data rates and offline storage requirements with and without the L3 filter.”

Driven by data volume, obviously.

Can L3 filter run on surface? Could it be done post-data quality?

Anders, Ray, Lauren