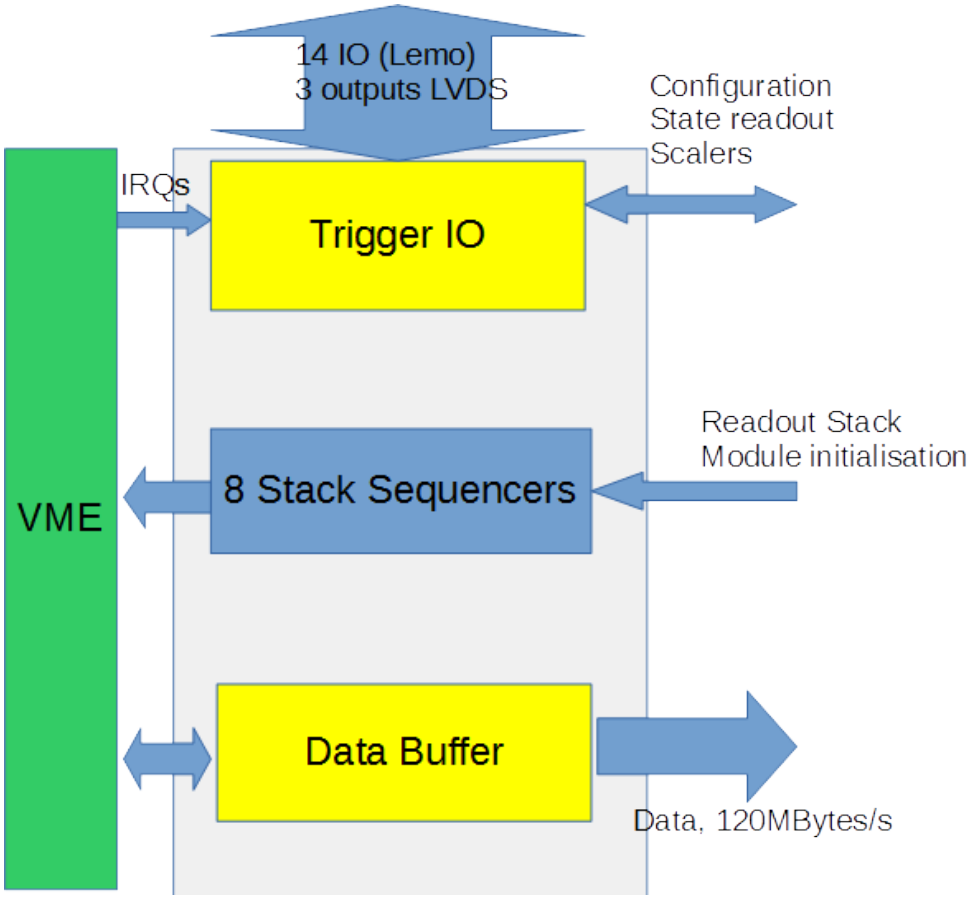
mesytec **MVLC** is a fast, low latency VME64x (compatible to VME64) controller with a powerful integrated trigger module. It provides USB2/3, and an SFP cage for Gbit copper or fiber connection. Sophisticated list processing allows low latency times. Highly optimized MBLT mode for up to 120 MByte/s via VME bus. MVLC can be used to synchronize multiple VME crates.

Hardware features:

- **High speed connectivity:**
  USB2 and USB3
  SFP cage for 1G copper or fiber Ethernet
- **Trigger options:**
  14 Lemo Trigger inputs/outputs (TTL or NIM in, NIM out),
  3 PECL/LVDS outputs
- **List processing for lowest latency** (Very fast response time from trigger to VME readout)**:**
  Trigger to first VME-access < 100ns.
- **Provides complex configurable trigger logic and gate generators:**
  timing delay and width of trigger inputs and outputs (precise to 1ns)
  14 IOs can be configured as trigger inputs, outputs, gate generators.
  Inputs can be delayed, coincidence units, logical operations.
  All signals can be output with delay and width (up to 64 us, 1 ns steps)
  Internal oscillators with different frequencies (synchronous over several crates when synchronization is active)
  8 x 64 bit universal counters with capture
- **High data throughput:**
  up to 120 MBytes/s with MBLT transfer,
  up to 2 MHz readout cycles: (cycle: Iack, MBLT- 16Bytes, End cycle)
- **Multi crate synchronisation** (Software Option, to come):
  One master MVLC synchronizes any number of MVLC slaves
  Results in synchronized VME back plane clock
  Synchronous command transmission to start synchronous VME actions on multi crate setups

mesytec GmbH & Co. KG
Wernher-von-Braun-Str. 1, 85640Putzbrunn, Germany
phone: +49 - 89 / 456007-30 fax: +49 - 89 / 456007-39
info@mesytec.com – www.mesytec.com

MVLC Block Overview



The left side shows the VME crate while the right side shows the control from PC.
The top shows input from the front panel IOs and the 6 IRQ lines from the VME backplane.

When the data acquisition is started, the VME modules are initialized, via a series of writes to internal registers.
Then a readout stack is built. It contains a list of VME read commands which are subsequently executed to get data from the modules. The received data is formatted to a well defined event structure. A stack runs once, then has to be started again. Start triggers can be initiated by VME IRQs or any signal constructed in the Trigger IO system. Usually a stack is started when there is enough data in the module buffers.
It is also possible to start stacks periodically when there are for example counters or scalers to read out.

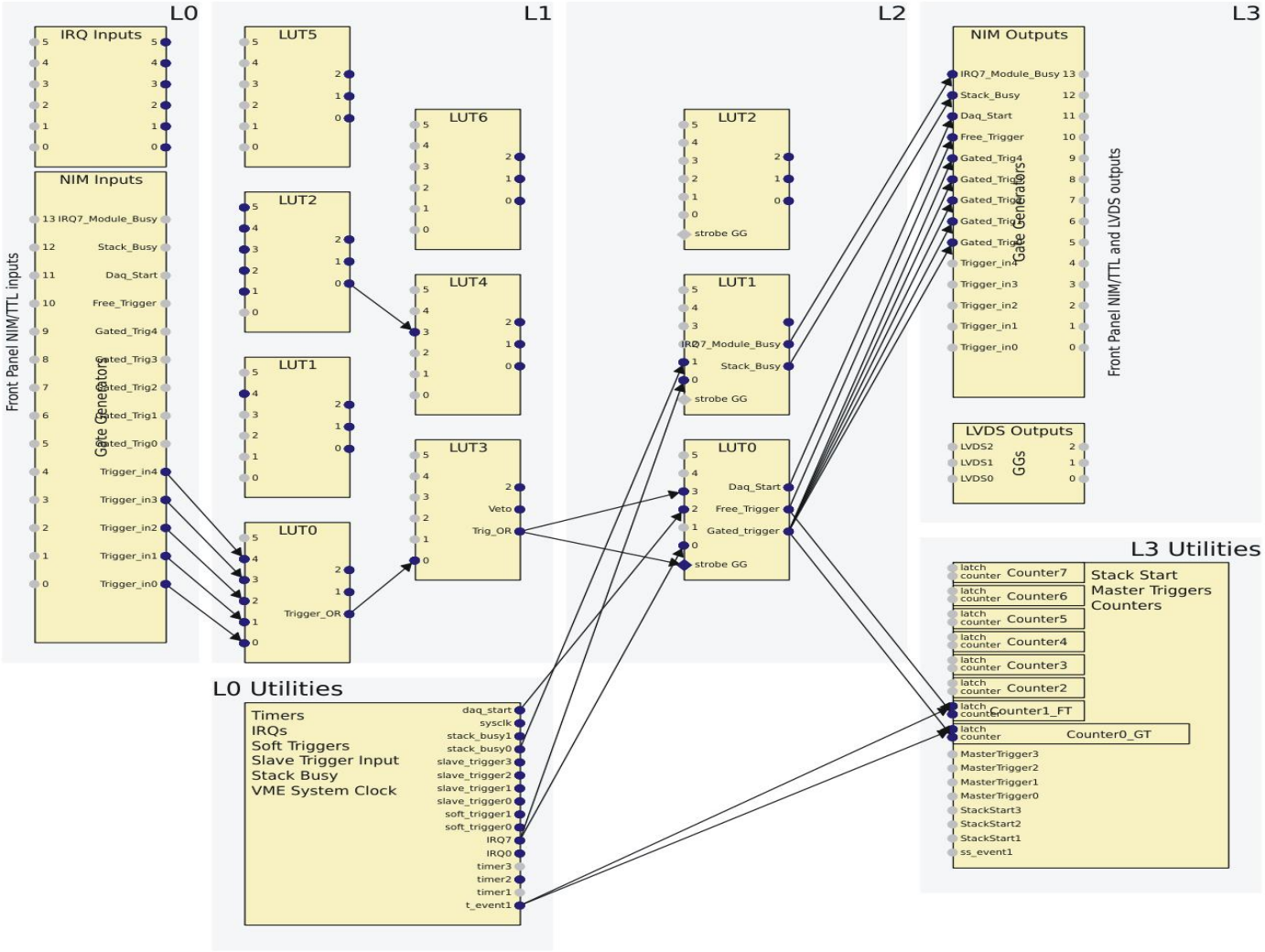
All registers of the MVLC module can be addressed by writing or reading data at the reserved high Address 0xFFFF.
This way also counters in the Trigger IO can be read via stack.

Trigger module

MVLC includes a powerful trigger module.

The internal logic samples the input signals with 1GHz, so a timing jitter of 1ns is introduced.
Output signals originating from the same logic source are strictly synchronous.

Internal Structure of the trigger IO



IOs

The left side (Level 0) shows the input side of the MVLC module, The right side (L3) the output side. The 14 Lemo I/Os can be configured as input or output. In addition there are 6 VME-IRQ inputs which sense the IRQ lines on the VME backplane. Some of the mesytec modules allow to put trigger signals to those lines. They behave like wired OR lines, so all modules in a crate can add signals to a line. As output 3 LVDS lines are available. They can be used to send the same signal (experiment trigger, Clock, reset) to all modules via cable chain.

Gate Generator

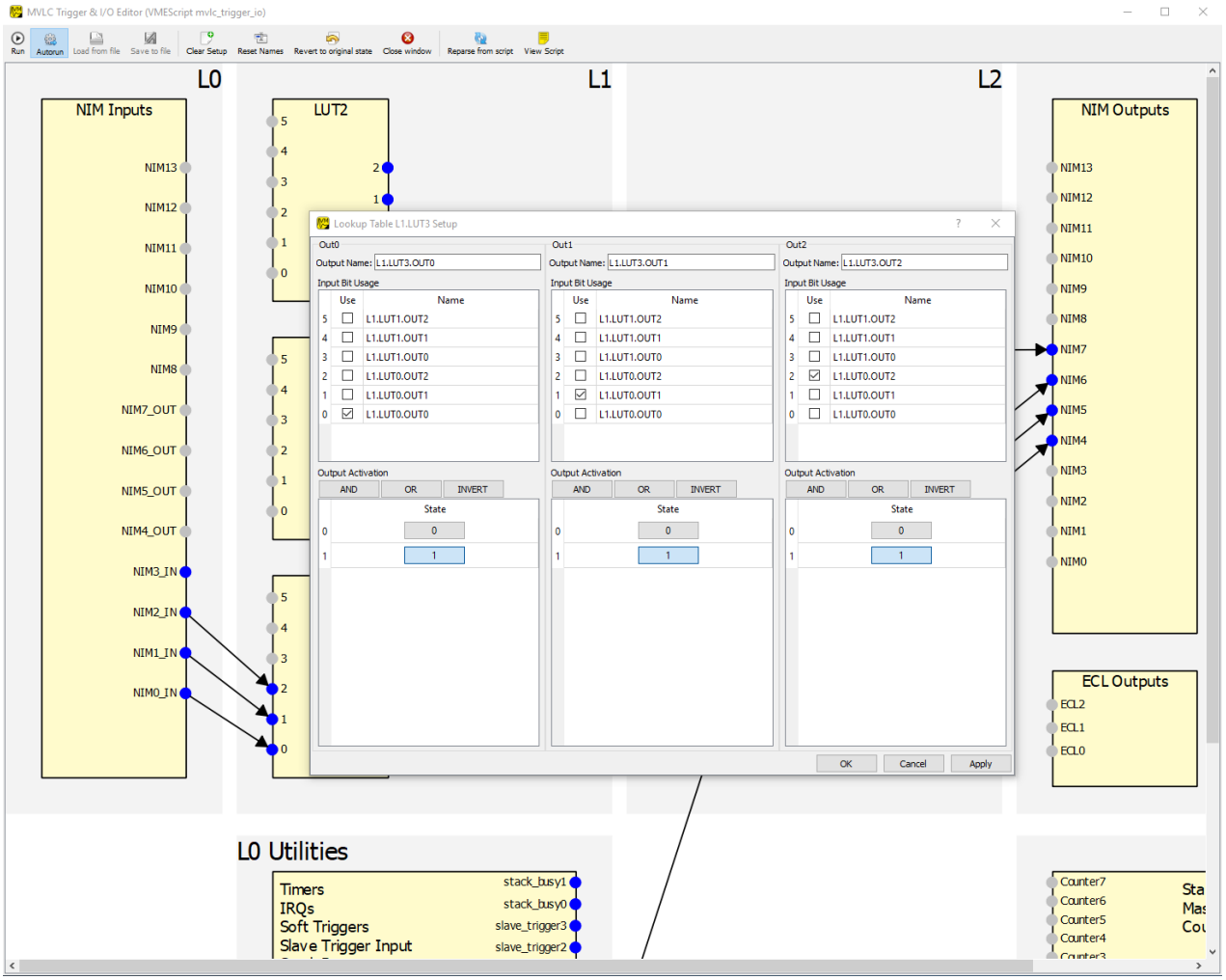
When configured as input the NIM IOs are configured on the left side, as output they are configured on the right side. Each I/O has a signal processing unit (“Gate Generator) which allows to delay the incoming or outgoing pulse, set its width and add a hold off time (blocking time). The hold off time is useful to block trigger bounce or limit a trigger rate. When configured as input the signal is scanned with a 1GHz clock. All delays width and hold off can be set with a resolution of 1ns in a range of 8ns to 64us. When the width is set to 0. the Gate generator is transparent, so inactive.

Lookup tables (LUT)

LUTs for logic processing of the input signals 10 units for universal logic are implemented. Each element can do any logic operation of 6 inputs providing 3 results on 3 outputs. (including “and”, “or”, counting input bits...)

On level 2 the LUTs also provide an optional strobe input. When an edge is detected at this input, the evaluated logic result is sent as a pulse to the output. So a “clocked” output with timing determined by the strobe input.
Lookup tables with strobe

Editing a Lookup table:



Shows the edit window for a L1 LUT.

L0 Utilities: (outputs)

some useful signals are available at this unit;
daq start: is activated after the module start scrips are executed.
This allows to free the event trigger when the modules are ready to process triggers.
Sysclk: The VME backplane clock
Stack_Busy0/1: Allows for example to latch a counter before readout by a stack.
Slave_trigger0...3: For multi crate use. Allows to get synchronous signals from the master MVLC to all slave MVLCs.
Soft_trigger0/1: can be set by MVLC-register, also from executed stacks. Can be configured to show a pulse or a state.
Irq0/1: gives access to two VME IRQ lines out of 7. Can be used to start a stack in the data acquisition.

The internal event stamping FIFO has reserved internal IRQs 8 and 9. They can be configured to “data available threshold” → start readout , and “FIFO almost full” → stop event trigger.
Timer0...3: allows to generate a frequency from 1 pulse per 18hours down to 41.66MHz.
They can be used to start stacks generating a 1s readout, or may be connected to a counter to create a time stamp.

L3 utilities: (inputs)

counters0...7: can count any input pulses, 64 bit wide. Can be read out in stack execution.

A latch input allows to hold a copy of the counter for read out. The counter continues to count.

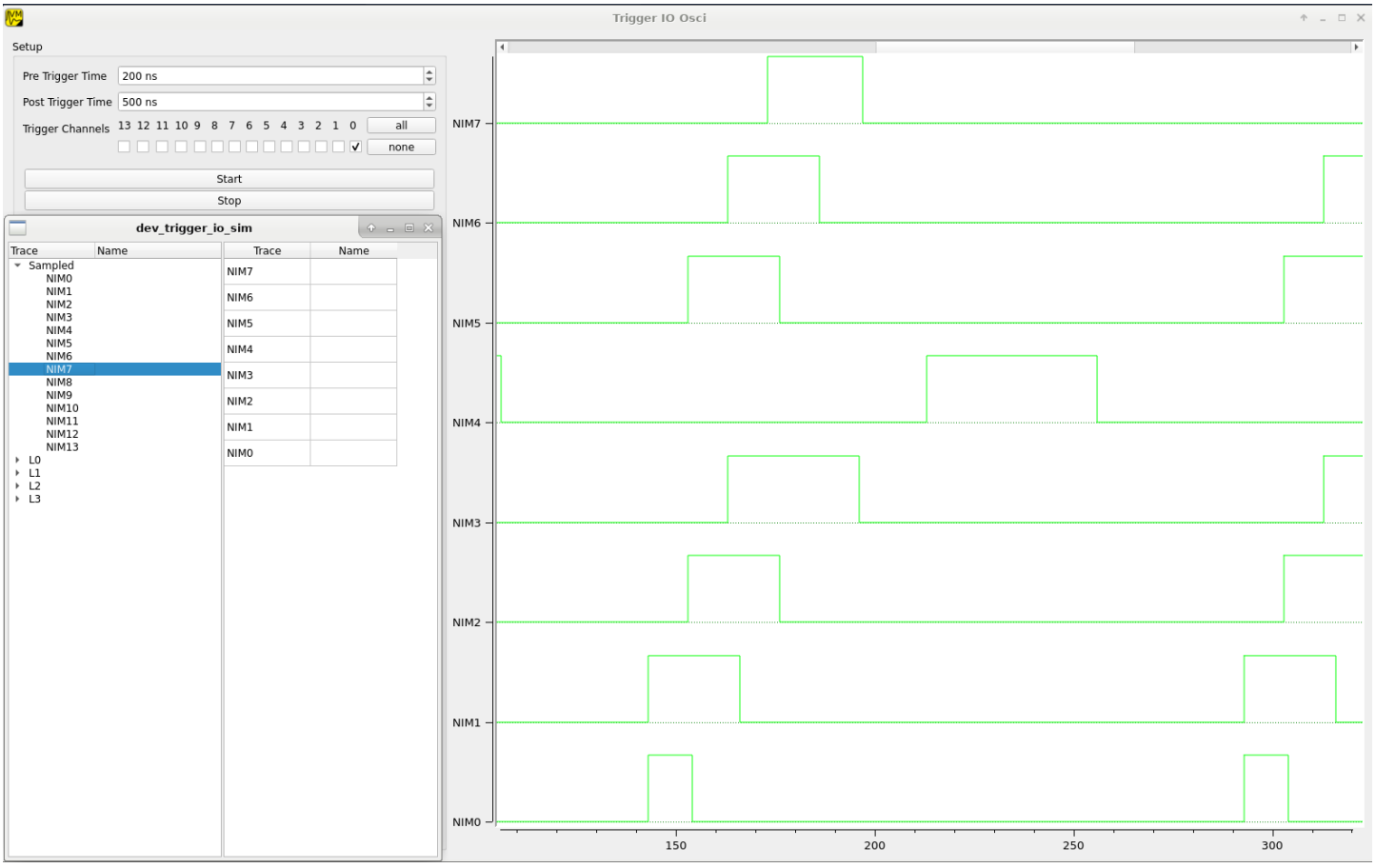
Counter0 has a special feature: it can generate time stamps and forward them into a 8k FIFO when latch detects a rising edge. The time stamp FIFO is described later.

Master_trigger0...3: signals are forwarded to the slave trigger input of all connected MVLCs. Also for the master module itself. Needed for multi crate operation.

Stack_start0..3: A logic=1 signal or edge starts the associated stack.

Digital signal Oscilloscope and simulator

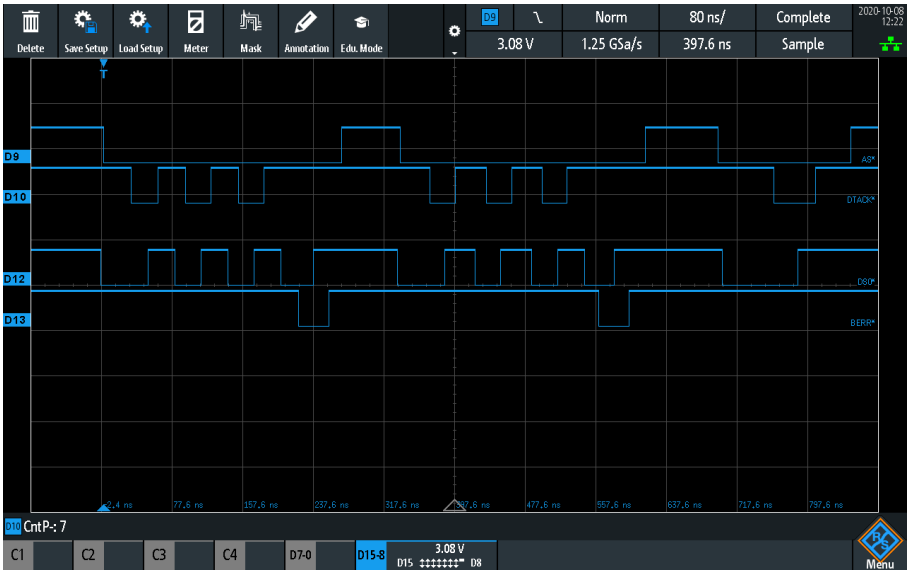
For debugging and adjustment of the timing relation of input triggers and trigger logic, MVLC together with mvme provide a digital signal oscilloscope including a simulation of the configured delays and logic. This allows to inspect a number of inputs, internal nodes and outputs.



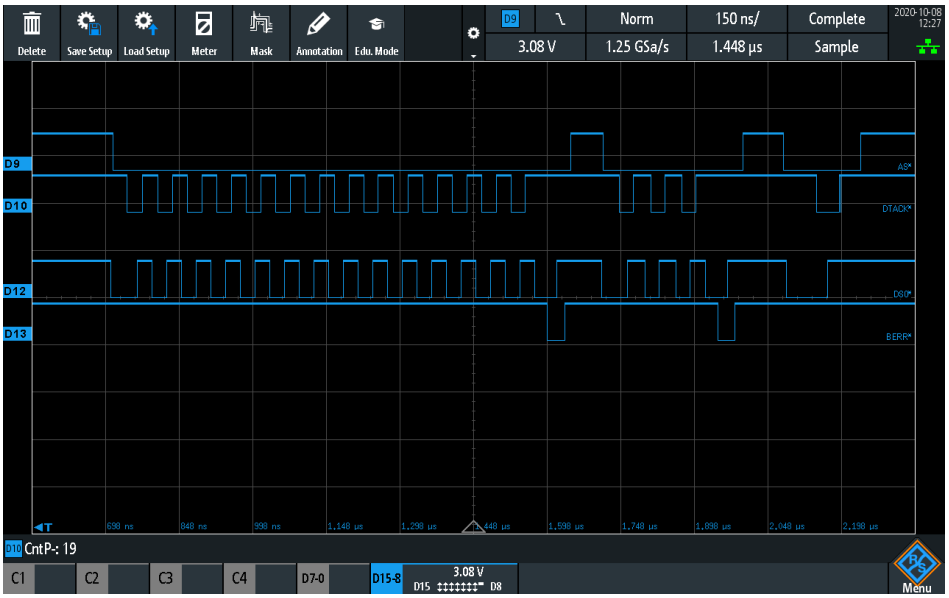
Digital oscilloscope and simulator for Inputs, outputs and any nodes in the logic

Timing performance:

Readout via fast MBLT cycle: 60ns cycle time. The MBLT transfer of MVLC and mesytec modules is fully VME64 conform. The theoretical VME specified data rate via MBLT is 160MBytes/s.



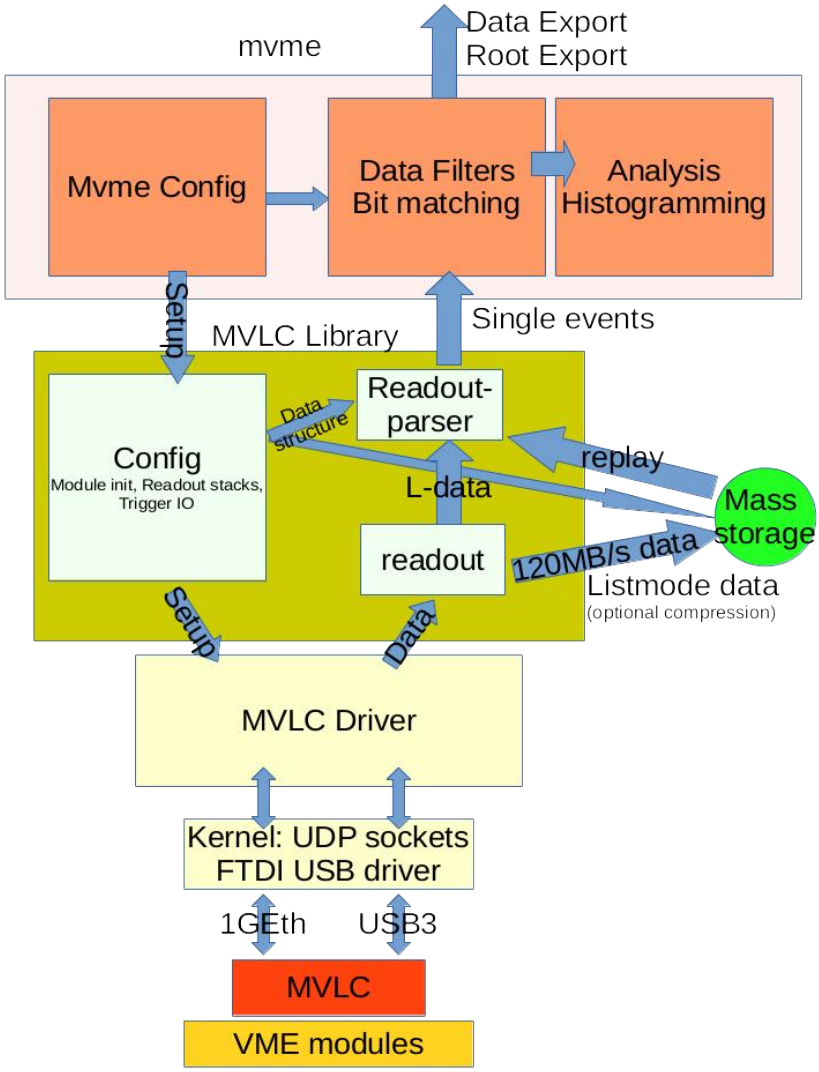
Cycle time for readout of 2 MTDCs with minimum event length, each transmits 4x 32bit words of data
Readout via VME MBLT mode. Total transmitted data rate: 20 MBytes/s



Cycle time for readout of 2 MTDCs. First one transmits 34 words, second one 4 words.
Readout via VME MBLT mode. Total transmitted data rate: 88 Mbytes/s

For long events (200 words) up to 120 Mbytes/s can be transferred from VME via Ethernet to the PC.
Via USB3, 126 MBytes/s were measured.

mvme software architecture overview



mvme

- mvme is the Qt-GUI based DAQ solution by mesytec
- Easy VME setup (includes templates for all mesytec VME modules)
- GUI based analysis system with flexible data extraction and basic histogramming and statistics functionality
- Tools for debugging the VME setup and monitoring the incoming data
- Built on top of the mesytec-mvlc library implementation
- Open Source (GPLv3) and actively developed

mesytec-mvlc library and driver

- C++ userspace driver library for the MVLC controller
- Low-level USB and UDP driver implementation
- Higher level abstractions for direct VME access and MVLC setup
- Simple VME module and readout configuration either in code or via YAML config files
- Multi threaded readout worker with ZIP or fast LZ4 data compression built-in
- Readout parser for interpreting the readout data (low-level “event building”) and handing the data to user-defined analysis code
- Self-describing list file format
- Interoperability with mvme
  - mvme setups can be converted to mesytec-mvlc YAML configs
  - List files recorded using the library can be opened and replayed in mvme
- The library contains example code showing how to build a minimal DAQ system
- Reference implementation for the MVLC, actively developed and maintained
- Permissive Open Source license (Boost V1)