GDDAQ

发行版本 *V20241025*

Hongyi Wu(吴鸿毅)

2024 年 10 月 25 日

Contents:

Welcome to GDDAQ’ s guides.

CHAPTER 1

Introduction

1.1 Version

- **GUI Qt: The program is in the stage of rapid version iteration. Welcome to download and use it.**

Download the latest version: PKUCAENDAQ

Manual: 简体中文/English

If you have any comments or suggestions for this project(function addition or improvement), please contact Hongyi Wu (wuhongyi@qq.com / wuhongyi@pku.edu.cn).

1.2 About

This program supports CAEN digital 2.0 data acquisition modules, including 2745/2740/2730. Support mixed use of PHA/PSD/ZLE/DAW/SCOPE/OPEN firmware for different modules. The implementation of general external logic requires the V2495 module.

Technical adviser:

- Zhihuan Li 李智焕

Software Developer:

- **2021 - now**
  - Hongyi Wu 吴鸿毅 (wuhongyi@qq.com / wuhongyi@pku.edu.cn)

CHAPTER 2

Program installation

This program installation requires reliance on third-party libraries such as Boost, OpenSSL, Qt5, ROOT 6, etc. The following versions have been tested.

- **OS**
  - Ubuntu 20.04/22.04/24.04
  - Fedora 40
  - Rocky 9
- **boost >= 1.67**
  - 1.71.0
  - 1.74.0
  - 1.75.0
  - 1.83.0
- **OpenSSL**
  - 1.1.1
  - 3.0.2
  - 3.0.7
  - 3.0.13
  - 3.2.2
- **Qt 5**
  - 5.12.8
  - 5.15.3
  - 5.15.9
  - 5.15.13
  - 5.15.17
- **ROOT 6**
  - 6.24.08 some bug

- 6.26.16
- 6.30.06 some bug
- 6.32.04 some bug

The systems tested by this program include Ubuntu20.04/22.04/24.04 Fedora40 Rocky9. Supports compilation of various LINUX operating systems. If your operating system compilation does not pass, please contact Hongyi Wu

```
# Ubuntu20.04 Dependency installation
sudo apt -y install libboost-dev libboost-all-dev libssl-dev openssl qt5-default␣
↪qtcreator libqt5charts5-dev
# Ubuntu22.04 Dependency installation
sudo apt -y install libboost-dev libboost-all-dev libssl-dev openssl libqt5charts5-
↪dev
# Ubuntu24.04 Dependency installation
sudo apt -y install libboost-dev libboost-all-dev libssl-dev openssl libqt5charts5-
↪dev
# Fedora 40 Dependency installation
sudo dnf -y install redhat-lsb-core boost boost-devel openssl openssl-devel qt5-
↪qtcharts qt5-qtcharts-devel qt5-qtbase qt5-qtbase-devel
# Rocky 9  Dependency installation
sudo dnf -y install boost boost-devel openssl openssl-devel qt5-qtcharts qt5-
↪qtcharts-devel qt5-qtbase qt5-qtbase-devel
# ROOT 6 recommend 6.26.16
```

2.1 Software installation steps

- Delete the old version PKUCAENDAQ folder from the personal directory
- Extract this package to your personal directory ($HOME)
- Compile and install drivers in the *driver* folder

```
cd driver
tar -zxvf CAENDGTZ-USB-Drv-1.2.tgz
cd CAENDGTZ-USB-Drv-1.2/
sudo ./install.sh

cd ..
tar -zxvf caen_felib-v1.3.1.tar.gz
cd caen_felib-v1.3.1/
./configure --disable-assert
make
sudo make install
sudo ldconfig

cd ..
tar -zxvf caen_dig2-v1.6.1.tar.gz
cd caen_dig2-v1.6.1/
./configure --disable-assert
make
sudo make install
sudo ldconfig
```

- Compile GUI software

```
cd GUI
chmod +x makefile.sh.x
./makefile.sh.x
```

(续下页)

(接上页)

```
# Waitting for the compilation to complete. After the compilation is passed, the␣
↪executable file gddaq will be generated in the folder
# Check if the executable file gddaq has been generated, and if so, compile␣
↪successfully. If not, please contact Hongyi Wu.
```

2.2 Registration of each module in LINUX OS

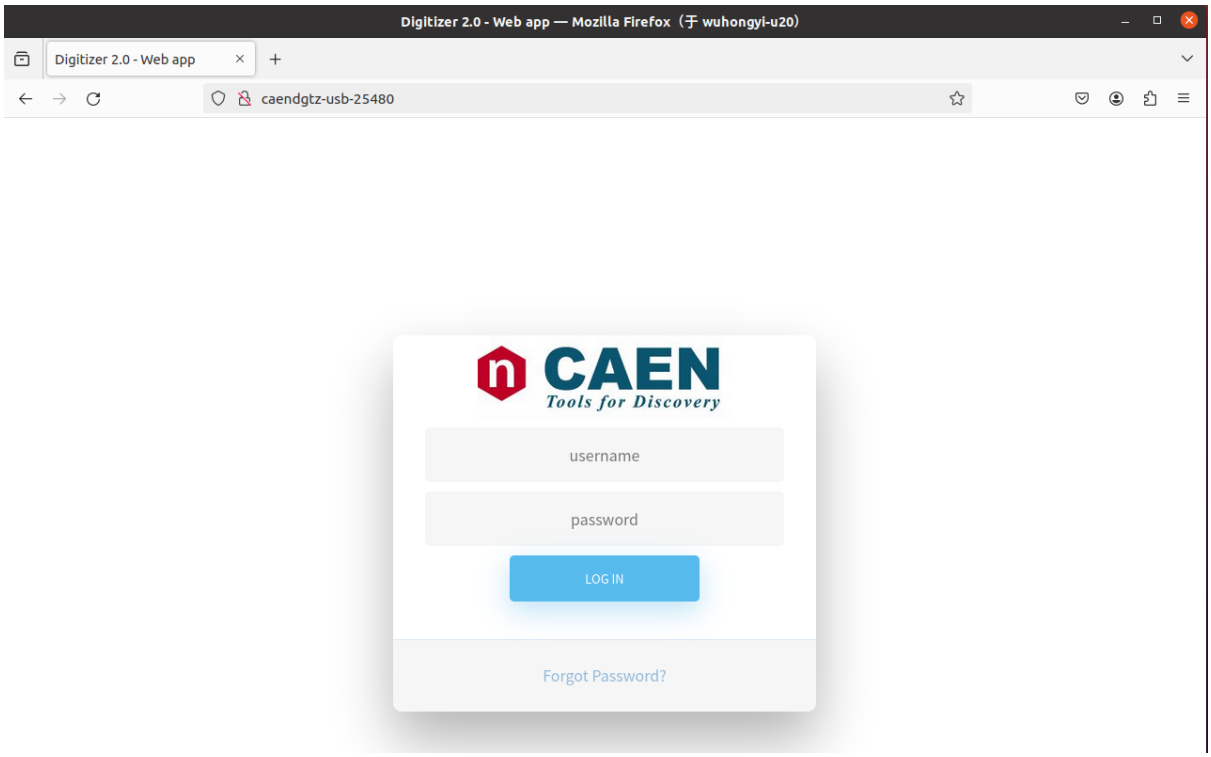
Due to the inability of some Linux distributions to automatically retrieve the USB name of the digitizer, registration is required when using this module for the first time. Module registration needs to be done module by module. During the registration process, only one module can be connected to the LINUX OS via USB.

Connect the Type-C end of the USB3.0 cable to the module and the other end to the computer. Then enter CAENDGTZ-USB-{PIDNUMBER} in the browser, where {PIDNUMBER} is replaced with the PID code of the module being used, for example: CAENDGTZ-USB-25480. Check if it can be accessed. If it cannot be accessed, it means that the module has not been registered as a driver yet.

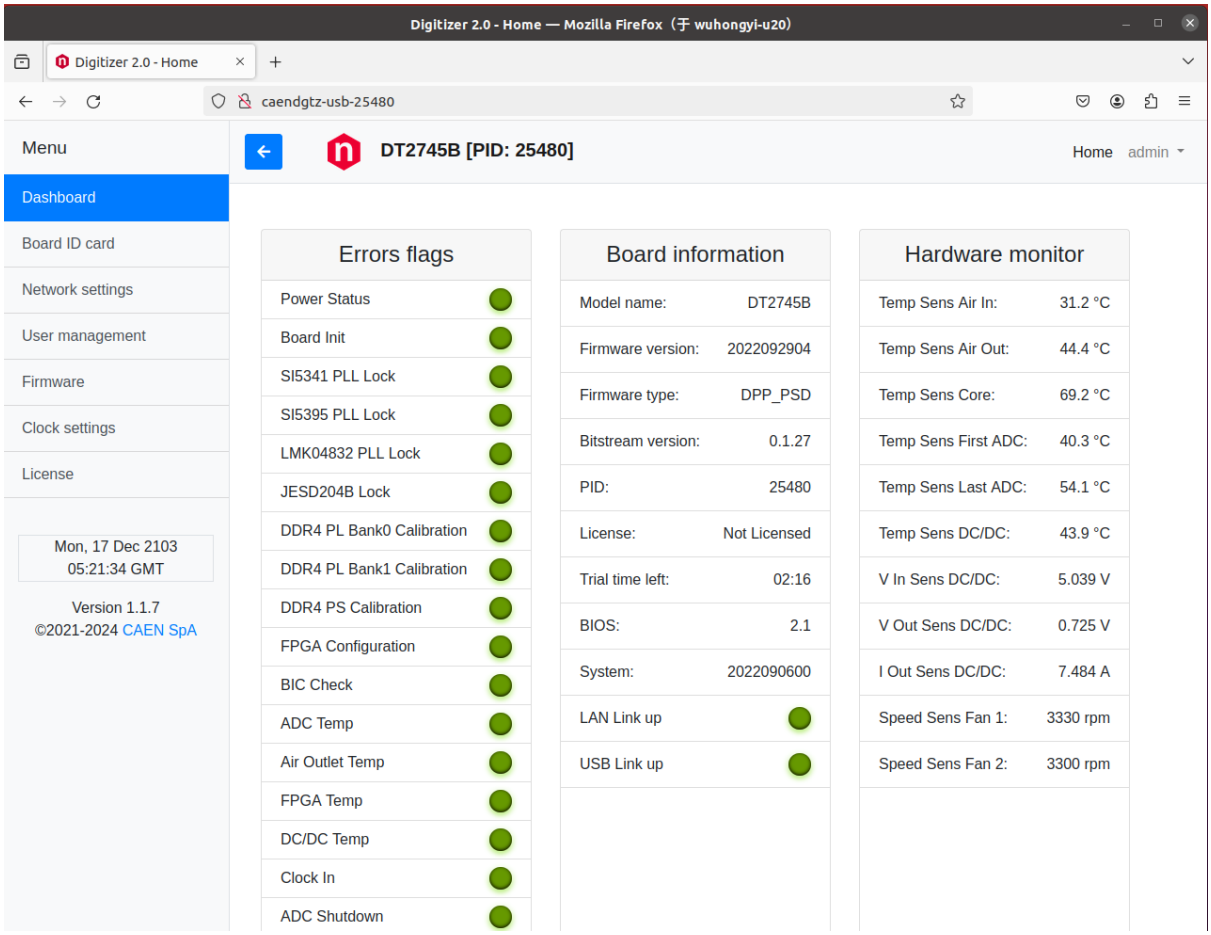
In the USB driver installation package, such as **CAENDGTZ-USB-Drv-1.2**, there is a driver registration script file *regPID.sh*, which can be run using the following command. After execution, you will see a prompt indicating successful registration.

```
sudo ./regPID.sh
```

Afterwards, by accessing CAENDGTZ-USB-{PIDNUMBER} through the webpage, you can see the following login page, with the default username and password being *admin*.



After logging in, the interface is shown below. In the left menu bar, network settings can be made. If you want to obtain data through Ethernet cable, you can set the IP through this page. According to the laboratory network situation, choose DHCP to automatically allocate IP or manually configure IP. After setting the IP, entering the module's IP in the browser can also access the settings page.


Digitizer 2.0 - Home — Mozilla Firefox（于 wuhongyi-u20）
Digitizer 2.0 - Home
caendgtz-usb-25480
Menu
Dashboard
Board ID card
Network settings
User management
Firmware
Clock settings
License
Mon, 17 Dec 2103
05:21:34 GMT
Version 1.1.7
©2021-2024 CAEN SpA
DT2745B [PID: 25480]
Home
admin
Errors flags
Power Status
Board Init
SI5341 PLL Lock
SI5395 PLL Lock
LMK04832 PLL Lock
JESD204B Lock
DDR4 PL Bank0 Calibration
DDR4 PL Bank1 Calibration
DDR4 PS Calibration
FPGA Configuration
BIC Check
ADC Temp
Air Outlet Temp
FPGA Temp
DC/DC Temp
Clock In
ADC Shutdown
Board information
Model name: DT2745B
Firmware version: 2022092904
Firmware type: DPP_PSD
Bitstream version: 0.1.27
PID: 25480
License: Not Licensed
Trial time left: 02:16
BIOS: 2.1
System: 2022090600
LAN Link up
USB Link up
Hardware monitor
Temp Sens Air In: 31.2 °C
Temp Sens Air Out: 44.4 °C
Temp Sens Core: 69.2 °C
Temp Sens First ADC: 40.3 °C
Temp Sens Last ADC: 54.1 °C
Temp Sens DC/DC: 43.9 °C
V In Sens DC/DC: 5.039 V
V Out Sens DC/DC: 0.725 V
I Out Sens DC/DC: 7.484 A
Speed Sens Fan 1: 3330 rpm
Speed Sens Fan 2: 3300 rpm

CHAPTER 3

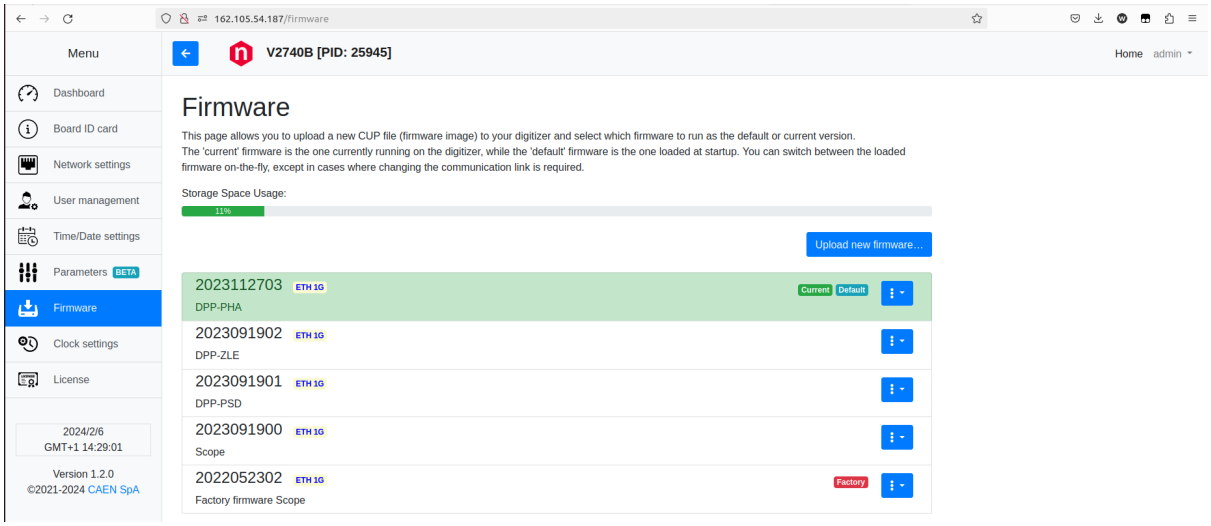
Firmware

3.1 Latest Firmware

- **2745**
  - V2745-scope-1G-2024100301.cup
  - V2745-dpp-pha-1G-2024102101.cup
  - V2745-dpp-psd-1G-2024101603.cup
  - V2745-dpp-zle-1G-2024102200.cup
  - V2745-OpenDPP-2024101303.cup
- **2740**
  - V2740-scope-1G-2024051505.cup
  - V2740-dpp-pha-1G-2024102101.cup
  - V2740-dpp-psd-1G-2024101603.cup
  - V2740-dpp-zle-1G-2024102200.cup
  - V2740-OpenDPP-2024101302.cup
- **2730**
  - V2730-scope-1G-2024100300.cup
  - V2730-dpp-psd-1G-2024092000.cup
  - V2730-OpenDPP-2024101301.cup

3.2 Firmware Update

Access the module configuration page via USB（CAENDGTZ-USB-{PIDNUMBER}）or IP. There is a “Fireware” option in the left menu bar, click to enter the page, as follows:



Check if the Scope/PHA/PSD/ZLE firmware version on this page matches the **firmware** folder in thepackage. If not, upload the new firmware through ‘Upload new firmware’. There is a blue bottom triangle button on the right end of each firmware in the webpage. By clicking this button, you can switch between the currently used firmware and set the default firmware to load when booting up. Additionally, promptly remove outdated firmware versions.

CHAPTER 4

GUI Guide

Before using graphical software, please read about time synchronization, GUI, and other related content.

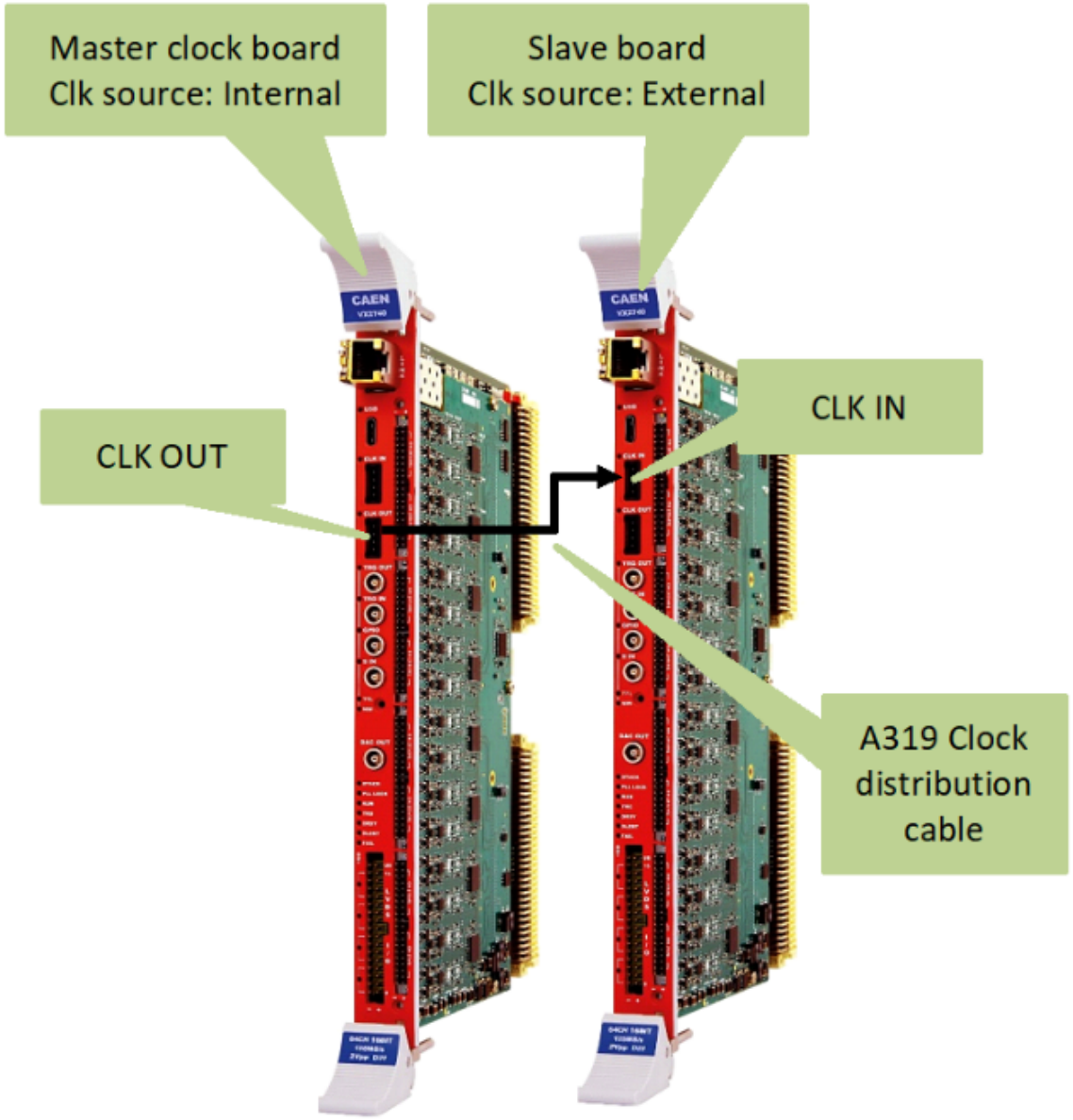
CHAPTER 5

Time synchronization

In a multi module acquisition system, a fundamental requirement is that all modules acquisition data synchronously. It requires all modules to share a clock source and then start and end data acquisition simultaneously.

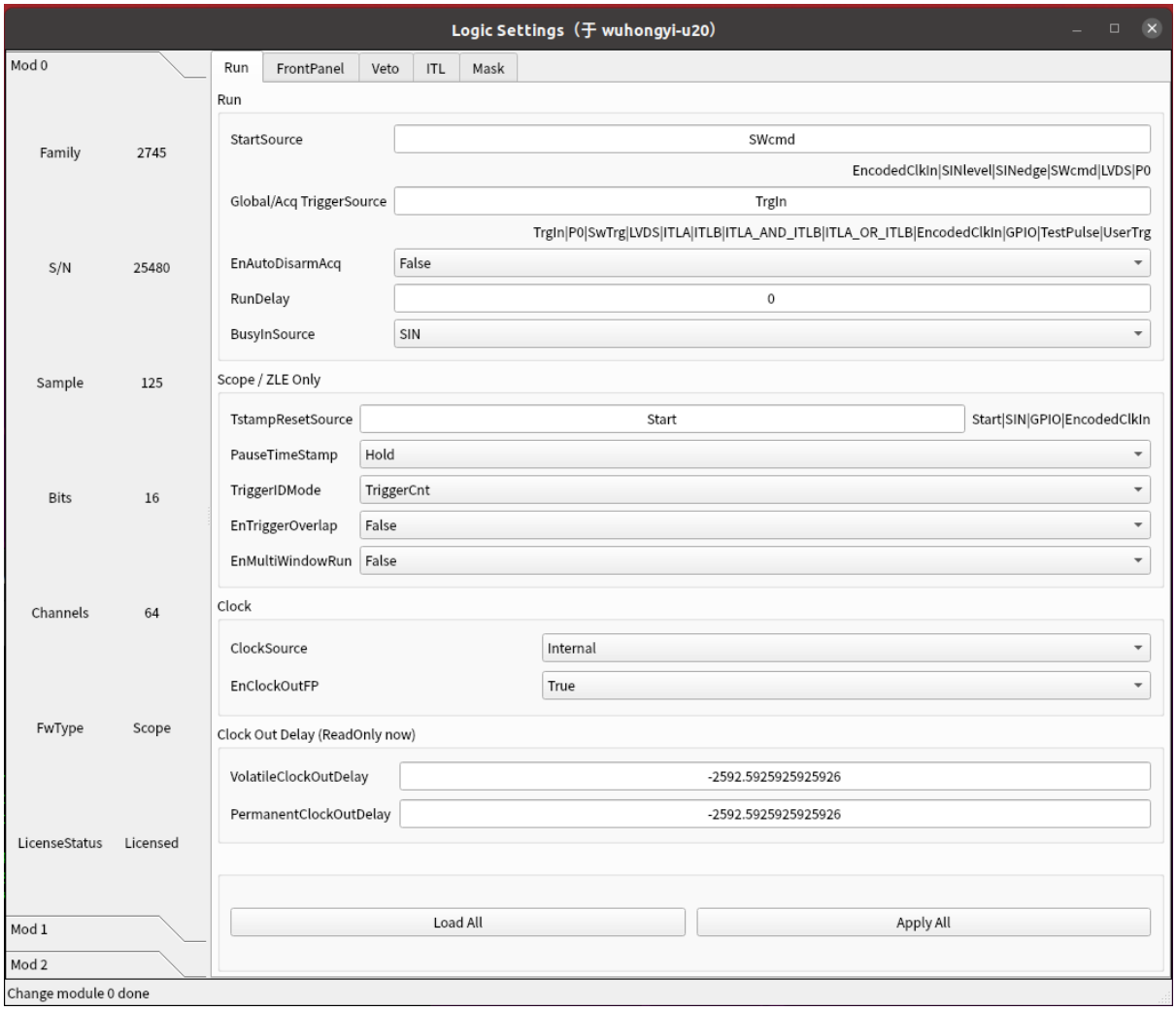
5.1 Clock source setting

CAEN’s clock can be connected in series through the A319 cable on the front panel or shared through the chassis backplane (under development). The following figure shows a typical clock synchronization configuration diagram, where the previous CLK OUT is connected to the next CLK IN through A319.

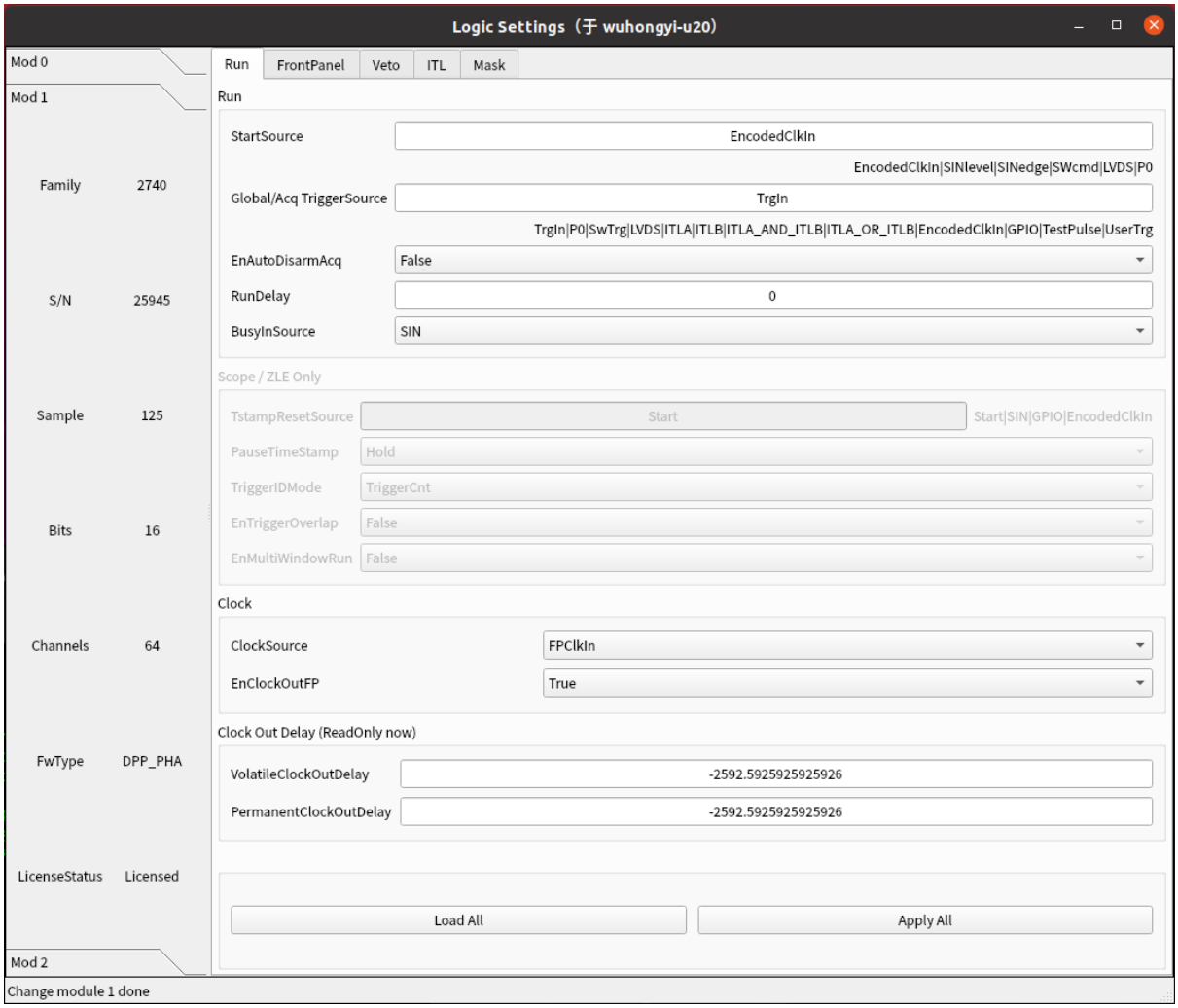


After connecting the clock synchronization cable, it is necessary to sequentially set the clock source for each module.

For the main module, it is necessary to set ClockSource to Internal and EnClockOutFP to True, as shown in the following figure:



For all slave modules, ClockSource needs to be FPClkIn, and EnClockOutFP needs to be set to True, as shown in the following figure:



The above settings are to output the internal clock of the main module through the front panel, and receive the clock signals transmitted from the previous module through the front panel in sequence from the sub modules.

5.2 Synchronous acquisition

For a synchronous system, the main module is usually set to control the start and end of acquisition, and other sub modules also need to make corresponding settings. There are three common ways to control the start and end of the acquisition through the front panel: SYNC-CLKIN、SIN-GPIO、SIN-TRGOUT。Among them, SYNC-CLKIN is the most widely used because it is included in the clock synchronization line of the front panel, while the other two require the SIN/GPIO/TRGOUT LEMO port on the front panel, commonly used for synchronous acquisition with other acquisition systems.

5.3 SYNC-CLKIN

The master module needs to set StartSource to SWcmd and SyncOutMode to Run. Set StartSource to EncodedClkIn and SyncOutMode to SyncIn for all slave modules.

5.4 SIN-GPIO

The master module needs to set StartSource to SWcmd and GPIOMode to Run. Set StartSource to SIN level and GPOMode to SIN for all slave modules.

5.5 SIN-TRGOUT

Connect the TRGOUT of the master module to the SIN of the first slave module, and so on.

The master module needs to set StartSource to SWcmd and TrgOutMode to Run. Set StartSource to SINlevel and TrgOutMode to Run for all slave modules.

CHAPTER 6

Graphical interactive interface

6.1 Configuration file

The configuration file is placed in the **pars** folder, JSON format file, mainly containing module PID or IP information.

```
{
"modules": 3,
"connecttype": "eth",
"pid": [25480, 25945, 24946],
"ip": ["162.105.54.162", "162.105.54.187", "162.105.54.90"],

"par": "setting.json",

"userpars": "../pars/init.txt"

}
```

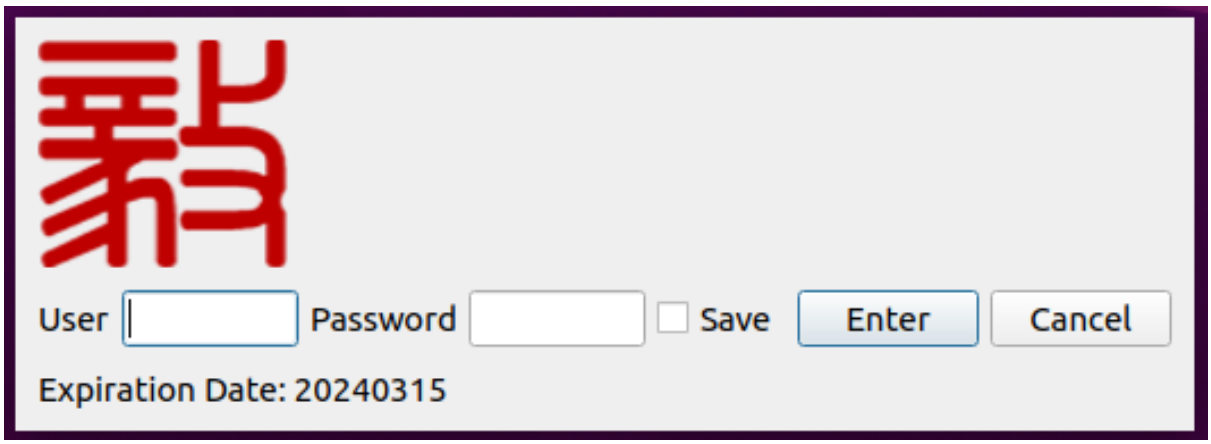
Among them, the parameter *modules* is the number of data acquisition modules used by the system, and the parameter *connecttype* is the connection method between the acquisition modules and the computer, which can be filled in with "usb" or "eth" . Using USB for reading, the maximum limit for a single module is 280 MB/s, while using network for reading, the 1G network has a maximum limit of 110 MB/s. If USB is used for reading, the parameter PID will take effect, which will be the PID of each module in the system in sequence. If network reading is used, the parameter IP takes effect, which is the IPV4 IP of each module in the system.

Enter the GUI directory and execute the following command to pop up the main control interface

```
./gddaq
# If the following error occurs, set the following environment variables
# ./gddaq: error while loading shared libraries: libCAEN_FELib.so.0: cannot open␣
↪shared object file: No such file or directory
# export LD_LIBRARY_PATH=/usr/local/lib:$LD_LIBRARY_PATH
```

6.2 login interface

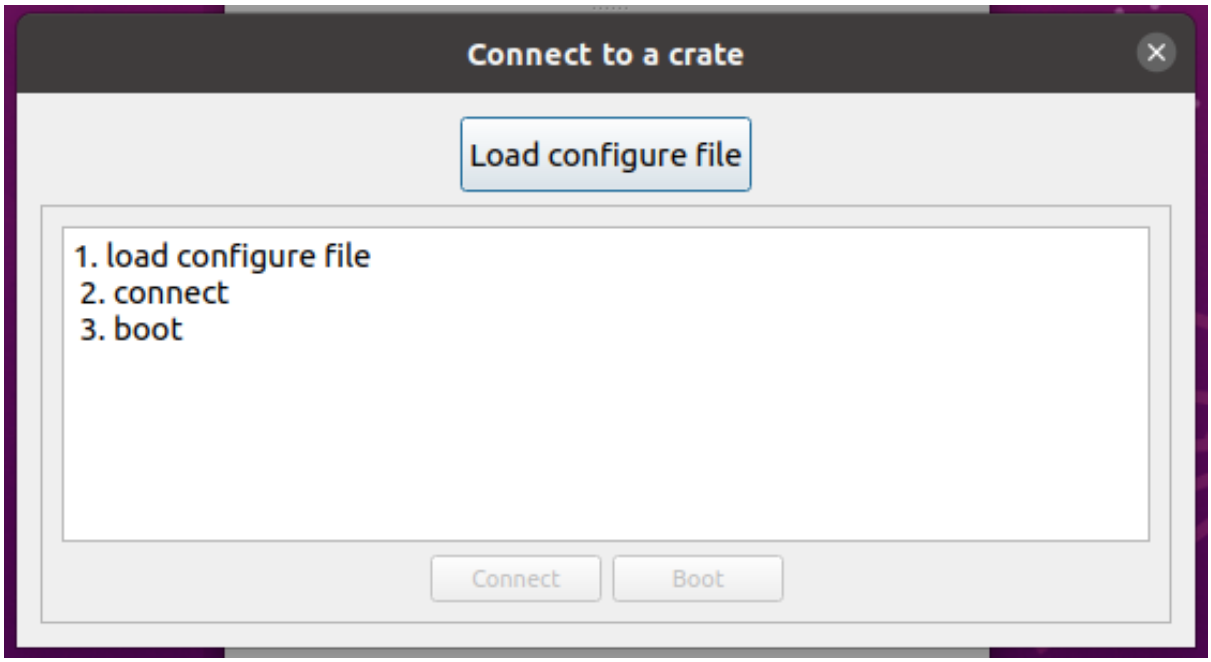
In order to achieve different operation permissions for different login accounts. Reducing the operational permissions of experimental duty personnel has not yet been implemented···



After entering the username (admin) and password (admin), the main control interface will pop up.

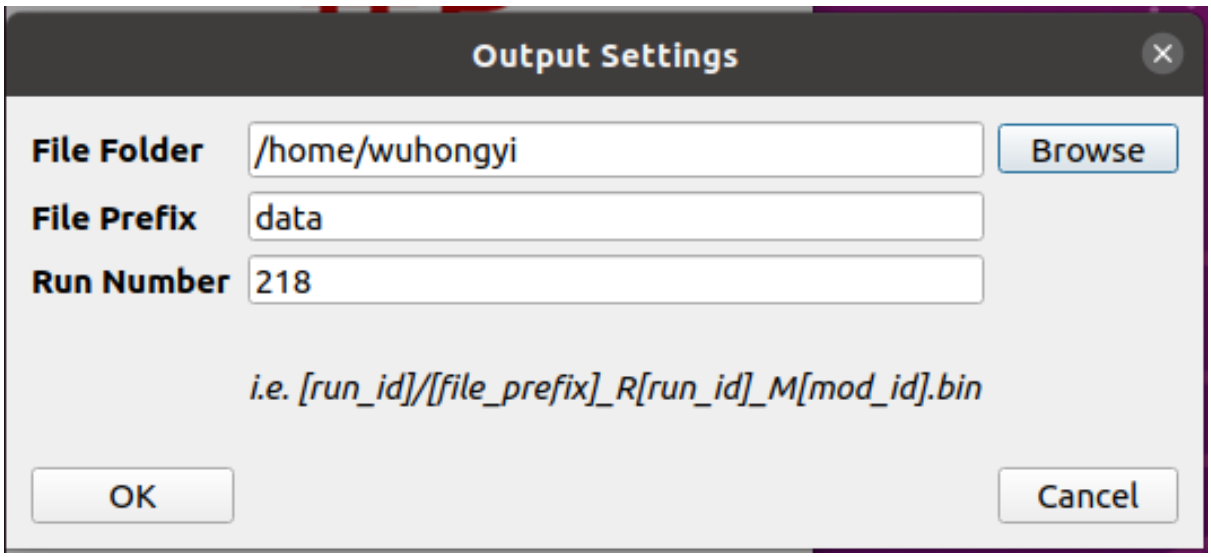
6.3 Main control interface

Select “connect to device” from the drop-down menu in the upper left corner of the main interface. The following interface will pop up. Click on the file selection box and select the pre configured parameter file.

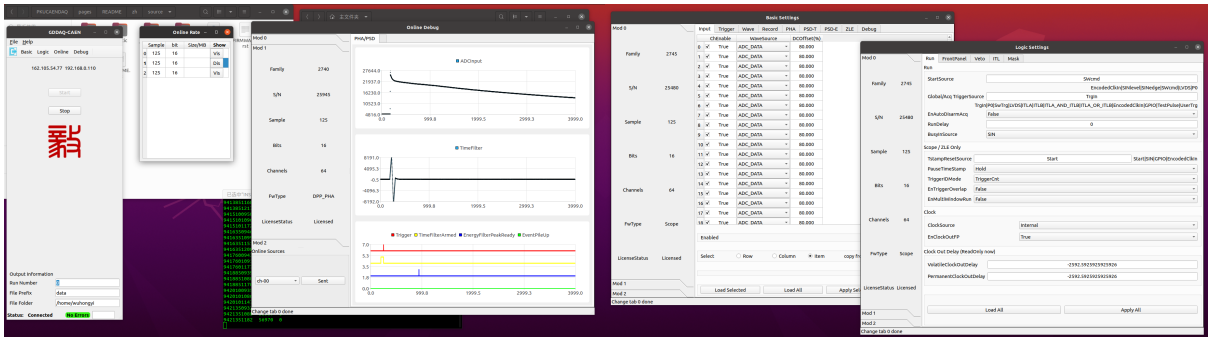


After the configuration file is loaded, the main configuration information, PID or IP information, will be displayed in the middle of the dialog box. Then click “connect”, and then click “boot”. After completing the progress bar, the pop-up interface will automatically close. All modules in the configuration file have completed initialization.

Select “Output Configure” from the drop-down menu in the upper left corner of the main interface, and the following interface will pop up. This interface is used to set the folder, file name, and run number for outputting data. Press the OK button to close the interface. The above information can only be modified on this interface and displayed on the main interface.



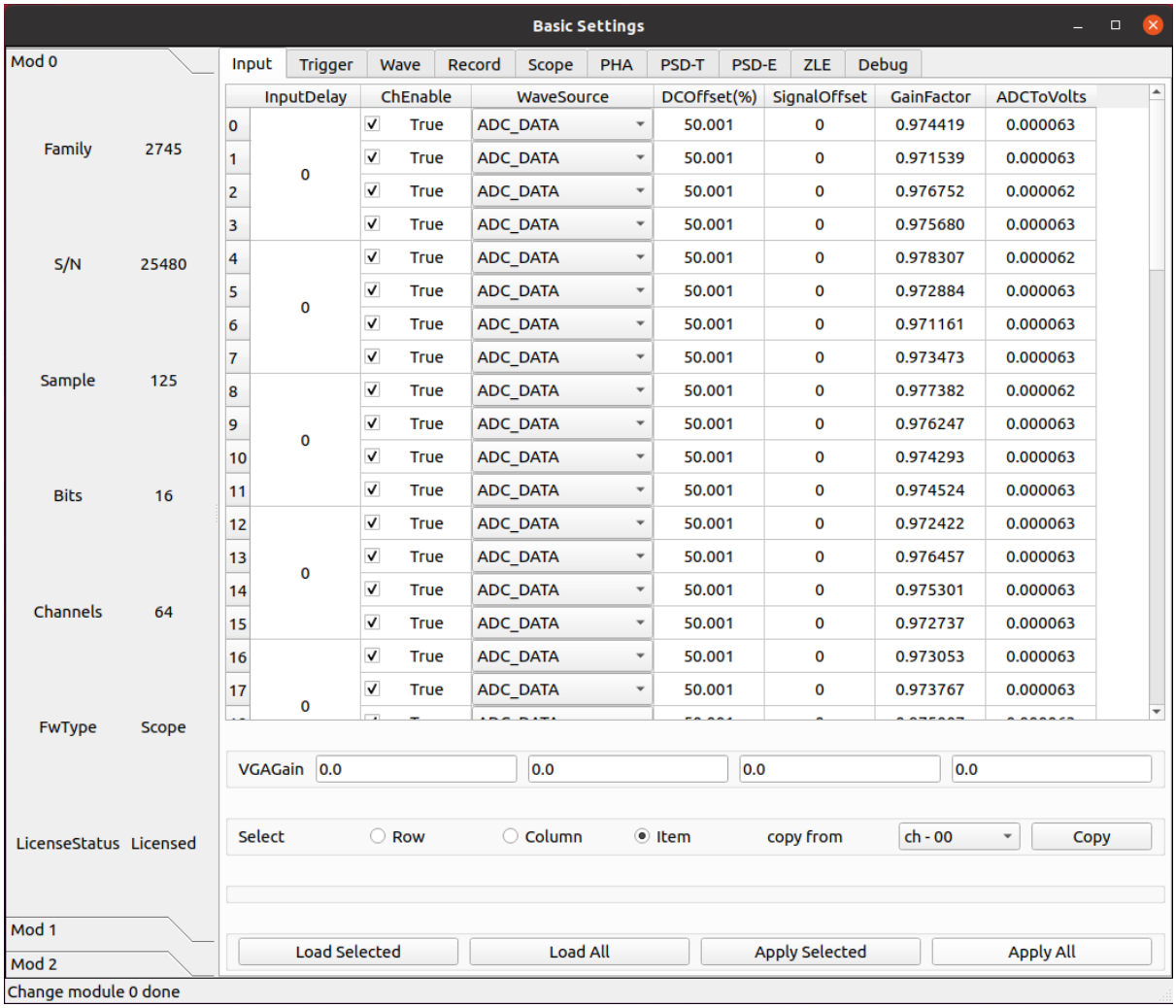
After completing the system initialization, the Basic, Logic, Online, and Debug buttons above the main control interface will float up. Clicking on them will pop up the corresponding sub interface, and clicking again will hide it.



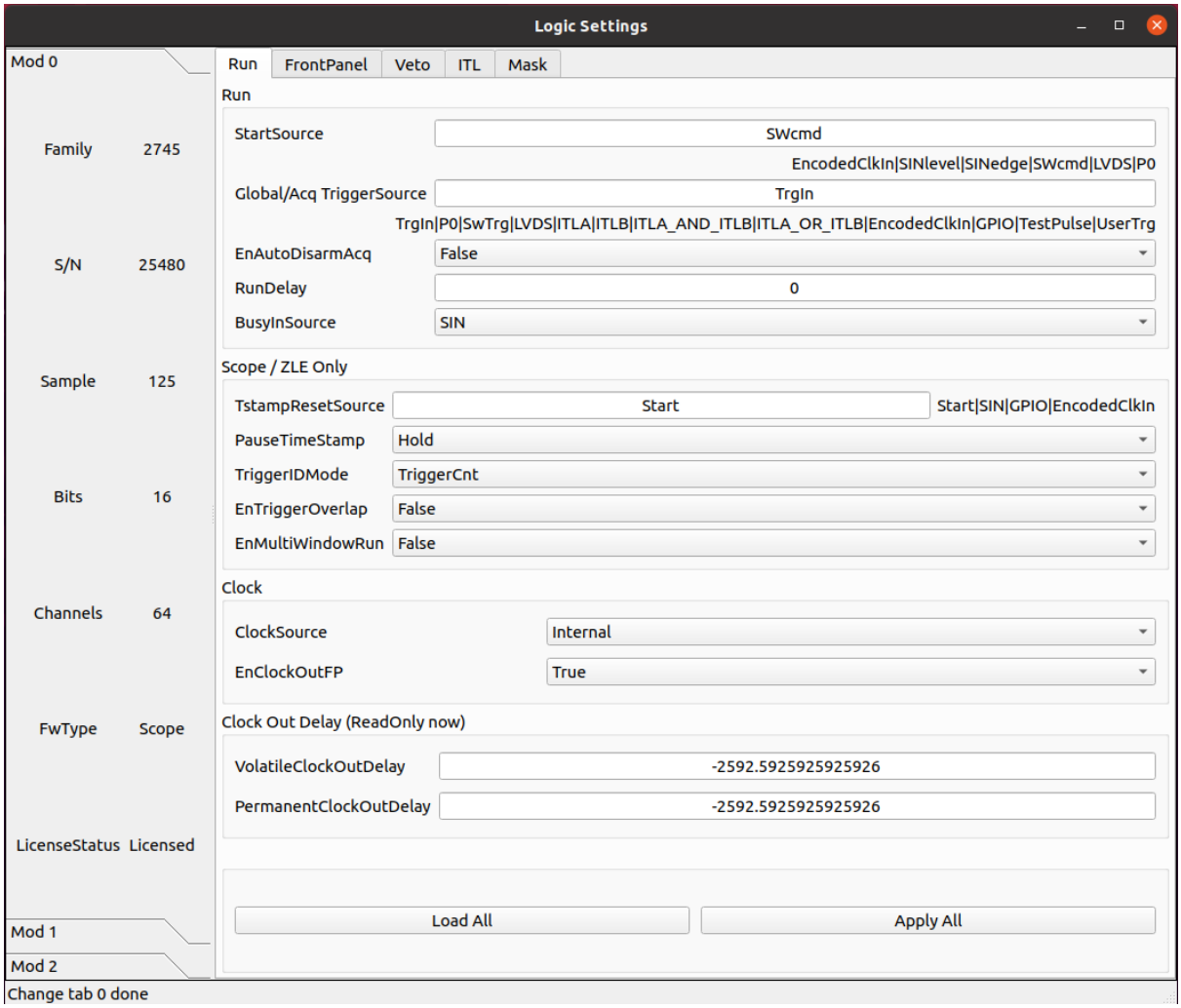
In the Basic, Logic, Debug sub interfaces that pop up, when the system has multiple modules, switch the acquisition module by clicking on the “Mod 0”, “Mod 1”, “Mod 2” and other switching tags on the left side. When switching, the information on the right page will be automatically read. There are multiple tabs above each sub interface, click on the tab to switch to the corresponding parameter configuration page.

Due to the adaptability of this program to different firmware, different configurable parameters will be displayed in different firmware on the same parameter settings tab.

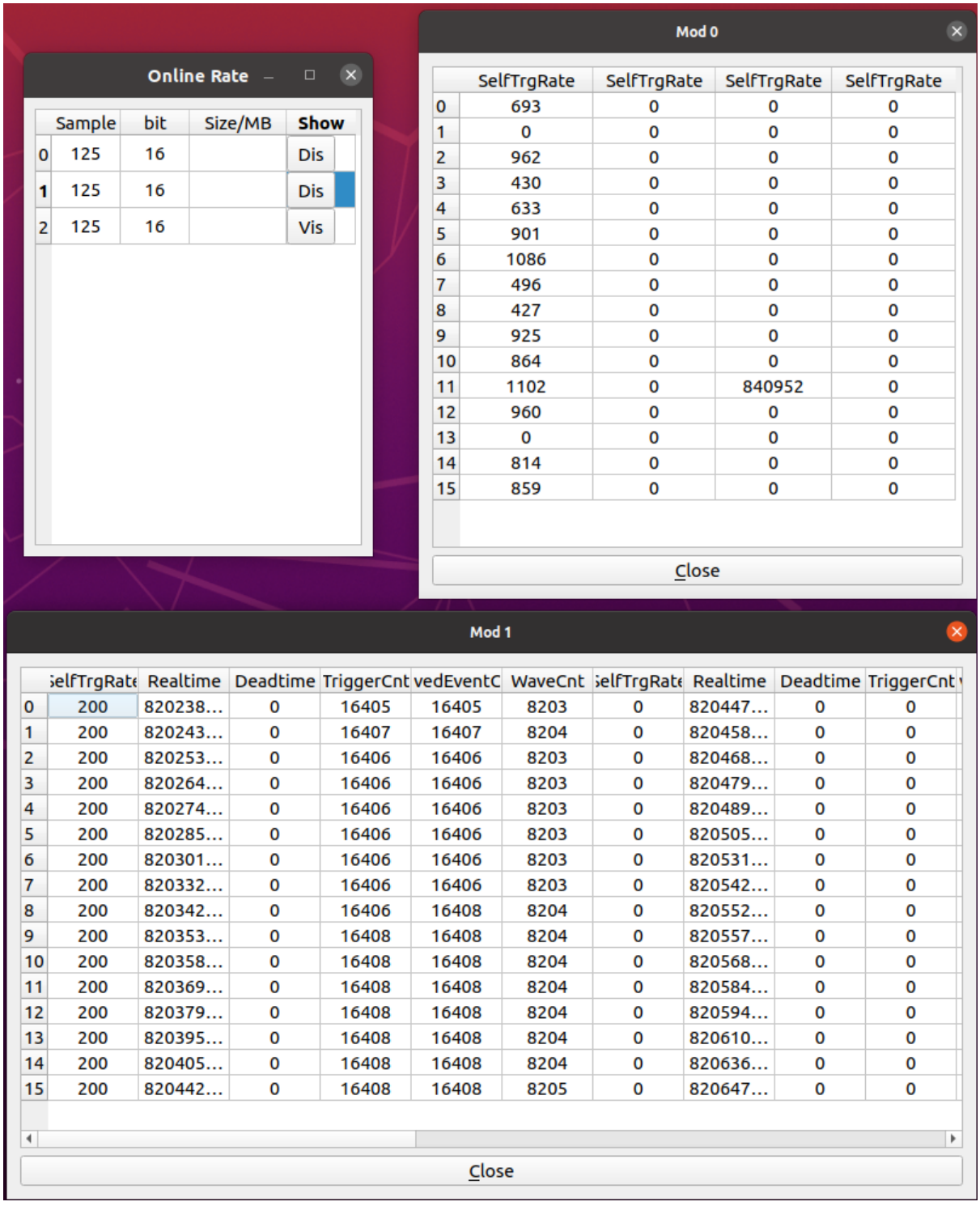
Basic parameter settings



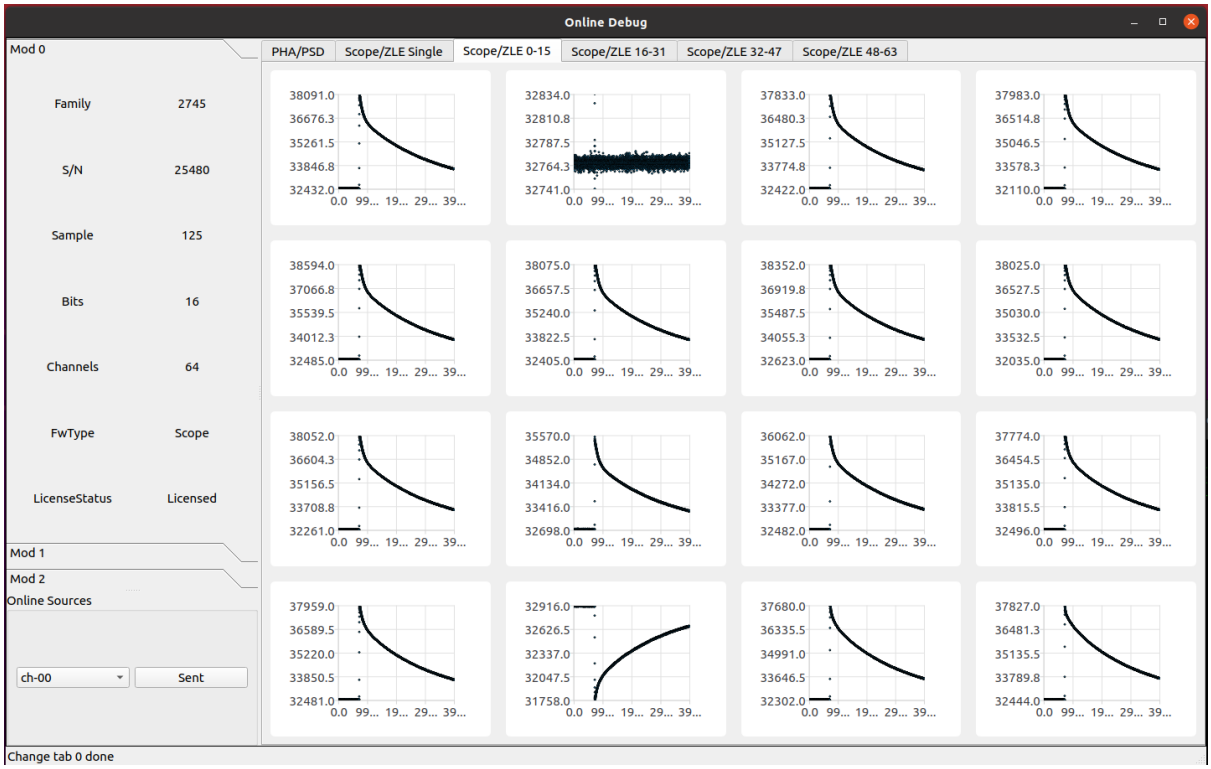
Logic parameter settings

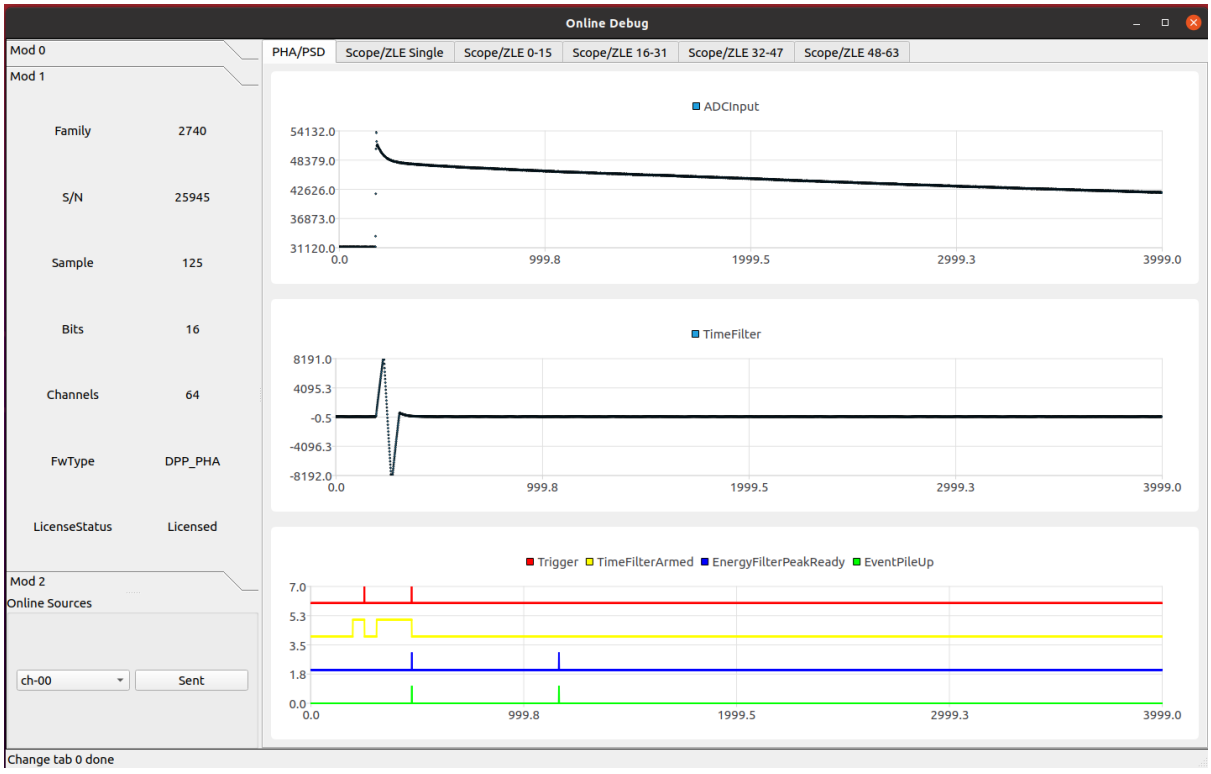


Real time count rate monitoring



Waveform monitoring and debugging


Online Debug
Mod 0
Family 2745
S/N 25480
Sample 125
Bits 16
Channels 64
FwType Scope
LicenseStatus Licensed
Mod 1
Mod 2
Online Sources
ch-00
Sent
PHA/PSD
Scope/ZLE Single
Scope/ZLE 0-15
Scope/ZLE 16-31
Scope/ZLE 32-47
Scope/ZLE 48-63
Change tab 0 done


Online Debug
Mod 0
Mod 1
Family 2740
S/N 25945
Sample 125
Bits 16
Channels 64
FwType DPP_PHA
LicenseStatus Licensed
Mod 2
Online Sources
ch-00
Sent
PHA/PSD
Scope/ZLE Single
Scope/ZLE 0-15
Scope/ZLE 16-31
Scope/ZLE 32-47
Scope/ZLE 48-63
ADCInput
TimeFilter
Trigger
TimeFilterArmed
EnergyFilterPeakReady
EventPileUp
Change tab 0 done

6.4 Online monitoring of web pages

The GUI program occupies port 8765 for online display, which can be accessed by users through IP 8765, and can be accessed locally using 127.0.0.1:8765.

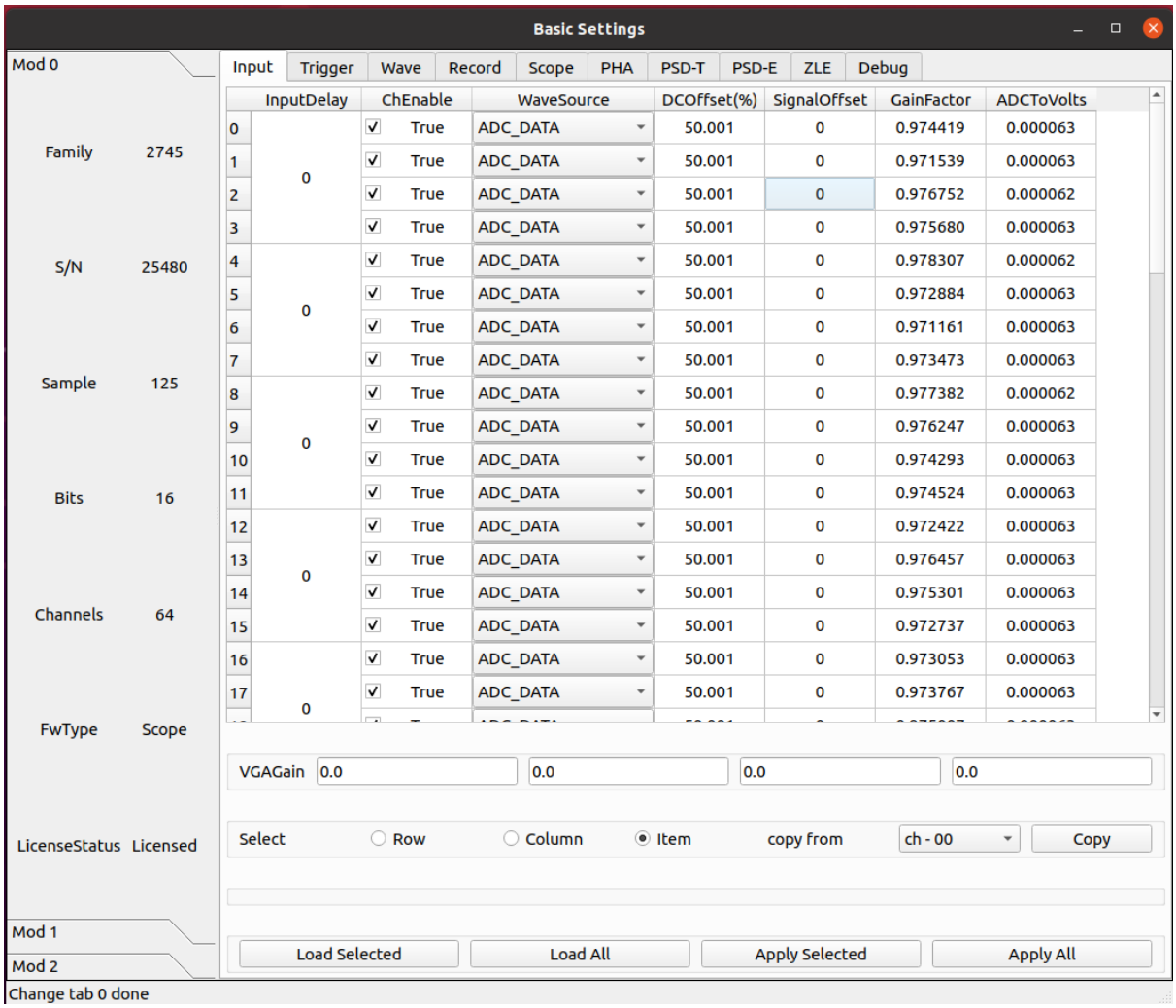
Accessing online monitoring requires logging in, with two users set up: admin and guest. The password for admin is admin, and this account has all access permissions and can perform operations such as clearing spectra at any time. The guest does not have a password and can only view online monitoring information.

CHAPTER 7

SCOPE firmware

7.1 Basic

7.1.1 Input



parameter ChGain

Unique to x2730.

Sets the gain of the Variable Gain Amplifiers (VGA). Unit of Measure: dB

parameter InputDelay

Set input delay. The value is set at groups of 4 channels for x2745/x2740.

Unit: sample

parameter ChEnable

Enable the channels for the acquisition, according to the Index. When the channel is disabled, it does not give any data and its self-trigger is off.

- True: The channel is enabled for the acquisition
- False: The channel is disabled for the acquisition

parameter WaveSource

In normal mode, the acquired waveform represents a sequence of ADC samples, resulting from the A/D conversion of the analog input. For test purposes, it is possible to replace the ADC data with internal data generators.

- **ADC_DATA**
  - Data from the ADC (normal operating mode)
- **ADC_TEST_TOGGLE**
  - Toggle between 0x5555 and 0xAAAA (test mode)
- **ADC_TEST_RAMP**
  - 16-bit ramp pattern (test mode)
- **ADC_TEST_SIN**
  - 8-point sine wave test pattern
- **ADC_TEST_PRBS**
  - 16-bit PRBS generated by a 23-bit PRBS pattern generator (test mode)
- **Ramp**
  - Data from a ramp generator. It is actually a 16-bit field, where the 6 most significant bits identify the channel and the 10 less significant bits are the samples of a ramp from 0x000 up to 0x3FF (i.e. 0 to 1023). It is so a 10-bit ramp with offset given by “channel*1024”. For channel 0, it is a counter from 0 to 1023; for channel 1, it is a counter from 1024 to 2047, and so on
- **IPE**
  - Not implemented
- **SquareWave**
  - Internally generated programmable square wave

parameter DCOffset

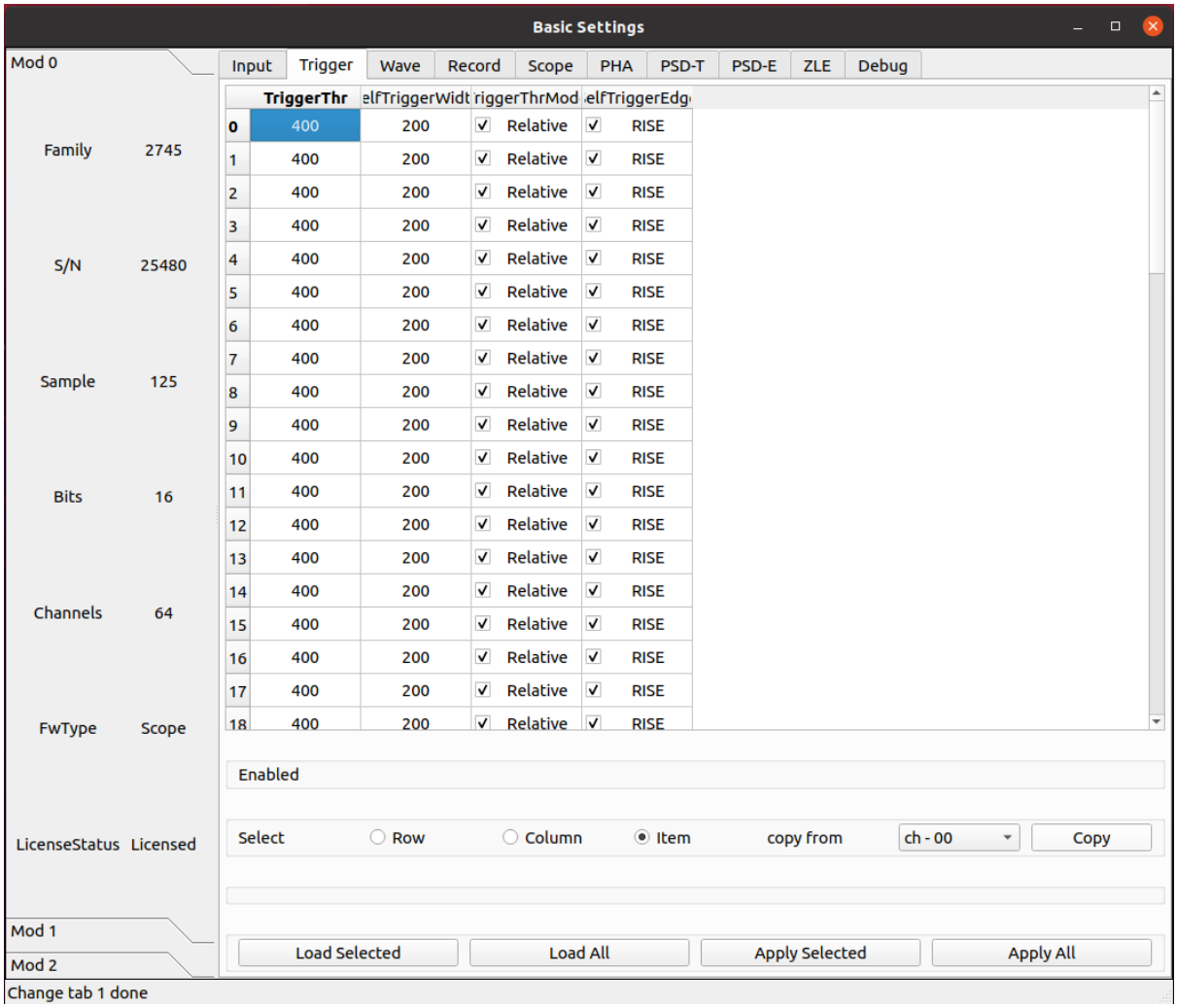
A constant DC offset (controlled by a 16-bit DAC) is added to the analog input, individually for each channel, to adjust the position of the signal baseline (that is the "zero volt" of the analog input) within the dynamic range of the ADC. Because of the tolerance of the components, it is necessary to calibrate the offset DAC. The calibration is done by factory testing and normally it is not necessary to recalibrate the digitizer. It is however possible to perform a new calibration. The calibration parameters are stored in the flash memory of the board and loaded at power on. They are automatically applied by the internal logic every time the DCoffset parameter is written or read. DCoffset is expressed as a NUMBER number, in percent of the full-scale. When the DCoffset is 0, the baseline of the input signal is at 0 ADC counts. When the DCoffset is 100, the baseline of the input signal is at $2^{NBIT}-1$ ADC counts.

parameter VGAGain

Unique to x2745.

Set the gain of the variable gain amplifier (VGA) in increments of 0.5 dB. Parameter settings are grouped every 16 channels, with 64 channels divided into 4 groups. The minimum can be set to 0 and the maximum to 40.

7.1.2 Trigger



parameter TriggerThr

Each channel of the digitizer has a digital leading-edge discriminator with programmable threshold able to self-trigger on the input pulses and generate a self-trigger signal (or an overthreshold signal) feeding the internal trigger logics or digitizer outputs.

This parameter sets the trigger threshold. Typically, the value is relative to the baseline of the signal and the threshold is a 17-bit signed NUMBER number; in this case, the threshold automatically follows the baseline when the DCoffset parameter changes. Sometimes, it is preferable to set an absolute value for the threshold, referred to the ADC range. In this case, the threshold is unsigned NUMBER number. The threshold is relative to the baseline whose value is determined by the DCOffset.

parameter SelfTriggerWidth

The output of the digital leading-edge comparator, that generates the self-trigger signal, can be used in "linear" mode, meaning that it lasts for the time the signal remains above (or below) the threshold, thus acting as an "Over-Threshold" signal, or can pass through a programmable gate generator that makes it a fixed-width pulse. The gate generator is a non-retriggerable monostable that goes high when the threshold is crossed and returns low after the programmed time. This parameter defines the fixed width of the overthreshold pulse.

parameter TriggerThrMode

Defines whether the trigger threshold is relative to the baseline or absolute.

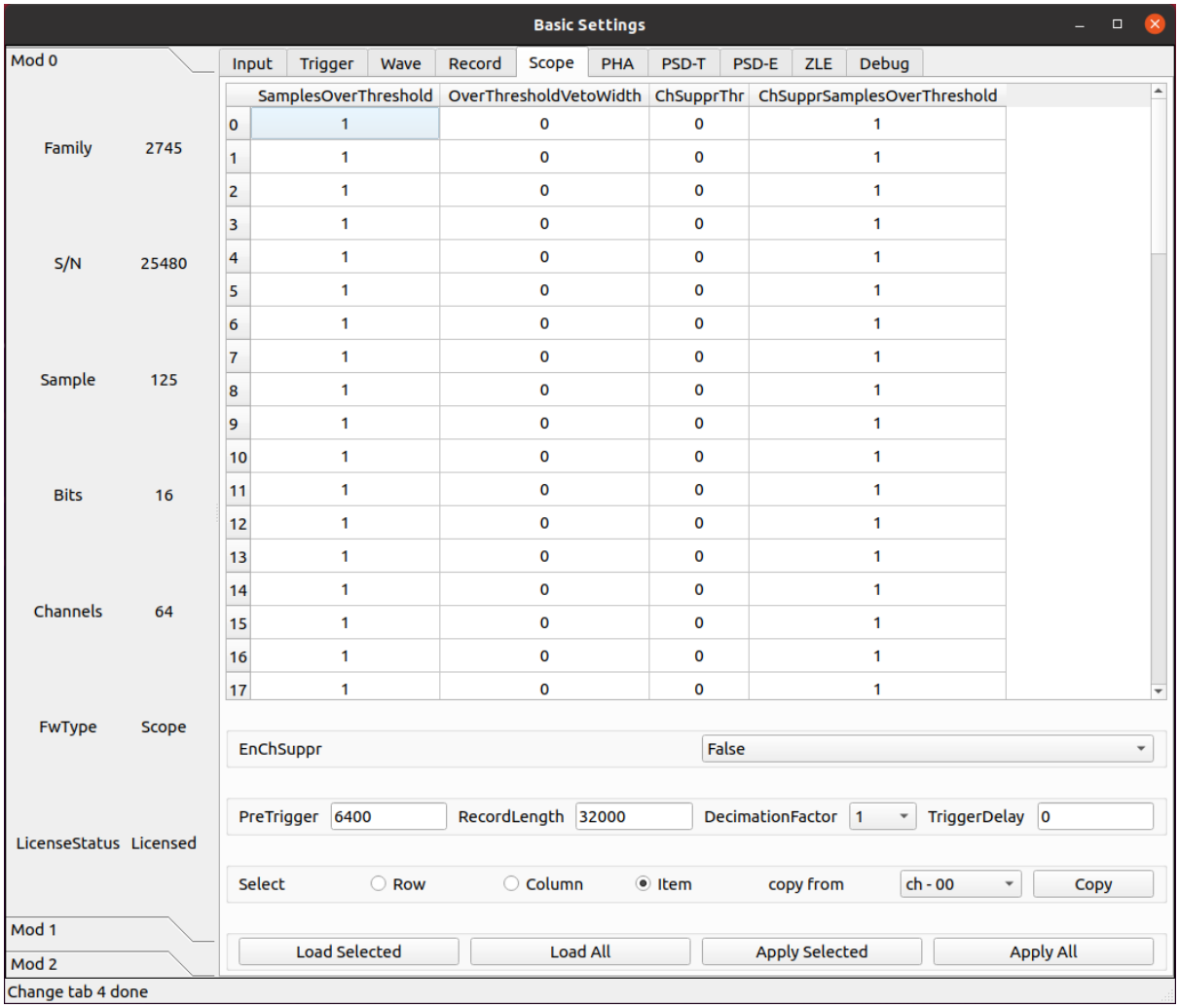
- **Relative**
  - The threshold is relative to the baseline and automatically follows it when the DCOffset parameter is changed.
- **Absolute**
  - The threshold is absolute, referred to the ADC input range. It does not follow the DCOffset setting.

parameter SelfTriggerEdge

Defines whether the self-trigger must be issued on the rising or falling crossing of the threshold. Likewise, the overthreshold signal will be TRUE when the signal is either above or below the threshold.

- **RISE**
  - The trigger occurs on the rising crossing of the threshold and the OverThr is true when the signal is above the threshold
- **FALL**
  - The trigger occurs on the falling crossing of the threshold and the OverThr is true when the signal is below the threshold

7.1.3 scope



parameter SamplesOverThreshold

Number of samples over threshold.

parameter OverThresholdVetoWidth

Veto width to discard triggers when crossing the threshold in the opposite direction to the trigger one

parameter ChSupperThr

Threshold for channel zero-suppression. Configuration parameters *TriggerThrMode* and *SelfTriggerEdge* shared with TriggerThr.

parameter ChSupperSamplesOverThreshold

Samples over threshold for channel zero-suppression

parameter EnChSupper

Enable channel zero-suppression

- **True**
  - Channel zero-suppression is enabled
- **False**
  - Channel zero-suppression is disabled

parameter PreTrigger

The time before the position of the trigger in the waveform (i.e. size of the pre-trigger window). The actual size of the waveform will be automatically rounded to the closest allowed value. It is possible to get the exact size by reading back the parameter

Unit of Measure: ns

parameter RecordLength

The waveform Size (i.e. size of the acquisition window). The actual size of the waveform will be automatically rounded to the closest allowed value. It is possible to get the exact size by reading back the parameter. The record length in time depends on wave resolution.

Unit of Measure: ns

parameter DecimationFactor

Sets the decimation factor to be applied to the Digitizer nominal sampling frequency.If enabled, the RecordLength, PreTrigger and TriggerDelay parameter versions should not be used.

- **1**
  - Decimation disabled (default)
- **2**
  - Sampling Frequency / 2
- **4**
  - Sampling Frequency / 4
- **8**
  - Sampling Frequency / 8
- **16**
  - Sampling Frequency / 16
- **32**
  - Sampling Frequency / 32
- **64**
  - Sampling Frequency / 64

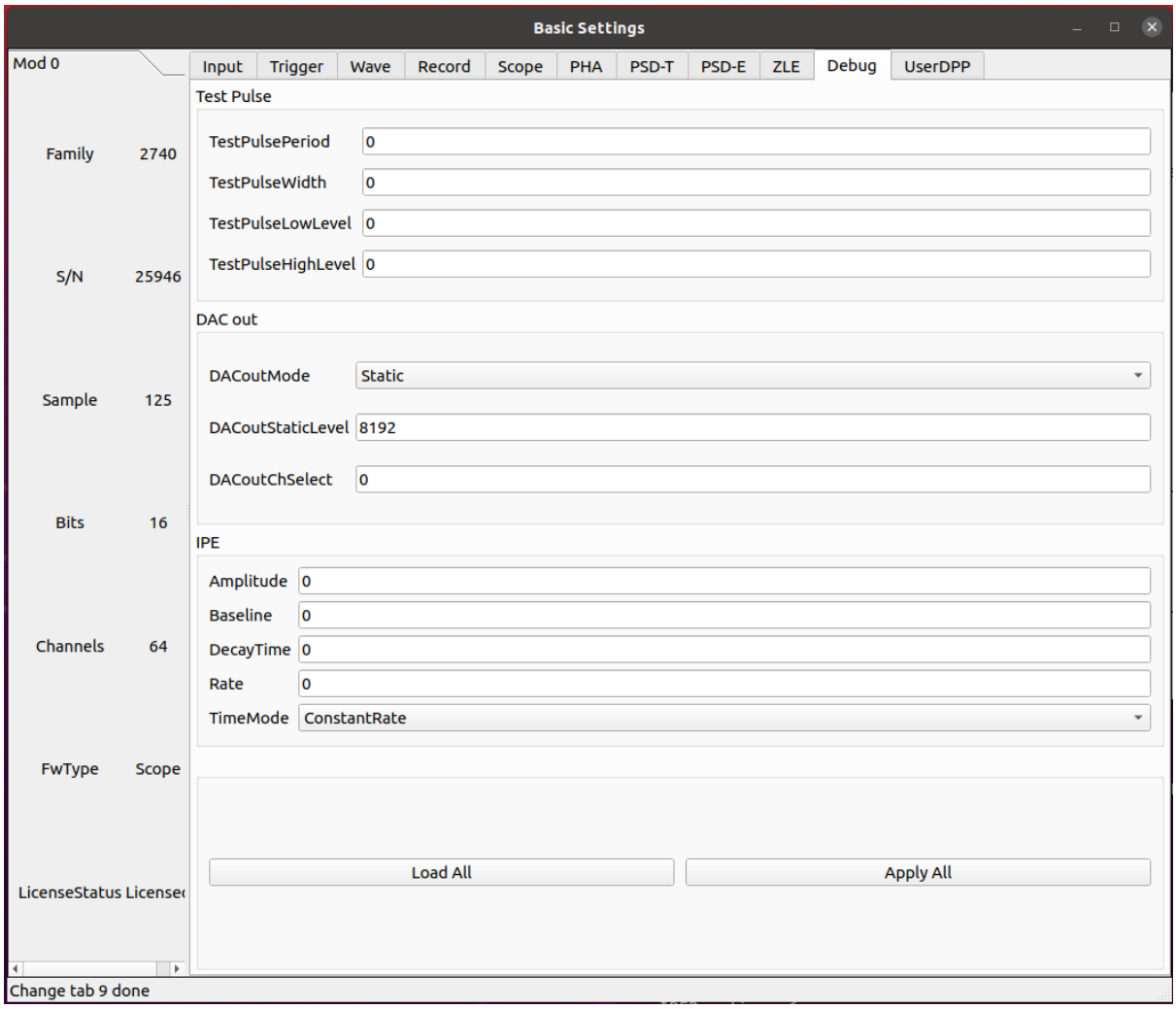
- **128**
  - Sampling Frequency / 128
- **256**
  - Sampling Frequency / 256
- **512**
  - Sampling Frequency / 512
- **1024**
  - Sampling Frequency / 1024

parameter TriggerDelay

Time representing the delay added to the acquisition trigger. This parameter can be used to acquire a window that starts after the position of the trigger

Unit of Measure: ns

7.1.4 Debug



parameter TestPulsePeriod

The Test Pulse is a programmable square wave that can be used as an internal periodic trigger (mainly for test purposes) or to generate a logic test pulse (TTL or NIM) on the TRGOUT and GPIO outputs. This parameter sets the period of the Test Pulse.

Unit of Measure: ns

parameter TestPulseWidth

Width of the Test Pulse (time that the signal stays high = 1).

Unit of Measure: ns

parameter TestPulseLowLevel

Low level of the Test Pulse expressed in ADC counts

Unit of Measure: ADC counts

parameter TestPulseHighLevel

High level of the Test Pulse expressed in ADC counts

Unit of Measure: ADC counts

parameter DACoutMode

Selects the signal type to be sent in output on the front panel DAC connector.

- **Static**
  - DAC output stays at a fixed level, given by the DACoutStaticLevel parameter
- **Ramp**
  - The DAC output is driven by a 14-bit counter
- **Sin5MHz**
  - The DAC output is a sine wave at 5 MHz with fixed amplitude
- **Square**
  - Square wave with period and with set by TestPulsePeriod and TestPulseWidth and amplitude between TestPulseLoweLevel and TestPulseHighLevel.
- **IPE**
  - Not implemented
- **ChInput**
  - The DAC reproduces the input signal received by one input channel, selected by the DACoutChSelect parameter
- **MemOccupancy**
  - Level of the memory occupancy (not yet implemented)
- **ChSum**
  - The DAC reproduces the “analog” sum of all the digitizer inputs (not yet implemented)
- **OverThrSum**

  - The DAC output is proportional to the number of channels that are currently above the threshold

parameter DACoutStaticLevel

When DACoutMode = Static, this parameter sets the 14-bit level of the DAC static output.

parameter DACoutChSelect

When DACoutMode = ChInput, the DAC reproduces the input signal of the channel selected by this parameter.

parameter IPEAmplitude

The new digitizers are equipped with an Internal Pulse Emulator capable of generating exponential pulses. This parameter determines the amplitude of the pulse.

Unit of Measure: ADC counts

parameter IPEBaseline

Sets the offset of the exponantial pulses generated by the Internal Pulse Emulator.

Unit of Measure: ADC counts

parameter IPEDecayTime

Sets the decay time of the exponantial pulses generated by the Internal Pulse Emulator.

Unit of Measure: ns

parameter IPERate

Sets the rate of the exponantial pulses generated by the Internal Pulse Emulator.

Unit of Measure: Hz

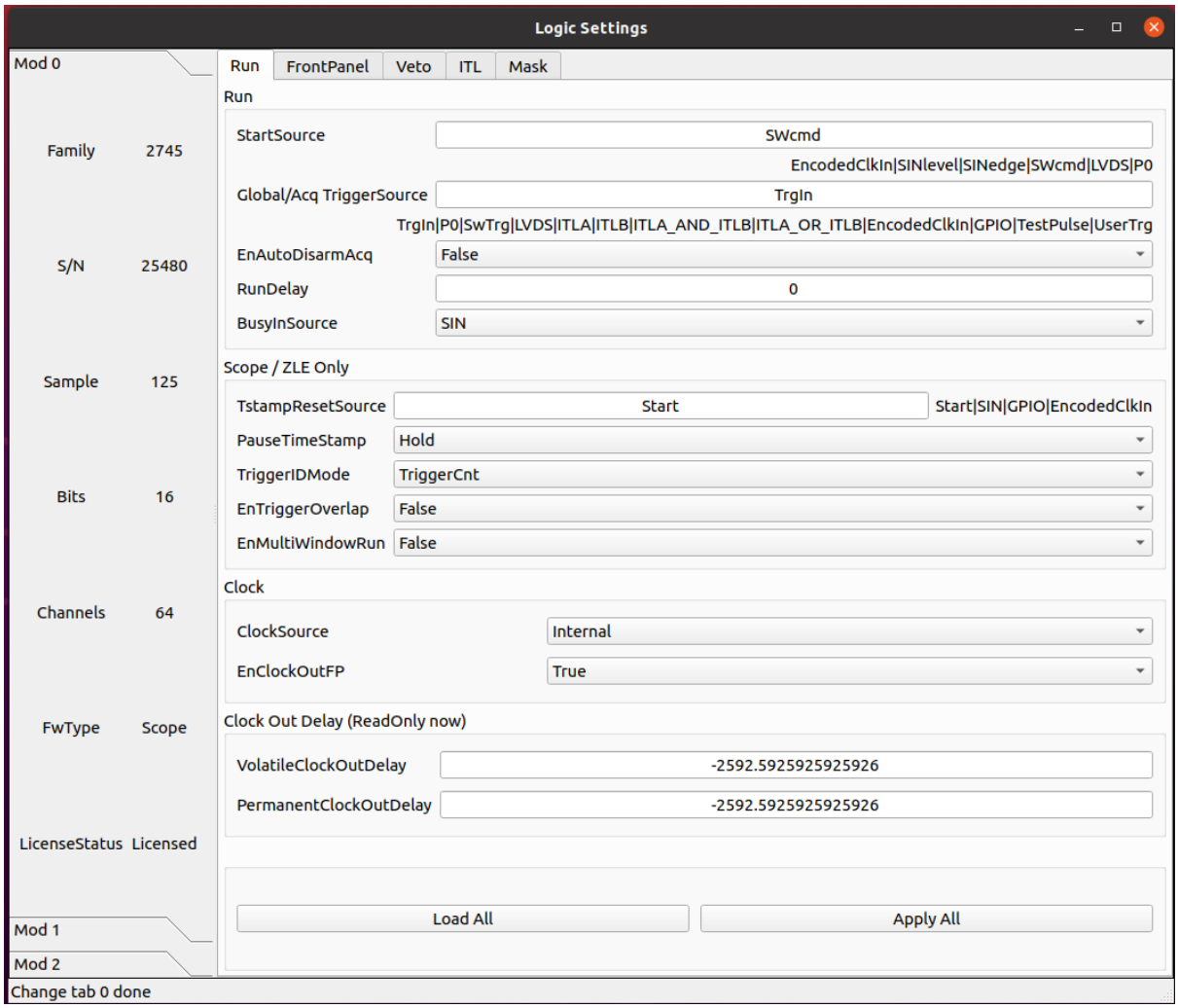
parameter IPETimeMode

Selectes the time distribution of the Internal Pulse Emulator.

- **ConstantRate**
  - Pulse shapes are constant over time. It is possible to set the frequency using the IPERate parameter
- **Poissonian**
  - The pulse rate follows a Poisson distribution. The average frequency value can be configured using the IPERate parameter

7.2 Logic parameter

7.2.1 Run



parameter StartSource

Defines the source for the start of run. Multiple options are allowed, separated by "|".

- **EncodedClkIn**
  - Start from CLK-IN/SYNC connector on the front panel. This is a 4-pin connector (LVDS signals) used to propagate the reference clock (typ. 62.5 MHz) and a Sync signal. The rising edge of the Sync starts the acquisition, that lasts until the Sync returns low (falling edge).
- **SINlevel**
  - Start from SIN (1=run, 0=stop)
- **SINedge**
  - Start from SIN (rising edge = run; stop from SW)
- **SWcmd**
  - Start from SW
- **LVDS**

  - Start from LVDS

- **FirstTrigger**
  - Start on 1st trigger (stop from SW)
- **P0**
  - Start from P0 (backplane)

parameter AcqTriggerSource

Defines the source for the Acquisition Trigger, which is the signal that opens the acquisition window and saves the waveforms in the memory buffers. Multiple options are allowed, separated by "|".

- **TrgIn**
  - Front Panel TRGIN
- **P0**
  - Trigger from P0 (backplane)
- **SwTrg**
  - Software trigger
- **LVDS**
  - LVDS trgin
- **ITLA**
  - Internal Trigger Logic A: combination of channel self-triggers
- **ITLB**
  - Internal Trigger Logic B: combination of channel self-triggers
- **ITLA_AND_ITLB**
  - Second level Trigger logic making the AND of ITL A and B
- **ITLA_OR_ITLB**
  - Second level Trigger logic making the OR of ITL A and B
- **EncodedClkIn**
  - Not implemented (encoded CLK-IN trigger)
- **GPIO**
  - Front Panel GPIO
- **TestPulse**
  - Internal Test Pulse
- **UserTrg**
  - User custom trigger source

parameter EnAutoDisarmAcq

When enabled, the Auto Disarm option disarms the acquisition at the stop of run. When the start of run is controlled by an external signal, this option prevents the digitizer to restart without the intervention of the software.

- **True**
  - The acquisition is automatically disarmed after the stop. It is therefore necessary to rearm the digitizer (with the relevant command sent by the software) before starting a new run.
- **False**
  - The acquisition is not disarmed after the stop. Multiple transition of the start signal will produce multiple runs.

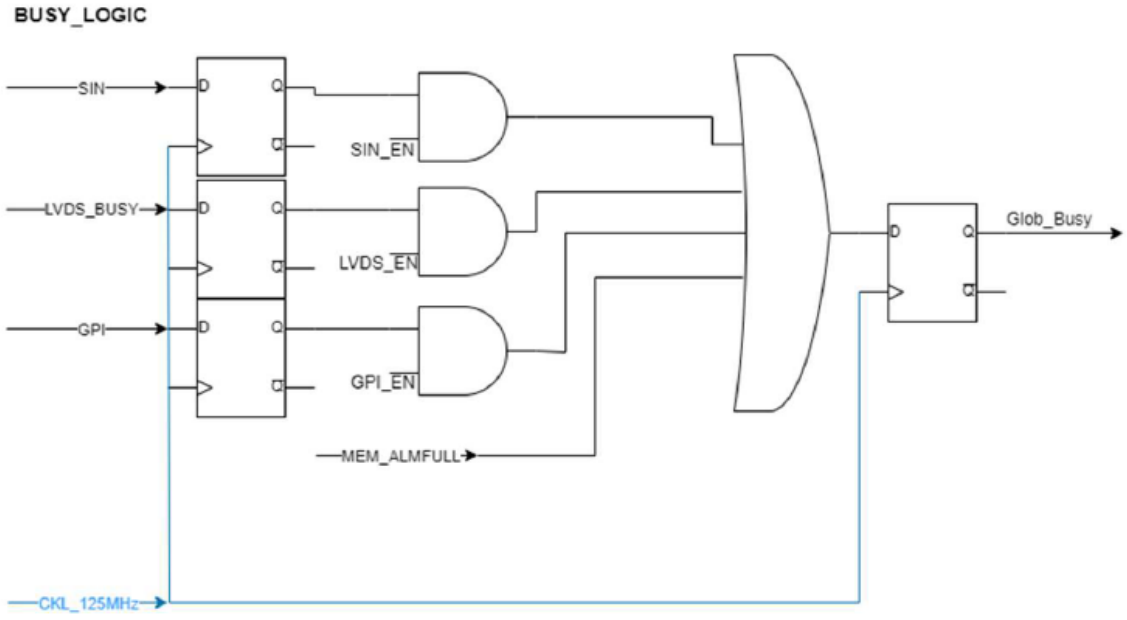
parameter RunDelay

When the start of run is controlled by a RUN signal that is propagated in daisy chain between the boards (for instance through the ClkIn- ClkOut or SIN-GPIO sync chain), it is necessary to compensate for the propagation delay and let the boards start exactly at the same time. The RunDelay parameter allows the start of the acquisition to be delayed by a given number of clock cycles with respect to the rising edge of the RUN signal. Assuming that the propagation delay is 2 cycles, the RunDelay setting will be 0 for the last board in the chain, 2 for the previous one, and so on up 2x(NB-1) for the first one.

Unit of Measure: ns

parameter BusyInSource

In a multi-board system, it might be necessary to prevent one board to accept a new trigger while another board is full and thus unable to accept the same trigger. For this reason, each board can generate a Busy signal to notify that it is unable to get a new trigger. If the busy/veto mechanism has some latency, it is advisable to generate the busy slightly before the digitizer become full. For this purpose, it is possible to assert the busy output when the acquisition memory reaches a certain level of occupancy (internally managed). The OR of the busy signals is typically used to stop the global trigger. It is possible to get the individual busy signals from each board and make an external OR logic or connect the boards with cables to propagate the Busy along the chain. Each board makes an OR between its internal busy and the busy input signal coming from the previous board, thus having a global Busy at the end of the line. This parameter defines the source of the Busy Input (schematized in the figure below)



- **Disabled**
  - The Busy is given by the Internal Busy only (Memory full or almost full)
- **SIN**
  - Busy input from SIN on front panel
- **GPIO**
  - Busy input coming from GPIO on front panel, used as a simple input. It is also possible to use GPIO as a wired OR (bidirectional). In this mode, the Busy line goes high as soon as one board drives it high. All the boards can read the Busy line and use it as a veto for the trigger
- **LVDS**
  - LVDS trgin

parameter TstampResetSource

Defines the source of the timestamp reset. Multiple options are allowed, separated by "|". The timestamp of the board (internal counter running at 125 MHz) is typically reset at the start of each run, which corresponds to the "zero" of the timestamps. In Multi-board systems, the synchronization of the clock and RUN signals allows event data coming from different boards to be merged and correlated by the time stamp. However, it is possible to configure different ways to control the reset of the time stamp in the cases where it is necessary to synchronize it with an external global time stamping system.

- **Start**
  - Time stamp reset at the start of run
- **SIN**
  - SIN input
- **GPIO**
  - GPIO used as input
- **EncodedClkIn**
  - Not implemented (encoded in CLK-IN/SYNC)

parameter PauseTimeStamp

Allows the time stamp to either stop or run during the pauses of the acquisition

- **Hold**
  - The timestamp stops while pausing the acquisition
- **Run**
  - The timestamp runs while pausing the acquisition

parameter TriggerIDMode

The event data packet contains a 24-bit identifier called TriggerID. This can be the total trigger counter, the saved event counter or a pattern coming from the LVDS I/Os.

- **TriggerCnt**
  - The Event triggerID is associated to the total trigger counter. This is a 24-bit counter that is reset at the start of run and increased with every received trigger, including those ones that are not accepted. In this mode, events coming from multiple boards can be correlated by the triggerID that is supposed to by synchronized. There might be gaps due to lost triggers.
- **EventCnt**
  - The Event triggerID is a sequential number of the saved event. In this case, there are no gaps in the sequence, but it is not guaranteed that the trigger ID of multiple boards are aligned, thus it is not possible to use it for data correlation.
- **LVDSpattern**
  - The triggerID is taken from the 16 LVDS inputs at the time of the trigger arrival. The user can provide an external trigger pattern to correlate the event data between boards or even with other readout electronics

parameter EnTriggerOverlap

Allows a trigger occurring within the acquisition window of a previous trigger to be either accepted or rejected. When accepted, the previous window is prematurely closed and the new window immediately opened, without any dead time between the two.

- **True**
  - Triggers with overlapped acquisition windows are accepted.
- **False**
  - Triggers with overlapped acquisition windows are not accepted. The rejected triggers are counted by the total trigger counter, so that the trigger-ID in the event header allows for tracing the rejected triggers.

parameter EnMultiWindowRun

When the acquisition start and stop are controlled by an external "RUN" signal (e.g. feeding SIN), it is possible to configure the digitizer to work in two different modes:

The RUN signal acts as a start (rising edge) and stop (falling edge). Therefore, multiple transitions of the RUN signal cause multiple runs (provided that the Auto Disarm option is disabled). At every start of run, the timestamp is reset, and all the statistics and counters are cleared. Typically, the software produces different output files.

The RUN signal acts as "enable" of the acquisition: once the digitizer has been armed, the first rising edge of RUN starts the acquisition. When RUN goes down, the acquisition is "paused" rather than stopped. This means that all data and statistics are frozen, and the timestamp can be either stopped or left running, depending on the PauseTimeStamp parameter. The RUN signal can toggle multiple times within the same acquisition. The stop of the acquisition will be done by a software command. It is necessary to disarm and rearm the acquisition before starting a new run with the rising edge of the RUN signal.

- **True**
  - MultiWindow run is enabled. The RUN signal acts as start (first rising edge) and pause (subsequent falling edges) for the acquisition. The stop of the acquisition is always given by a software command
- **False**

  - The RUN signal acts as start and stop for the acquisition

parameter ClockSource

This is the source of the system clock. Multiple options are not allowed

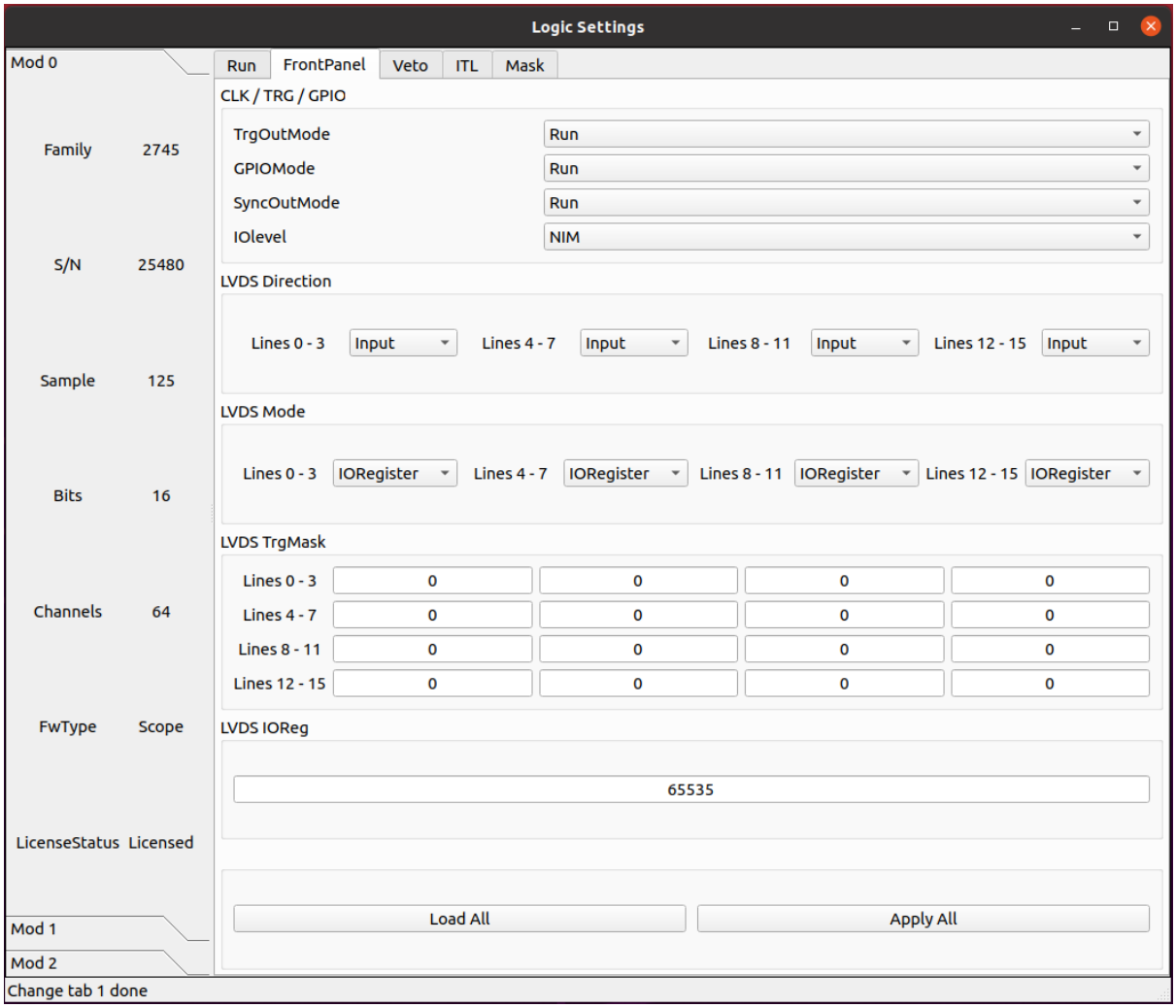
- **Internal**
  - Local oscillator, 62.5 MHz
- **FPClkIn**
  - Front Panel Clock input

parameter EnClockOutFP

Enables clock output on Front Panel for the daisy chain propagation of the clock between multiple boards.

- **True**
  - Enabled
- **False**
  - Disabled

7.2.2 FrontPanel



parameter TrgOutMode

Selects the signal that is routed to the TRGOUT output. Multiple options are not allowed.

- **Disabled**
  - TRGOUT output disabled
- **TrgIn**
  - Propagation of Front Panel TRGIN (TRGOUT is a replica, with some delay, of the TRGIN signal)
- **P0**
  - Propagation of P0 trigger
- **SwTrg**
  - Software trigger
- **LVDS**
  - LVDS trgin
- **ITLA**
  - Internal Trigger Logic A: combination of channel self-triggers

- **ITLB**
  - Internal Trigger Logic B: combination of channel self-triggers
- **ITLA_AND_ITLB**
  - Second level Trigger logic making the AND of ITL A and B
- **ITLA_OR_ITLB**
  - Second level Trigger logic making the OR of ITL A and B
- **EncodedClkIn**
  - Not implemented (propagation of the Encoded CLK-IN trigger)
- **Run**
  - Propagation of the RUN signal (acquisition start/stop), before applying the delay given by the Run-Delay parameter
- **RefClk**
  - Monitor of the 62.5 MHz clock (used for phase alignment)
- **TestPulse**
  - Internal Test Pulse
- **Busy**
  - Busy of the board
- **UserTrgout**
  - Trgout coming from the User Logic (open FPGA)
- **Fixed0**
  - 0 level signal
- **Fixed1**
  - 1 level signal
- **SyncIn**
  - SyncIn signal
- **SIN**
  - SIN connector signal
- **GPIO**
  - GPIO connector signal
- **LBinClk**
  - Internal Logic B clock signal
- **AcceptTrg**
  - Accepted triggers signal
- **TrgClk**
  - Tigger clock signal

parameter GPIOMode

Selects the signal that is routed to the GPIO, when this is used as output. Multiple options are not allowed. The GPIO on the front panel is a bidirectional signal that can used in three different ways:

As independent board output (each board drives its own GPIO)

As a shared input for the boards: the signal is driven high (= 1) or low (= 0) by an external source and connected in "short circuit" among multiple boards using "T" connectors at the inputs. The GPIO is not internally terminated, thus it is necessary to put a 50 Ohm terminator at the end of the line (last "T" of the chain)

As a shared bidirectional line, making a "wired OR". One or more boards can simultaneously drive the signal high (= 1). If no board drives the GPIO, it remains low (= 0). All boards can read back the signal. It is necessary to put a 50 Ohm terminator at both ends of the line (first and last "T" of the chain). This mode can be used to generate, for instance, the global Busy and Veto logic for multiple boards.

- **Disabled**
  - GPIO disabled
- **TrgIn**
  - Propagation of Front Panel TRGIN (GPIO is a replica, with some delay, of the TRGIN signal)
- **P0**
  - Propagation of P0 trigger
- **SIN**
  - Propagation of SIN
- **LVDS**
  - LVDS trgin
- **ITLA**
  - Internal Trigger Logic A: combination of channel self-triggers
- **ITLB**
  - Internal Trigger Logic B: combination of channel self-triggers
- **ITLA_AND_ITLB**
  - Second level Trigger logic making the AND of ITL A and B
- **ITLA_OR_ITLB**
  - Second level Trigger logic making the OR of ITL A and B
- **EncodedClkIn**
  - Not implemented (propagation of the Encoded CLK-IN trigger)
- **SwTrg**
  - Software trigger
- **Run**
  - Propagation of RUN
- **RefClk**
  - Monitor of the 62.5 MHz clock (used for phase alignment)
- **TestPulse**
  - Internal Test Pulse

- **Busy**
  - Busy of the board
- **UserGPO**
  - GPO coming from the User Logic (open FPGA)
- **Fixed0**
  - 0 level signal
- **Fixed1**
  - 1 level signal

parameter SyncOutMode

In a multi-board system, it can be useful to propagate a synchronous signal together with the clock (to synchronize the start of the run, for example) on CLK OUT front panel connector. This parameter defines which signal must be sent out. Multiple options are not allowed.

- **Disabled**
  - SyncoutMode is disabled
- **SyncIn**
  - SyncIn signal (if provided with clkIn on CLK IN connector)
- **TestPulse**
  - Internal Test Pulse
- **IntClk**
  - Internal 62.5 MHz clock (for test purposes)
- **Run**
  - Propagation of RUN signal
- **User**
  - User customSyncoutMode

parameter IOlevel

Sets the electrical logic level of the LEMO I/Os (TRGIN, SIN, TRGOUT, GPIO).

Note that TRGIN and SIN are internally terminated to 50 Ohm, while GPIO and TRGOUT require the termination to 50 Ohms at the receiver

- **NIM**
  - NIM logic (0 = 0V, 1 = -0.8V, that is -16mA)
- **TTL**
  - Low Voltage TLL logic (0 = 0V, 1 = 3.3V)

parameter LVDSDirection

Assigns the direction to a quartet of LVDS I/Os.

- **Input**
  - The LVDS lines of the relevant quartet are used as input. The relevant LED on the front panel is OFF.
- **Output**
  - The LVDS lines of the relevant quartet are used as output. The relevant LED on the front panel lights-up.

parameter LVDSMode

The digitizer is equipped with 16 LVDS I/Os that can be programmed to be inputs or outputs by groups of 4 (quartets), depending on the LVDSDirection parameter. Once the direction has been selected, it is possible to select the functionality of the LVDS lines, individually for each quartet.

- **SelfTriggers**
  - This option is available only when the LVDS are set as outputs. Each LVDS line can be assigned to a combination of the 64 self-triggers, implemented as a masked OR, where the mask is set by the LVDSTrgMask parameter(16 independent masks, one per LVDS line)
- **Sync**
  - Whatever is the direction of the quartet, the 4 lines are rigidly assigned to specific acquisition signals: 0 = Run 1 = Trigger 2 = Busy 3= Veto It is possible to implement a daisy chain distribution of these signals using one quartet as input and another one as output
- **IORegister**
  - The LVDS lines of the quartet are statically controlled by the LVDSIOReg parameter. Use the SetValue function to set the relevant LVDS lines when programmed as output. Use GetValue to read the status of the LVDS lines when programmed as inputs.
- **User**
  - User custom.

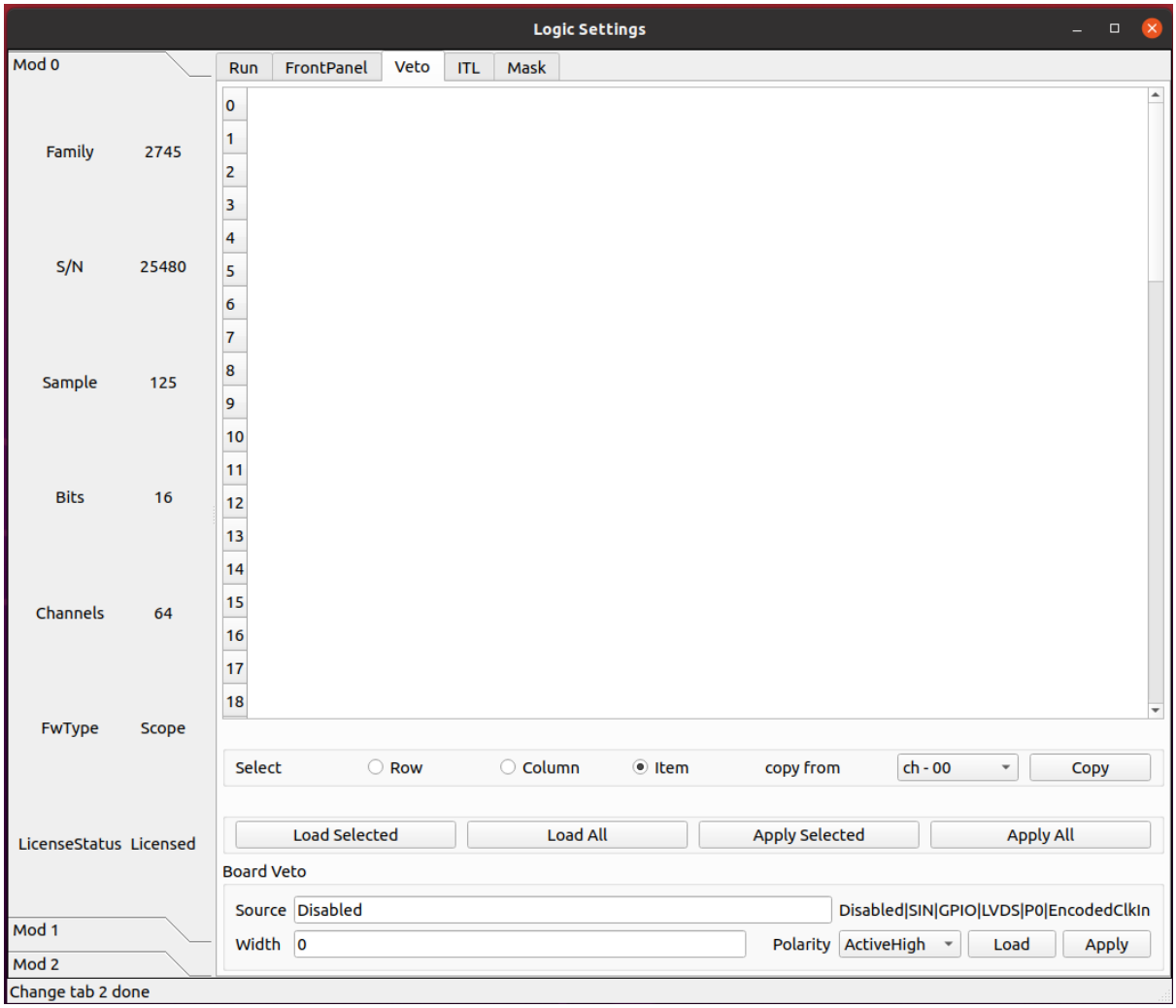
parameter LVDSTrgMask

Each LVDS line can be assigned to a combination of the 64 self-triggers, implemented as a masked OR, where the mask is set by this parameter. There are 16 independent masks, one per LVDS line. Note that the trigger mask assignment does not imply the LVDS direction and mode settings. It is therefore necessary to set the Direction = Output and Mode = SelfTriggers to use the Self-Trigger propagation to the LVDS I/Os.

parameter LVDSIOReg

Set the status of the LVDS I/O for the quartets when they are programmed to be output and Mode = IORegister. This parameter reads out the status of the quartets in the case the LVDS I/O are programmed as inputs (possibly externally driven).

7.2.3 Veto



parameter VetoSource

Defines the source for the Veto, which is the signal that inhibits the acquisition trigger. Multiple options are allowed, separated by "|". The VETO signal can be either active high or low, depending on the VetoPolarity parameter. When active low, it acts as a GATE for the trigger. It is possible to stretch the duration of the VETO by means of the parameter VetoWidth.

- **Disabled**
  - VETO is always OFF
- **SIN**
  - SIN on the front panel
- **GPIO**
  - GPIO on the front panel (used as input)
- **LVDS**
  - LVDS trgin
- **P0**
  - P0 (signal from the backplane)

- **EncodedClkIn**
  - Not implemented (encoded CLK-IN veto)

parameter VetoWidth

Whatever is the source of the VETO signal, it is possible to stretch the duration of the veto up to a given time by means of a re-triggerable monostable. When 0, the monostable is disabled and the veto lasts as long as the selected source is active.

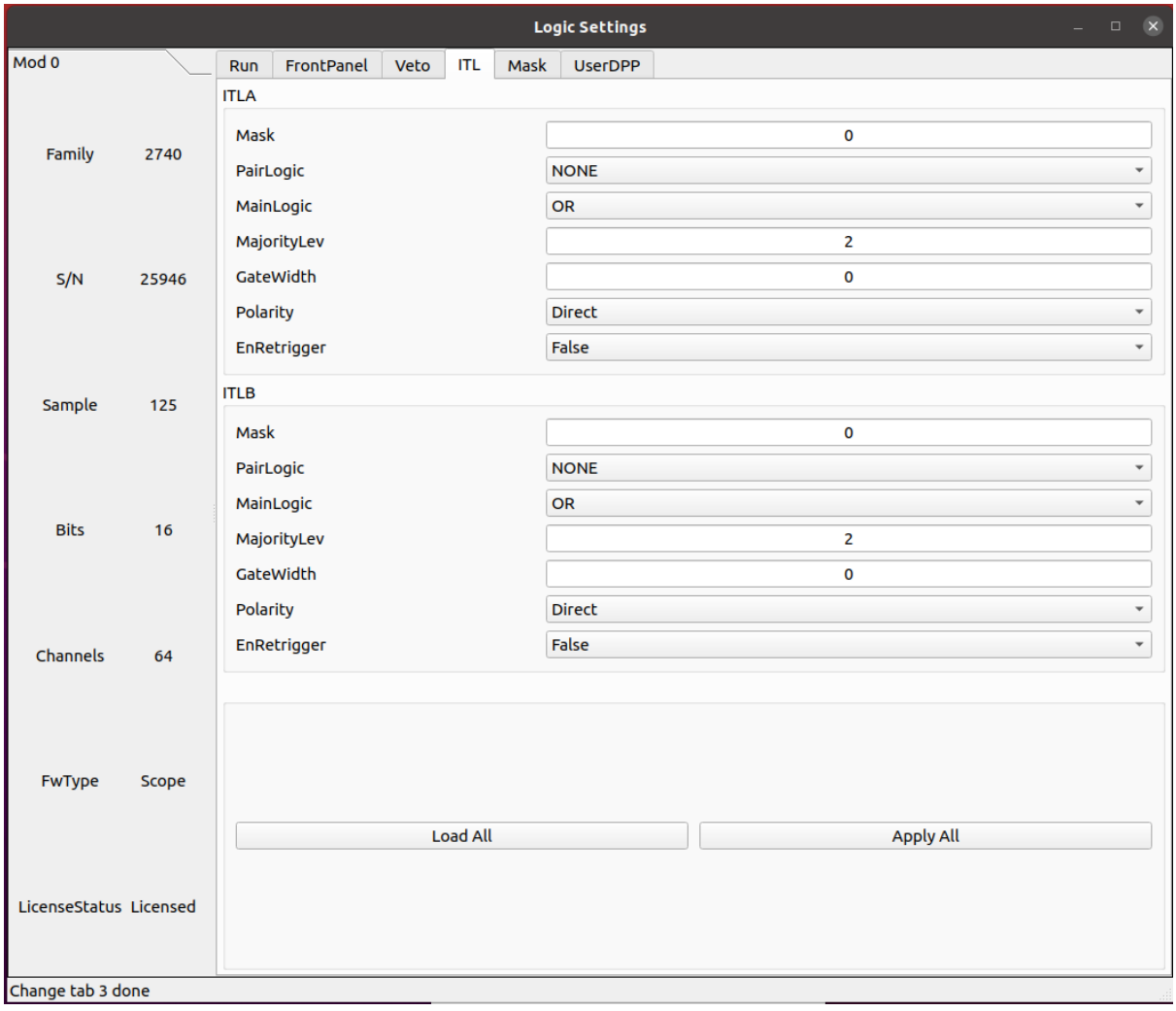
Unit of Measure: ns

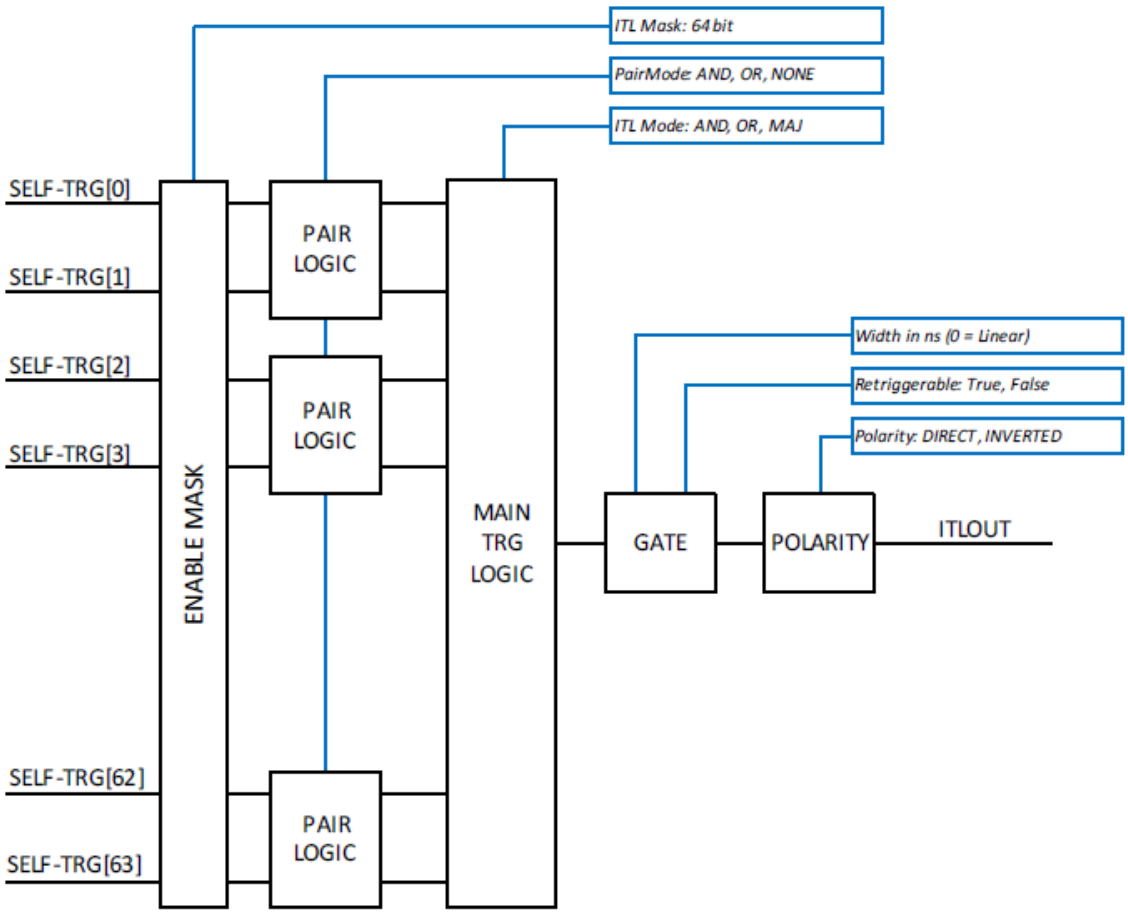
parameter VetoPolarity

Defines the polarity of the Veto

- **ActiveHigh**
  - Veto is active high. The signal acts as an "Inhibit" for the trigger
- **ActiveLow**
  - Veto is active low. The signal acts as a "Gate" the trigger

7.2.4 ITL





parameter ITLA/BMask

Enable Mask at the input of the ITLA/B.

parameter ITLA/BPairLogic

Pairs of channels can be combined with an OR or AND before feeding in the Main trigger Logic. This is typically used in the readout of tubes or scintillator bars, where the two ends are read in coincidence, for instance in position sensitive detectors (the coincidence window will be set by the SelfTriggerWidth parameter). When the AND/OR logic is applied, the two outputs of the Pair Logic blocks are identical.

Note that they are counted twice in the following Majority logic. If the Pair Logic is disabled ("NONE" option), the block is transparent, and the two outputs are just a replica of the inputs.

- **OR**
  - Both Pair Logic Outputs = OR of two consecutive self-triggers
- **AND**
  - Both Pair Logic Outputs = AND of two consecutive self-triggers
- **NONE**
  - Outputs = Inputs

parameter ITLA/BMainLogic

Each channel of the digitizer feature a digital bipolar triangular filter discriminator with programmable rise time and threshold able to self-trigger on the input pulses and generate a self-trigger signal. In DPP Mode, the channels acquire independently, so the channel self-trigger is used locally to acquire a waveform. The trigger threshold is then referred to the bipolar triangular filter, and the threshold crossing arms the event selection. The trigger fires at the zero crossing of the time filter signal. The user can see the derivative trace on the signal inspector. It is also possible to combine all the self-triggers of the board, according to a specific trigger logic. There are two independent logic blocks, ITLA and ITLB. Their output can be used separately to feed, for instance, AcqTrigger and TrgOut, or combined in a second level trigger logic to implement more complex trigger schemes. Therefore, the ITLs can either generate the local acquisition trigger, common to all the channels, for the acquisition of the waveform, or propagate the signal outside, through the TRGOUT, thus making it possible to combine triggers of multiple boards in an external trigger logic, that eventually feeds back the TRGIN of the digitizers. Each ITL is made of an input enable mask (64 bits, one per channel), an optional pairing logic that combines the self triggers of two consecutive channels (e.g. paired coincidence) and the main trigger logic that combines the 64 selftriggers with an OR, AND or Majority logic. The output can be linear (no stretching) or reshaped by a programmable gate generator, either re-triggerable or not and finally programmed for polarity (direct or inverted).

- **OR**
  - ITLOUT = masked OR of channel self-triggers
- **AND**
  - ITLOUT = masked AND of channel self-triggers
- **Majority**
  - ITLOUT = masked Majority of channel self-triggers

parameter ITLA/BMajorityLev

Defines the majority level of the Main Logic of the ITL A/B block. The majority output is calculated at every clock cycle, and it becomes TRUE when Nch >= MajLev, where Nch is the number of self-triggers active in that clock cycle and MajLev is the programmed majority level.

Note that when the Pair Logic is used to combine the self triggers two by two (AND/OR), each pair produces two identical signals that will be counted twice in the majority level.

parameter ITLA/BGateWidth

Width of the gate generator at the output of the ITLA/B block.

Unit of Measure: ns

parameter ITLA/BPolarity

Polarity of the gate generator output.

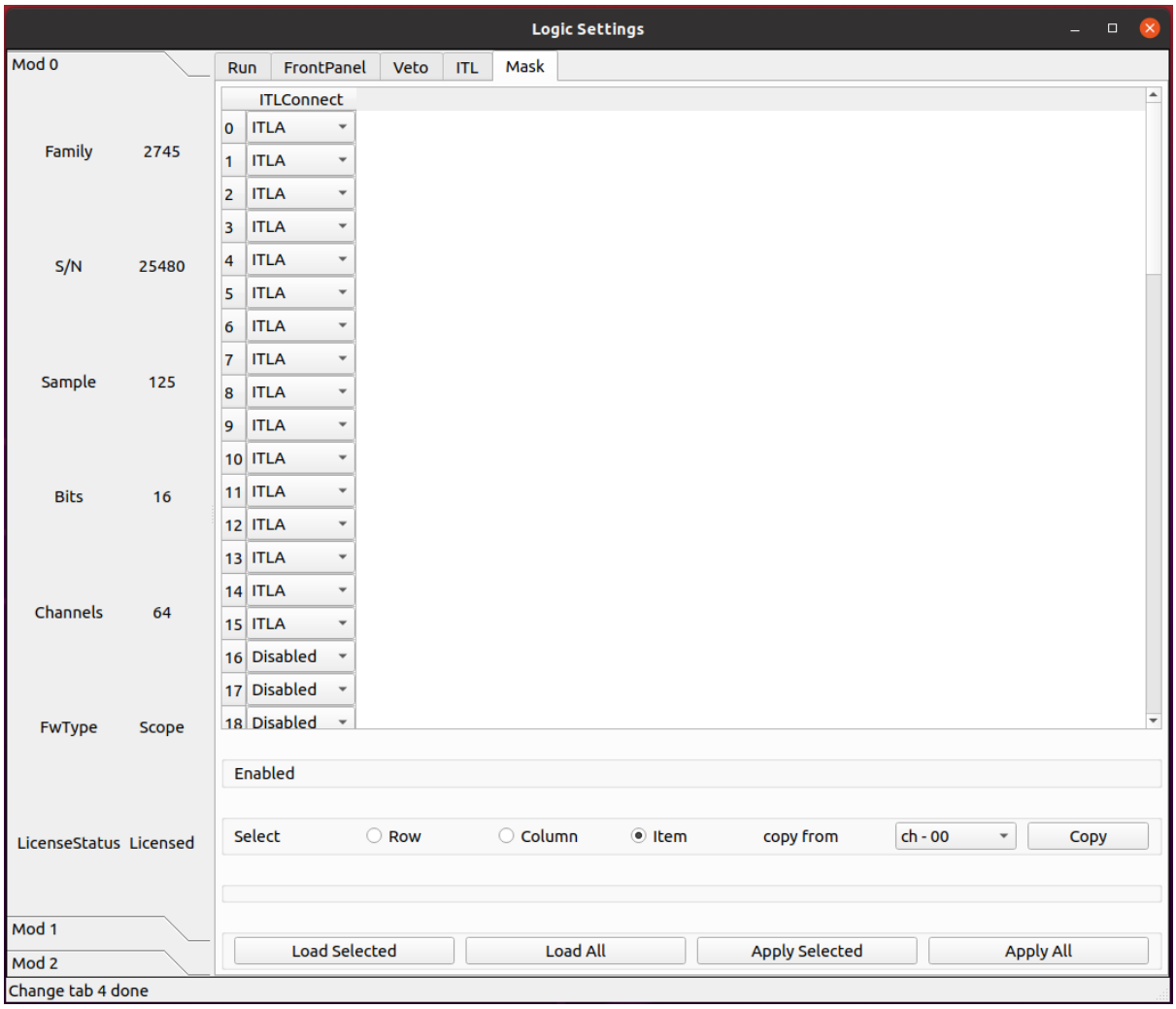
- **Direct**
  - Direct polarity
- **Inverted**
  - Inverted polarity

parameter ITLA/BEnRetrigger

Set the ITLA/B to be retriggerable.

- **True**
  - The ITLA/B is retriggerable
- **False**
  - The ITLA/B is not retriggerable

7.2.5 mask



parameter ITLConnect

Alternative to ITLAMask, ITLBMask. Determines if the channel partecipate in ITLA or ITLB

- **Disabled**
  - The channel is disabled
- **ITLA**
  - The channel participates in ITLA logic block
- **ITLB**

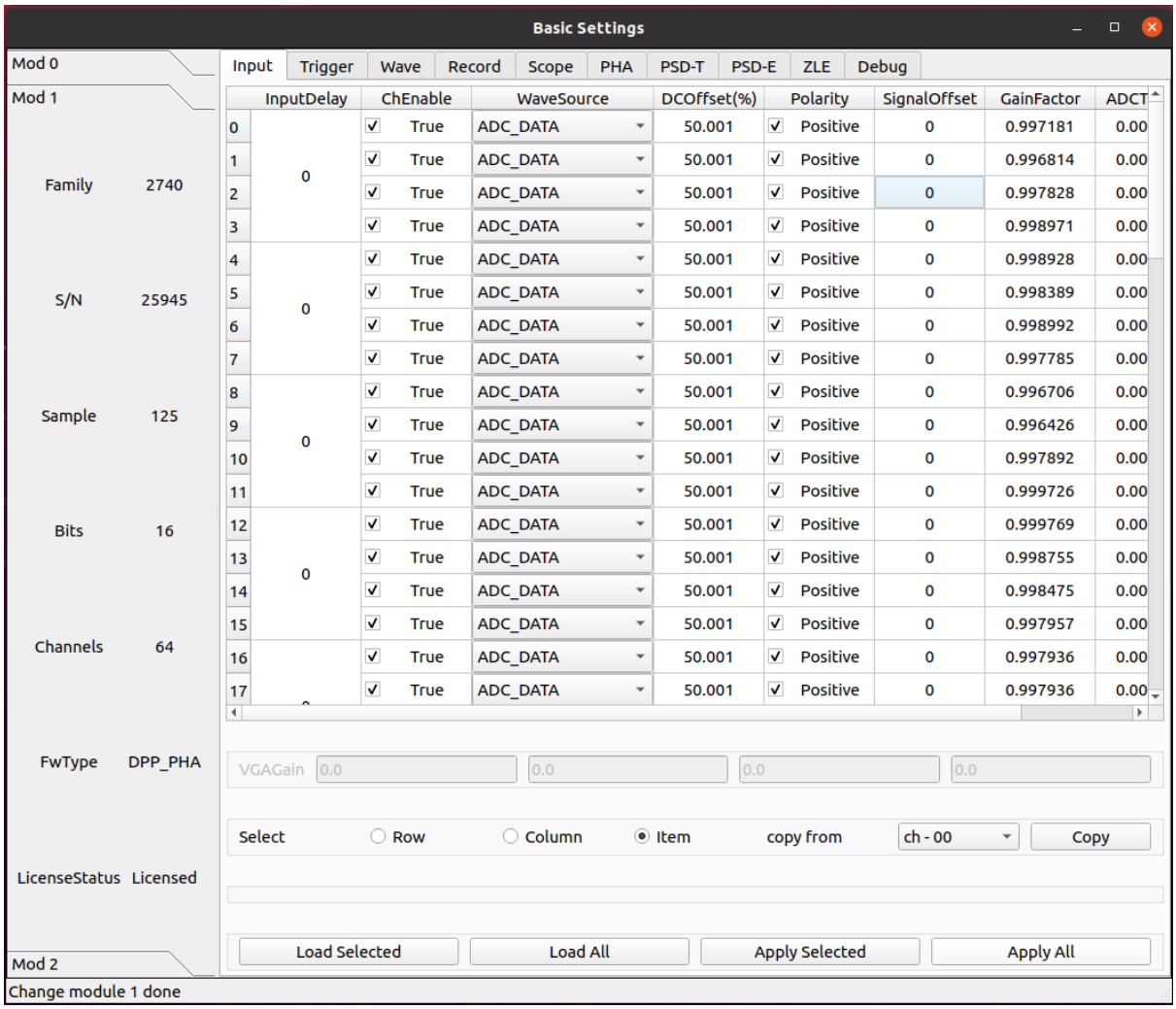
- The channel participates in ITLB logic block

CHAPTER 8

PHA firmware

8.1 Basic

8.1.1 Input



parameter ChGain

Unique to x2730.

Sets the gain of the Variable Gain Amplifiers (VGA). Unit of Measure: dB

parameter InputDelay

Set input delay. The value is set at groups of 4 channels for x2745/x2740.

Unit: sample

parameter ChEnable

Enable the channels for the acquisition, according to the Index. When the channel is disabled, it does not give any data and its self-trigger is off.

- True: The channel is enabled for the acquisition
- False: The channel is disabled for the acquisition

parameter WaveSource

In normal mode, the acquired waveform represents a sequence of ADC samples, resulting from the A/D conversion of the analog input. For test purposes, it is possible to replace the ADC data with internal data generators.

- **ADC_DATA**
  - Data from the ADC (normal operating mode)
- **ADC_TEST_TOGGLE**
  - Toggle between 0x5555 and 0xAAAA (test mode)
- **ADC_TEST_RAMP**
  - 16-bit ramp pattern (test mode)
- **ADC_TEST_SIN**
  - 8-point sine wave test pattern
- **ADC_TEST_PRBS**
  - 16-bit PRBS generated by a 23-bit PRBS pattern generator (test mode)
- **Ramp**
  - Data from a ramp generator. It is actually a 16-bit field, where the 6 most significant bits identify the channel and the 10 less significant bits are the samples of a ramp from 0x000 up to 0x3FF (i.e. 0 to 1023). It is so a 10-bit ramp with offset given by “channel*1024”. For channel 0, it is a counter from 0 to 1023; for channel 1, it is a counter from 1024 to 2047, and so on
- **IPE**
  - Not implemented
- **SquareWave**
  - Internally generated programmable square wave

parameter DCOffset

A constant DC offset (controlled by a 16-bit DAC) is added to the analog input, individually for each channel, to adjust the position of the signal baseline (that is the "zero volt" of the analog input) within the dynamic range of the ADC. Because of the tolerance of the components, it is necessary to calibrate the offset DAC. The calibration is done by factory testing and normally it is not necessary to recalibrate the digitizer. It is however possible to perform a new calibration. The calibration parameters are stored in the flash memory of the board and loaded at power on. They are automatically applied by the internal logic every time the DCoffset parameter is written or read. DCoffset is expressed as a NUMBER number, in percent of the full-scale. When the DCoffset is 0, the baseline of the input signal is at 0 ADC counts. When the DCoffset is 100, the baseline of the input signal is at $2^{NBIT}-1$ ADC counts.

parameter Polarity

Allows to set the polarity of the input pulse.

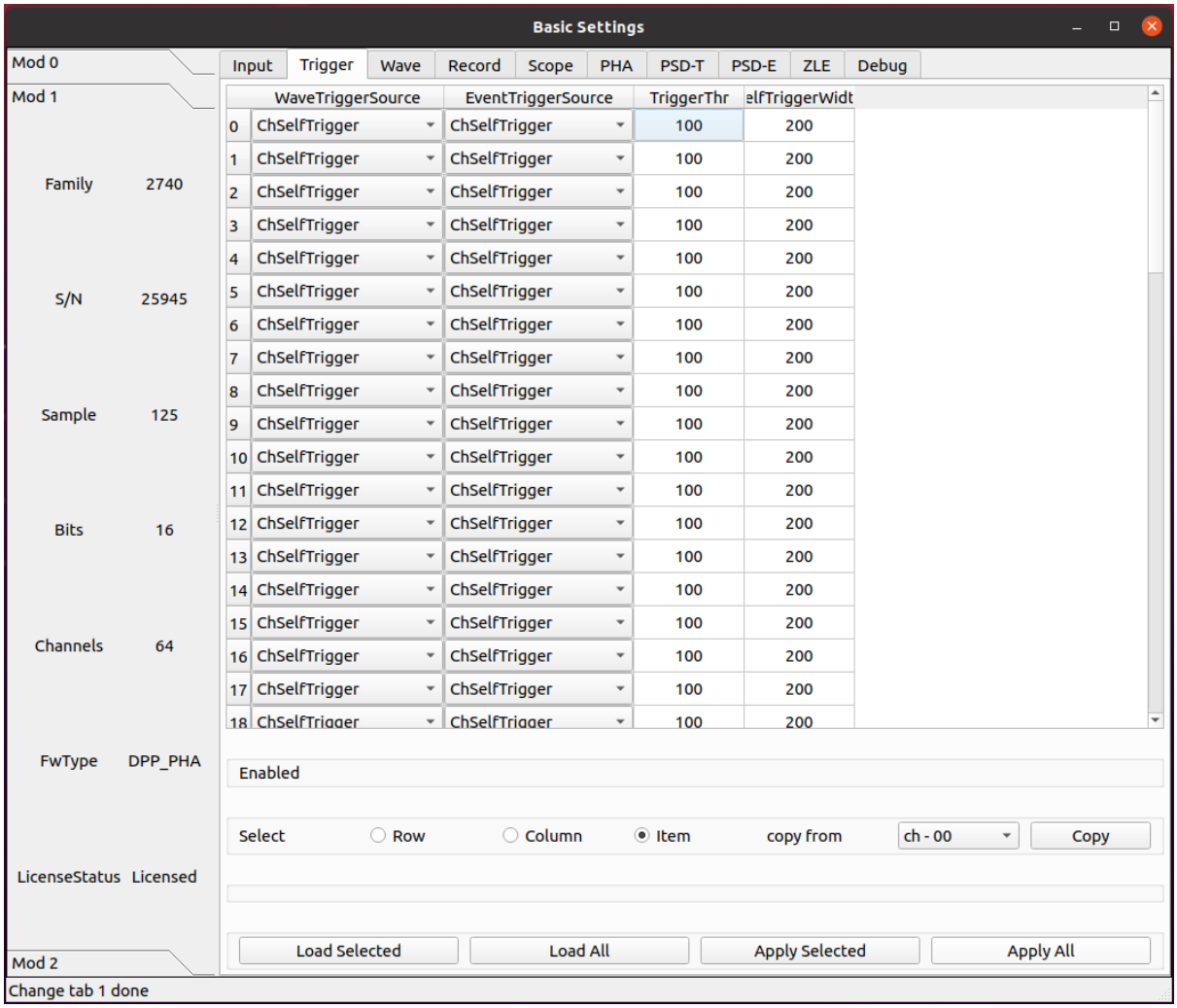
- **Positive**
  - Positive polarity
- **Negative**
  - Negative polarity

parameter VGAGain

Unique to x2745.

Set the gain of the variable gain amplifier (VGA) in increments of 0.5 dB. Parameter settings are grouped every 16 channels, with 64 channels divided into 4 groups. The minimum can be set to 0 and the maximum to 40.

8.1.2 Trigger



parameter WaveTriggerSource

Allows to set the trigger source for the waveform. Setting this parameter means to get an event including the waveform and the associated time stamp and energy information..

- **Disabled**
  - No trigger source enabled for the waveform
- **Ch64Trigger**
  - One (or more) channel self-trigger can generate a trigger for a waveform
- **ChSelfTrigger**
  - Channel self-trigger can generate a trigger for a waveform
- **SwTrg**
  - Software Trigger can generate a trigger for a waveform
- **ADCOverSaturation**
  - ADC Oversaturation can generate a trigger for a waveform
- **ADCUnderSaturation**
  - ADC Undersaturation can generate a trigger for a waveform

- **ExternalInhibit**
  - Inhibit can generate a trigger for a waveform
- **TRGIN**
  - External TRGIN can generate a trigger for a waveform
- **GlobalTriggerSource**
  - Acquisition Trigger Source (the same of the Scope mode) can generate a trigger for a waveform
- **LVDS**
  - A signal on the LVDS connectors can generate a trigger for a waveform
- **ITLA**
  - Internal Trigger Logic A can generate a trigger for a waveform
- **ITLB**
  - Internal Trigger Logic B can generate a trigger for a waveform

parameter EventTriggerSource

Allows to set the trigger source for a Time-Energy (T-E) event. Setting this parameter means to get an event including time stamp and energy information

- **Disabled**
  - No trigger source enabled for the T-E event
- **Ch64Trigger**
  - One (or more) channel self-trigger can generate a trigger for a T-E event
- **ChSelfTrigger**
  - Channel self-trigger can generate a trigger for a T-E event
- **SwTrg**
  - Software Trigger can generate a trigger for a T-E event
- **TRGIN**
  - External TRGIN can generate a trigger for a T-E event
- **GlobalTriggerSource**
  - Acquisition Trigger Source (the same of the Scope mode) can generate a trigger for a T-E event
- **LVDS**
  - A signal on the LVDS connectors can generate a trigger for a T-E event
- **ITLA**
  - Internal Trigger Logic A can generate a trigger for a T-E event
- **ITLB**
  - Internal Trigger Logic B can generate a trigger for a T-E event

parameter TriggerThr

Each channel of the digitizer has a digital leading-edge discriminator with programmable threshold able to self-trigger on the input pulses and generate a self-trigger signal (or an overthreshold signal) feeding the internal trigger logics or digitizer outputs. This parameter sets the trigger threshold. Typically, the value is relative to the baseline of the signal and the threshold is a 17-bit signed NUMBER number; in this case, the threshold automatically follows the baseline when the DCoffset parameter changes. Sometimes, it is preferable to set an absolute value for the threshold, referred to the ADC range. In this case, the threshold is unsigned NUMBER number.

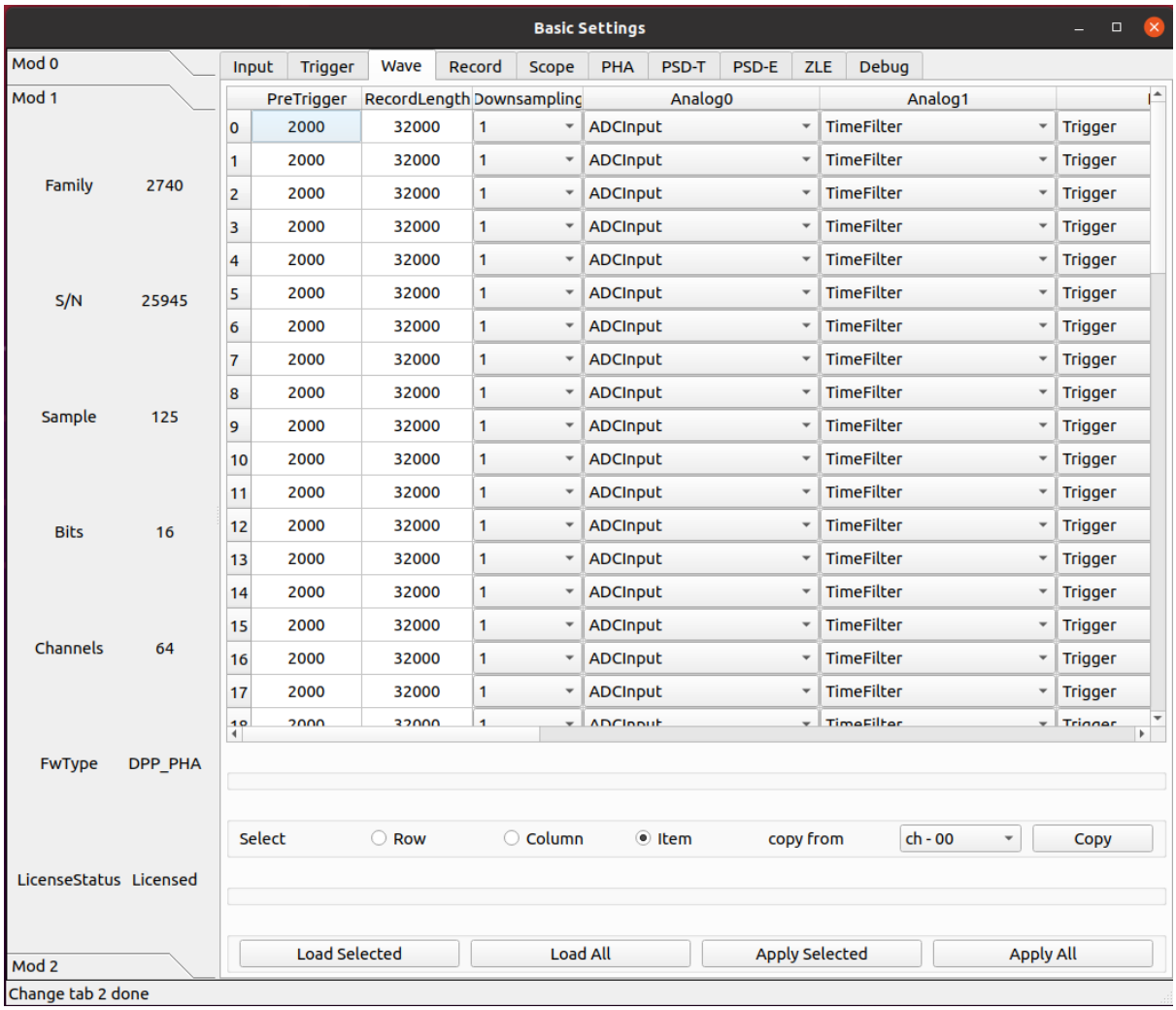
Unit of Measure: ADC counts

parameter SelfTriggerWidth

The output of the digital leading-edge comparator, that generates the self-trigger signal, can be used in "linear" mode, meaning that it lasts for the time the signal remains above (or below) the threshold, thus acting as an "Over-Threshold" signal, or can pass through a programmable gate generator that makes it a fixed-width pulse. The gate generator is a non-retriggerable monostable that goes high when the threshold is crossed and returns low after the programmed time. This parameter defines the fixed width of the overthreshold pulse.

Unit of Measure: ns

8.1.3 Wave



parameter PreTrigger

Time coming before the position of the trigger in the waveform (i.e. size of the pre-trigger window).

Unit of Measure: ns

parameter RecordLength

The waveform size. Integer representing the time in ns. The actual size of the waveform will be automatically rounded to the closest allowed value. It is possible to get the exact size by reading back the parameter. The record length in time depends on wave resolution.

Unit of Measure: ns

parameter DownsampingFactor

Downsampling factor for the waveform resolution.

- **1**
  - x1
- **2**
  - x2
- **4**
  - x4
- **8**
  - x8

parameter Analog0/1

Waveform Analog Probe0/1

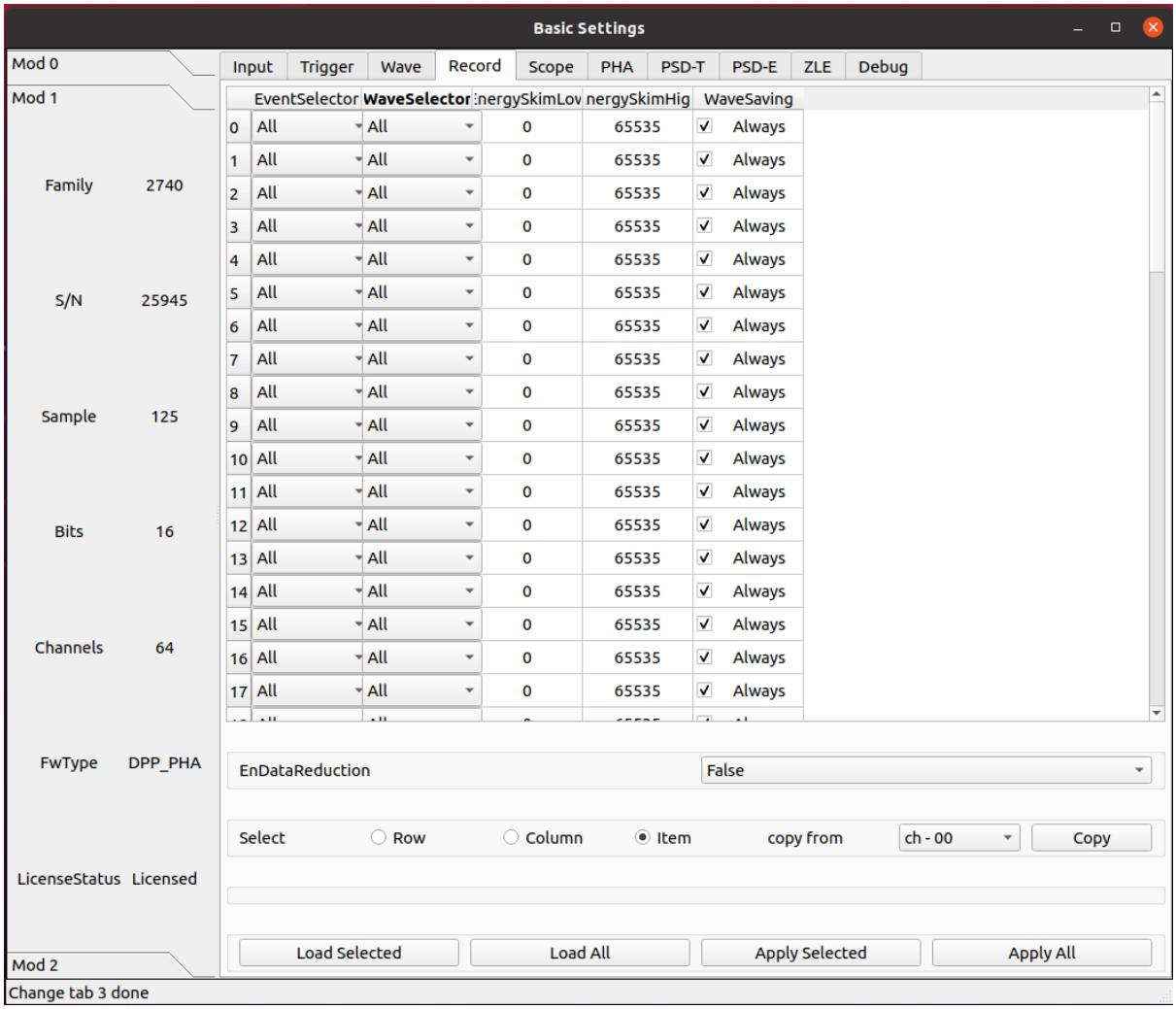
- **ADCInput**
  - ADC input probe
- **TimeFilter**
  - Time Filter probe
- **EnergyFilter**
  - Energy Filter probe
- **EnergyFilterBaseline**
  - Energy Filter Baseline
- **EnergyFilterMinusBaseline**
  - [Energy Filter –Baseline] probe

parameter Digital0/1/2/3

Waveform Digital Probe

- **Trigger**
  - Trigger probe
- **TimeFilterArmed**
  - Time Filter Armed probe
- **ReTriggerGuard**
  - ReTrigger Guard probe
- **EnergyFilterBaselineFreeze**
  - Energy Filter Baseline Freeze probe
- **EnergyFilterPeaking**
  - Energy Filter Peaking probe
- **EnergyFilterPeakReady**
  - Energy Filter Peak Ready probe
- **EnergyFilterPileupGuard**
  - Energy Filter Pile Up Guard probe
- **EventPileUp**
  - Event Pile Up probe
- **ADCSaturation**
  - ADC Saturation probe
- **ADCSaturationProtection**
  - ADC Saturation Protection probe
- **PostSaturationEvent**
  - Post Saturation Event probe
- **EnergyFilterSaturation**
  - Energy Filter Saturation probe
- **AcquisitionInhibit**
  - Acquisition Inhibit probe

8.1.4 Record



parameter EventSelector

Allows to set which events have to be saved.

- **All**
  - All events are saved
- **PileUp**
  - Only pileup events are saved
- **EnergySkim**
  - Save only the events in the Energy Skim range

parameter WaveSelector

Allows to set which waveform have to be saved.

- **All**
  - All waves are saved
- **PileUp**
  - Only pileup waves are saved
- **EnergySkim**
  - Save only waves in the Energy Skim range

parameter EnergySkimLowDiscriminator

Allows to flag events with energy higher than the low skim threshold. 16-bit value.

Unit of Measure: bin

parameter EnergySkimHighDiscriminator

Allows to flag events with energy lower than the high skim threshold. 16-bit value.

Unit of Measure: bin

parameter WaveSaving

Allows to save waveforms always or on request only.

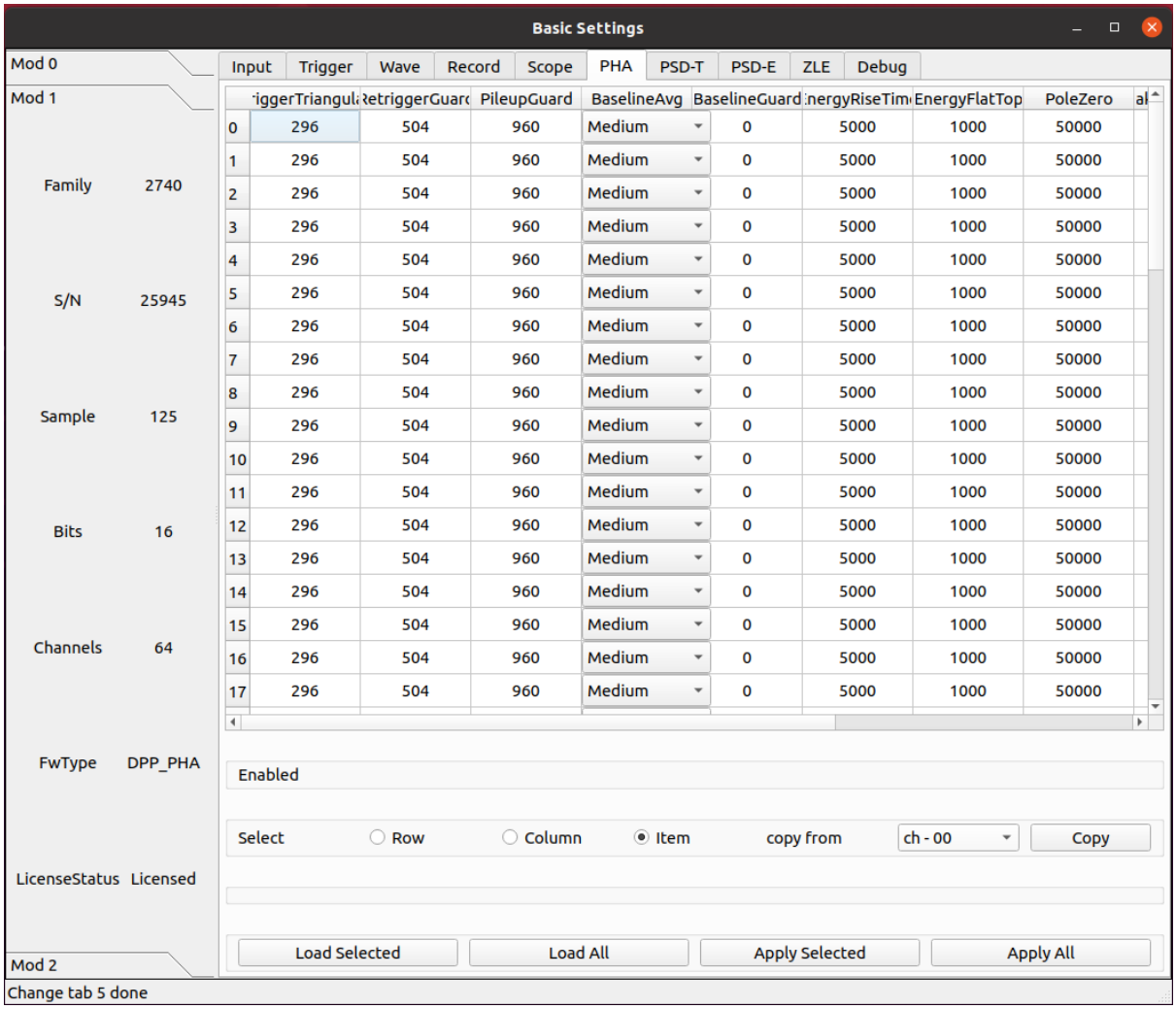
- **Always**
  - Waveforms are always saved
- **OnRequest**
  - Waveforms are saved on request

parameter EnDataReduction

If enabled, events consisting of 2 words are compressed in a single word event.

- **True**
  - Option enabled
- **False**
  - Option disabled

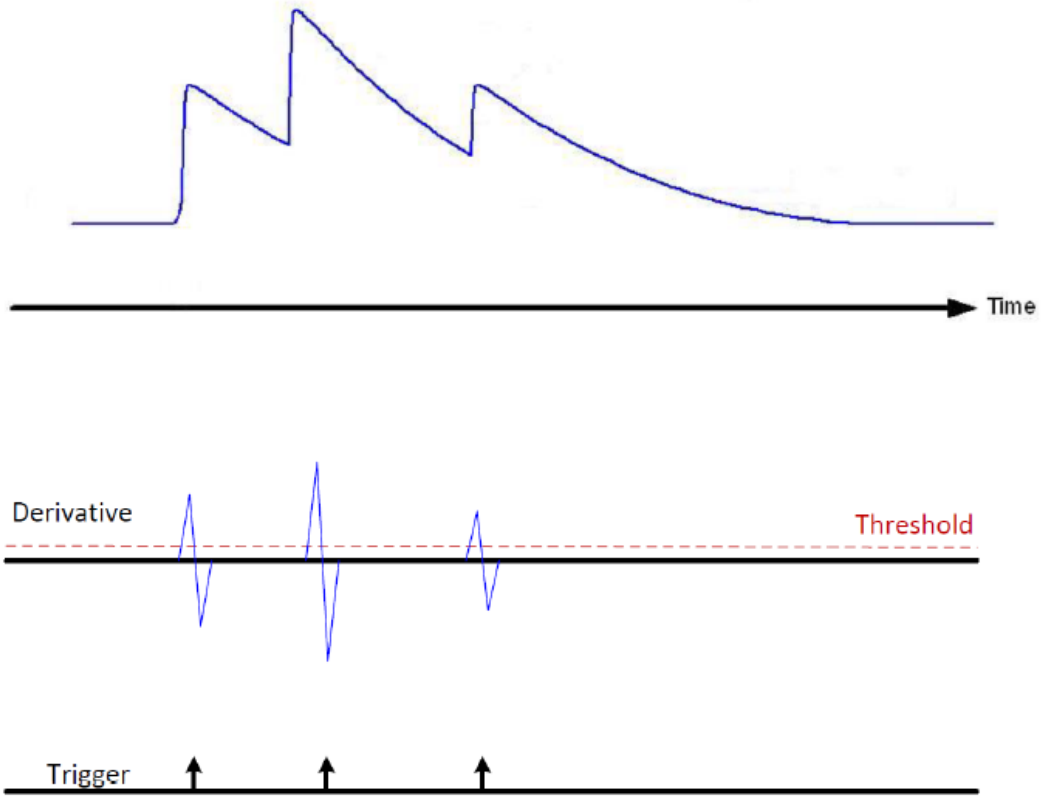
8.1.5 PHA



parameter TriggerTriangular

The x27xx digitizer running the DPP firmware discriminates events based on a triangular signal, whose rise time can be defined by the user. The TriggerThreshold is then referred to the derivative of the triangle itself, and the threshold crossing arms the event selection. The trigger fires at the zero crossing of the derivative signal itself. This parameter allows to set the rise time of the time filter.

Unit of Measure: ns



parameter RetriggerGuard

In case of fast signal such as those coming from PMTs possible overshoots in the fast discriminator signal may occur causing a retrigger and so possible fake pile-up. This parameter allows to set a retrigger inhibit guard (in ns). 10-bit value.

Unit of Measure: ns

parameter PileupGuard

If two events are separated by less than the trapezoid duration, then the relevant trapezoids overlap. The trapezoid duration is defined as RT + FT + PileUpGuard, where RT is the trapezoid Rise Time, FT is the trapezoid Flat Top, the PileUpGuard starts at the end of the Peaking time (see EnergyFilterPeakingPosition). This parameter allows to set the trapezoid filter pileup guard (in ns). 10-bit value.

Unit of Measure: ns

parameter BaselineAvg

Allows to enable a low-frequency filter for the energy filter

- **Fixed**
  - Baseline fixed at 0
- **VeryLow**
  - Baseline samples for average = 16
- **Low**
  - Baseline samples for average = 64

- **MediumLow**
  - Baseline samples for average = 256
- **Medium**
  - Baseline samples for average = 1024
- **MediumHigh**
  - Baseline samples for average = 4096
- **High**
  - Baseline samples for average = 16384

parameter BaselineGuard

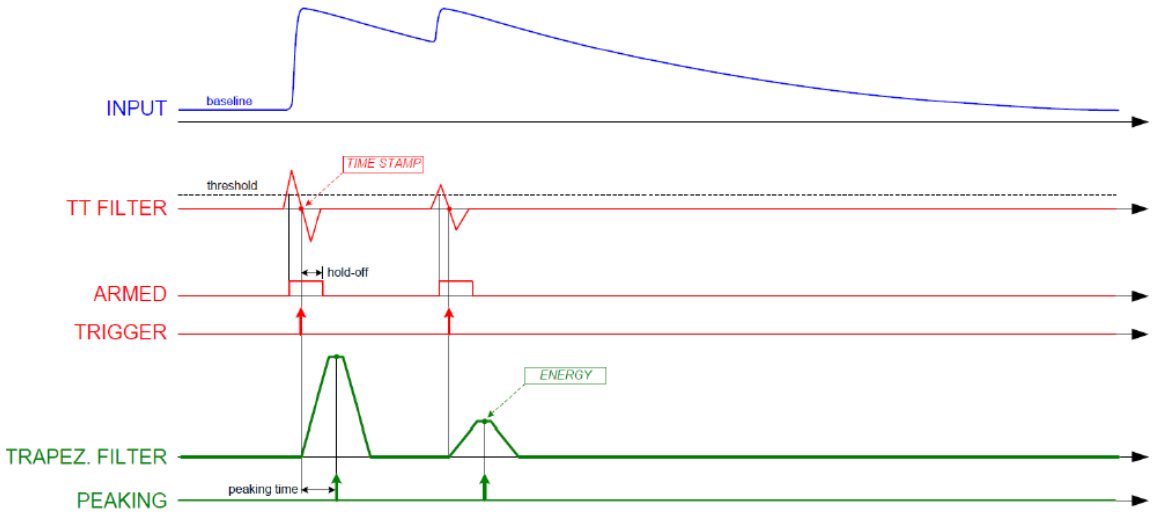
In addition to the Baseline Average, the user can also set the Baseline Hold-Off or Baseline Guard value to freeze the baseline calculation beyond the trapezoid end, thus reducing the noise on the baseline calculation. This parameter allows to set the trapezoid filter baseline protection after peak (in ns). 10-bit value.

Unit of Measure: ns

parameter EnergyRiseTime

The digitizer x27xx running the DPP-PHA firmware evaluates the energy value using a Trapezoidal filter. As in the traditional analog chain, the Shaping Amplifier is able to convert the exponential shape from the Charge Sensitive Preamplifier into a Gaussian shape whose height is proportional to the pulse energy, in the same way the Trapezoidal filter is able to transform it into a trapezoidal signal whose amplitude is proportional to the input pulse height (energy). In this analogy, the Energy Filter Rise Time corresponds to the Shaping Time times a factor of 2/2.5.

Unit of Measure: ns



parameter EnergyFlatTop

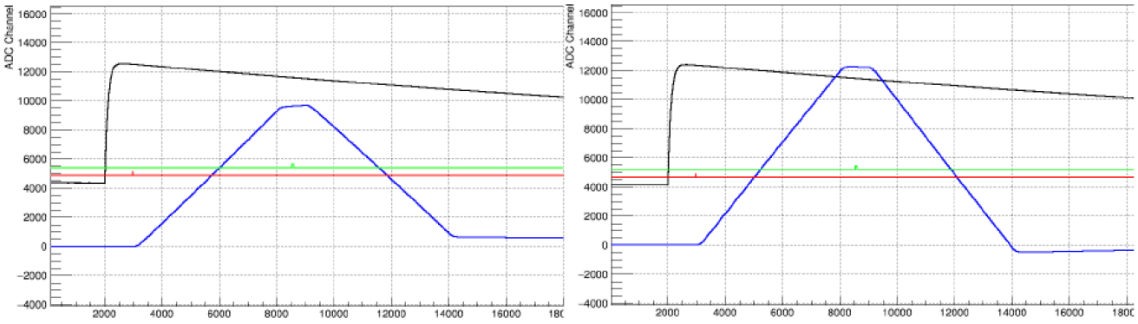
The energy value of the input pulse is evaluated as the height of the trapezoid in its Flat Top region. The user must take care that the flat top is really flat and that the Peaking (i.e. the samples used for the energy calculation) is in the flat region. Moreover the correct setting of flat top and peaking helps in the correct evaluation of the energy especially when large volume detectors are involved and the ballistic deficit may cause a significant error in the energy calculation. In this case it may be convenient to increase the flat top duration and delay the peaking time to wait for the full charge collection. This parameter allows to set the trapezoid flat top (in ns or samples). 9-bit value.

Unit of Measure: ns

parameter PoleZero

Like the Gaussian pulse of the Shaping Amplifier, also the trapezoid requires an accurate pole-zero adjustment to guarantee the correct return to the baseline at the end of the falling edge. To correctly set the pole-zero the user must take care of setting the proper Trapezoid Decay Time value (which corresponds also to the Input Decay Time) to avoid either undershoot or overshoot effects. Pole Zero Adjustment can reduce signal artifacts due to pulses pile up occurring when the counting rate is high compared to the pulse decay.

Unit of Measure: ns



parameter PeakingPosition

The trapezoid Peaking Position in percentage (%) of the flat top.

Unit of Measure: %

parameter PeakingAvg

The number of samples used to evaluate the peak.

- **OneShot**
  - 1 sample
- **LowAVG**
  - 4 samples
- **MediumAVG**
  - 16 samples
- **HighAVG**
  - 64 samples

parameter FineGain

Allows to set the energy fine gain. The energy fine gain is a digital multiplication factor and does not change the Full Scale Range. 16-bit value.

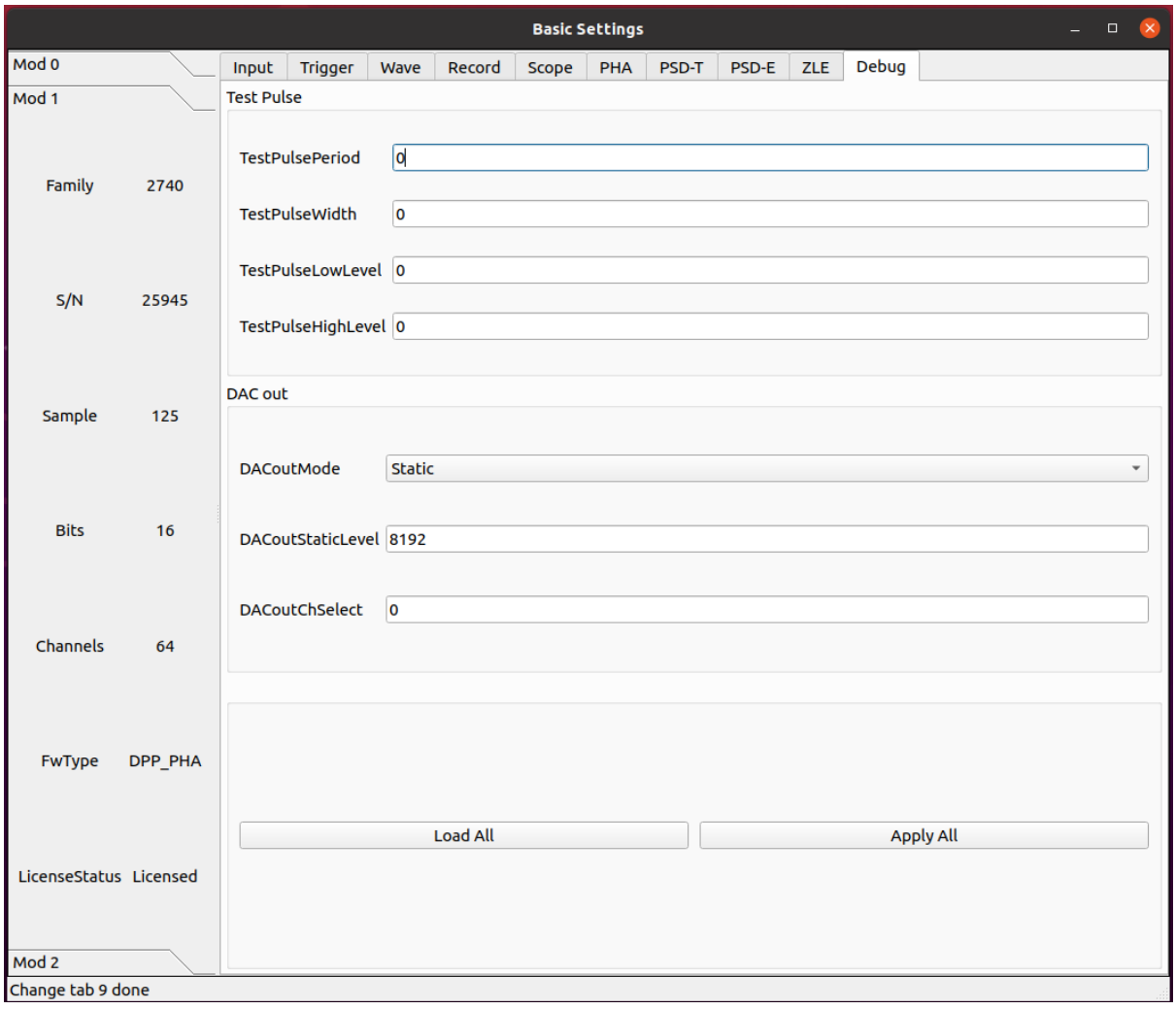
Increment: 0.001

parameter LFLimitation

Allows to enable a low-frequency filter for the energy filter

- **On**
  - Enabled
- **Off**
  - Disabled

8.1.6 Debug



parameter TestPulsePeriod

The Test Pulse is a programmable square wave that can be used as an internal periodic trigger (mainly for test purposes) or to generate a logic test pulse (TTL or NIM) on the TRGOUT and GPIO outputs. This parameter sets the period of the Test Pulse.

Unit of Measure: ns

parameter TestPulseWidth

Width of the Test Pulse (time that the signal stays high = 1).

Unit of Measure: ns

parameter TestPulseLowLevel

Low level of the Test Pulse expressed in ADC counts

Unit of Measure: ADC counts

parameter TestPulseHighLevel

High level of the Test Pulse expressed in ADC counts

Unit of Measure: ADC counts

parameter DACoutMode

Selects the signal type to be sent in output on the front panel DAC connector.

- **Static**
  - DAC output stays at a fixed level, given by the DACoutStaticLevel parameter
- **Ramp**
  - The DAC output is driven by a 14-bit counter
- **Sin5MHz**
  - The DAC output is a sine wave at 5 MHz with fixed amplitude
- **Square**
  - Square wave with period and with set by TestPulsePeriod and TestPulseWidth and amplitude between TestPulseLoweLevel and TestPulseHighLevel.
- **IPE**
  - Not implemented
- **ChInput**
  - The DAC reproduces the input signal received by one input channel, selected by the DACoutChSelect parameter
- **MemOccupancy**
  - Level of the memory occupancy (not yet implemented)
- **ChSum**
  - The DAC reproduces the “analog” sum of all the digitizer inputs (not yet implemented)
- **OverThrSum**

  - The DAC output is proportional to the number of channels that are currently above the threshold

parameter DACoutStaticLevel

When DACoutMode = Static, this parameter sets the 14-bit level of the DAC static output.

parameter DACoutChSelect

When DACoutMode = ChInput, the DAC reproduces the input signal of the channel selected by this parameter.

parameter IPEAmplitude

The new digitizers are equipped with an Internal Pulse Emulator capable of generating exponential pulses. This parameter determines the amplitude of the pulse.

Unit of Measure: ADC counts

parameter IPEBaseline

Sets the offset of the exponantial pulses generated by the Internal Pulse Emulator.

Unit of Measure: ADC counts

parameter IPEDecayTime

Sets the decay time of the exponantial pulses generated by the Internal Pulse Emulator.

Unit of Measure: ns

parameter IPERate

Sets the rate of the exponantial pulses generated by the Internal Pulse Emulator.

Unit of Measure: Hz

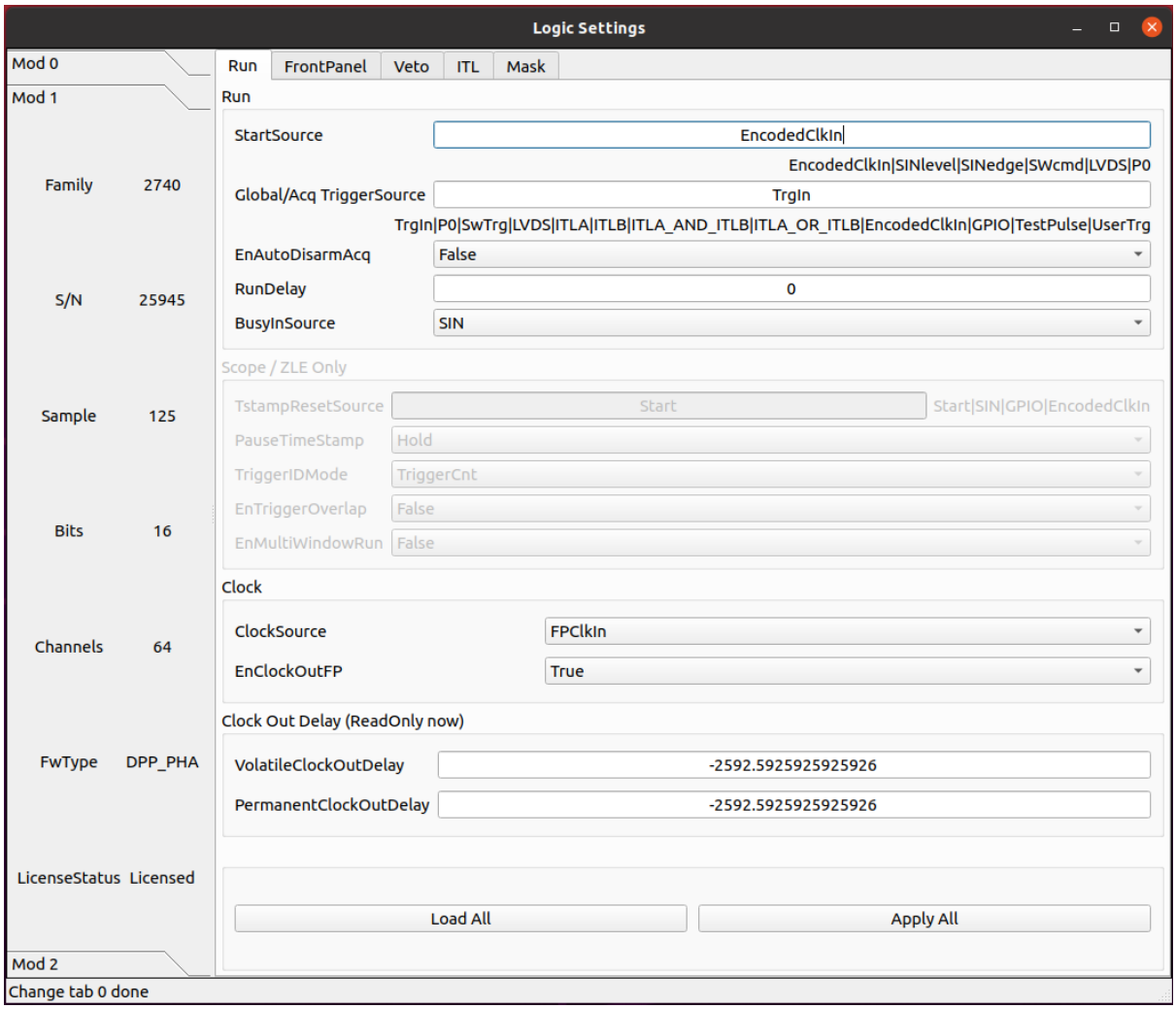
parameter IPETimeMode

Selectes the time distribution of the Internal Pulse Emulator.

- **ConstantRate**
  - Pulse shapes are constant over time. It is possible to set the frequency using the IPERate parameter
- **Poissonian**
  - The pulse rate follows a Poisson distribution. The average frequency value can be configured using the IPERate parameter

8.2 Logic

8.2.1 Run



parameter StartSource

Defines the source for the start of run. Multiple options are allowed, separated by "|" .

- **EncodedClkIn**
  - Start from CLK-IN/SYNC connector on the front panel. This is a 4-pin connector (LVDS signals) used to propagate the reference clock (typ. 62.5 MHz) and a Sync signal. The rising edge of the Sync starts the acquisition, that lasts until the Sync returns low (falling edge).
- **SINlevel**
  - Start from SIN (1=run, 0=stop)
- **SINedge**
  - Start from SIN (rising edge = run; stop from SW)
- **SWcmd**
  - Start from SW
- **LVDS**

  - Start from LVDS

- **P0**
  - Start from P0 (backplane)

parameter GlobalTriggerSource

Defines the source for the Acquisition Trigger, which is the signal that opens the acquisition window and saves the waveforms in the memory buffers. Multiple options are allowed, separated by "|".

- **TrgIn**
  - Front Panel TRGIN
- **P0**
  - Trigger from P0 (backplane)
- **SwTrg**
  - Software trigger
- **LVDS**
  - LVDS trgin
- **ITLA**
  - Internal Trigger Logic A: combination of channel self-triggers
- **ITLB**
  - Internal Trigger Logic B: combination of channel self-triggers
- **ITLA_AND_ITLB**
  - Second level Trigger logic making the AND of ITL A and B
- **ITLA_OR_ITLB**
  - Second level Trigger logic making the OR of ITL A and B
- **EncodedClkIn**
  - Not implemented (encoded CLK-IN trigger)
- **GPIO**
  - Front Panel GPIO
- **TestPulse**
  - Internal Test Pulse
- **UserTrg**
  - User custom trigger source

parameter EnAutoDisarmAcq

When enabled, the Auto Disarm option disarms the acquisition at the stop of run. When the start of run is controlled by an external signal, this option prevents the digitizer to restart without the intervention of the software.

- **True**
  - The acquisition is automatically disarmed after the stop. It is therefore necessary to rearm the digitizer (with the relevant command sent by the software) before starting a new run.
- **False**
  - The acquisition is not disarmed after the stop. Multiple transition of the start signal will produce multiple runs.

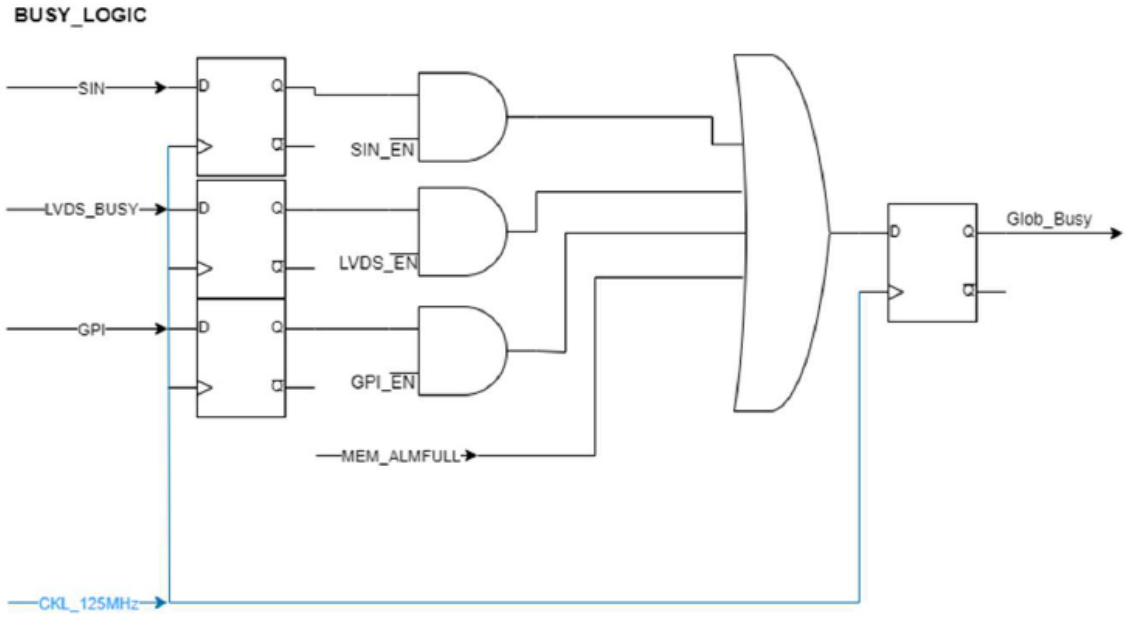
parameter RunDelay

When the start of run is controlled by a RUN signal that is propagated in daisy chain between the boards (for instance through the ClkIn- ClkOut or SIN-GPIO sync chain), it is necessary to compensate for the propagation delay and let the boards start exactly at the same time. The RunDelay parameter allows the start of the acquisition to be delayed by a given number of clock cycles with respect to the rising edge of the RUN signal. Assuming that the propagation delay is 2 cycles, the RunDelay setting will be 0 for the last board in the chain, 2 for the previous one, and so on up 2x(NB-1) for the first one.

Unit of Measure: ns

parameter BusyInSource

In a multi-board system, it might be necessary to prevent one board to accept a new trigger while another board is full and thus unable to accept the same trigger. For this reason, each board can generate a Busy signal to notify that it is unable to get a new trigger. If the busy/veto mechanism has some latency, it is advisable to generate the busy slightly before the digitizer become full. For this purpose, it is possible to assert the busy output when the acquisition memory reaches a certain level of occupancy (internally managed). The OR of the busy signals is typically used to stop the global trigger. It is possible to get the individual busy signals from each board and make an external OR logic or connect the boards with cables to propagate the Busy along the chain. Each board makes an OR between its internal busy and the busy input signal coming from the previous board, thus having a global Busy at the end of the line. This parameter defines the source of the Busy Input (schematized in the figure below)



- **Disabled**
  - The Busy is given by the Internal Busy only (Memory full or almost full)
- **SIN**
  - Busy input from SIN on front panel
- **GPIO**
  - Busy input coming from GPIO on front panel, used as a simple input. It is also possible to use GPIO as a wired OR (bidirectional). In this mode, the Busy line goes high as soon as one board drives it high. All the boards can read the Busy line and use it as a veto for the trigger
- **LVDS**
  - LVDS trgin

parameter ClockSource

This is the source of the system clock. Multiple options are not allowed

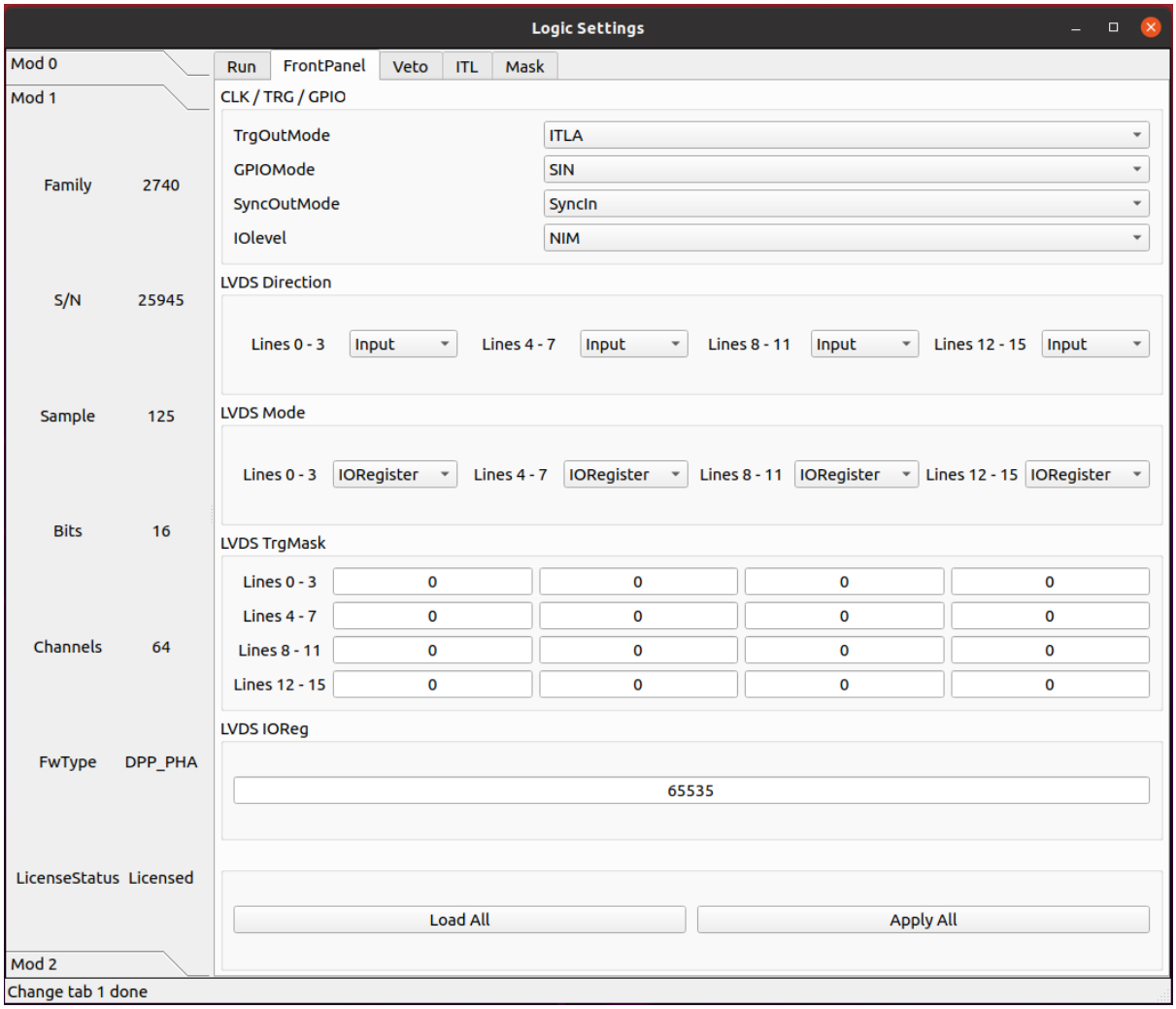
- **Internal**
  - Local oscillator, 62.5 MHz
- **FPClkIn**
  - Front Panel Clock input

parameter EnClockOutFP

Enables clock output on Front Panel for the daisy chain propagation of the clock between multiple boards.

- **True**
  - Enabled
- **False**
  - Disabled

8.2.2 FrontPanel



parameter TrgOutMode

Selects the signal that is routed to the TRGOUT output. Multiple options are not allowed.

- **Disabled**
  - TRGOUT output disabled
- **TrgIn**
  - Propagation of Front Panel TRGIN (TRGOUT is a replica, with some delay, of the TRGIN signal)
- **P0**
  - Propagation of P0 trigger
- **SwTrg**
  - Software trigger
- **LVDS**
  - LVDS trgin
- **ITLA**
  - Internal Trigger Logic A: combination of channel self-triggers

- **ITLB**
  - Internal Trigger Logic B: combination of channel self-triggers
- **ITLA_AND_ITLB**
  - Second level Trigger logic making the AND of ITL A and B
- **ITLA_OR_ITLB**
  - Second level Trigger logic making the OR of ITL A and B
- **EncodedClkIn**
  - Not implemented (propagation of the Encoded CLK-IN trigger)
- **Run**
  - Propagation of the RUN signal (acquisition start/stop), before applying the delay given by the Run-Delay parameter
- **RefClk**
  - Monitor of the 62.5 MHz clock (used for phase alignment)
- **TestPulse**
  - Internal Test Pulse
- **Busy**
  - Busy of the board
- **UserTrgout**
  - Trgout coming from the User Logic (open FPGA)
- **Fixed0**
  - 0 level signal
- **Fixed1**
  - 1 level signal
- **SyncIn**
  - SyncIn signal
- **SIN**
  - SIN connector signal
- **GPIO**
  - GPIO connector signal
- **LBinClk**
  - Internal Logic B clock signal
- **AcceptTrg**
  - Accepted triggers signal
- **TrgClk**
  - Tigger clock signal

parameter GPIOMode

Selects the signal that is routed to the GPIO, when this is used as output. Multiple options are not allowed. The GPIO on the front panel is a bidirectional signal that can used in three different ways:

As independent board output (each board drives its own GPIO)

As a shared input for the boards: the signal is driven high (= 1) or low (= 0) by an external source and connected in "short circuit" among multiple boards using "T" connectors at the inputs. The GPIO is not internally terminated, thus it is necessary to put a 50 Ohm terminator at the end of the line (last "T" of the chain)

As a shared bidirectional line, making a "wired OR". One or more boards can simultaneously drive the signal high (= 1). If no board drives the GPIO, it remains low (= 0). All boards can read back the signal. It is necessary to put a 50 Ohm terminator at both ends of the line (first and last "T" of the chain). This mode can be used to generate, for instance, the global Busy and Veto logic for multiple boards.

- **Disabled**
  - GPIO disabled
- **TrgIn**
  - Propagation of Front Panel TRGIN (GPIO is a replica, with some delay, of the TRGIN signal)
- **P0**
  - Propagation of P0 trigger
- **SIN**
  - Propagation of SIN
- **LVDS**
  - LVDS trgin
- **ITLA**
  - Internal Trigger Logic A: combination of channel self-triggers
- **ITLB**
  - Internal Trigger Logic B: combination of channel self-triggers
- **ITLA_AND_ITLB**
  - Second level Trigger logic making the AND of ITL A and B
- **ITLA_OR_ITLB**
  - Second level Trigger logic making the OR of ITL A and B
- **EncodedClkIn**
  - Not implemented (propagation of the Encoded CLK-IN trigger)
- **SwTrg**
  - Software trigger
- **Run**
  - Propagation of RUN
- **RefClk**
  - Monitor of the 62.5 MHz clock (used for phase alignment)
- **TestPulse**
  - Internal Test Pulse

- **Busy**
  - Busy of the board
- **UserGPO**
  - GPO coming from the User Logic (open FPGA)
- **Fixed0**
  - 0 level signal
- **Fixed1**
  - 1 level signal

parameter SyncOutMode

In a multi-board system, it can be useful to propagate a synchronous signal together with the clock (to synchronize the start of the run, for example) on CLK OUT front panel connector. This parameter defines which signal must be sent out. Multiple options are not allowed.

- **Disabled**
  - SyncoutMode is disabled
- **SyncIn**
  - SyncIn signal (if provided with clkIn on CLK IN connector)
- **TestPulse**
  - Internal Test Pulse
- **IntClk**
  - Internal 62.5 MHz clock (for test purposes)
- **Run**
  - Propagation of RUN signal
- **User**
  - User customSyncoutMode

parameter IOlevel

Sets the electrical logic level of the LEMO I/Os (TRGIN, SIN, TRGOUT, GPIO).

Note that TRGIN and SIN are internally terminated to 50 Ohm, while GPIO and TRGOUT require the termination to 50 Ohms at the receiver

- **NIM**
  - NIM logic (0 = 0V, 1 = -0.8V, that is -16mA)
- **TTL**
  - Low Voltage TLL logic (0 = 0V, 1 = 3.3V)

parameter LVDSDirection

Assigns the direction to a quartet of LVDS I/Os.

- **Input**
  - The LVDS lines of the relevant quartet are used as input. The relevant LED on the front panel is OFF.
- **Output**
  - The LVDS lines of the relevant quartet are used as output. The relevant LED on the front panel lights-up.

parameter LVDSMode

The digitizer is equipped with 16 LVDS I/Os that can be programmed to be inputs or outputs by groups of 4 (quartets), depending on the LVDSDirection parameter. Once the direction has been selected, it is possible to select the functionality of the LVDS lines, individually for each quartet.

- **SelfTriggers**
  - This option is available only when the LVDS are set as outputs. Each LVDS line can be assigned to a combination of the 64 self-triggers, implemented as a masked OR, where the mask is set by the LVDSTrgMask parameter(16 independent masks, one per LVDS line)
- **Sync**
  - Whatever is the direction of the quartet, the 4 lines are rigidly assigned to specific acquisition signals: 0 = Run 1 = Trigger 2 = Busy 3= Veto It is possible to implement a daisy chain distribution of these signals using one quartet as input and another one as output
- **IORegister**
  - The LVDS lines of the quartet are statically controlled by the LVDSIOReg parameter. Use the SetValue function to set the relevant LVDS lines when programmed as output. Use GetValue to read the status of the LVDS lines when programmed as inputs.
- **User**
  - User custom.

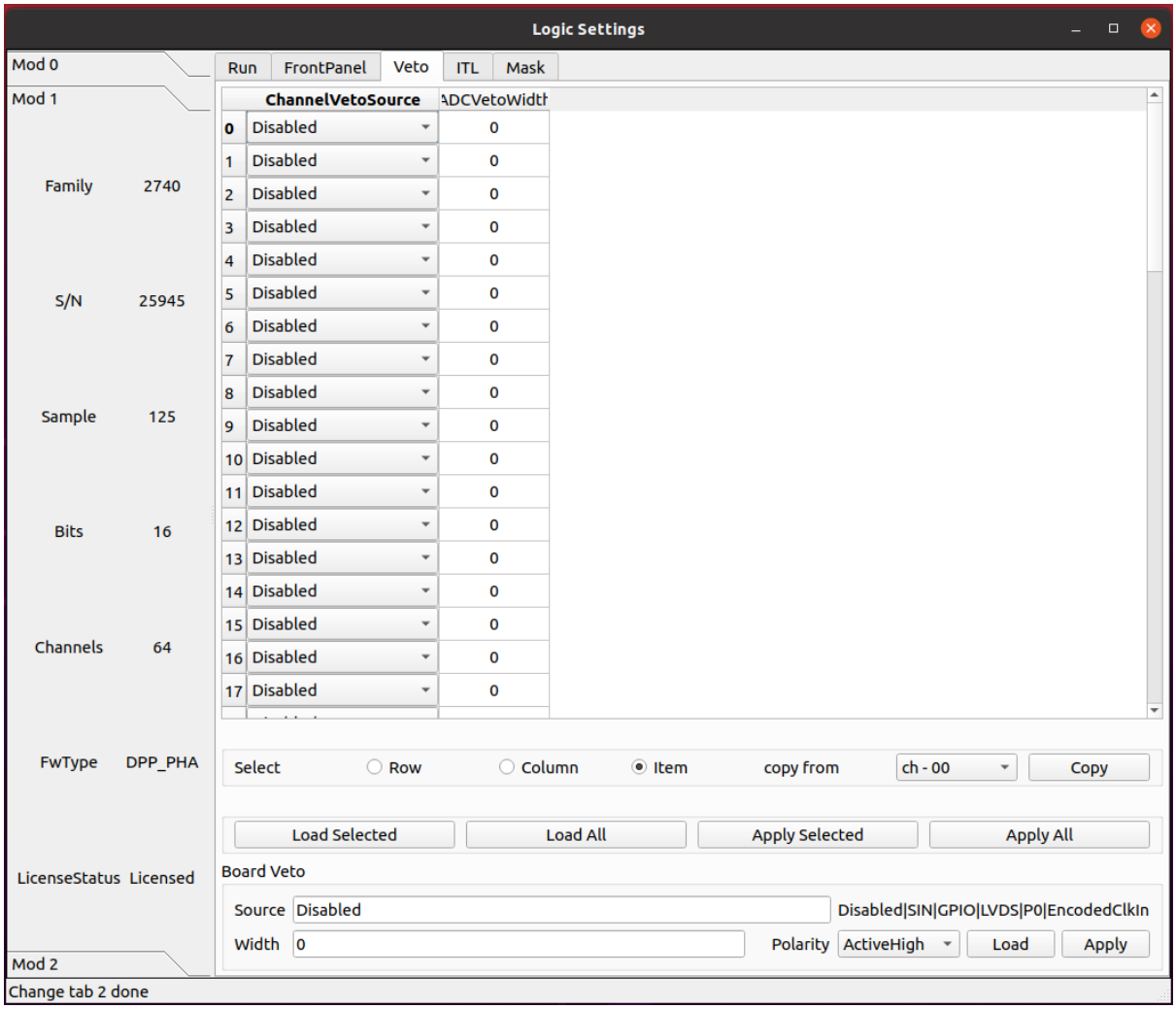
parameter LVDSTrgMask

Each LVDS line can be assigned to a combination of the 64 self-triggers, implemented as a masked OR, where the mask is set by this parameter. There are 16 independent masks, one per LVDS line. Note that the trigger mask assignment does not imply the LVDS direction and mode settings. It is therefore necessary to set the Direction = Output and Mode = SelfTriggers to use the Self-Trigger propagation to the LVDS I/Os.

parameter LVDSIOReg

Set the status of the LVDS I/O for the quartets when they are programmed to be output and Mode = IORegister. This parameter reads out the status of the quartets in the case the LVDS I/O are programmed as inputs (possibly externally driven).

8.2.3 Veto



parameter ChannelVetoSource

Allows to set the veto for each channel; it can be external (which means one of the veto options in the previous table), or it can be on a channel base.

- **Disabled**
  - Any channel veto source is disabled
- **BoardVeto**
  - Enables board veto
- ADCOverSaturation: Enables veto due to ADC oversaturation
- ADCUnderSaturation: Enables veto due to ADC undersaturation

parameter ADCVetoWidth

It is the width of the ADC veto (undersaturation and oversaturation width) expressed in ns.

Unit of Measure: ns

parameter VetoSource

Defines the source for the Veto, which is the signal that inhibits the acquisition trigger. Multiple options are allowed, separated by "|". The VETO signal can be either active high or low, depending on the VetoPolarity parameter. When active low, it acts as a GATE for the trigger. It is possible to stretch the duration of the VETO by means of the parameter VetoWidth.

- **Disabled**
  - VETO is always OFF
- **SIN**
  - SIN on the front panel
- **GPIO**
  - GPIO on the front panel (used as input)
- **LVDS**
  - LVDS trgin
- **P0**
  - P0 (signal from the backplane)
- **EncodedClkIn**
  - Not implemented (encoded CLK-IN veto)

parameter VetoWidth

Whatever is the source of the VETO signal, it is possible to stretch the duration of the veto up to a given time by means of a re-triggerable monostable. When 0, the monostable is disabled and the veto lasts as long as the selected source is active.

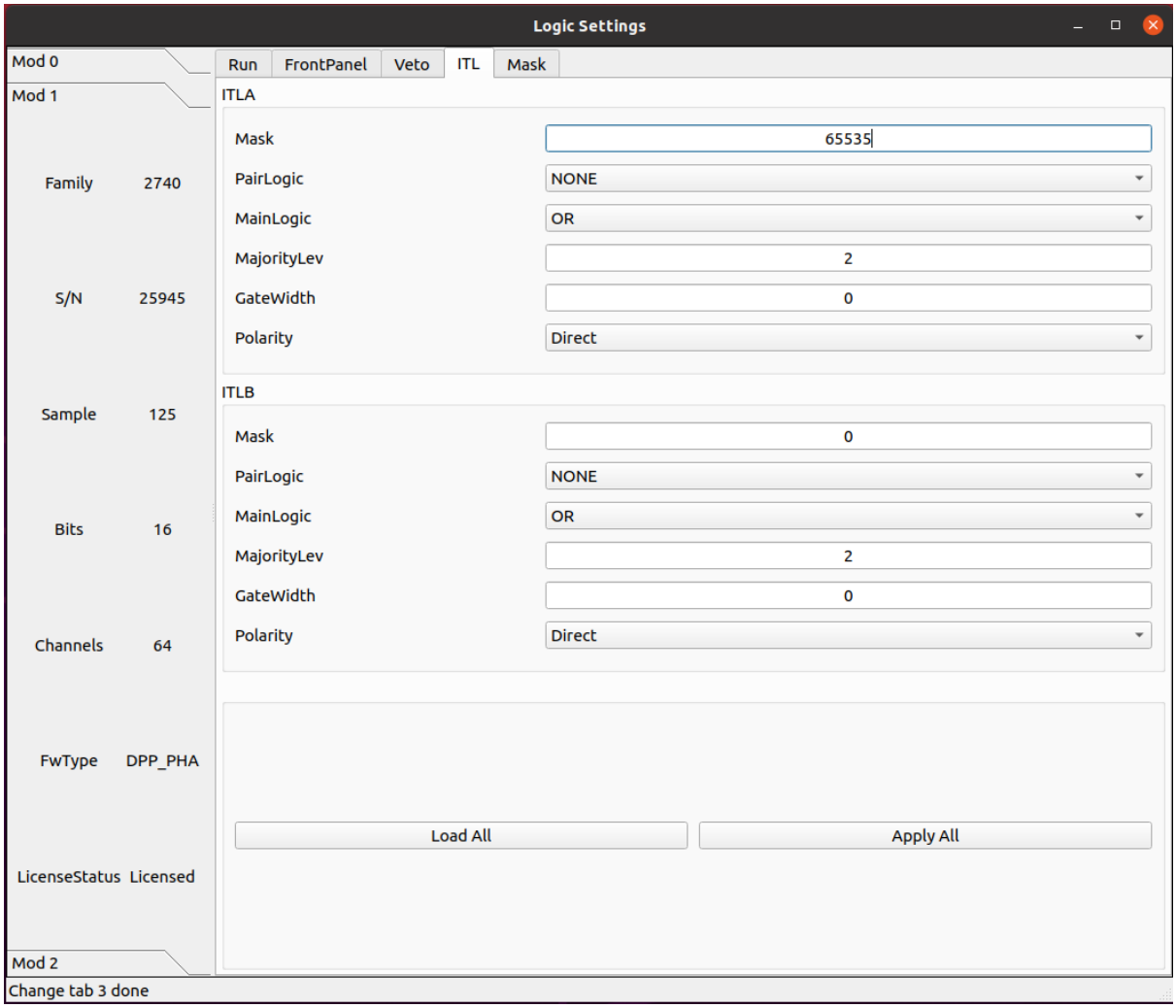
Unit of Measure: ns

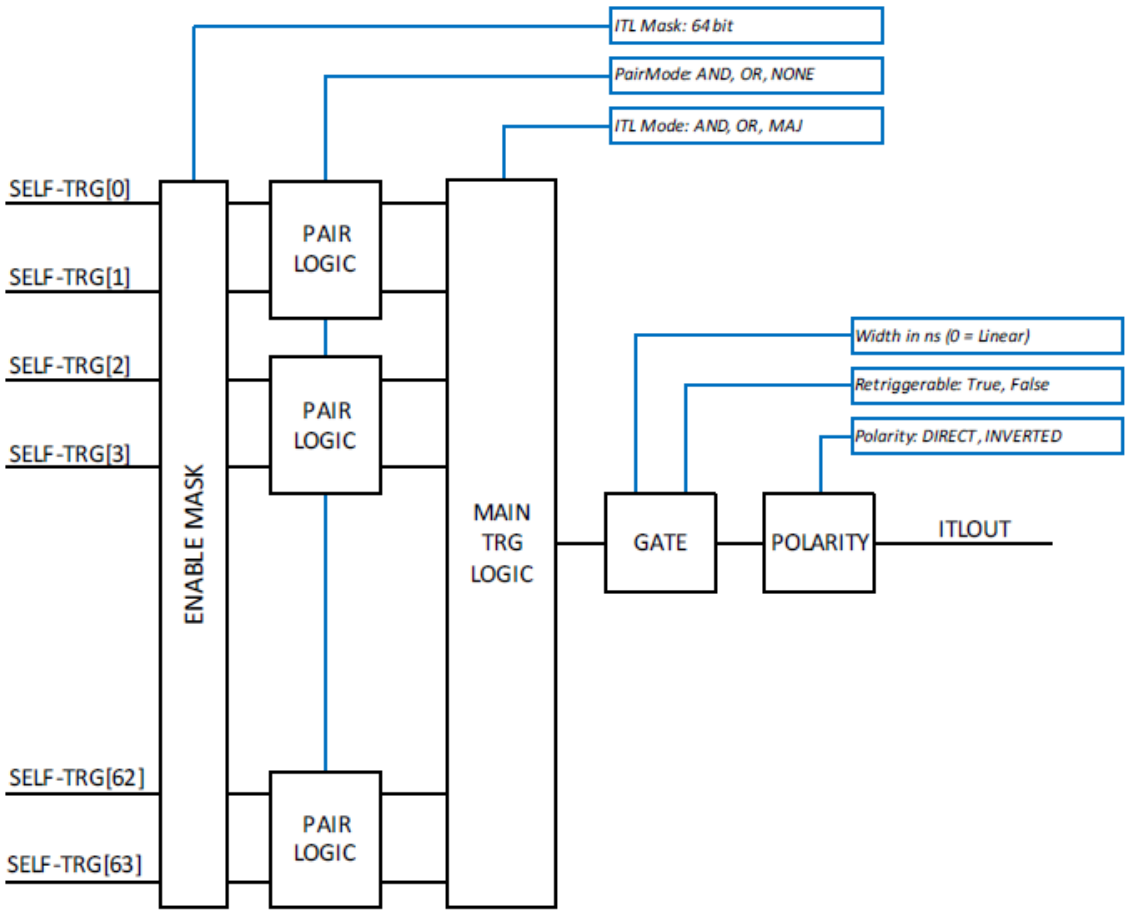
parameter VetoPolarity

Defines the polarity of the Veto

- **ActiveHigh**
  - Veto is active high. The signal acts as an "Inhibit" for the trigger
- **ActiveLow**
  - Veto is active low. The signal acts as a "Gate" the trigger

8.2.4 ITL


Logic Settings
Mod 0
Mod 1
Family 2740
S/N 25945
Sample 125
Bits 16
Channels 64
FwType DPP_PHA
LicenseStatus Licensed
Mod 2
Run
FrontPanel
Veto
ITL
Mask
ITLA
Mask 65535
PairLogic NONE
MainLogic OR
MajorityLev 2
GateWidth 0
Polarity Direct
ITLB
Mask 0
PairLogic NONE
MainLogic OR
MajorityLev 2
GateWidth 0
Polarity Direct
Load All
Apply All
Change tab 3 done



parameter ITLA/BMask

Enable Mask at the input of the ITLA/B.

parameter ITLA/BPairLogic

Pairs of channels can be combined with an OR or AND before feeding in the Main trigger Logic. This is typically used in the readout of tubes or scintillator bars, where the two ends are read in coincidence, for instance in position sensitive detectors (the coincidence window will be set by the SelfTriggerWidth parameter). When the AND/OR logic is applied, the two outputs of the Pair Logic blocks are identical.

Note that they are counted twice in the following Majority logic. If the Pair Logic is disabled ("NONE" option), the block is transparent, and the two outputs are just a replica of the inputs.

- **OR**
  - Both Pair Logic Outputs = OR of two consecutive self-triggers
- **AND**
  - Both Pair Logic Outputs = AND of two consecutive self-triggers
- **NONE**
  - Outputs = Inputs

parameter ITLA/BMainLogic

Each channel of the digitizer feature a digital bipolar triangular filter discriminator with programmable rise time and threshold able to self-trigger on the input pulses and generate a self-trigger signal. In DPP Mode, the channels acquire independently, so the channel self-trigger is used locally to acquire a waveform. The trigger threshold is then referred to the bipolar triangular filter, and the threshold crossing arms the event selection. The trigger fires at the zero crossing of the time filter signal. The user can see the derivative trace on the signal inspector. It is also possible to combine all the self-triggers of the board, according to a specific trigger logic. There are two independent logic blocks, ITLA and ITLB. Their output can be used separately to feed, for instance, AcqTrigger and TrgOut, or combined in a second level trigger logic to implement more complex trigger schemes. Therefore, the ITLs can either generate the local acquisition trigger, common to all the channels, for the acquisition of the waveform, or propagate the signal outside, through the TRGOUT, thus making it possible to combine triggers of multiple boards in an external trigger logic, that eventually feeds back the TRGIN of the digitizers. Each ITL is made of an input enable mask (64 bits, one per channel), an optional pairing logic that combines the self triggers of two consecutive channels (e.g. paired coincidence) and the main trigger logic that combines the 64 selftriggers with an OR, AND or Majority logic. The output can be linear (no stretching) or reshaped by a programmable gate generator, either re-triggerable or not and finally programmed for polarity (direct or inverted).

- **OR**
  - ITLOUT = masked OR of channel self-triggers
- **AND**
  - ITLOUT = masked AND of channel self-triggers
- **Majority**
  - ITLOUT = masked Majority of channel self-triggers

parameter ITLA/BMajorityLev

Defines the majority level of the Main Logic of the ITL A/B block. The majority output is calculated at every clock cycle, and it becomes TRUE when Nch >= MajLev, where Nch is the number of self-triggers active in that clock cycle and MajLev is the programmed majority level.

Note that when the Pair Logic is used to combine the self triggers two by two (AND/OR), each pair produces two identical signals that will be counted twice in the majority level.

parameter ITLA/BGateWidth

Width of the gate generator at the output of the ITLA/B block.

Unit of Measure: ns

parameter ITLA/BPolarity

Polarity of the gate generator output.

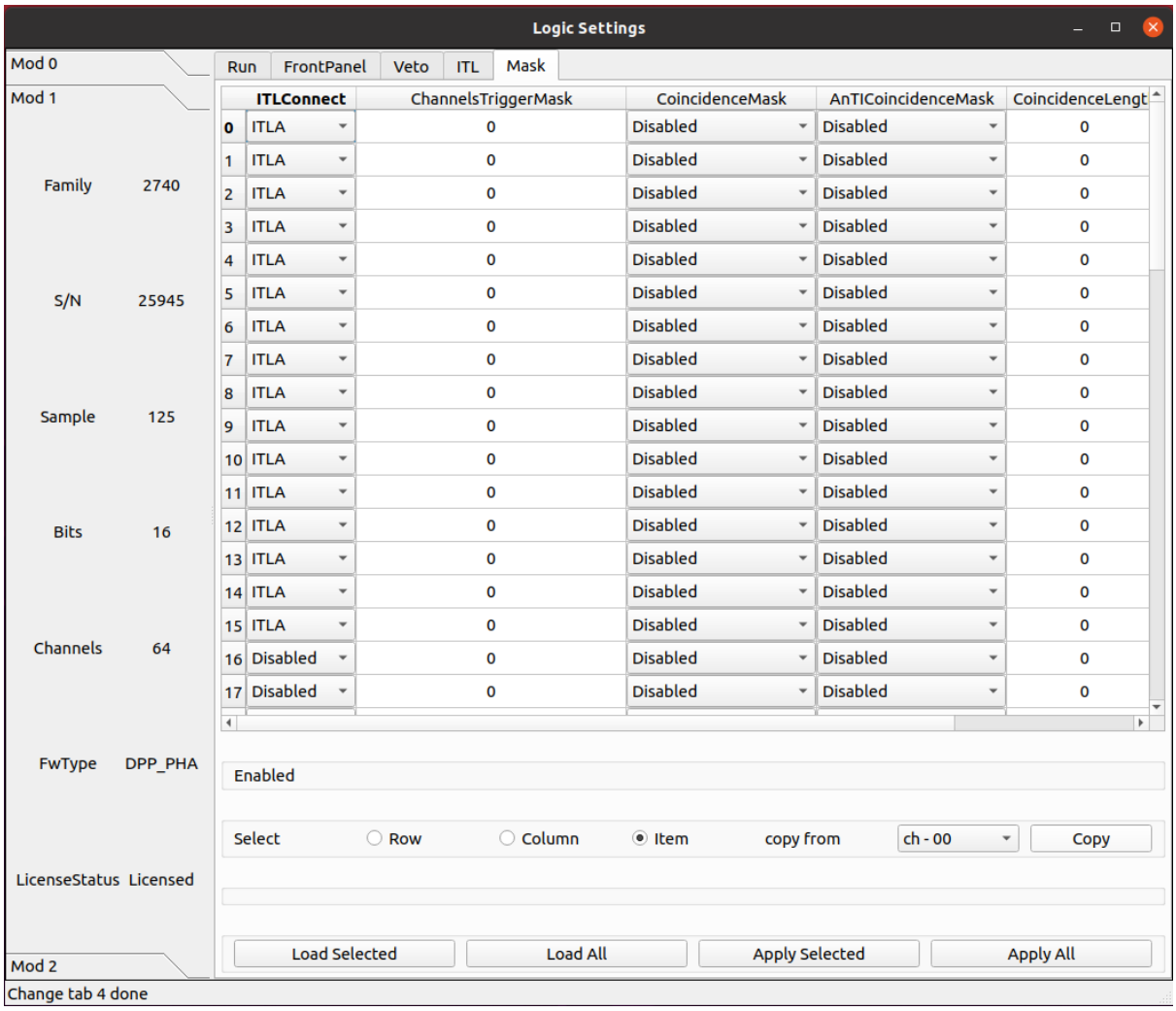
- **Direct**
  - Direct polarity
- **Inverted**
  - Inverted polarity

parameter ITLA/BEnRetrigger

Set the ITLA/B to be retriggerable.

- **True**
  - The ITLA/B is retriggerable
- **False**
  - The ITLA/B is not retriggerable

8.2.5 mask



parameter ITLConnect

Alternative to ITLAMask, ITLBMask. Determines if the channel partecipate in ITLA or ITLB

- **Disabled**
  - The channel is disabled
- **ITLA**
  - The channel participates in ITLA logic block
- **ITLB**

  - The channel participates in ITLB logic block

parameter ChannelsTriggerMask

Allows to set the mask over 64 bits to generate a channel trigger. It can be used to trigger a channel using a trigger coming from another channel. It also allows to set the mask over 64 bits to enable the channel to participate in the coincidence logic defined in CoincidenceMask and AntiCoincidenceMask (option Channel64Trg). 64-bit enable mask, each bit representing a channel.

parameter CoincidenceMask

Allows to set the coincidence mask that generates a trigger on the specified channel.

- **Disabled**
  - All the coincidence sources are disabled
- **Ch64Trigger**
  - One of the 64 channels can generate a coincidence signal
- **TRGIN**
  - TRGIN can generate a coincidence signal
- **GlobalTriggerSource**
  - Acquisition Trigger can generate a coincidence signal
- **ITLA**
  - ITLA can generate a coincidence signal
- **ITLB**
  - ITLB can generate a coincidence signal

parameter AntiCoincidenceMask

Allows to set the anticoincidence mask that generates a trigger on the specified channel.

- **Disabled**
  - All the coincidence sources are disabled
- **Ch64Trigger**
  - One of the 64 channels can generate a coincidence signal
- **TRGIN**
  - TRGIN can generate a coincidence signal
- **GlobalTriggerSource**
  - Acquisition Trigger can generate a coincidence signal
- **ITLA**
  - ITLA can generate a coincidence signal
- **ITLB**
  - ITLB can generate a coincidence signal

parameter CoincidenceLength

Coincidence window length in nanoseconds (ns). 16-bit value.

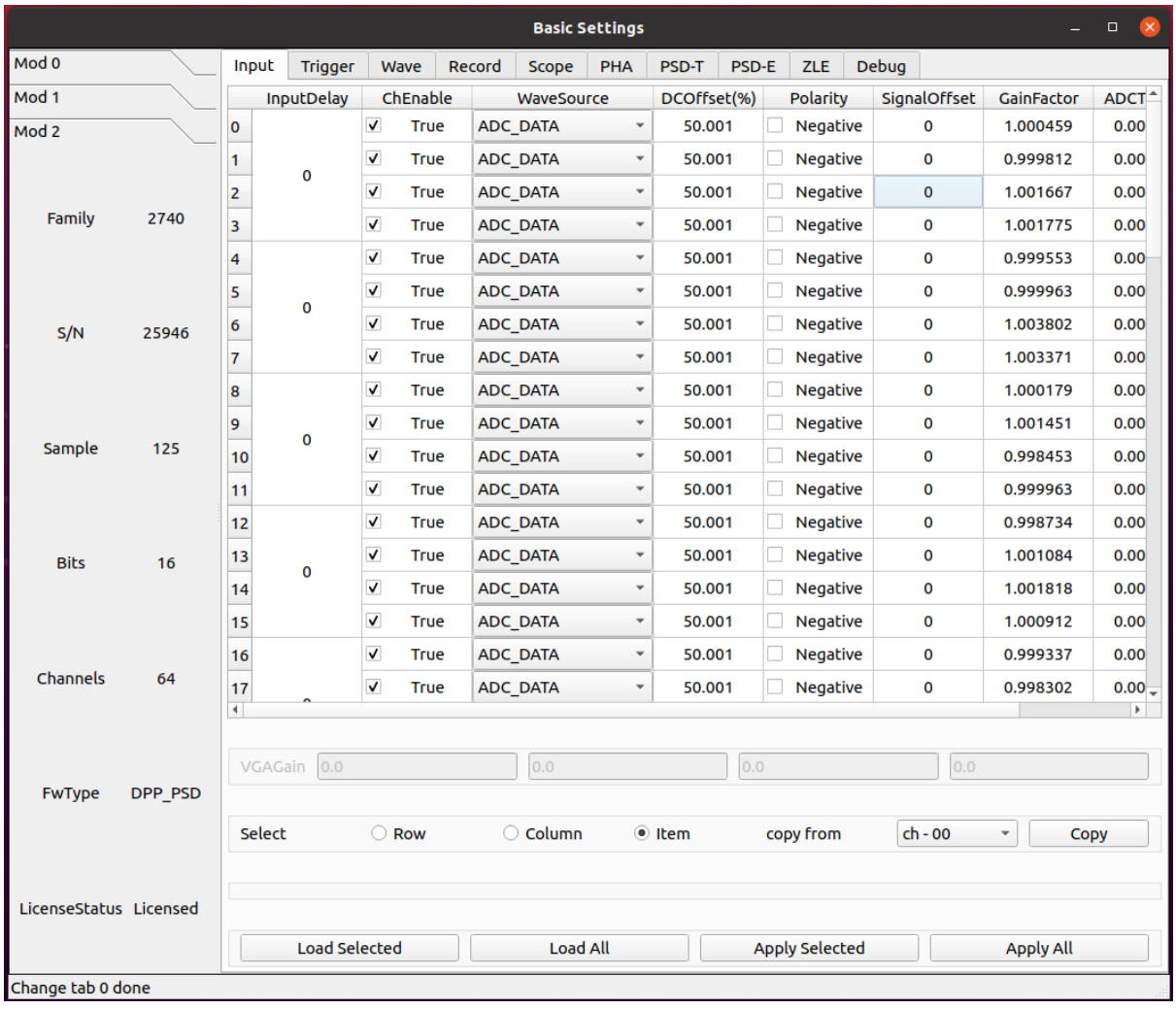
Unit of Measure: ns

CHAPTER 9

PSD firmware

9.1 Basic

9.1.1 Input



parameter ChGain

Unique to x2730.

Sets the gain of the Variable Gain Amplifiers (VGA). Unit of Measure: dB

parameter InputDelay

Set input delay. The value is set at groups of 4 channels for x2745/x2740.

Unit: sample

parameter ChEnable

Enable the channels for the acquisition, according to the Index. When the channel is disabled, it does not give any data and its self-trigger is off.

- True: The channel is enabled for the acquisition
- False: The channel is disabled for the acquisition

parameter WaveSource

In normal mode, the acquired waveform represents a sequence of ADC samples, resulting from the A/D conversion of the analog input. For test purposes, it is possible to replace the ADC data with internal data generators.

- **ADC_DATA**
  - Data from the ADC (normal operating mode)
- **ADC_TEST_TOGGLE**
  - Toggle between 0x5555 and 0xAAAA (test mode)
- **ADC_TEST_RAMP**
  - 16-bit ramp pattern (test mode)
- **ADC_TEST_SIN**
  - 8-point sine wave test pattern
- **ADC_TEST_PRBS**
  - 16-bit PRBS generated by a 23-bit PRBS pattern generator (test mode)
- **Ramp**
  - Data from a ramp generator. It is actually a 16-bit field, where the 6 most significant bits identify the channel and the 10 less significant bits are the samples of a ramp from 0x000 up to 0x3FF (i.e. 0 to 1023). It is so a 10-bit ramp with offset given by “channel*1024”. For channel 0, it is a counter from 0 to 1023; for channel 1, it is a counter from 1024 to 2047, and so on
- **IPE**
  - Not implemented
- **SquareWave**
  - Internally generated programmable square wave

parameter DCOffset

A constant DC offset (controlled by a 16-bit DAC) is added to the analog input, individually for each channel, to adjust the position of the signal baseline (that is the "zero volt" of the analog input) within the dynamic range of the ADC. Because of the tolerance of the components, it is necessary to calibrate the offset DAC. The calibration is done by factory testing and normally it is not necessary to recalibrate the digitizer. It is however possible to perform a new calibration. The calibration parameters are stored in the flash memory of the board and loaded at power on. They are automatically applied by the internal logic every time the DCoffset parameter is written or read. DCoffset is expressed as a NUMBER number, in percent of the full-scale. When the DCoffset is 0, the baseline of the input signal is at 0 ADC counts. When the DCoffset is 100, the baseline of the input signal is at $2^{NBIT}-1$ ADC counts.

parameter Polarity

Allows to set the polarity of the input pulse.

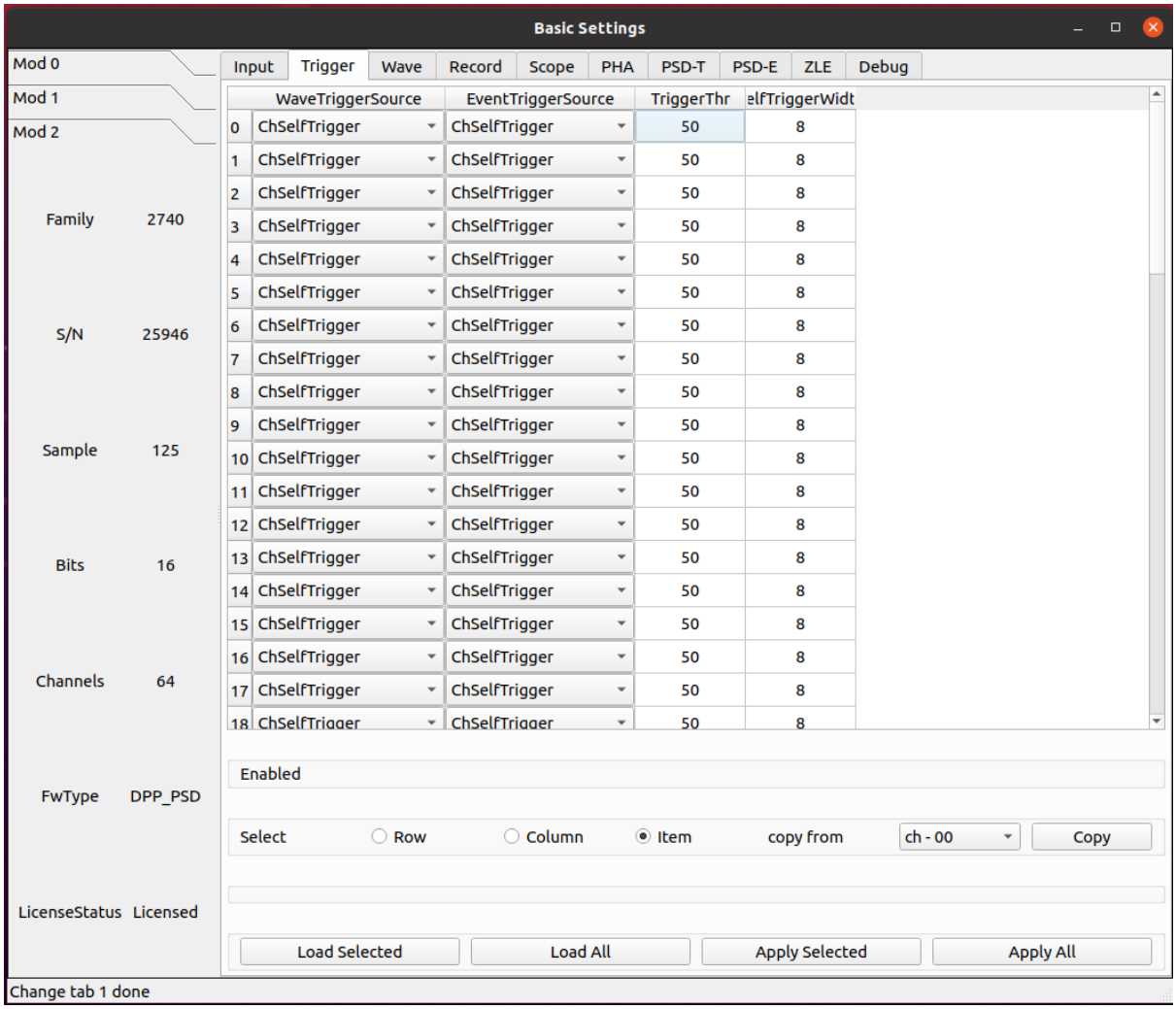
- **Positive**
  - Positive polarity
- **Negative**
  - Negative polarity

parameter VGAGain

Unique to x2745.

Set the gain of the variable gain amplifier (VGA) in increments of 0.5 dB. Parameter settings are grouped every 16 channels, with 64 channels divided into 4 groups. The minimum can be set to 0 and the maximum to 40.

9.1.2 Trigger



parameter WaveTriggerSource

Allows to set the trigger source for the waveform. Setting this parameter means to get an event including the waveform and the associated time stamp and energy information..

- **Disabled**
  - No trigger source enabled for the waveform
- **Ch64Trigger**
  - One (or more) channel self-trigger can generate a trigger for a waveform
- **ChSelfTrigger**
  - Channel self-trigger can generate a trigger for a waveform
- **SwTrg**
  - Software Trigger can generate a trigger for a waveform
- **ADCOverSaturation**
  - ADC Oversaturation can generate a trigger for a waveform
- **ADCUnderSaturation**
  - ADC Undersaturation can generate a trigger for a waveform

- **ExternalInhibit**
  - Inhibit can generate a trigger for a waveform
- **TRGIN**
  - External TRGIN can generate a trigger for a waveform
- **GlobalTriggerSource**
  - Acquisition Trigger Source (the same of the Scope mode) can generate a trigger for a waveform
- **LVDS**
  - A signal on the LVDS connectors can generate a trigger for a waveform
- **ITLA**
  - Internal Trigger Logic A can generate a trigger for a waveform
- **ITLB**
  - Internal Trigger Logic B can generate a trigger for a waveform

parameter EventTriggerSource

Allows to set the trigger source for a Time-Energy (T-E) event. Setting this parameter means to get an event including time stamp and energy information

- **Disabled**
  - No trigger source enabled for the T-E event
- **Ch64Trigger**
  - One (or more) channel self-trigger can generate a trigger for a T-E event
- **ChSelfTrigger**
  - Channel self-trigger can generate a trigger for a T-E event
- **SwTrg**
  - Software Trigger can generate a trigger for a T-E event
- **TRGIN**
  - External TRGIN can generate a trigger for a T-E event
- **GlobalTriggerSource**
  - Acquisition Trigger Source (the same of the Scope mode) can generate a trigger for a T-E event
- **LVDS**
  - A signal on the LVDS connectors can generate a trigger for a T-E event
- **ITLA**
  - Internal Trigger Logic A can generate a trigger for a T-E event
- **ITLB**
  - Internal Trigger Logic B can generate a trigger for a T-E event

parameter TriggerThr

Each channel of the digitizer has a digital leading-edge discriminator with programmable threshold able to self-trigger on the input pulses and generate a self-trigger signal (or an overthreshold signal) feeding the internal trigger logics or digitizer outputs. This parameter sets the trigger threshold. Typically, the value is relative to the baseline of the signal and the threshold is a 17-bit signed NUMBER number; in this case, the threshold automatically follows the baseline when the DCoffset parameter changes. Sometimes, it is preferable to set an absolute value for the threshold, referred to the ADC range. In this case, the threshold is unsigned NUMBER number.

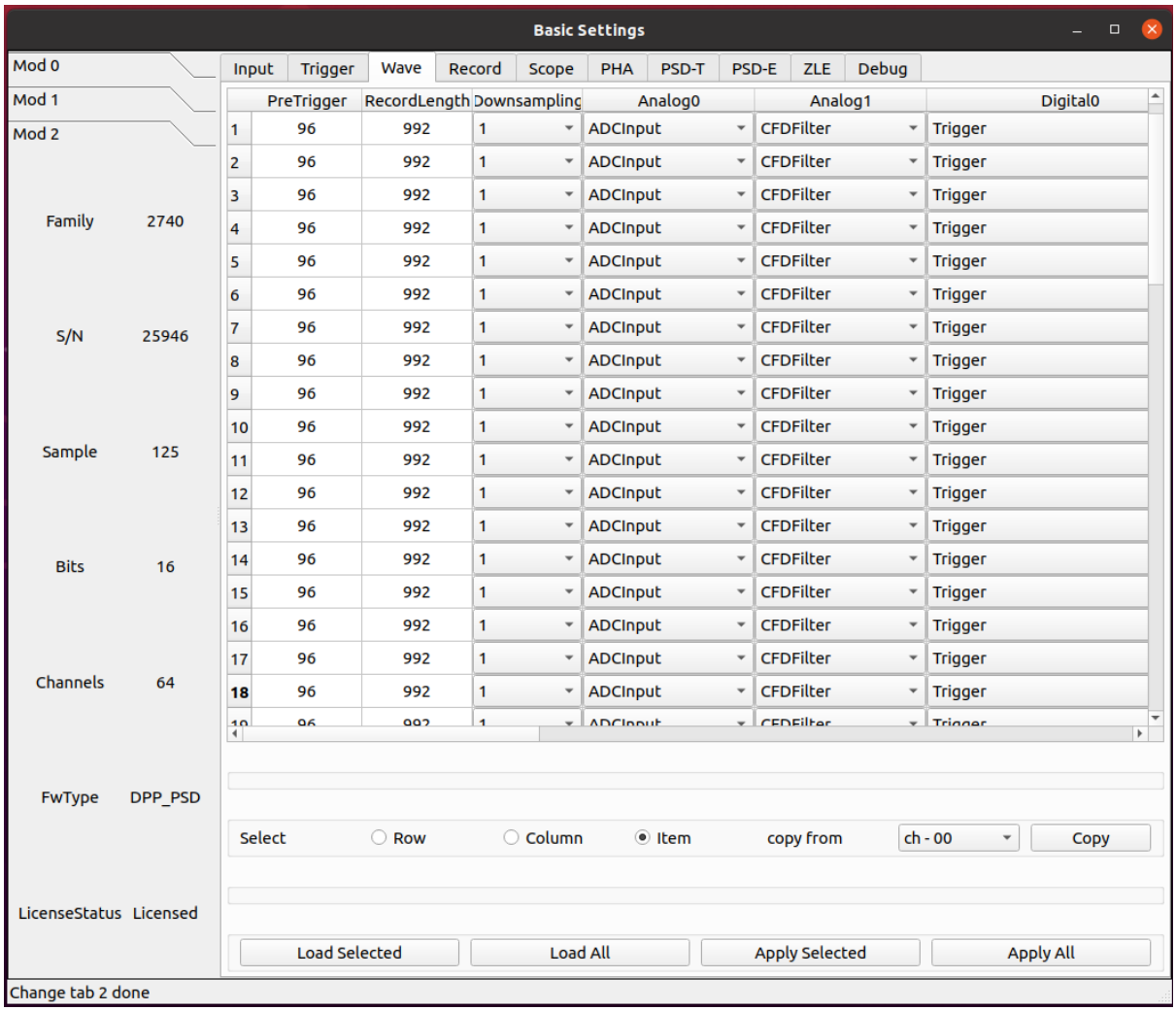
Unit of Measure: ADC counts

parameter SelfTriggerWidth

The output of the digital leading-edge comparator, that generates the self-trigger signal, can be used in "linear" mode, meaning that it lasts for the time the signal remains above (or below) the threshold, thus acting as an "Over-Threshold" signal, or can pass through a programmable gate generator that makes it a fixed-width pulse. The gate generator is a non-retriggerable monostable that goes high when the threshold is crossed and returns low after the programmed time. This parameter defines the fixed width of the overthreshold pulse.

Unit of Measure: ns

9.1.3 Wave



parameter PreTrigger

Time coming before the position of the trigger in the waveform (i.e. size of the pre-trigger window).

Unit of Measure: ns

parameter RecordLength

The waveform size. Integer representing the time in ns. The actual size of the waveform will be automatically rounded to the closest allowed value. It is possible to get the exact size by reading back the parameter. The record length in time depends on wave resolution.

Unit of Measure: ns

parameter DownsampingFactor

Downsampling factor for the waveform resolution.

- **1**
  - x1
- **2**
  - x2
- **4**
  - x4
- **8**
  - x8

parameter Analog0/1

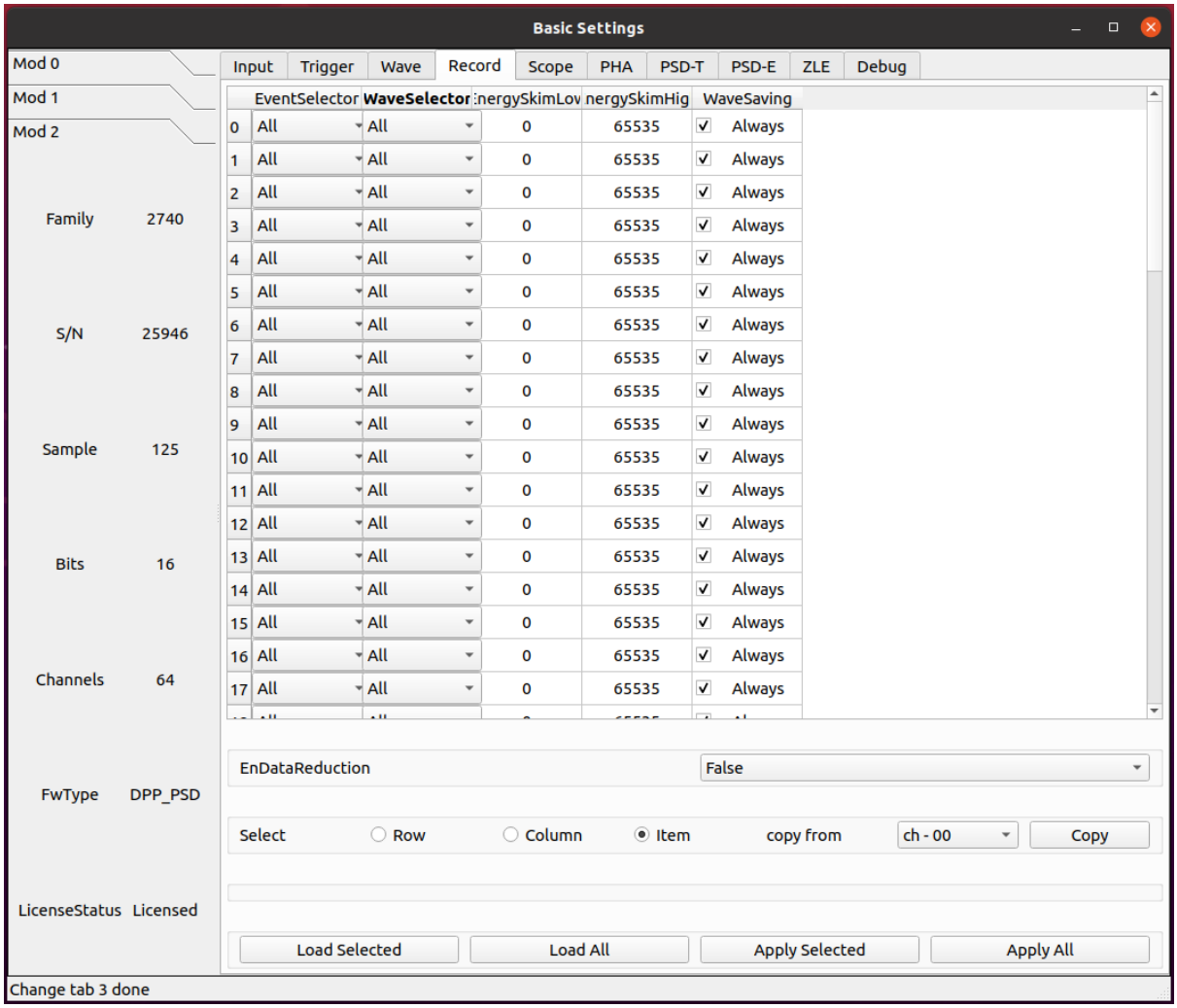
- **ADCInput**
  - ADC input probe
- **ADCInputBaseline**
  - ADC input baseline probe
- **CFDFilter**
  - Constant Fraction Discriminator filter probe

parameter Digital0/1/2/3

- **Trigger**
  - Trigger probe
- **CFDFilterArmed**
  - Constant Fraction Discriminator Filter Armed probe
- **ReTriggerGuard**
  - ReTrigger Guard probe
- **ADCInputBaselineFreeze**
  - ADC Input Baseline Freeze probe

- **ADCInputOverthreshold**
  - ADCInputOverthreshold
- **ChargeReady**
  - Charge Ready probe
- **LongGate**
  - Long Gate probe
- **PileUpTrigger**
  - Pile Up Trigger probe
- **ShortGate**
  - Short Gate probe
- **ChargeOverRange**
  - Integrated Charge Over Range probe
- **ADCSaturation**
  - ADC Saturation probe
- **ADCInputNegativeOverthreshold**
  - ADC Input Negative Overthreshold probe

9.1.4 Record



parameter EventSelector

Allows to set which events have to be saved.

- **All**
  - All events are saved
- **PileUp**
  - Only pileup events are saved
- **EnergySkim**
  - Save only the events in the Energy Skim range

parameter WaveSelector

Allows to set which waveform have to be saved.

- **All**
  - All waves are saved
- **PileUp**
  - Only pileup waves are saved
- **EnergySkim**
  - Save only waves in the Energy Skim range

parameter EnergySkimLowDiscriminator

Allows to flag events with energy higher than the low skim threshold. 16-bit value.

Unit of Measure: bin

parameter EnergySkimHighDiscriminator

Allows to flag events with energy lower than the high skim threshold. 16-bit value.

Unit of Measure: bin

parameter WaveSaving

Allows to save waveforms always or on request only.

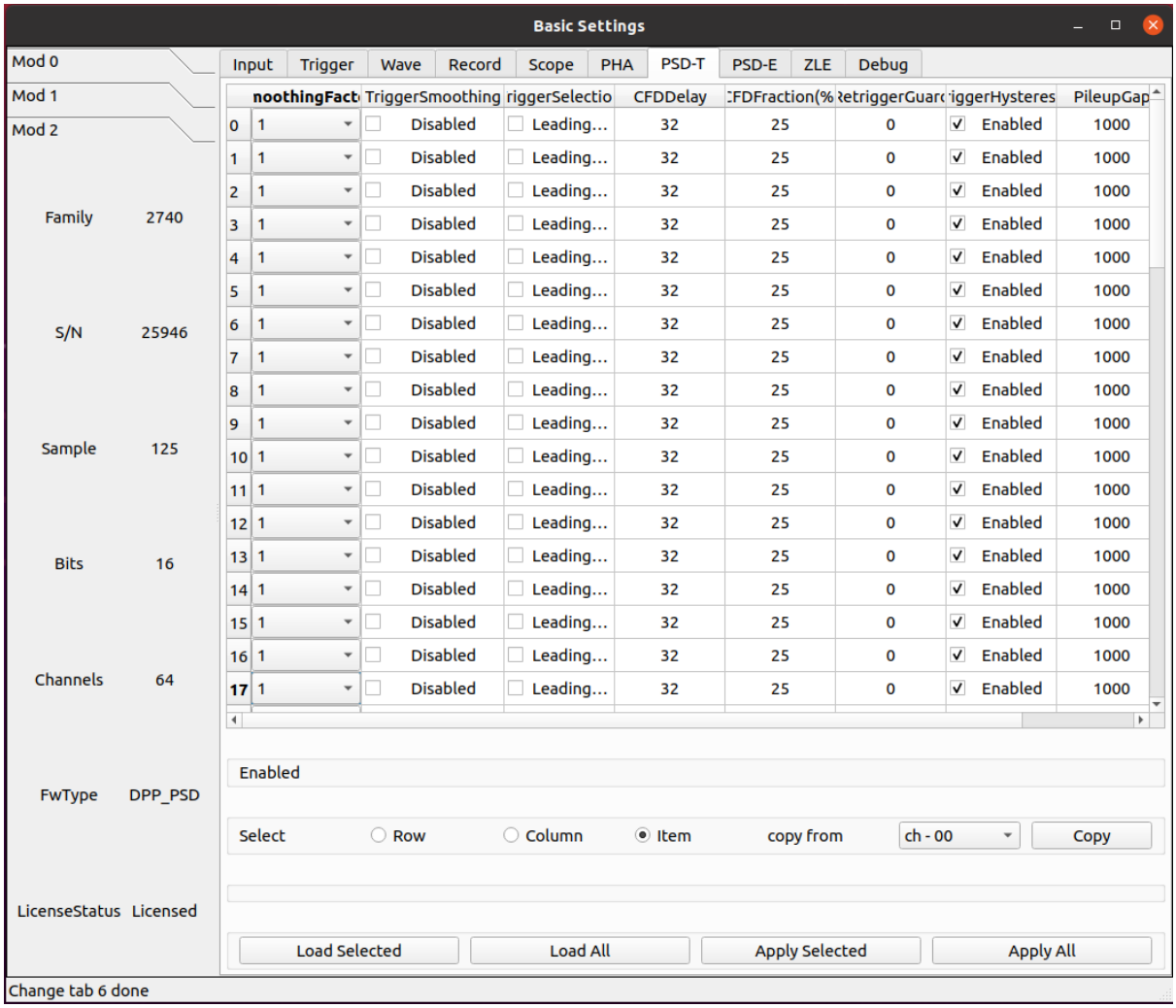
- **Always**
  - Waveforms are always saved
- **OnRequest**
  - Waveforms are saved on request

parameter EnDataReduction

If enabled, events consisting of 2 words are compressed in a single word event.

- **True**
  - Option enabled
- **False**
  - Option disabled

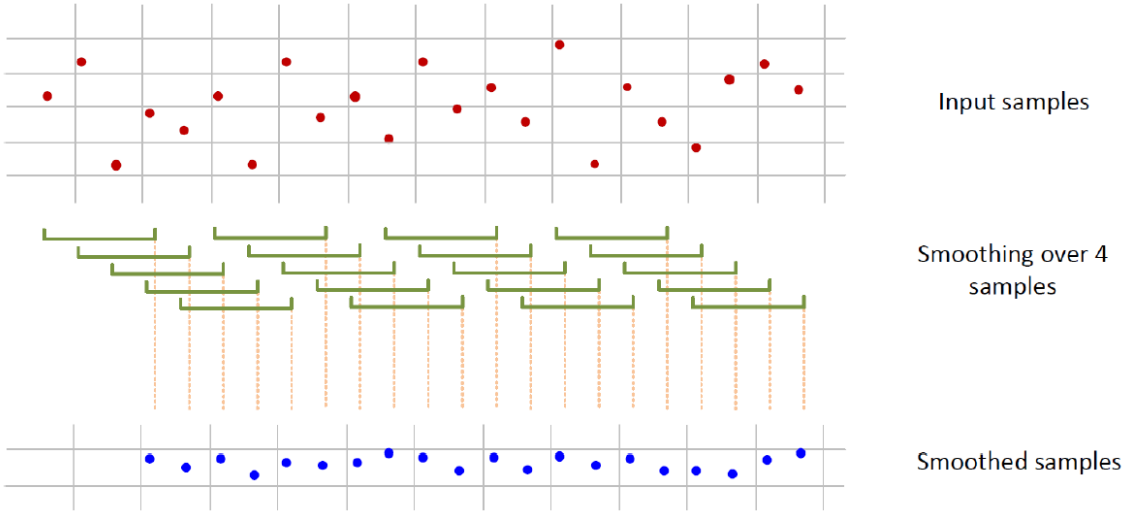
9.1.5 PSD-T



parameter SmoothingFactor

The smoothing is a moving average filter, where the input samples are replaced by the mean value of the previous n samples, where n is: 2, 4, 8 and 16 samples. When enabled (see TimeFilterSmoothing), the trigger is applied on the smoothed samples, thus reducing triggering on noise. Both CFD and LED triggering modes can be used on the smoothed input. The charge integration is either performed on the input samples and/or on the smoothed samples, according the ChargeSmoothing parameter.

- **1**
  - Smoothing is disabled.
- **2**
  - Smoothing is done averaging 2 samples.
- **4**
  - Smoothing is done averaging 4 samples.
- **8**
  - Smoothing is done averaging 8 samples.
- **16**
  - Smoothing is done averaging 16 samples.



parameter TriggerSmoothing

Enable/Disable Smoothing factor for the time filter.

- **Enabled**
  - Smoothing factor is enabled for the time filter.
- **Disabled**
  - Smoothing factor is disabled for the time filter.

parameter TriggerSelection

The DPP-PSD allows the user to select the pulses according to two methods: leading edge, where a pulse is identified when its samples crosses a programmable threshold value, or through a digital constant fraction discrimination to have a better timing information. In both cases once the event is selected, the signal is delayed by a programmable number of samples (corresponding to the "pre-trigger" value in ns) to be able to integrate the pulse before the trigger ("Pre-Gate"). The gates for charge integration are then generated and received by the charge accumulator before the signal. While the gates are active, the baseline remains frozen until the last averaged value and its value is used as charge integration reference. For the whole duration of a programmable "retrigger guard" (see TimeFilterRetriggerGuardT, TimeFilterRetriggerGuardS) value, other trigger signals are inhibited. It is recommended to set a trigger hold-off value compatible with the signal width. The baseline remains frozen for the whole trigger hold-off duration. This parameter allows to set the Leading Edge or Constant Fraction Discriminator Filter selection

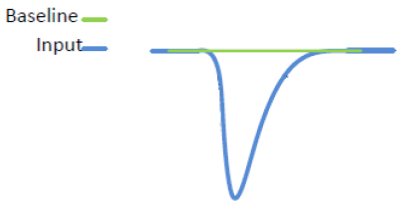
- **LeadingEdge**
  - Set the Leading Edge discriminator
- **CFD**
  - Set the Constant Fraction discriminator

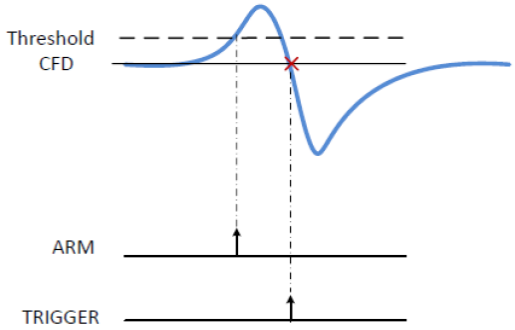
parameter CFDDelay

The x27xx digitizer running the DPP_PSD firmware discriminates events based on a CFD signal. The digital CFD signal has been implemented in the classical way except for the input signal inversion. The input waveform is first inverted, then attenuated by a factor f equal to the desired timing fraction (see CFDFraction) of full amplitude, then the signal is inverted again and delayed by a time d equal to the time it takes the pulse to rise from the constant fraction level to the pulse peak; the latest two signals are summed to produce a bipolar pulse, the CFD, and its zero crossing –corresponding to the fraction f of the input pulse –is taken as the trigger time.

The delay of the CFD signal can be defined by the user. The TriggerThreshold is then referred to the CFD itself, and the threshold crossing arms the event selection. The trigger fires at the zero crossing of the derivative signal itself.

Unit of Measure: ns





parameter CFDFraction

CFD Fraction. Unit of Measure: %

- **25**
  - 25%
- **50**
  - 50%
- **75**
  - 75%
- **100**
  - 100%

parameter RetriggerGuard

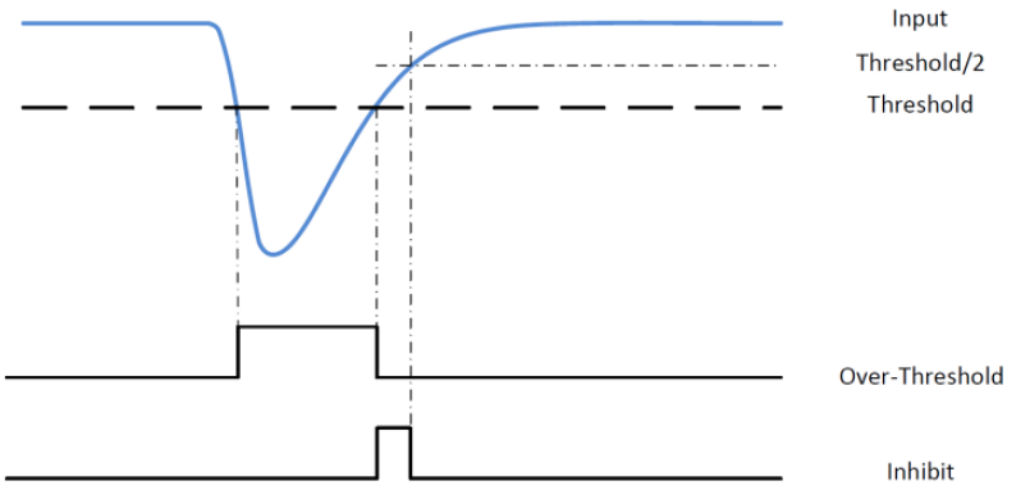
In case of fast signal such as those coming from PMTs possible overshoots in the fast discriminator signal may occur causing a retrigger and so possible fake pile-up. This parameter allows to set a retrigger inhibit guard (in ns). 10-bit value.

Unit of Measure: ns

parameter TriggerHysteresis

When the input signal is no more over-threshold, the trigger could fire again in the tail of the pulse, especially in case the tail contains spikes or noise. The “Trigger Hysteresis” feature inhibits any trigger until the input pulse reaches half of the threshold value itself. See for a diagram of this feature. This parameters allows to Enable/Disable Trigger hysteresis mechanism.

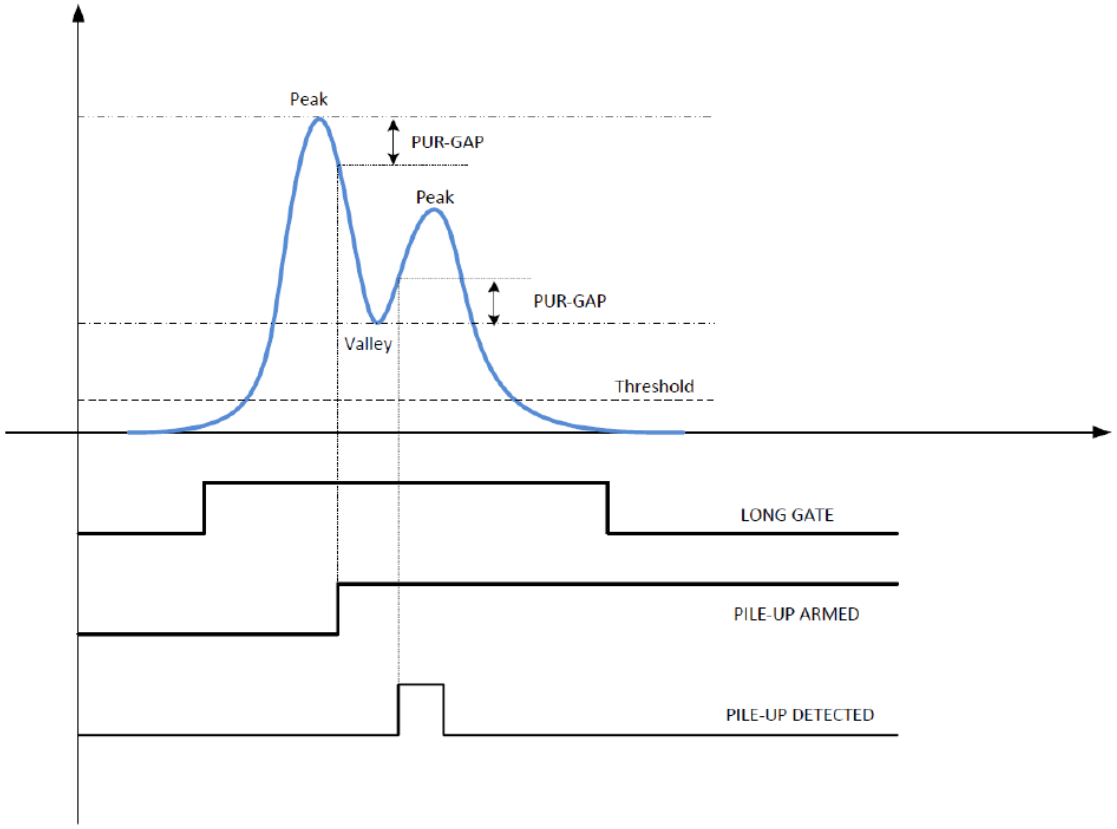
- **Disabled**
  - Trigger hysteresis mechanism is disabled.
- **Enabled**
  - Trigger hysteresis mechanism is enabled.



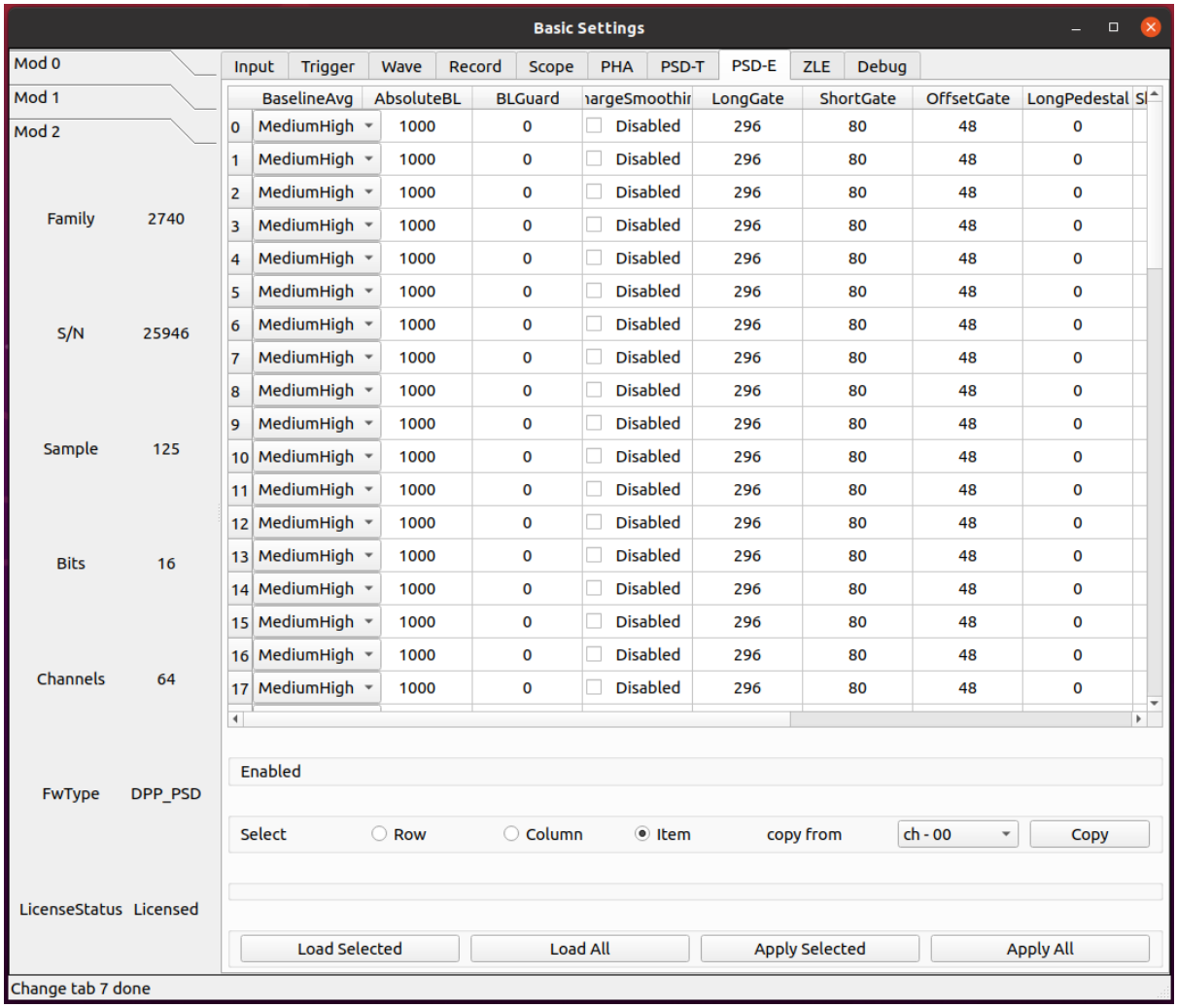
parameter PileupGap

The DPP-PSD firmware is mainly designed to work with fast signals like those coming from scintillation detectors coupled with Photomultiplier Tubes. The relevant output signals do not show long decay tails as in the case of charge sensitive preamplifiers, and the probability of pile-up between two pulses is quite low. In particular, the case of a second pulse sitting on the exponential tail of the previous one is rather rare. However, with the PSD algorithm, it is important to separate fast and slow components of the light emitted by the scintillation detector. Typically, the fast component is a quick pulse (few tens of ns) while the slow component is a quite long tail (typically a few µs) having amplitude much smaller than the fast component. To get the best results in the pulse shape discrimination, it is necessary to set the “Long Gate” as long as the full duration of the slow component. Under these conditions, most likely the events in pile-up occur during the long gate and cause an error in the calculation of the charge of the slow component. For this reason, it is important to detect these cases. In the DPP-PSD firmware, two events are considered in pile-up when there is a situation of peak-valley-peak inside the same gate, where the gap between the valley and the peak is a programmable value. Referring to Figure 6, when the peak value is reached the algorithm evaluates the point corresponding to the PileupGap (PUR-GAP) value and gets ready to detect a pile-up event (PILE-UP ARMED). If there is a condition of “valley”, and the input signal overcomes the PUR-GAP threshold, then the event is tagged as pile-up. In the default configuration the firmware does not take any action and the total charge of the event is evaluated within the gate and saved into memory.

This parameter allows to set the Peak Gap to identify pile up.


Peak
PUR-GAP
Peak
PUR-GAP
Valley
Threshold
LONG GATE
PILE-UP ARMED
PILE-UP DETECTED

9.1.6 PSD-E

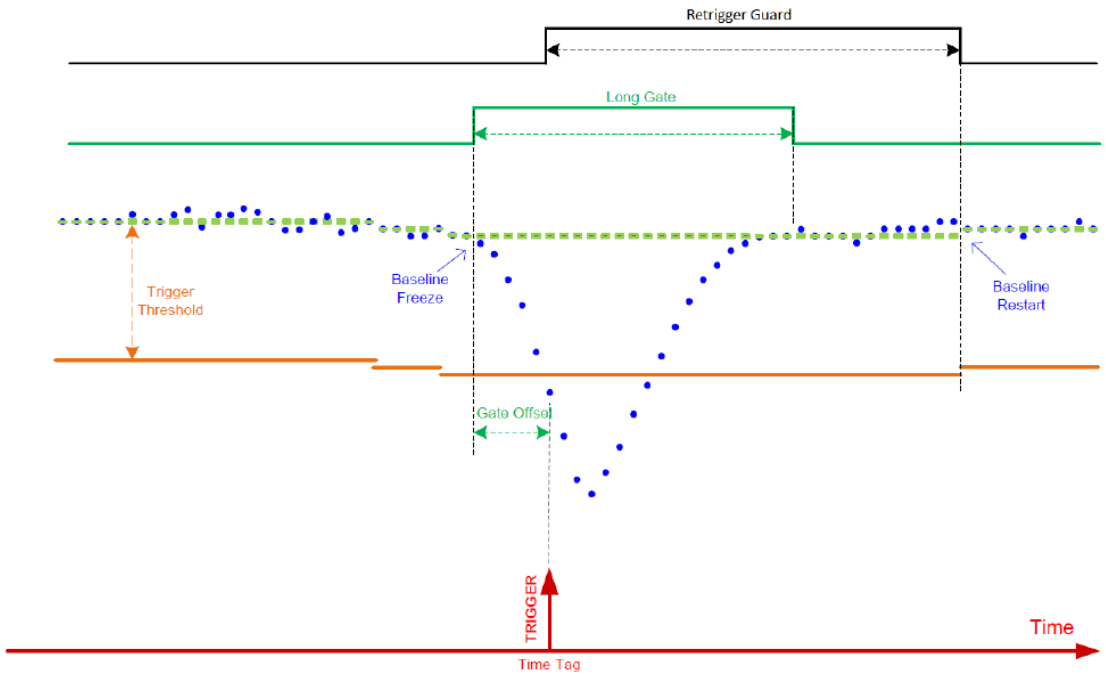


parameter BaselineAvg

The digitizer continuously receives the input signal and digitizes it. The position of the signal baseline can be adjusted in the ADC scale to exploit the full dynamics of the digitizer using the DCOffset parameter. The baseline value is an important parameter of the DPP-PSD firmware, since its value is used as a reference value for the charge integration of the input pulses. Moreover, most of the DPP parameters are related to the baseline value, like the trigger threshold. The user can set either a fixed value for the baseline or let the DPP firmware calculate it dynamically. In the first case the user must set the baseline value in LSB units through the option Fixed. This value remains fixed for the entire acquisition run. In the latter case, the firmware dynamically evaluates the baseline as the mean value of N points inside a moving time window. The user can choose one of the below listed option each of them corresponding to a predefined number of samples. The baseline is then frozen from few clocks before the gates start, up to the end of the maximum value between the long gate and the trigger hold-off (see GateLongLength and TimeFilterRetriggerGuard). This parameter allows to set the Number of samples used to average the baseline of the Energy Filter.

- **Fixed**
  - Baseline fixed at AbsoluteBaseline value
- **Low**
  - Baseline samples for average = 16
- **MediumLow**
  - Baseline samples for average = 64

- **MediumHigh**
  - Baseline samples for average = 256
- **High**
  - Baseline samples for average = 1024



parameter AbsoluteBL

Absolute value of the ADCInput signal baseline.

Unit of Measure: ADC counts

parameter BLGuard

Energy Filter Baseline Evaluation Guard before the integration gate open in nanoseconds (ns).

parameter ChargeSmoothing

Enable/Disable Smoothing factor for the charge evaluation.

- **Enabled**
  - Smoothing factor is enabled in the charge evaluation
- **Disabled**
  - Smoothing factor is disabled in the charge evaluation

parameter ShortGate

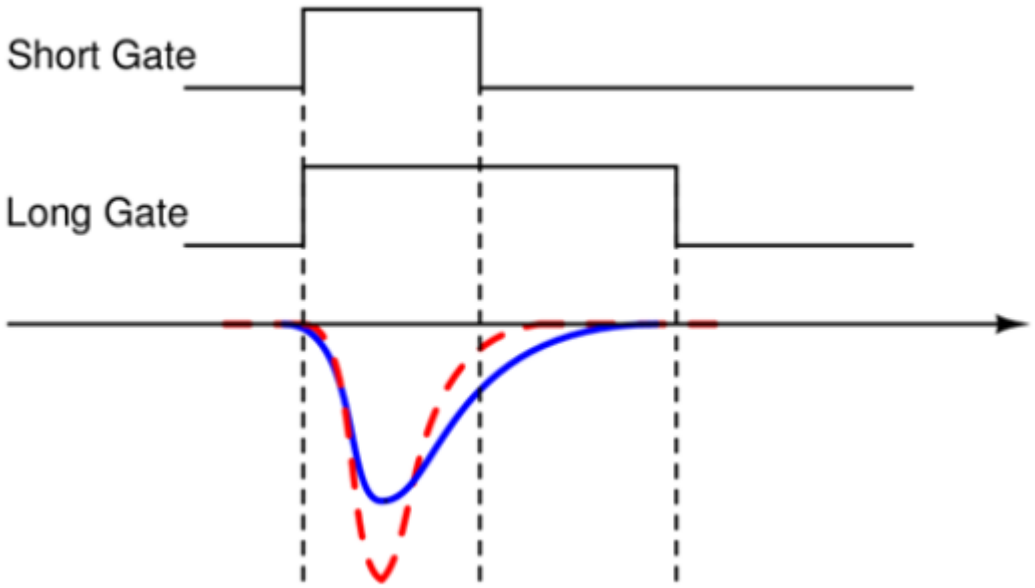
This parameter allows to set the Gate Short Length.

Unit of Measure: ns

parameter LongGate

The aim of the DPP-PSD firmware is to perform a charge integration of the input signal and to calculate the PSD factor performing a double gate integration of the input (Qshort and Qlong). Figure below shows the short and long gates position for two signals of different shapes. This parameter allows to set the Gate Long Length.

Unit of Measure: ns



parameter OffsetGate

This parameter allows to set the Gate Offset with respect to the trigger signal.

Unit of Measure: ns

parameter LongPedestal

This parameter allows to set the Long Charge Integrator Pedestal. This feature is useful in case of energies close to zero.

parameter ShortPedestal

This parameter allows to set the Long Charge Integrator Pedestal. This feature is useful in case of energies close to zero.

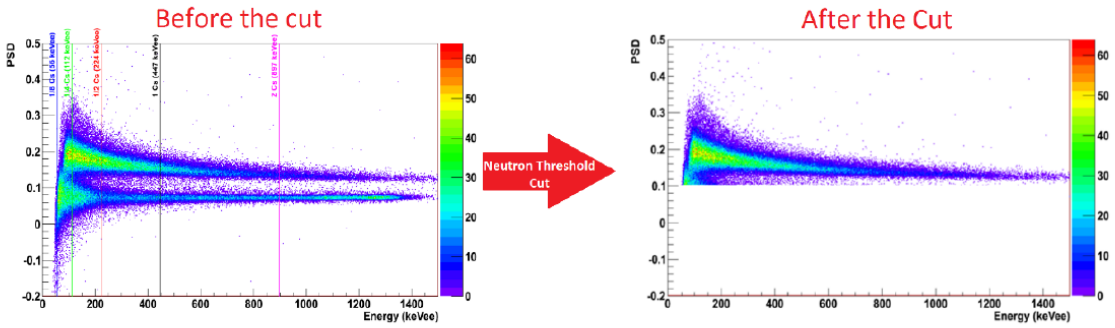
parameter EnergyGain

This parameter allows to set the Energy Gain i.e. to rescale the signal charge.

- **x1**
  - Charge value is multiplied x1.
- **x4**
  - Charge value is multiplied x4.
- **x16**
  - Charge value is multiplied x16.
- **x64**
  - Charge value is multiplied x64.
- **x256**
  - Charge value is multiplied x256.

parameter NeutronThr

This parameter allows to set the Neutron Energy Threshold for Neutron Discriminator. The FW compare the difference "Energy Long" – "Energy Short" (i.e. the difference between the charge integrated in the Long gate and in the Short gate) with the threshold set by this parameter to decide whether the event has to be rejected. Referring to the example of neutron/gamma discrimination shown in the figure below, the cut on PSD allows the user to reject most of the gamma events, thus recording only neutrons and the small amount of gamma overlapping with the neutrons.



parameter EventReject

Enable Neutron Rejection for Events. See NeutronThreshold parameter

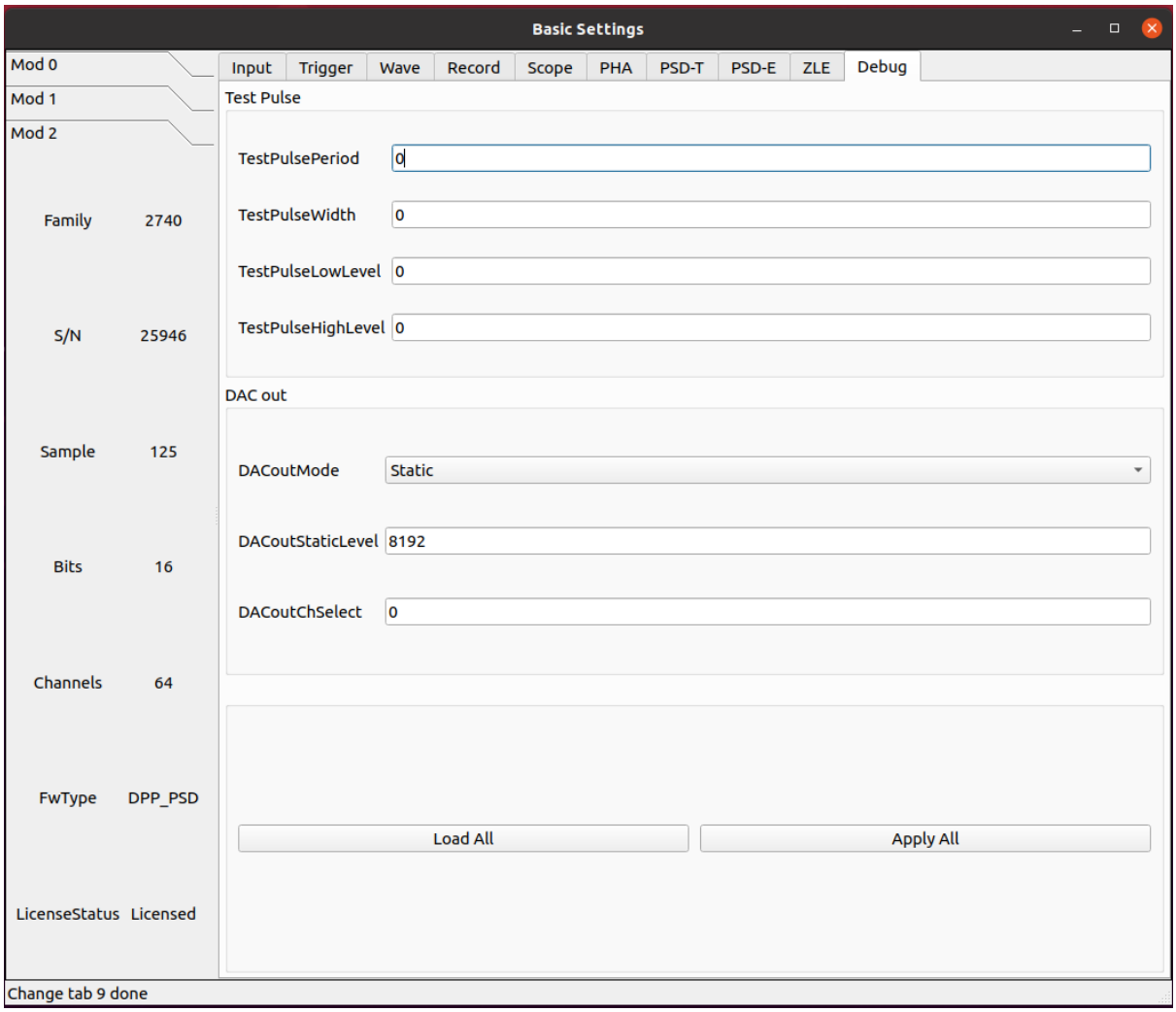
- **Disabled**
  - Neutron rejection for events is disabled.
- **Enabled**
  - Neutron rejection for events is enabled.

parameter WaveReject

Enable Neutron Rejection for Waves. See NeutronThreshold parameter

- **Disabled**
  - Neutron rejection for waves is disabled.
- **Enabled**
  - Neutron rejection for waves is enabled.

9.1.7 Debug



parameter TestPulsePeriod

The Test Pulse is a programmable square wave that can be used as an internal periodic trigger (mainly for test purposes) or to generate a logic test pulse (TTL or NIM) on the TRGOUT and GPIO outputs. This parameter sets the period of the Test Pulse.

Unit of Measure: ns

parameter TestPulseWidth

Width of the Test Pulse (time that the signal stays high = 1).

Unit of Measure: ns

parameter TestPulseLowLevel

Low level of the Test Pulse expressed in ADC counts

Unit of Measure: ADC counts

parameter TestPulseHighLevel

High level of the Test Pulse expressed in ADC counts

Unit of Measure: ADC counts

parameter DACoutMode

Selects the signal type to be sent in output on the front panel DAC connector.

- **Static**
  - DAC output stays at a fixed level, given by the DACoutStaticLevel parameter
- **Ramp**
  - The DAC output is driven by a 14-bit counter
- **Sin5MHz**
  - The DAC output is a sine wave at 5 MHz with fixed amplitude
- **Square**
  - Square wave with period and with set by TestPulsePeriod and TestPulseWidth and amplitude between TestPulseLoweLevel and TestPulseHighLevel.
- **IPE**
  - Not implemented
- **ChInput**
  - The DAC reproduces the input signal received by one input channel, selected by the DACoutChSelect parameter
- **MemOccupancy**
  - Level of the memory occupancy (not yet implemented)
- **ChSum**
  - The DAC reproduces the “analog” sum of all the digitizer inputs (not yet implemented)
- **OverThrSum**

  - The DAC output is proportional to the number of channels that are currently above the threshold

parameter DACoutStaticLevel

When DACoutMode = Static, this parameter sets the 14-bit level of the DAC static output.

parameter DACoutChSelect

When DACoutMode = ChInput, the DAC reproduces the input signal of the channel selected by this parameter.

parameter IPEAmplitude

The new digitizers are equipped with an Internal Pulse Emulator capable of generating exponential pulses. This parameter determines the amplitude of the pulse.

Unit of Measure: ADC counts

parameter IPEBaseline

Sets the offset of the exponantial pulses generated by the Internal Pulse Emulator.

Unit of Measure: ADC counts

parameter IPEDecayTime

Sets the decay time of the exponantial pulses generated by the Internal Pulse Emulator.

Unit of Measure: ns

parameter IPERate

Sets the rate of the exponantial pulses generated by the Internal Pulse Emulator.

Unit of Measure: Hz

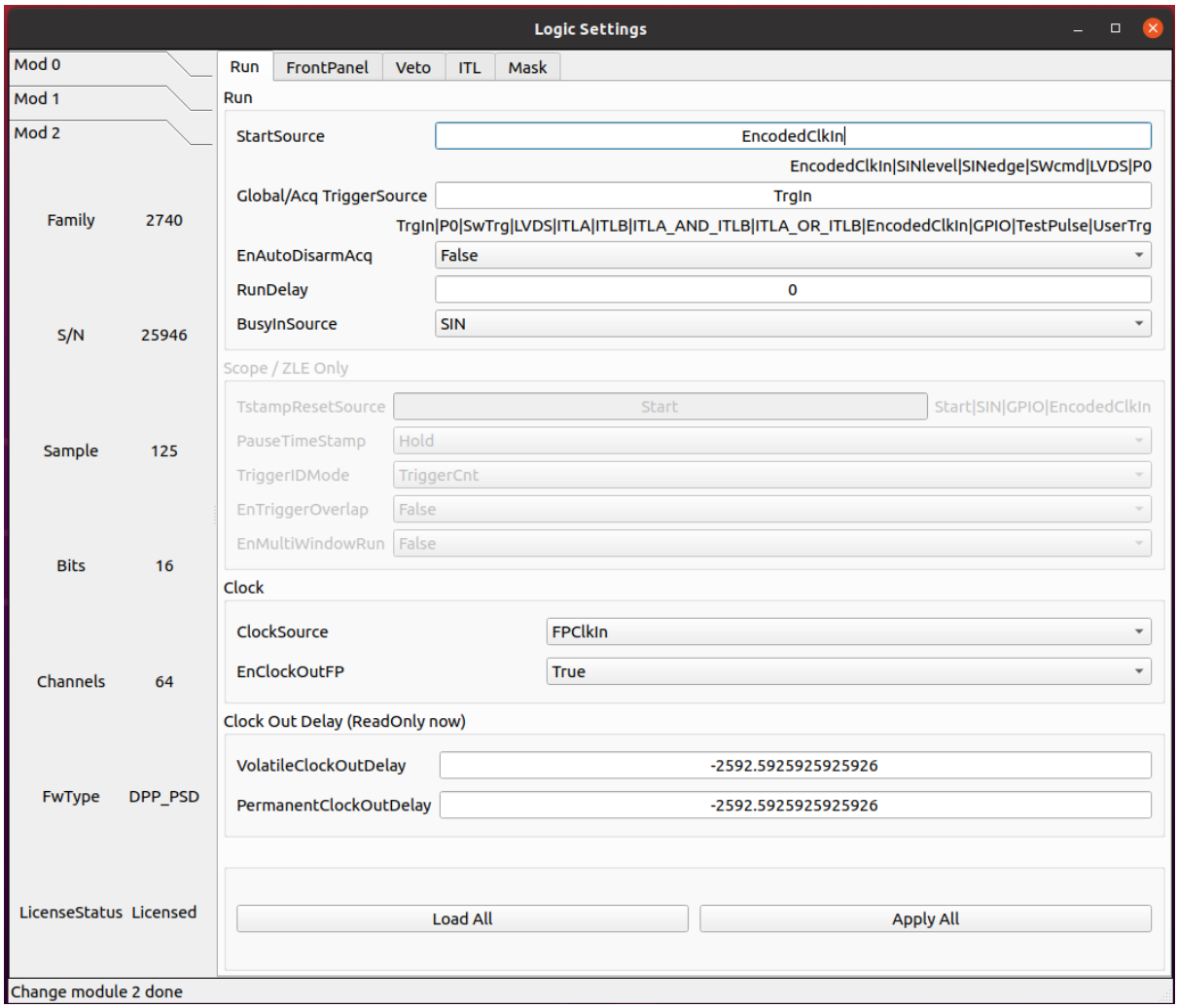
parameter IPETimeMode

Selectes the time distribution of the Internal Pulse Emulator.

- **ConstantRate**
  - Pulse shapes are constant over time. It is possible to set the frequency using the IPERate parameter
- **Poissonian**
  - The pulse rate follows a Poisson distribution. The average frequency value can be configured using the IPERate parameter

9.2 Logic parameter

9.2.1 Run



parameter StartSource

Defines the source for the start of run. Multiple options are allowed, separated by "|".

- **EncodedClkIn**
  - Start from CLK-IN/SYNC connector on the front panel. This is a 4-pin connector (LVDS signals) used to propagate the reference clock (typ. 62.5 MHz) and a Sync signal. The rising edge of the Sync starts the acquisition, that lasts until the Sync returns low (falling edge).
- **SINlevel**
  - Start from SIN (1=run, 0=stop)
- **SINedge**
  - Start from SIN (rising edge = run; stop from SW)
- **SWcmd**
  - Start from SW
- **LVDS**

  - Start from LVDS

- **P0**
  - Start from P0 (backplane)

parameter GlobalTriggerSource

Defines the source for the Acquisition Trigger, which is the signal that opens the acquisition window and saves the waveforms in the memory buffers. Multiple options are allowed, separated by "|".

- **TrgIn**
  - Front Panel TRGIN
- **P0**
  - Trigger from P0 (backplane)
- **SwTrg**
  - Software trigger
- **LVDS**
  - LVDS trgin
- **ITLA**
  - Internal Trigger Logic A: combination of channel self-triggers
- **ITLB**
  - Internal Trigger Logic B: combination of channel self-triggers
- **ITLA_AND_ITLB**
  - Second level Trigger logic making the AND of ITL A and B
- **ITLA_OR_ITLB**
  - Second level Trigger logic making the OR of ITL A and B
- **EncodedClkIn**
  - Not implemented (encoded CLK-IN trigger)
- **GPIO**
  - Front Panel GPIO
- **TestPulse**
  - Internal Test Pulse
- **UserTrg**
  - User custom trigger source

parameter EnAutoDisarmAcq

When enabled, the Auto Disarm option disarms the acquisition at the stop of run. When the start of run is controlled by an external signal, this option prevents the digitizer to restart without the intervention of the software.

- **True**
  - The acquisition is automatically disarmed after the stop. It is therefore necessary to rearm the digitizer (with the relevant command sent by the software) before starting a new run.
- **False**
  - The acquisition is not disarmed after the stop. Multiple transition of the start signal will produce multiple runs.

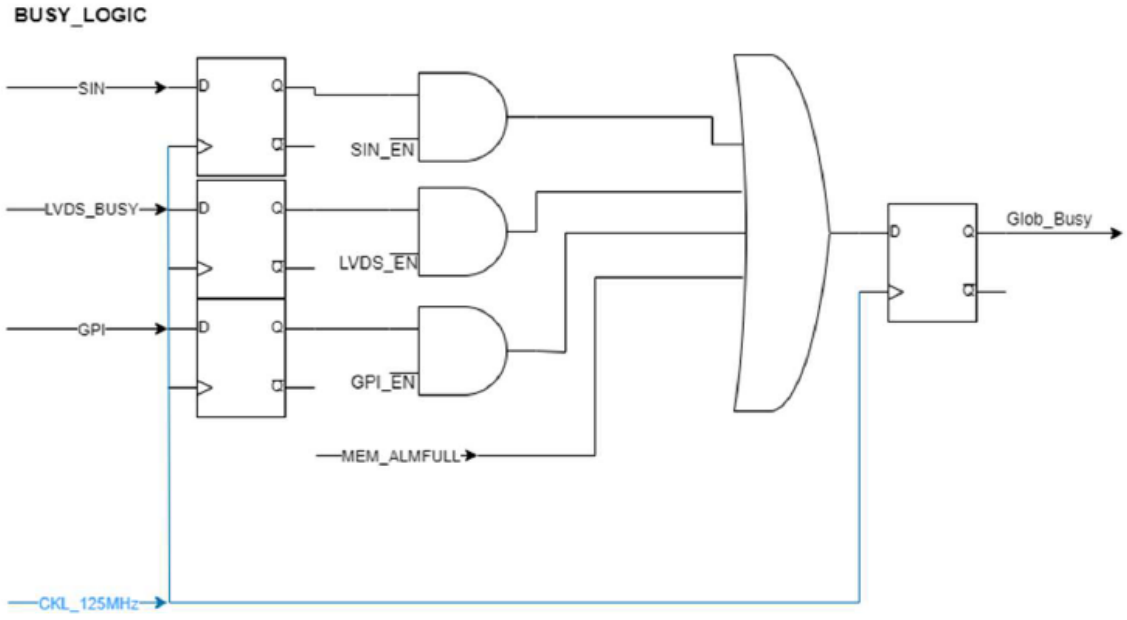
parameter RunDelay

When the start of run is controlled by a RUN signal that is propagated in daisy chain between the boards (for instance through the ClkIn- ClkOut or SIN-GPIO sync chain), it is necessary to compensate for the propagation delay and let the boards start exactly at the same time. The RunDelay parameter allows the start of the acquisition to be delayed by a given number of clock cycles with respect to the rising edge of the RUN signal. Assuming that the propagation delay is 2 cycles, the RunDelay setting will be 0 for the last board in the chain, 2 for the previous one, and so on up 2x(NB-1) for the first one.

Unit of Measure: ns

parameter BusyInSource

In a multi-board system, it might be necessary to prevent one board to accept a new trigger while another board is full and thus unable to accept the same trigger. For this reason, each board can generate a Busy signal to notify that it is unable to get a new trigger. If the busy/veto mechanism has some latency, it is advisable to generate the busy slightly before the digitizer become full. For this purpose, it is possible to assert the busy output when the acquisition memory reaches a certain level of occupancy (internally managed). The OR of the busy signals is typically used to stop the global trigger. It is possible to get the individual busy signals from each board and make an external OR logic or connect the boards with cables to propagate the Busy along the chain. Each board makes an OR between its internal busy and the busy input signal coming from the previous board, thus having a global Busy at the end of the line. This parameter defines the source of the Busy Input (schematized in the figure below)



- **Disabled**
  - The Busy is given by the Internal Busy only (Memory full or almost full)
- **SIN**
  - Busy input from SIN on front panel
- **GPIO**
  - Busy input coming from GPIO on front panel, used as a simple input. It is also possible to use GPIO as a wired OR (bidirectional). In this mode, the Busy line goes high as soon as one board drives it high. All the boards can read the Busy line and use it as a veto for the trigger
- **LVDS**
  - LVDS trgin

parameter ClockSource

This is the source of the system clock. Multiple options are not allowed

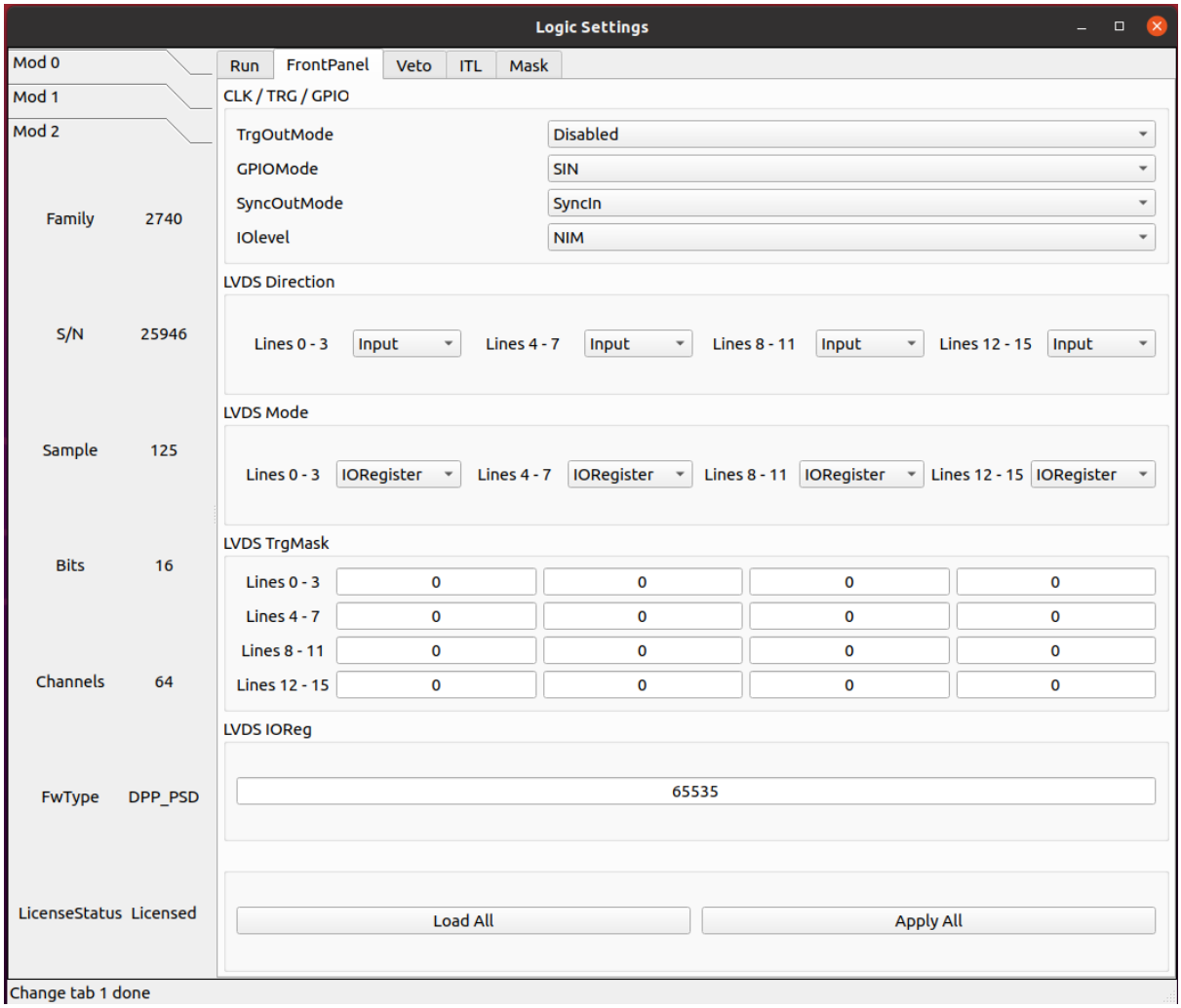
- **Internal**
  - Local oscillator, 62.5 MHz
- **FPClkIn**
  - Front Panel Clock input

parameter EnClockOutFP

Enables clock output on Front Panel for the daisy chain propagation of the clock between multiple boards.

- **True**
  - Enabled
- **False**
  - Disabled

9.2.2 FrontPanel



parameter TrgOutMode

Selects the signal that is routed to the TRGOUT output. Multiple options are not allowed.

- **Disabled**
  - TRGOUT output disabled
- **TrgIn**
  - Propagation of Front Panel TRGIN (TRGOUT is a replica, with some delay, of the TRGIN signal)
- **P0**
  - Propagation of P0 trigger
- **SwTrg**
  - Software trigger
- **LVDS**
  - LVDS trgin
- **ITLA**
  - Internal Trigger Logic A: combination of channel self-triggers

- **ITLB**
  - Internal Trigger Logic B: combination of channel self-triggers
- **ITLA_AND_ITLB**
  - Second level Trigger logic making the AND of ITL A and B
- **ITLA_OR_ITLB**
  - Second level Trigger logic making the OR of ITL A and B
- **EncodedClkIn**
  - Not implemented (propagation of the Encoded CLK-IN trigger)
- **Run**
  - Propagation of the RUN signal (acquisition start/stop), before applying the delay given by the Run-Delay parameter
- **RefClk**
  - Monitor of the 62.5 MHz clock (used for phase alignment)
- **TestPulse**
  - Internal Test Pulse
- **Busy**
  - Busy of the board
- **UserTrgout**
  - Trgout coming from the User Logic (open FPGA)
- **Fixed0**
  - 0 level signal
- **Fixed1**
  - 1 level signal
- **SyncIn**
  - SyncIn signal
- **SIN**
  - SIN connector signal
- **GPIO**
  - GPIO connector signal
- **LBinClk**
  - Internal Logic B clock signal
- **AcceptTrg**
  - Accepted triggers signal
- **TrgClk**
  - Tigger clock signal

parameter GPIOMode

Selects the signal that is routed to the GPIO, when this is used as output. Multiple options are not allowed. The GPIO on the front panel is a bidirectional signal that can used in three different ways:

> As independent board output (each board drives its own GPIO)
>
> As a shared input for the boards: the signal is driven high (= 1) or low (= 0) by an external source and connected in "short circuit" among multiple boards using "T" connectors at the inputs. The GPIO is not internally terminated, thus it is necessary to put a 50 Ohm terminator at the end of the line (last "T" of the chain)
>
> As a shared bidirectional line, making a "wired OR". One or more boards can simultaneously drive the signal high (= 1). If no board drives the GPIO, it remains low (= 0). All boards can read back the signal. It is necessary to put a 50 Ohm terminator at both ends of the line (first and last "T" of the chain). This mode can be used to generate, for instance, the global Busy and Veto logic for multiple boards.

- **Disabled**
  - GPIO disabled
- **TrgIn**
  - Propagation of Front Panel TRGIN (GPIO is a replica, with some delay, of the TRGIN signal)
- **P0**
  - Propagation of P0 trigger
- **SIN**
  - Propagation of SIN
- **LVDS**
  - LVDS trgin
- **ITLA**
  - Internal Trigger Logic A: combination of channel self-triggers
- **ITLB**
  - Internal Trigger Logic B: combination of channel self-triggers
- **ITLA_AND_ITLB**
  - Second level Trigger logic making the AND of ITL A and B
- **ITLA_OR_ITLB**
  - Second level Trigger logic making the OR of ITL A and B
- **EncodedClkIn**
  - Not implemented (propagation of the Encoded CLK-IN trigger)
- **SwTrg**
  - Software trigger
- **Run**
  - Propagation of RUN
- **RefClk**
  - Monitor of the 62.5 MHz clock (used for phase alignment)
- **TestPulse**
  - Internal Test Pulse

- **Busy**
  - Busy of the board
- **UserGPO**
  - GPO coming from the User Logic (open FPGA)
- **Fixed0**
  - 0 level signal
- **Fixed1**
  - 1 level signal

parameter SyncOutMode

In a multi-board system, it can be useful to propagate a synchronous signal together with the clock (to synchronize the start of the run, for example) on CLK OUT front panel connector. This parameter defines which signal must be sent out. Multiple options are not allowed.

- **Disabled**
  - SyncoutMode is disabled
- **SyncIn**
  - SyncIn signal (if provided with clkIn on CLK IN connector)
- **TestPulse**
  - Internal Test Pulse
- **IntClk**
  - Internal 62.5 MHz clock (for test purposes)
- **Run**
  - Propagation of RUN signal
- **User**
  - User customSyncoutMode

parameter IOlevel

Sets the electrical logic level of the LEMO I/Os (TRGIN, SIN, TRGOUT, GPIO).

Note that TRGIN and SIN are internally terminated to 50 Ohm, while GPIO and TRGOUT require the termination to 50 Ohms at the receiver

- **NIM**
  - NIM logic (0 = 0V, 1 = -0.8V, that is -16mA)
- **TTL**
  - Low Voltage TLL logic (0 = 0V, 1 = 3.3V)

parameter LVDSDirection

Assigns the direction to a quartet of LVDS I/Os.

- **Input**
  - The LVDS lines of the relevant quartet are used as input. The relevant LED on the front panel is OFF.
- **Output**
  - The LVDS lines of the relevant quartet are used as output. The relevant LED on the front panel lights-up.

parameter LVDSMode

The digitizer is equipped with 16 LVDS I/Os that can be programmed to be inputs or outputs by groups of 4 (quartets), depending on the LVDSDirection parameter. Once the direction has been selected, it is possible to select the functionality of the LVDS lines, individually for each quartet.

- **SelfTriggers**
  - This option is available only when the LVDS are set as outputs. Each LVDS line can be assigned to a combination of the 64 self-triggers, implemented as a masked OR, where the mask is set by the LVDSTrgMask parameter(16 independent masks, one per LVDS line)
- **Sync**
  - Whatever is the direction of the quartet, the 4 lines are rigidly assigned to specific acquisition signals: 0 = Run 1 = Trigger 2 = Busy 3= Veto It is possible to implement a daisy chain distribution of these signals using one quartet as input and another one as output
- **IORegister**
  - The LVDS lines of the quartet are statically controlled by the LVDSIOReg parameter. Use the SetValue function to set the relevant LVDS lines when programmed as output. Use GetValue to read the status of the LVDS lines when programmed as inputs.
- **User**
  - User custom.

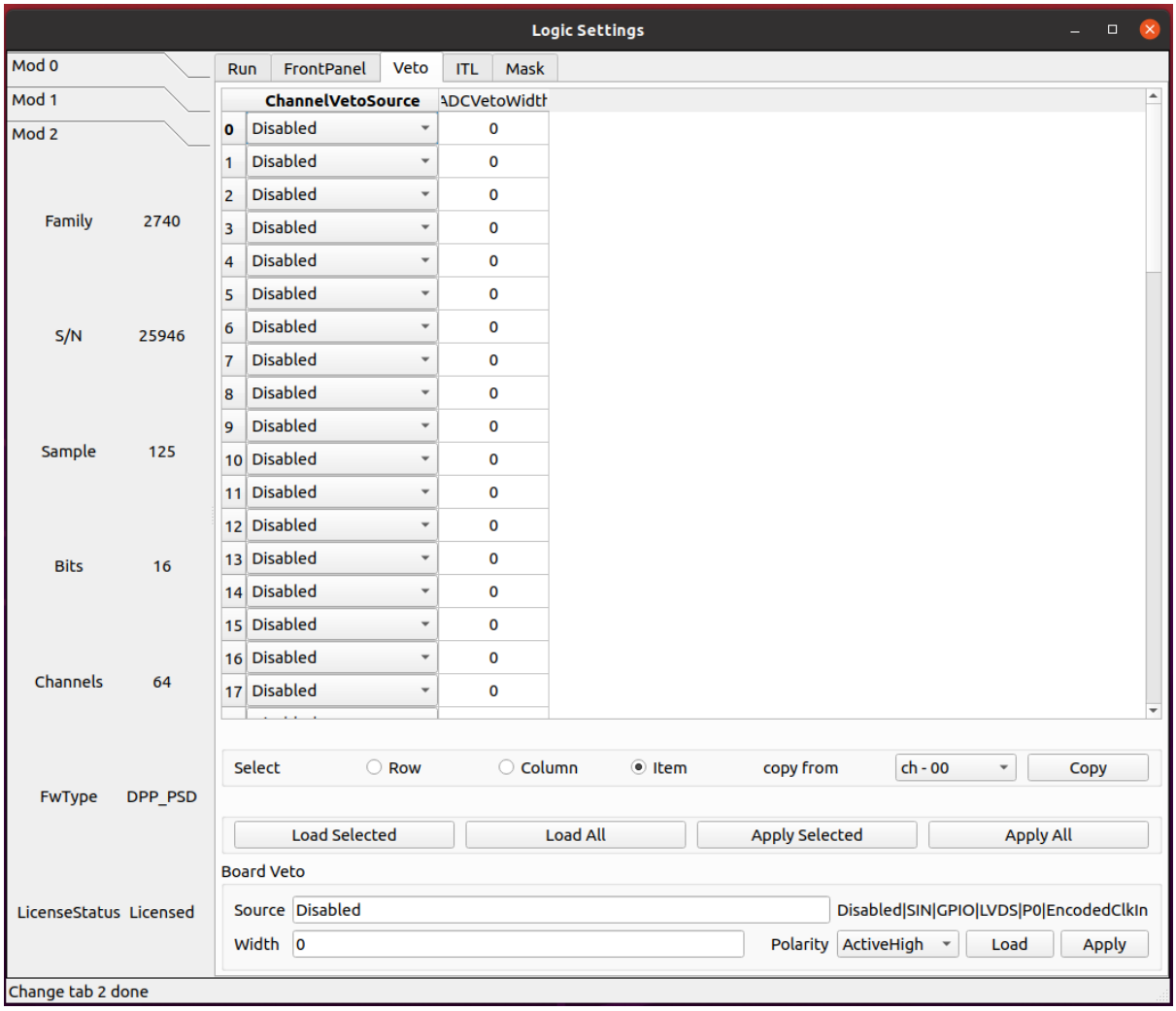
parameter LVDSTrgMask

Each LVDS line can be assigned to a combination of the 64 self-triggers, implemented as a masked OR, where the mask is set by this parameter. There are 16 independent masks, one per LVDS line. Note that the trigger mask assignment does not imply the LVDS direction and mode settings. It is therefore necessary to set the Direction = Output and Mode = SelfTriggers to use the Self-Trigger propagation to the LVDS I/Os.

parameter LVDSIOReg

Set the status of the LVDS I/O for the quartets when they are programmed to be output and Mode = IORegister. This parameter reads out the status of the quartets in the case the LVDS I/O are programmed as inputs (possibly externally driven).

9.2.3 Veto



parameter ChannelVetoSource

Allows to set the veto for each channel; it can be external (which means one of the veto options in the previous table), or it can be on a channel base.

- **Disabled**
  - Any channel veto source is disabled
- **BoardVeto**
  - Enables board veto
- ADCOverSaturation: Enables veto due to ADC oversaturation
- ADCUnderSaturation: Enables veto due to ADC undersaturation

parameter ADCVetoWidth

It is the width of the ADC veto (undersaturation and oversaturation width) expressed in ns.

Unit of Measure: ns

parameter VetoSource

Defines the source for the Veto, which is the signal that inhibits the acquisition trigger. Multiple options are allowed, separated by "|". The VETO signal can be either active high or low, depending on the VetoPolarity parameter. When active low, it acts as a GATE for the trigger. It is possible to stretch the duration of the VETO by means of the parameter VetoWidth.

- **Disabled**
  - VETO is always OFF
- **SIN**
  - SIN on the front panel
- **GPIO**
  - GPIO on the front panel (used as input)
- **LVDS**
  - LVDS trgin
- **P0**
  - P0 (signal from the backplane)
- **EncodedClkIn**
  - Not implemented (encoded CLK-IN veto)

parameter VetoWidth

Whatever is the source of the VETO signal, it is possible to stretch the duration of the veto up to a given time by means of a re-triggerable monostable. When 0, the monostable is disabled and the veto lasts as long as the selected source is active.

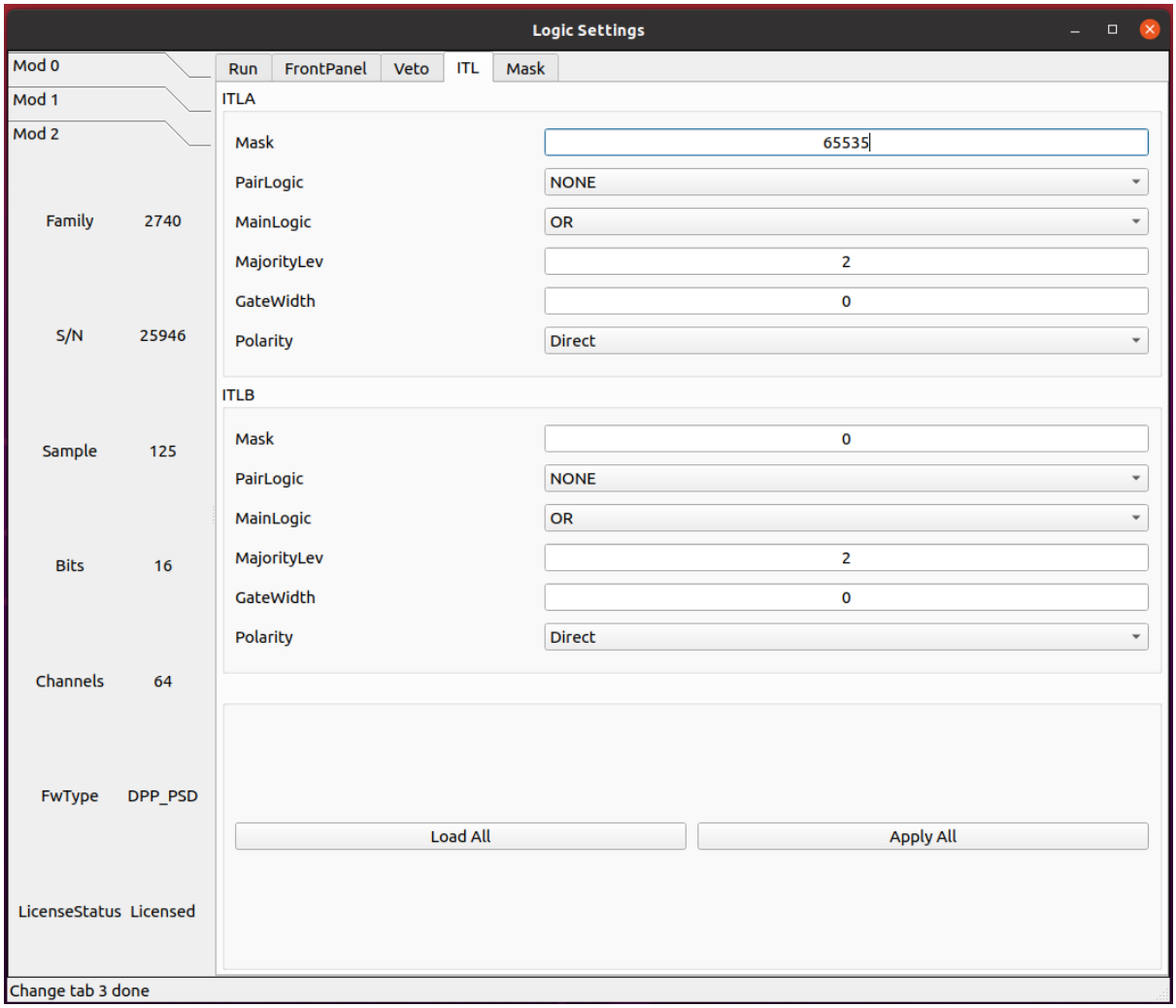
Unit of Measure: ns

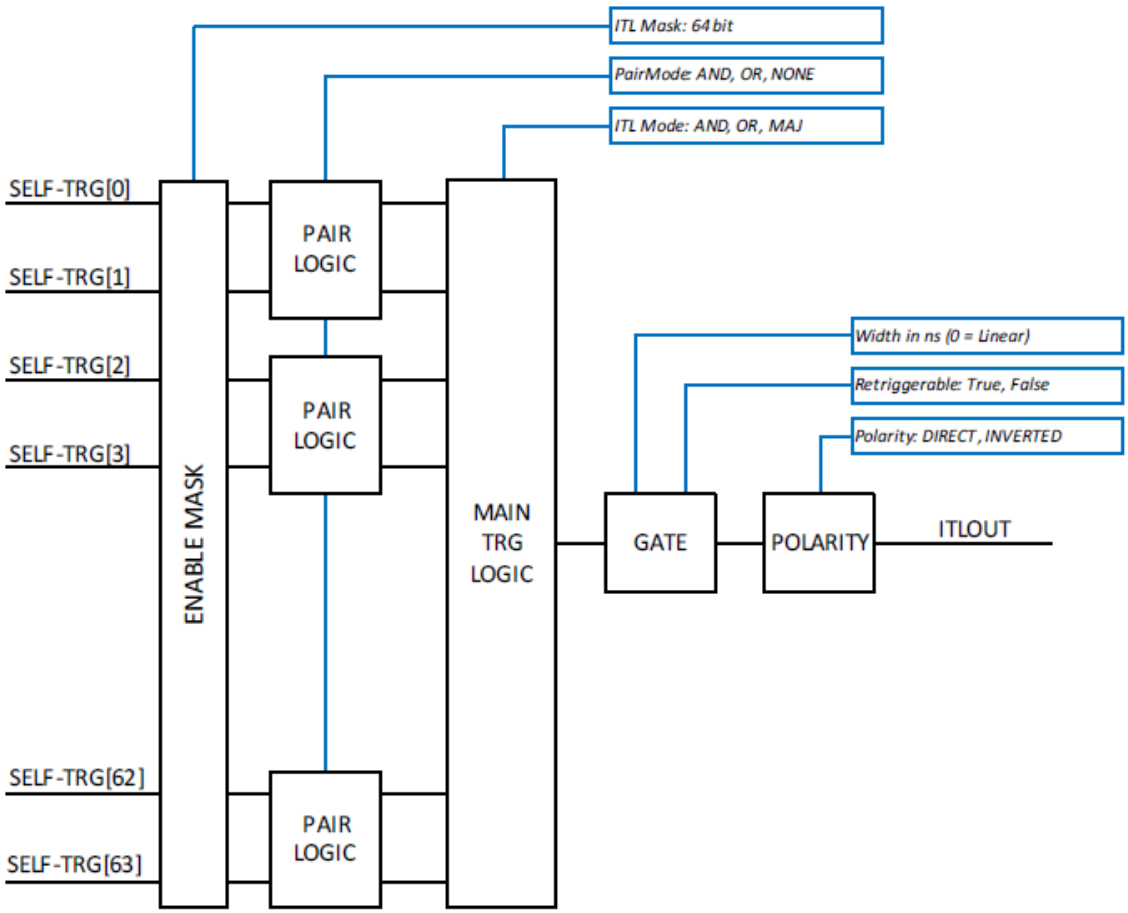
parameter VetoPolarity

Defines the polarity of the Veto

- **ActiveHigh**
  - Veto is active high. The signal acts as an "Inhibit" for the trigger
- **ActiveLow**
  - Veto is active low. The signal acts as a "Gate" the trigger

9.2.4 ITL


Logic Settings
Mod 0
Mod 1
Mod 2
Family 2740
S/N 25946
Sample 125
Bits 16
Channels 64
FwType DPP_PSD
LicenseStatus Licensed
Run
FrontPanel
Veto
ITL
Mask
ITLA
Mask 65535
PairLogic NONE
MainLogic OR
MajorityLev 2
GateWidth 0
Polarity Direct
ITLB
Mask 0
PairLogic NONE
MainLogic OR
MajorityLev 2
GateWidth 0
Polarity Direct
Load All
Apply All
Change tab 3 done



parameter ITLA/BMask

Enable Mask at the input of the ITLA/B.

parameter ITLA/BPairLogic

Pairs of channels can be combined with an OR or AND before feeding in the Main trigger Logic. This is typically used in the readout of tubes or scintillator bars, where the two ends are read in coincidence, for instance in position sensitive detectors (the coincidence window will be set by the SelfTriggerWidth parameter). When the AND/OR logic is applied, the two outputs of the Pair Logic blocks are identical.

Note that they are counted twice in the following Majority logic. If the Pair Logic is disabled ("NONE" option), the block is transparent, and the two outputs are just a replica of the inputs.

- **OR**
  - Both Pair Logic Outputs = OR of two consecutive self-triggers
- **AND**
  - Both Pair Logic Outputs = AND of two consecutive self-triggers
- **NONE**
  - Outputs = Inputs

parameter ITLA/BMainLogic

Each channel of the digitizer feature a digital bipolar triangular filter discriminator with programmable rise time and threshold able to self-trigger on the input pulses and generate a self-trigger signal. In DPP Mode, the channels acquire independently, so the channel self-trigger is used locally to acquire a waveform. The trigger threshold is then referred to the bipolar triangular filter, and the threshold crossing arms the event selection. The trigger fires at the zero crossing of the time filter signal. The user can see the derivative trace on the signal inspector. It is also possible to combine all the self-triggers of the board, according to a specific trigger logic. There are two independent logic blocks, ITLA and ITLB. Their output can be used separately to feed, for instance, AcqTrigger and TrgOut, or combined in a second level trigger logic to implement more complex trigger schemes. Therefore, the ITLs can either generate the local acquisition trigger, common to all the channels, for the acquisition of the waveform, or propagate the signal outside, through the TRGOUT, thus making it possible to combine triggers of multiple boards in an external trigger logic, that eventually feeds back the TRGIN of the digitizers. Each ITL is made of an input enable mask (64 bits, one per channel), an optional pairing logic that combines the self triggers of two consecutive channels (e.g. paired coincidence) and the main trigger logic that combines the 64 selftriggers with an OR, AND or Majority logic. The output can be linear (no stretching) or reshaped by a programmable gate generator, either re-triggerable or not and finally programmed for polarity (direct or inverted).

- **OR**
  - ITLOUT = masked OR of channel self-triggers
- **AND**
  - ITLOUT = masked AND of channel self-triggers
- **Majority**
  - ITLOUT = masked Majority of channel self-triggers

parameter ITLA/BMajorityLev

Defines the majority level of the Main Logic of the ITL A/B block. The majority output is calculated at every clock cycle, and it becomes TRUE when Nch >= MajLev, where Nch is the number of self-triggers active in that clock cycle and MajLev is the programmed majority level.

Note that when the Pair Logic is used to combine the self triggers two by two (AND/OR), each pair produces two identical signals that will be counted twice in the majority level.

parameter ITLA/BGateWidth

Width of the gate generator at the output of the ITLA/B block.

Unit of Measure: ns

parameter ITLA/BPolarity

Polarity of the gate generator output.

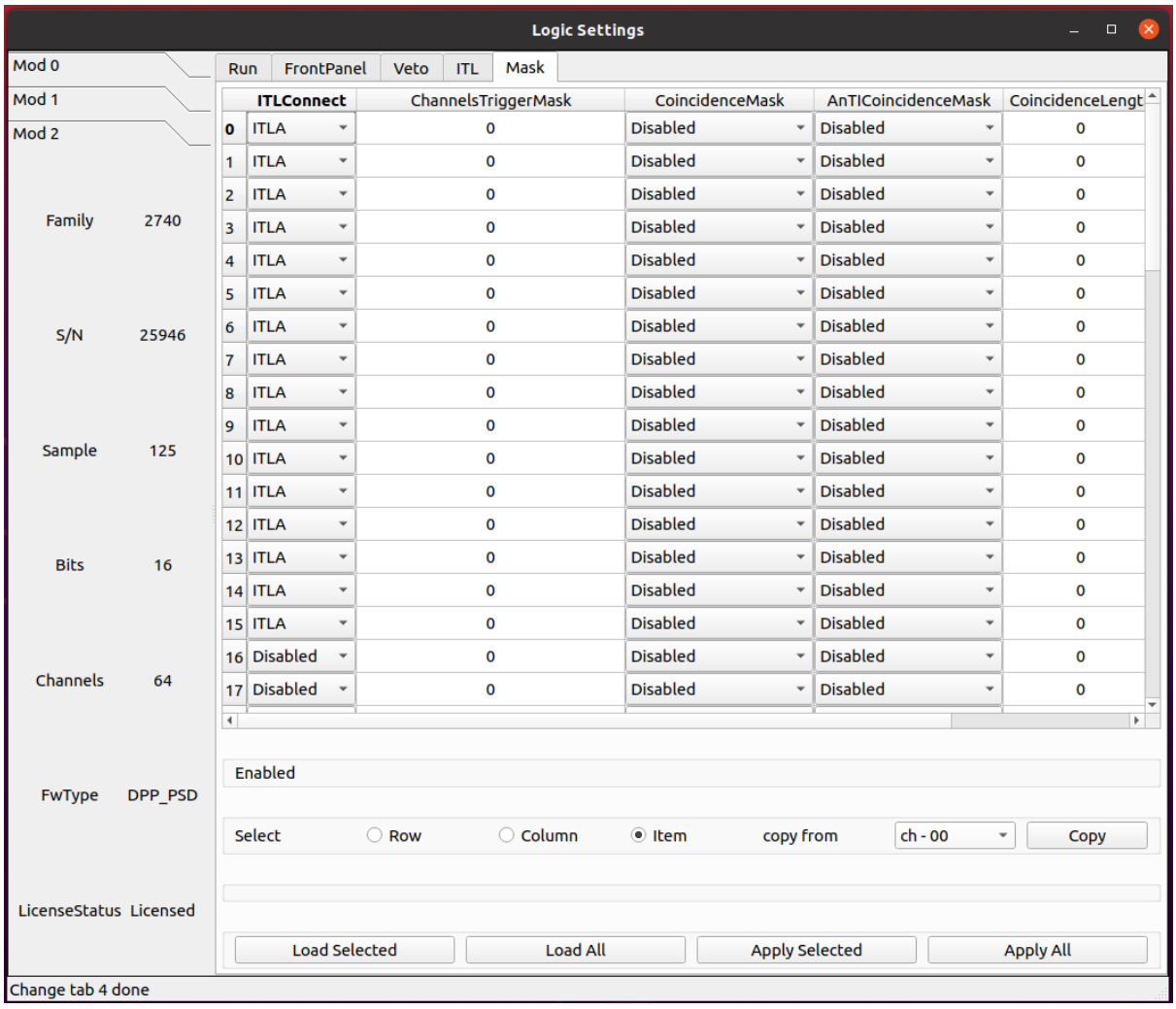
- **Direct**
  - Direct polarity
- **Inverted**
  - Inverted polarity

parameter ITLA/BEnRetrigger

Set the ITLA/B to be retriggerable.

- **True**
  - The ITLA/B is retriggerable
- **False**
  - The ITLA/B is not retriggerable

9.2.5 mask



parameter ITLConnect

Alternative to ITLAMask, ITLBMask. Determines if the channel partecipate in ITLA or ITLB

- **Disabled**
  - The channel is disabled
- **ITLA**
  - The channel participates in ITLA logic block
- **ITLB**

  - The channel participates in ITLB logic block

parameter ChannelsTriggerMask

Allows to set the mask over 64 bits to generate a channel trigger. It can be used to trigger a channel using a trigger coming from another channel. It also allows to set the mask over 64 bits to enable the channel to participate in the coincidence logic defined in CoincidenceMask and AntiCoincidenceMask (option Channel64Trg). 64-bit enable mask, each bit representing a channel.

parameter CoincidenceMask

Allows to set the coincidence mask that generates a trigger on the specified channel.

- **Disabled**
  - All the coincidence sources are disabled
- **Ch64Trigger**
  - One of the 64 channels can generate a coincidence signal
- **TRGIN**
  - TRGIN can generate a coincidence signal
- **GlobalTriggerSource**
  - Acquisition Trigger can generate a coincidence signal
- **ITLA**
  - ITLA can generate a coincidence signal
- **ITLB**
  - ITLB can generate a coincidence signal

parameter AntiCoincidenceMask

Allows to set the anticoincidence mask that generates a trigger on the specified channel.

- **Disabled**
  - All the coincidence sources are disabled
- **Ch64Trigger**
  - One of the 64 channels can generate a coincidence signal
- **TRGIN**
  - TRGIN can generate a coincidence signal
- **GlobalTriggerSource**
  - Acquisition Trigger can generate a coincidence signal
- **ITLA**
  - ITLA can generate a coincidence signal
- **ITLB**
  - ITLB can generate a coincidence signal

parameter CoincidenceLength

Coincidence window length in nanoseconds (ns). 16-bit value.

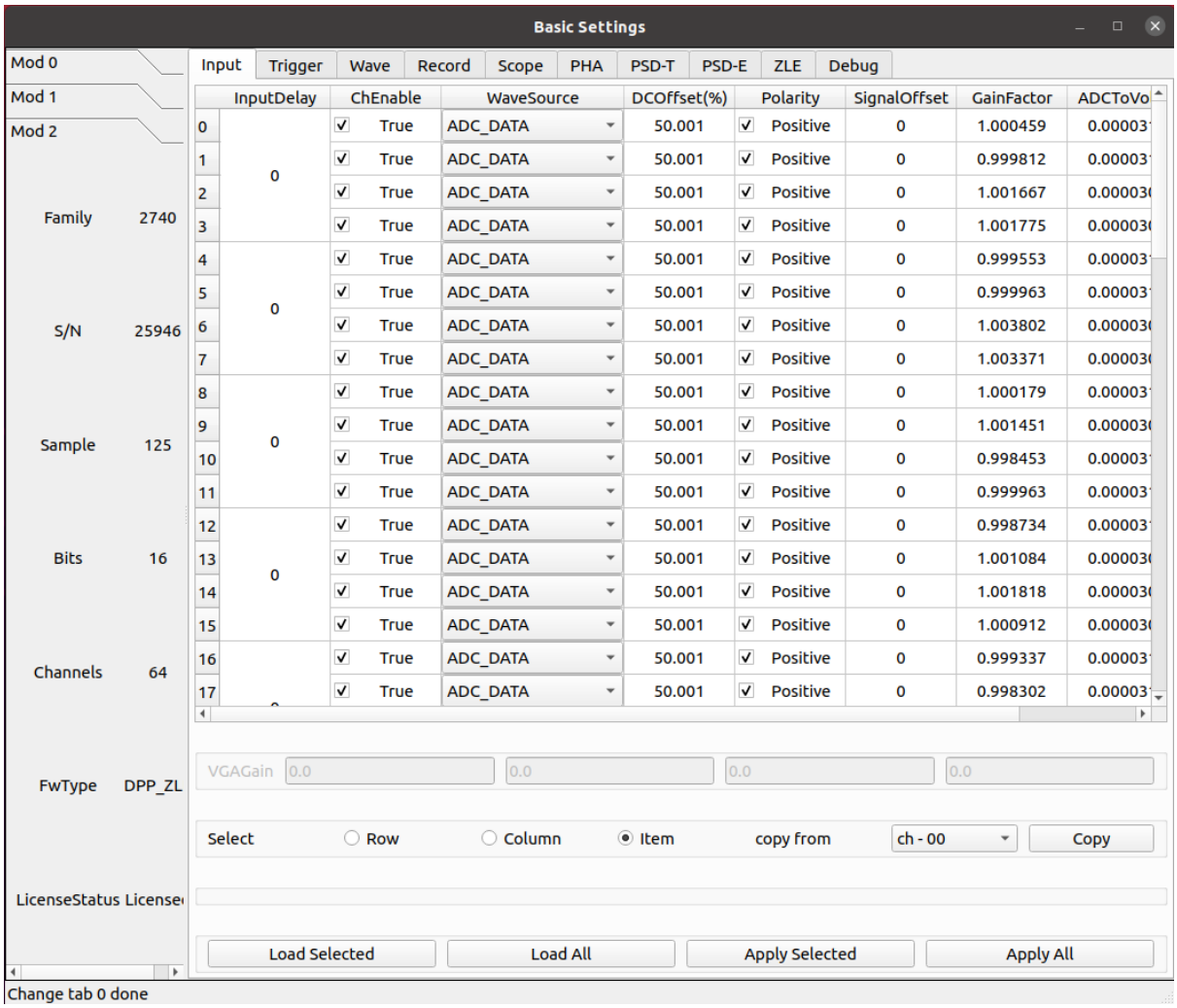
Unit of Measure: ns

CHAPTER 10

ZLE firmware

10.1 Basic

10.1.1 Input



parameter ChGain

Unique to x2730.

Sets the gain of the Variable Gain Amplifiers (VGA). Unit of Measure: dB

parameter InputDelay

Set input delay. The value is set at groups of 4 channels for x2745/x2740.

Unit: sample

parameter ChEnable

Enable the channels for the acquisition, according to the Index. When the channel is disabled, it does not give any data and its self-trigger is off.

- True: The channel is enabled for the acquisition
- False: The channel is disabled for the acquisition

parameter WaveSource

In normal mode, the acquired waveform represents a sequence of ADC samples, resulting from the A/D conversion of the analog input. For test purposes, it is possible to replace the ADC data with internal data generators.

- **ADC_DATA**
  - Data from the ADC (normal operating mode)
- **ADC_TEST_TOGGLE**
  - Toggle between 0x5555 and 0xAAAA (test mode)
- **ADC_TEST_RAMP**
  - 16-bit ramp pattern (test mode)
- **ADC_TEST_SIN**
  - 8-point sine wave test pattern
- **ADC_TEST_PRBS**
  - 16-bit PRBS generated by a 23-bit PRBS pattern generator (test mode)
- **Ramp**
  - Data from a ramp generator. It is actually a 16-bit field, where the 6 most significant bits identify the channel and the 10 less significant bits are the samples of a ramp from 0x000 up to 0x3FF (i.e. 0 to 1023). It is so a 10-bit ramp with offset given by “channel*1024”. For channel 0, it is a counter from 0 to 1023; for channel 1, it is a counter from 1024 to 2047, and so on
- **IPE**
  - Not implemented
- **SquareWave**
  - Internally generated programmable square wave

parameter DCOffset

A constant DC offset (controlled by a 16-bit DAC) is added to the analog input, individually for each channel, to adjust the position of the signal baseline (that is the "zero volt" of the analog input) within the dynamic range of the ADC. Because of the tolerance of the components, it is necessary to calibrate the offset DAC. The calibration is done by factory testing and normally it is not necessary to recalibrate the digitizer. It is however possible to perform a new calibration. The calibration parameters are stored in the flash memory of the board and loaded at power on. They are automatically applied by the internal logic every time the DCoffset parameter is written or read. DCoffset is expressed as a NUMBER number, in percent of the full-scale. When the DCoffset is 0, the baseline of the input signal is at 0 ADC counts. When the DCoffset is 100, the baseline of the input signal is at $2^{NBIT}-1$ ADC counts.

Unit of Measure: %

parameter Polarity

Allows to set the polarity of the input pulse.

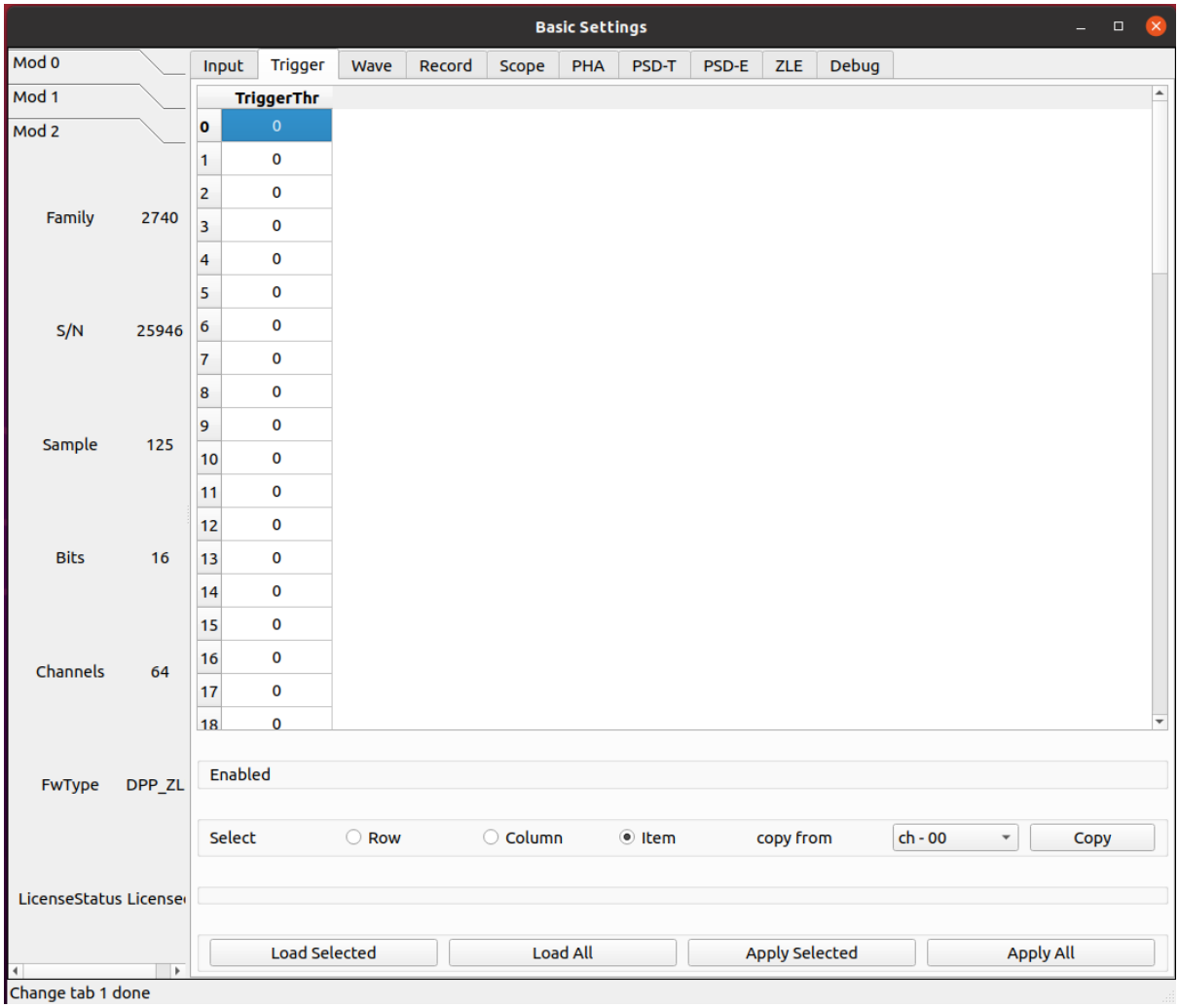
- **Positive**
  - Positive polarity
- **Negative**
  - Negative polarity

parameter VGAGain

Unique to x2745.

Set the gain of the variable gain amplifier (VGA) in increments of 0.5 dB. Parameter settings are grouped every 16 channels, with 64 channels divided into 4 groups. The minimum can be set to 0 and the maximum to 40.

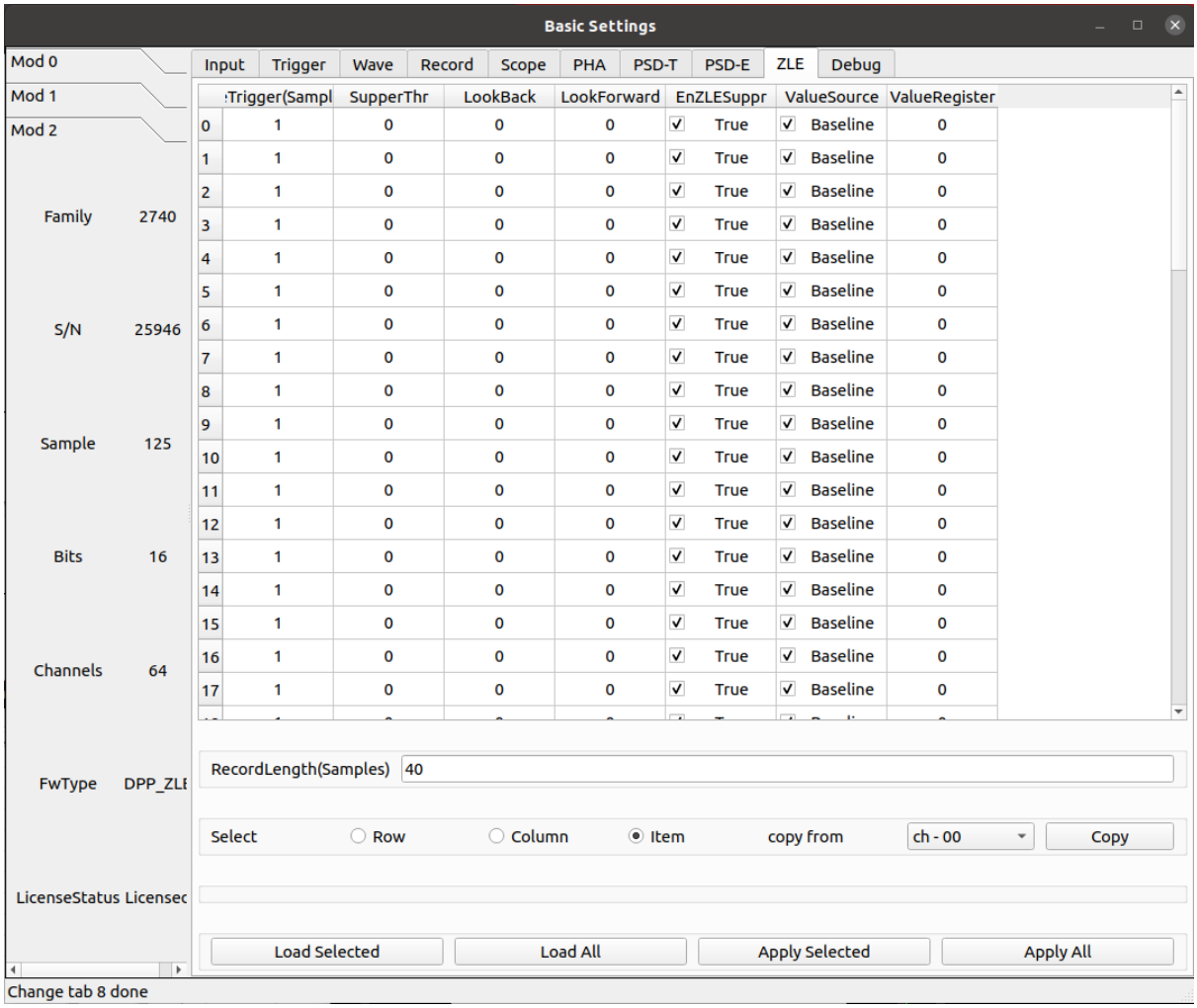
10.1.2 Trigger



parameter TriggerThr

Each channel of the digitizer has a digital leading-edge discriminator with programmable threshold able to self-trigger on the input pulses and generate a self-trigger signal (or an overthreshold signal) feeding the internal trigger logics or digitizer outputs. This parameter sets the trigger threshold. Typically, the value is relative to the baseline of the signal and the threshold is a 17-bit signed NUMBER number; in this case, the threshold automatically follows the baseline when the DCoffset parameter changes. Sometimes, it is preferable to set an absolute value for the threshold, referred to the ADC range. In this case, the threshold is unsigned NUMBER number.

10.1.3 ZLE



parameter PreTrigger

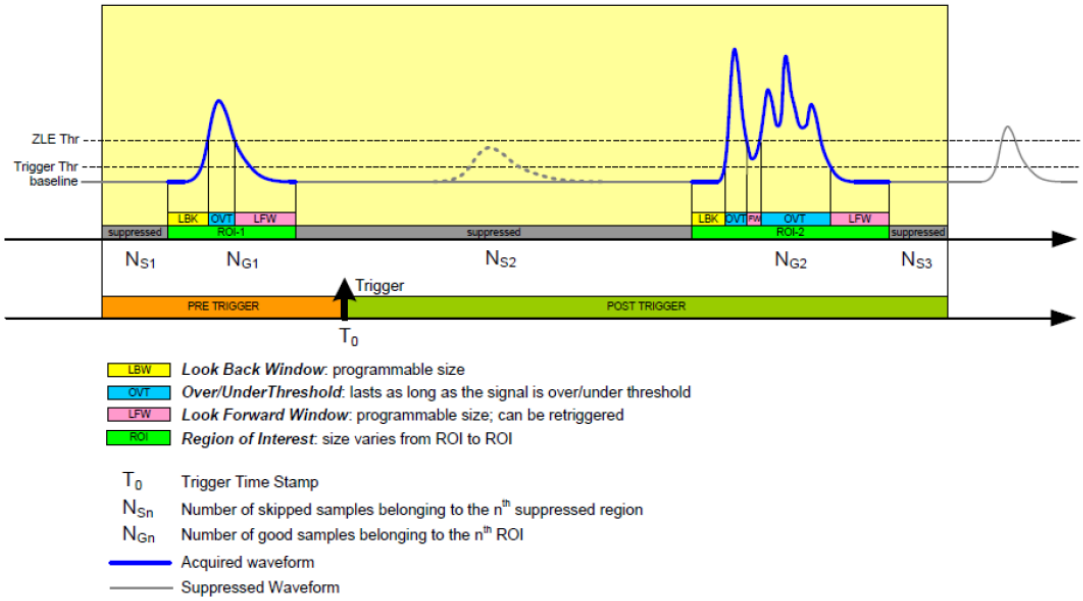
Number of samples coming before the position of the trigger in the waveform (i.e. size of the pre-trigger window).

Unit of Measure: Samples

parameter SupperThr

The ZLE algorithm further checks whether the ZLE suppression threshold (ZLESupprThreshold) is crossed and saves the overthreshold samples (under threshold in case of negative polarity), which are referred as "good" samples. In addition, also a programmable number of samples can be acquired before and after the over/under threshold (ChLookBackSamples and ChLookForwardSamples respectively). "Skipped" samples are discarded; only a word reporting the number of skipped samples is present in the final data. This parameter allows to set the ZLE data suppression threshold.

The figure below shows the parameters of the ZLE algorithm. The signal recording starts ChLookBackSamples samples before the ZLE threshold crossing (LBW in the picture) and stops ChLookForwardSamples samples after the crossing in the opposite direction (LFW in the picture). The recorded regions define the ROI (region of interest) of the algorithm. Outside the ROI the algorithm reports the number of skipped samples.



parameter LookBack

This parameter allows to set the ZLE look-back samples.

Unit of Measure: Samples

parameter LookForward

This parameter allows to set the ZLE look-forward samples.

Unit of Measure: Samples

parameter EnZLESuppr

When this bit is set, no ZLE data reduction is applied.

- **True**
  - ZLE Data reduction is enabled.
- **False**
  - ZLE Data reduction is disabled.

parameter ValueSource

Selection of the default sample value in the payload (baseline or register).

- **Baseline**
  - Default sample value is Baseline.
- **Register**
  - Default sample value is the value.

parameter ValueRegister

ZLE default sample value register.

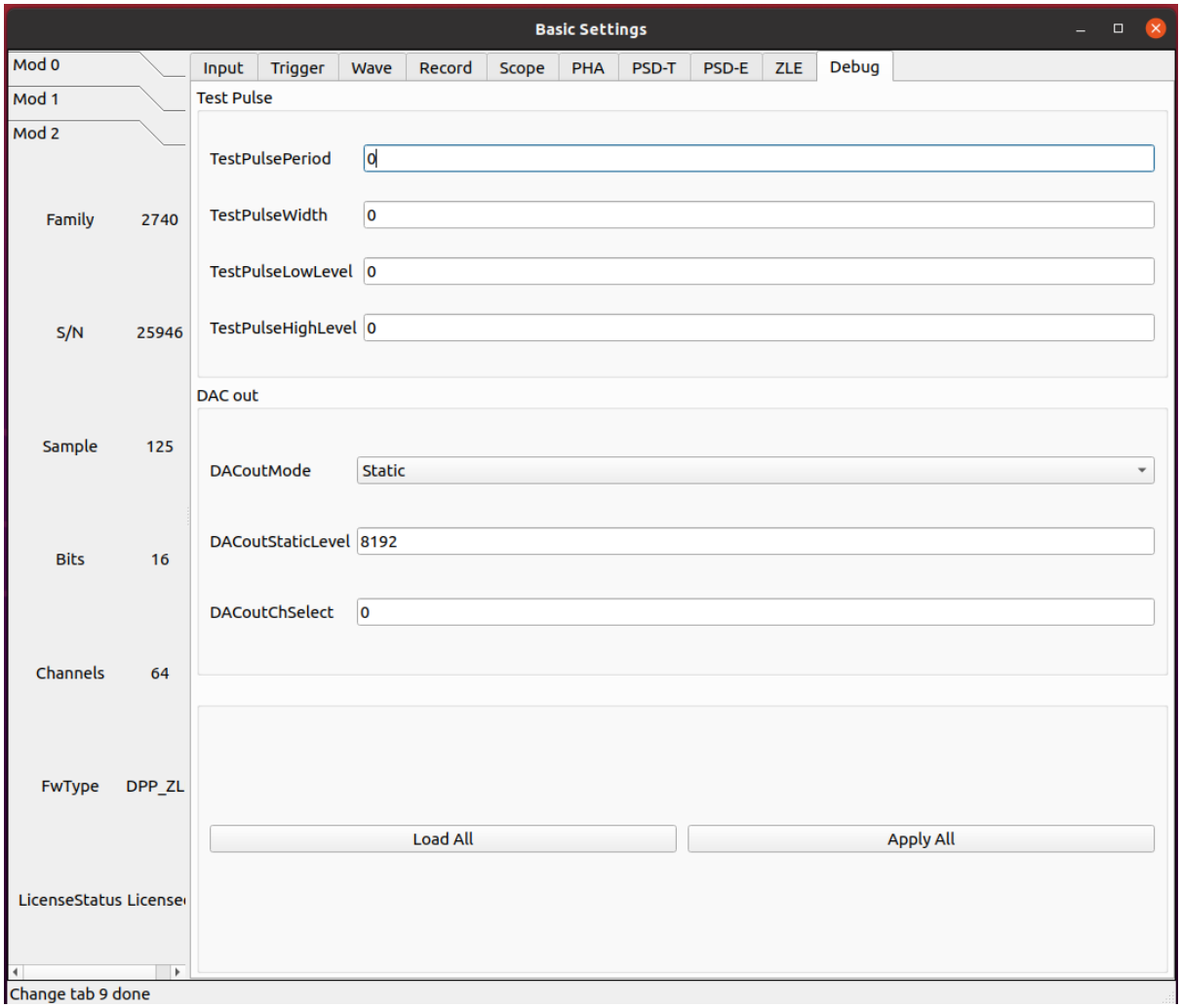
Unit of Measure: ADC counts

parameter RecordLength

The waveform Size (i.e. size of the acquisition window). The actual size of the waveform will be automatically rounded to the closest allowed value. It is possible to get the exact size by reading back the parameter. The record length in time depends on wave resolution.

Unit of Measure: Samples

10.1.4 Debug



parameter TestPulsePeriod

The Test Pulse is a programmable square wave that can be used as an internal periodic trigger (mainly for test purposes) or to generate a logic test pulse (TTL or NIM) on the TRGOUT and GPIO outputs. This parameter sets the period of the Test Pulse.

Unit of Measure: ns

parameter TestPulseWidth

Width of the Test Pulse (time that the signal stays high = 1).

Unit of Measure: ns

parameter TestPulseLowLevel

Low level of the Test Pulse expressed in ADC counts

Unit of Measure: ADC counts

parameter TestPulseHighLevel

High level of the Test Pulse expressed in ADC counts

Unit of Measure: ADC counts

parameter DACoutMode

Selects the signal type to be sent in output on the front panel DAC connector.

- **Static**
  - DAC output stays at a fixed level, given by the DACoutStaticLevel parameter
- **Ramp**
  - The DAC output is driven by a 14-bit counter
- **Sin5MHz**
  - The DAC output is a sine wave at 5 MHz with fixed amplitude
- **Square**
  - Square wave with period and with set by TestPulsePeriod and TestPulseWidth and amplitude between TestPulseLoweLevel and TestPulseHighLevel.
- **IPE**
  - Not implemented
- **ChInput**
  - The DAC reproduces the input signal received by one input channel, selected by the DACoutChSelect parameter
- **MemOccupancy**
  - Level of the memory occupancy (not yet implemented)
- **ChSum**
  - The DAC reproduces the “analog” sum of all the digitizer inputs (not yet implemented)
- **OverThrSum**

  - The DAC output is proportional to the number of channels that are currently above the threshold

parameter DACoutStaticLevel

When DACoutMode = Static, this parameter sets the 14-bit level of the DAC static output.

parameter DACoutChSelect

When DACoutMode = ChInput, the DAC reproduces the input signal of the channel selected by this parameter.

parameter IPEAmplitude

The new digitizers are equipped with an Internal Pulse Emulator capable of generating exponential pulses. This parameter determines the amplitude of the pulse.

Unit of Measure: ADC counts

parameter IPEBaseline

Sets the offset of the exponantial pulses generated by the Internal Pulse Emulator.

Unit of Measure: ADC counts

parameter IPEDecayTime

Sets the decay time of the exponantial pulses generated by the Internal Pulse Emulator.

Unit of Measure: ns

parameter IPERate

Sets the rate of the exponantial pulses generated by the Internal Pulse Emulator.

Unit of Measure: Hz

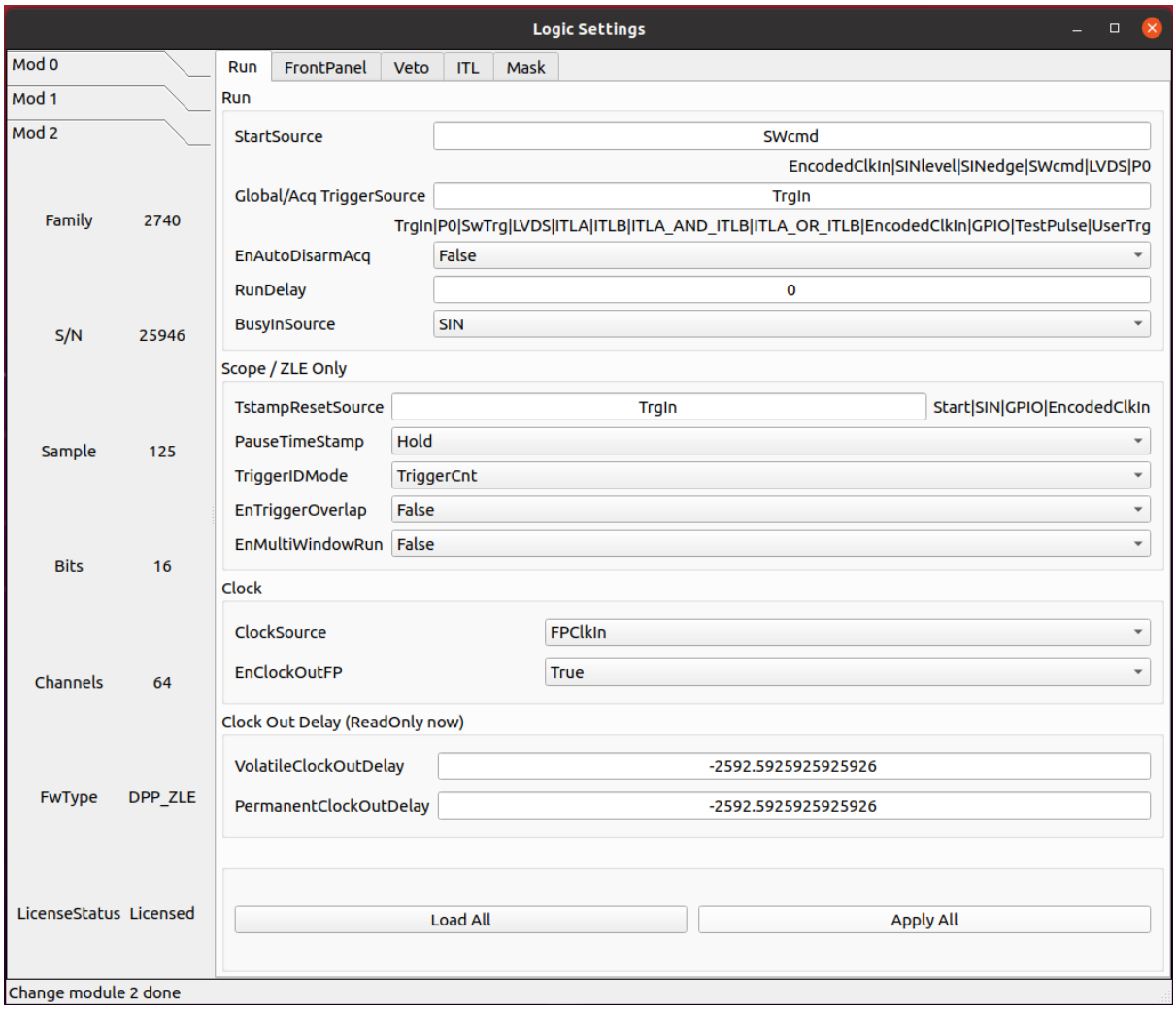
parameter IPETimeMode

Selectes the time distribution of the Internal Pulse Emulator.

- **ConstantRate**
  - Pulse shapes are constant over time. It is possible to set the frequency using the IPERate parameter
- **Poissonian**
  - The pulse rate follows a Poisson distribution. The average frequency value can be configured using the IPERate parameter

10.2 Logic parameter

10.2.1 Run



parameter StartSource

Defines the source for the start of run. Multiple options are allowed, separated by "|".

- **EncodedClkIn**
  - Start from CLK-IN/SYNC connector on the front panel. This is a 4-pin connector (LVDS signals) used to propagate the reference clock (typ. 62.5 MHz) and a Sync signal. The rising edge of the Sync starts the acquisition, that lasts until the Sync returns low (falling edge).
- **SINlevel**
  - Start from SIN (1=run, 0=stop)
- **SINedge**
  - Start from SIN (rising edge = run; stop from SW)
- **SWcmd**
  - Start from SW
- **LVDS**

  - Start from LVDS

- **FirstTrigger**
  - Start on 1st trigger (stop from SW)
- **P0**
  - Start from P0 (backplane)

parameter AcqTriggerSource

Defines the source for the Acquisition Trigger, which is the signal that opens the acquisition window and saves the waveforms in the memory buffers. Multiple options are allowed, separated by "|".

- **TrgIn**
  - Front Panel TRGIN
- **P0**
  - Trigger from P0 (backplane)
- **SwTrg**
  - Software trigger
- **LVDS**
  - LVDS trgin
- **ITLA**
  - Internal Trigger Logic A: combination of channel self-triggers
- **ITLB**
  - Internal Trigger Logic B: combination of channel self-triggers
- **ITLA_AND_ITLB**
  - Second level Trigger logic making the AND of ITL A and B
- **ITLA_OR_ITLB**
  - Second level Trigger logic making the OR of ITL A and B
- **EncodedClkIn**
  - Not implemented (encoded CLK-IN trigger)
- **GPIO**
  - Front Panel GPIO
- **TestPulse**
  - Internal Test Pulse
- **UserTrg**
  - User custom trigger source

parameter EnAutoDisarmAcq

When enabled, the Auto Disarm option disarms the acquisition at the stop of run. When the start of run is controlled by an external signal, this option prevents the digitizer to restart without the intervention of the software.

- **True**
  - The acquisition is automatically disarmed after the stop. It is therefore necessary to rearm the digitizer (with the relevant command sent by the software) before starting a new run.
- **False**
  - The acquisition is not disarmed after the stop. Multiple transition of the start signal will produce multiple runs.

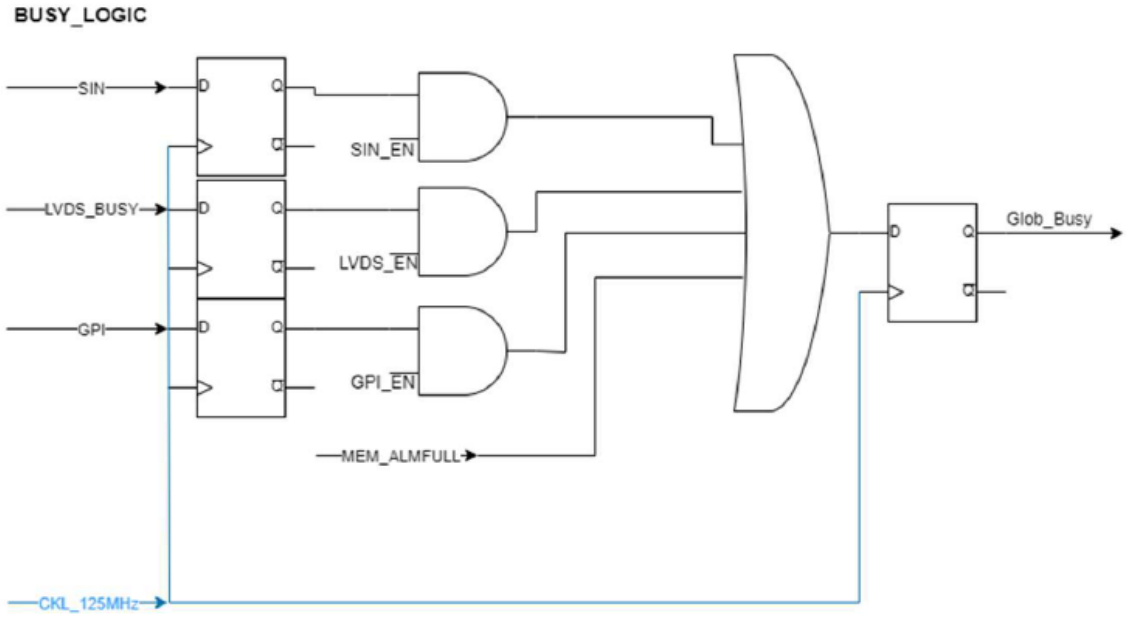
parameter RunDelay

When the start of run is controlled by a RUN signal that is propagated in daisy chain between the boards (for instance through the ClkIn- ClkOut or SIN-GPIO sync chain), it is necessary to compensate for the propagation delay and let the boards start exactly at the same time. The RunDelay parameter allows the start of the acquisition to be delayed by a given number of clock cycles with respect to the rising edge of the RUN signal. Assuming that the propagation delay is 2 cycles, the RunDelay setting will be 0 for the last board in the chain, 2 for the previous one, and so on up 2x(NB-1) for the first one.

Unit of Measure: ns

parameter BusyInSource

In a multi-board system, it might be necessary to prevent one board to accept a new trigger while another board is full and thus unable to accept the same trigger. For this reason, each board can generate a Busy signal to notify that it is unable to get a new trigger. If the busy/veto mechanism has some latency, it is advisable to generate the busy slightly before the digitizer become full. For this purpose, it is possible to assert the busy output when the acquisition memory reaches a certain level of occupancy (internally managed). The OR of the busy signals is typically used to stop the global trigger. It is possible to get the individual busy signals from each board and make an external OR logic or connect the boards with cables to propagate the Busy along the chain. Each board makes an OR between its internal busy and the busy input signal coming from the previous board, thus having a global Busy at the end of the line. This parameter defines the source of the Busy Input (schematized in the figure below)



- **Disabled**
  - The Busy is given by the Internal Busy only (Memory full or almost full)
- **SIN**
  - Busy input from SIN on front panel
- **GPIO**
  - Busy input coming from GPIO on front panel, used as a simple input. It is also possible to use GPIO as a wired OR (bidirectional). In this mode, the Busy line goes high as soon as one board drives it high. All the boards can read the Busy line and use it as a veto for the trigger
- **LVDS**
  - LVDS trgin

parameter TstampResetSource

Defines the source of the timestamp reset. Multiple options are allowed, separated by "|". The timestamp of the board (internal counter running at 125 MHz) is typically reset at the start of each run, which corresponds to the "zero" of the timestamps. In Multi-board systems, the synchronization of the clock and RUN signals allows event data coming from different boards to be merged and correlated by the time stamp. However, it is possible to configure different ways to control the reset of the time stamp in the cases where it is necessary to synchronize it with an external global time stamping system.

- **Start**
  - Time stamp reset at the start of run
- **SIN**
  - SIN input
- **GPIO**
  - GPIO used as input
- **EncodedClkIn**
  - Not implemented (encoded in CLK-IN/SYNC)

parameter PauseTimeStamp

Allows the time stamp to either stop or run during the pauses of the acquisition

- **Hold**
  - The timestamp stops while pausing the acquisition
- **Run**
  - The timestamp runs while pausing the acquisition

parameter TriggerIDMode

The event data packet contains a 24-bit identifier called TriggerID. This can be the total trigger counter, the saved event counter or a pattern coming from the LVDS I/Os.

- **TriggerCnt**
  - The Event triggerID is associated to the total trigger counter. This is a 24-bit counter that is reset at the start of run and increased with every received trigger, including those ones that are not accepted. In this mode, events coming from multiple boards can be correlated by the triggerID that is supposed to by synchronized. There might be gaps due to lost triggers.
- **EventCnt**
  - The Event triggerID is a sequential number of the saved event. In this case, there are no gaps in the sequence, but it is not guaranteed that the trigger ID of multiple boards are aligned, thus it is not possible to use it for data correlation.
- **LVDSpattern**
  - The triggerID is taken from the 16 LVDS inputs at the time of the trigger arrival. The user can provide an external trigger pattern to correlate the event data between boards or even with other readout electronics

parameter EnTriggerOverlap

Allows a trigger occurring within the acquisition window of a previous trigger to be either accepted or rejected. When accepted, the previous window is prematurely closed and the new window immediately opened, without any dead time between the two.

- **True**
  - Triggers with overlapped acquisition windows are accepted.
- **False**
  - Triggers with overlapped acquisition windows are not accepted. The rejected triggers are counted by the total trigger counter, so that the trigger-ID in the event header allows for tracing the rejected triggers.

parameter EnMultiWindowRun

When the acquisition start and stop are controlled by an external "RUN" signal (e.g. feeding SIN), it is possible to configure the digitizer to work in two different modes:

The RUN signal acts as a start (rising edge) and stop (falling edge). Therefore, multiple transitions of the RUN signal cause multiple runs (provided that the Auto Disarm option is disabled). At every start of run, the timestamp is reset, and all the statistics and counters are cleared. Typically, the software produces different output files.

The RUN signal acts as "enable" of the acquisition: once the digitizer has been armed, the first rising edge of RUN starts the acquisition. When RUN goes down, the acquisition is "paused" rather than stopped. This means that all data and statistics are frozen, and the timestamp can be either stopped or left running, depending on the PauseTimeStamp parameter. The RUN signal can toggle multiple times within the same acquisition. The stop of the acquisition will be done by a software command. It is necessary to disarm and rearm the acquisition before starting a new run with the rising edge of the RUN signal.

- **True**
  - MultiWindow run is enabled. The RUN signal acts as start (first rising edge) and pause (subsequent falling edges) for the acquisition. The stop of the acquisition is always given by a software command
- **False**

  - The RUN signal acts as start and stop for the acquisition

parameter ClockSource

This is the source of the system clock. Multiple options are not allowed

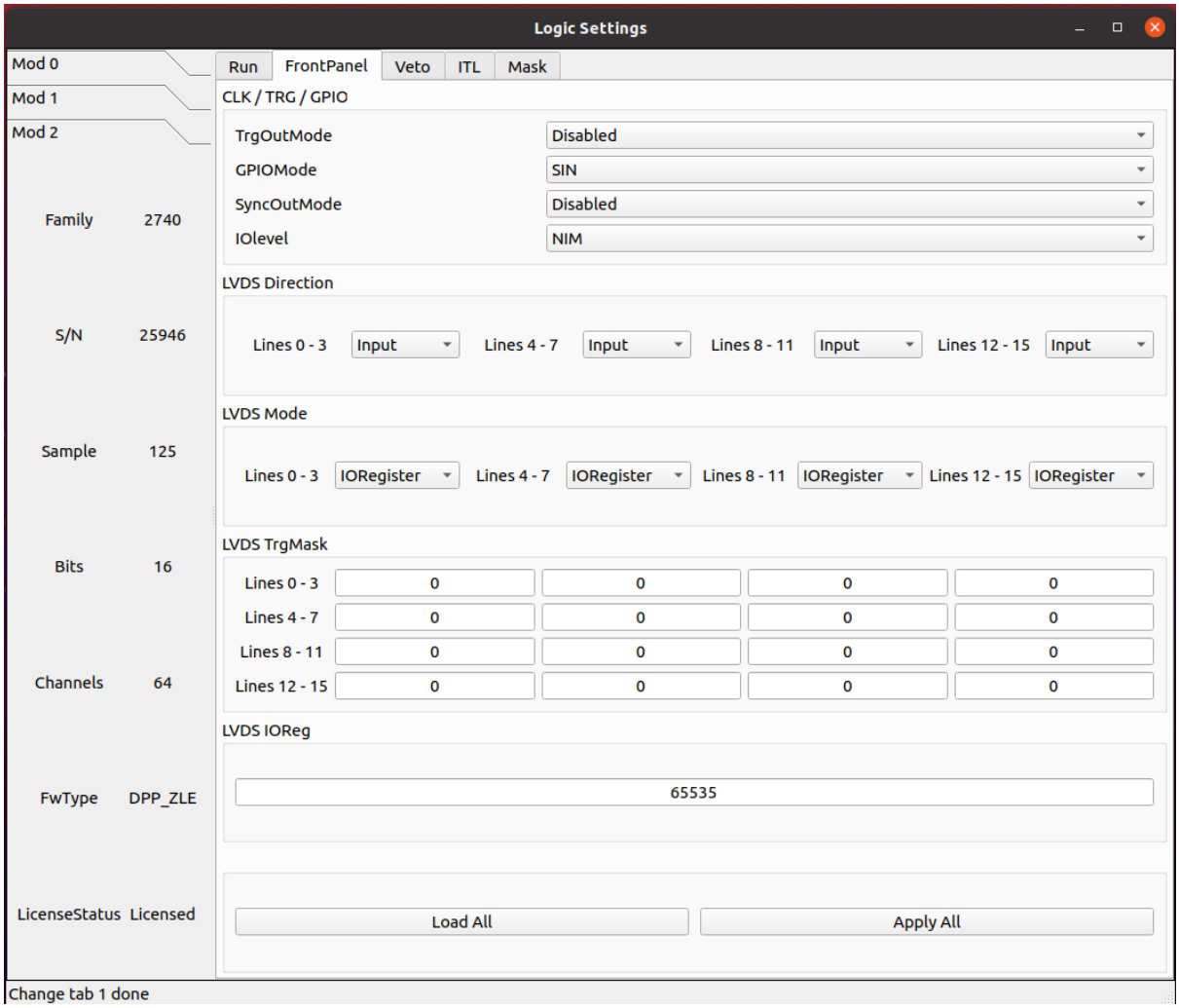
- **Internal**
  - Local oscillator, 62.5 MHz
- **FPClkIn**
  - Front Panel Clock input

parameter EnClockOutFP

Enables clock output on Front Panel for the daisy chain propagation of the clock between multiple boards.

- **True**
  - Enabled
- **False**
  - Disabled

10.2.2 FrontPanel



parameter TrgOutMode

Selects the signal that is routed to the TRGOUT output. Multiple options are not allowed.

- **Disabled**
  - TRGOUT output disabled
- **TrgIn**
  - Propagation of Front Panel TRGIN (TRGOUT is a replica, with some delay, of the TRGIN signal)
- **P0**
  - Propagation of P0 trigger
- **SwTrg**
  - Software trigger
- **LVDS**
  - LVDS trgin
- **ITLA**
  - Internal Trigger Logic A: combination of channel self-triggers

- **ITLB**
  - Internal Trigger Logic B: combination of channel self-triggers
- **ITLA_AND_ITLB**
  - Second level Trigger logic making the AND of ITL A and B
- **ITLA_OR_ITLB**
  - Second level Trigger logic making the OR of ITL A and B
- **EncodedClkIn**
  - Not implemented (propagation of the Encoded CLK-IN trigger)
- **Run**
  - Propagation of the RUN signal (acquisition start/stop), before applying the delay given by the Run-Delay parameter
- **RefClk**
  - Monitor of the 62.5 MHz clock (used for phase alignment)
- **TestPulse**
  - Internal Test Pulse
- **Busy**
  - Busy of the board
- **UserTrgout**
  - Trgout coming from the User Logic (open FPGA)
- **Fixed0**
  - 0 level signal
- **Fixed1**
  - 1 level signal
- **SyncIn**
  - SyncIn signal
- **SIN**
  - SIN connector signal
- **GPIO**
  - GPIO connector signal
- **LBinClk**
  - Internal Logic B clock signal
- **AcceptTrg**
  - Accepted triggers signal
- **TrgClk**
  - Tigger clock signal

parameter GPIOMode

Selects the signal that is routed to the GPIO, when this is used as output. Multiple options are not allowed. The GPIO on the front panel is a bidirectional signal that can used in three different ways:

As independent board output (each board drives its own GPIO)

As a shared input for the boards: the signal is driven high (= 1) or low (= 0) by an external source and connected in "short circuit" among multiple boards using "T" connectors at the inputs. The GPIO is not internally terminated, thus it is necessary to put a 50 Ohm terminator at the end of the line (last "T" of the chain)

As a shared bidirectional line, making a "wired OR". One or more boards can simultaneously drive the signal high (= 1). If no board drives the GPIO, it remains low (= 0). All boards can read back the signal. It is necessary to put a 50 Ohm terminator at both ends of the line (first and last "T" of the chain). This mode can be used to generate, for instance, the global Busy and Veto logic for multiple boards.

- **Disabled**
  - GPIO disabled
- **TrgIn**
  - Propagation of Front Panel TRGIN (GPIO is a replica, with some delay, of the TRGIN signal)
- **P0**
  - Propagation of P0 trigger
- **SIN**
  - Propagation of SIN
- **LVDS**
  - LVDS trgin
- **ITLA**
  - Internal Trigger Logic A: combination of channel self-triggers
- **ITLB**
  - Internal Trigger Logic B: combination of channel self-triggers
- **ITLA_AND_ITLB**
  - Second level Trigger logic making the AND of ITL A and B
- **ITLA_OR_ITLB**
  - Second level Trigger logic making the OR of ITL A and B
- **EncodedClkIn**
  - Not implemented (propagation of the Encoded CLK-IN trigger)
- **SwTrg**
  - Software trigger
- **Run**
  - Propagation of RUN
- **RefClk**
  - Monitor of the 62.5 MHz clock (used for phase alignment)
- **TestPulse**
  - Internal Test Pulse

- **Busy**
  - Busy of the board
- **UserGPO**
  - GPO coming from the User Logic (open FPGA)
- **Fixed0**
  - 0 level signal
- **Fixed1**
  - 1 level signal

parameter SyncOutMode

In a multi-board system, it can be useful to propagate a synchronous signal together with the clock (to synchronize the start of the run, for example) on CLK OUT front panel connector. This parameter defines which signal must be sent out. Multiple options are not allowed.

- **Disabled**
  - SyncoutMode is disabled
- **SyncIn**
  - SyncIn signal (if provided with clkIn on CLK IN connector)
- **TestPulse**
  - Internal Test Pulse
- **IntClk**
  - Internal 62.5 MHz clock (for test purposes)
- **Run**
  - Propagation of RUN signal
- **User**
  - User customSyncoutMode

parameter IOlevel

Sets the electrical logic level of the LEMO I/Os (TRGIN, SIN, TRGOUT, GPIO).

Note that TRGIN and SIN are internally terminated to 50 Ohm, while GPIO and TRGOUT require the termination to 50 Ohms at the receiver

- **NIM**
  - NIM logic (0 = 0V, 1 = -0.8V, that is -16mA)
- **TTL**
  - Low Voltage TLL logic (0 = 0V, 1 = 3.3V)

parameter LVDSDirection

Assigns the direction to a quartet of LVDS I/Os.

- **Input**
  - The LVDS lines of the relevant quartet are used as input. The relevant LED on the front panel is OFF.
- **Output**
  - The LVDS lines of the relevant quartet are used as output. The relevant LED on the front panel lights-up.

parameter LVDSMode

The digitizer is equipped with 16 LVDS I/Os that can be programmed to be inputs or outputs by groups of 4 (quartets), depending on the LVDSDirection parameter. Once the direction has been selected, it is possible to select the functionality of the LVDS lines, individually for each quartet.

- **SelfTriggers**
  - This option is available only when the LVDS are set as outputs. Each LVDS line can be assigned to a combination of the 64 self-triggers, implemented as a masked OR, where the mask is set by the LVDSTrgMask parameter(16 independent masks, one per LVDS line)
- **Sync**
  - Whatever is the direction of the quartet, the 4 lines are rigidly assigned to specific acquisition signals: 0 = Run 1 = Trigger 2 = Busy 3= Veto It is possible to implement a daisy chain distribution of these signals using one quartet as input and another one as output
- **IORegister**
  - The LVDS lines of the quartet are statically controlled by the LVDSIOReg parameter. Use the SetValue function to set the relevant LVDS lines when programmed as output. Use GetValue to read the status of the LVDS lines when programmed as inputs.
- **User**
  - User custom.

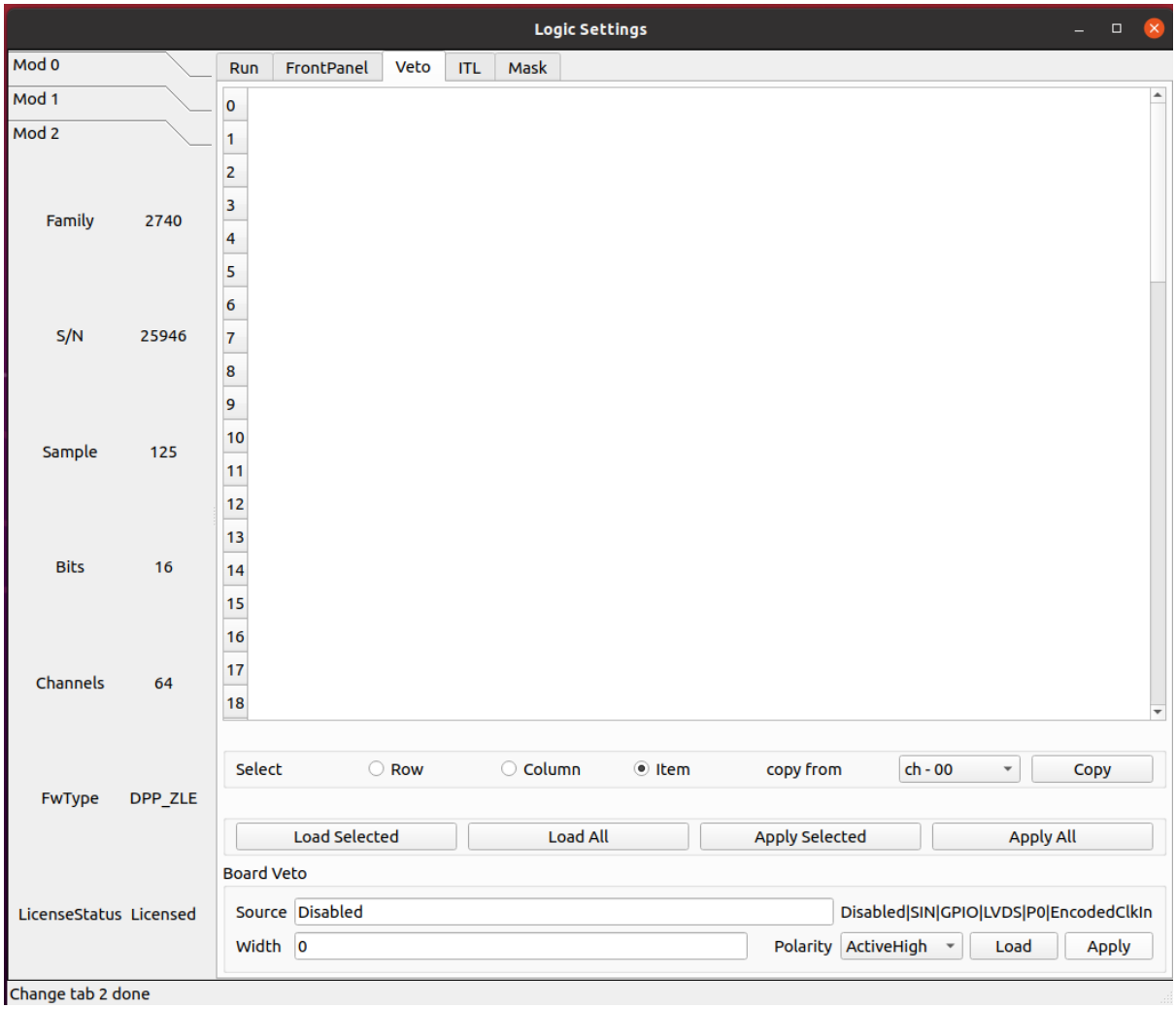
parameter LVDSTrgMask

Each LVDS line can be assigned to a combination of the 64 self-triggers, implemented as a masked OR, where the mask is set by this parameter. There are 16 independent masks, one per LVDS line. Note that the trigger mask assignment does not imply the LVDS direction and mode settings. It is therefore necessary to set the Direction = Output and Mode = SelfTriggers to use the Self-Trigger propagation to the LVDS I/Os.

parameter LVDSIOReg

Set the status of the LVDS I/O for the quartets when they are programmed to be output and Mode = IORegister. This parameter reads out the status of the quartets in the case the LVDS I/O are programmed as inputs (possibly externally driven).

10.2.3 Veto



parameter VetoSource

Defines the source for the Veto, which is the signal that inhibits the acquisition trigger. Multiple options are allowed, separated by "|". The VETO signal can be either active high or low, depending on the VetoPolarity parameter. When active low, it acts as a GATE for the trigger. It is possible to stretch the duration of the VETO by means of the parameter VetoWidth.

- **Disabled**
  - VETO is always OFF
- **SIN**
  - SIN on the front panel
- **GPIO**
  - GPIO on the front panel (used as input)
- **LVDS**
  - LVDS trgin
- **P0**
  - P0 (signal from the backplane)

- **EncodedClkIn**
  - Not implemented (encoded CLK-IN veto)

parameter VetoWidth

Whatever is the source of the VETO signal, it is possible to stretch the duration of the veto up to a given time by means of a re-triggerable monostable. When 0, the monostable is disabled and the veto lasts as long as the selected source is active.

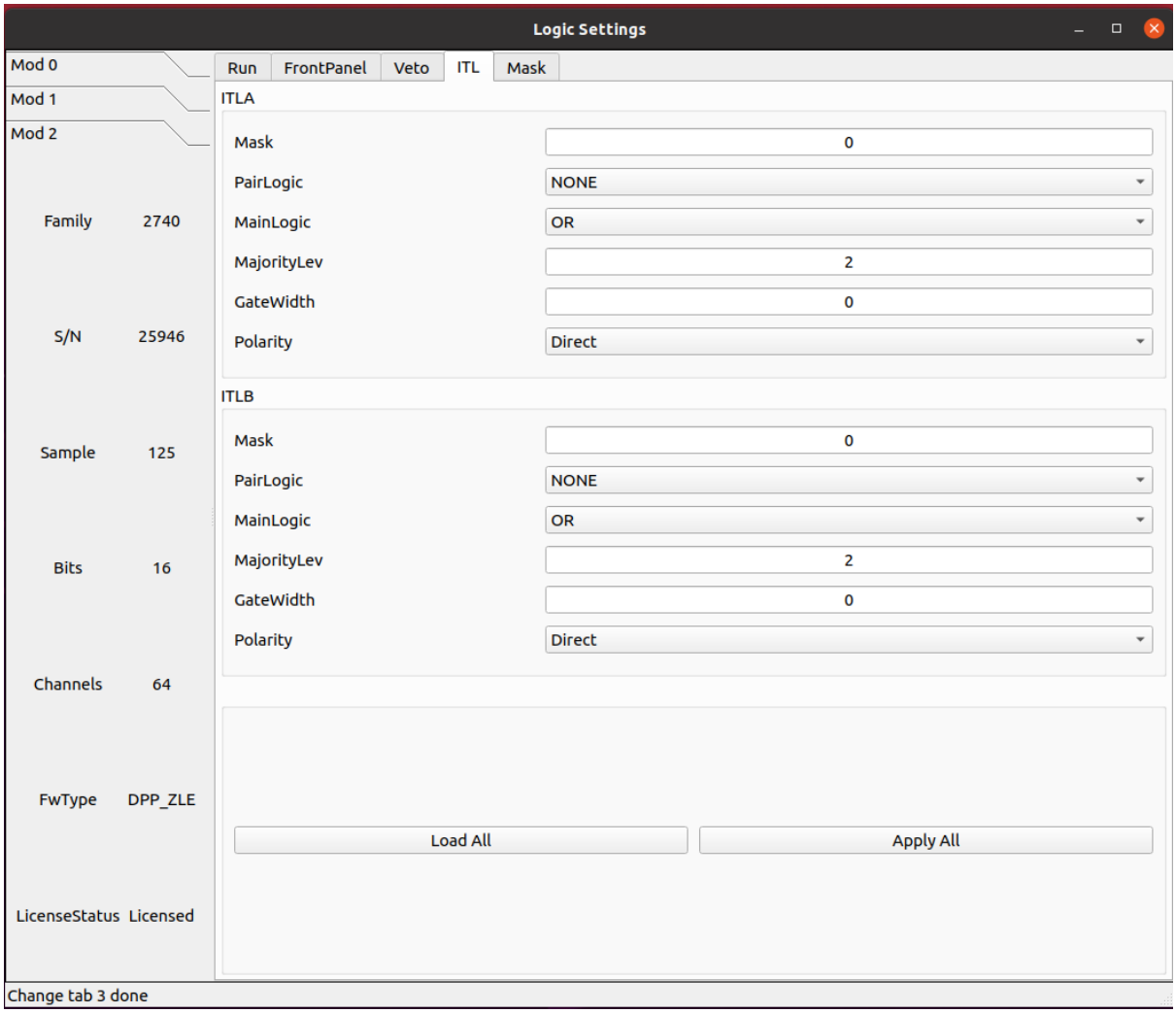
Unit of Measure: ns

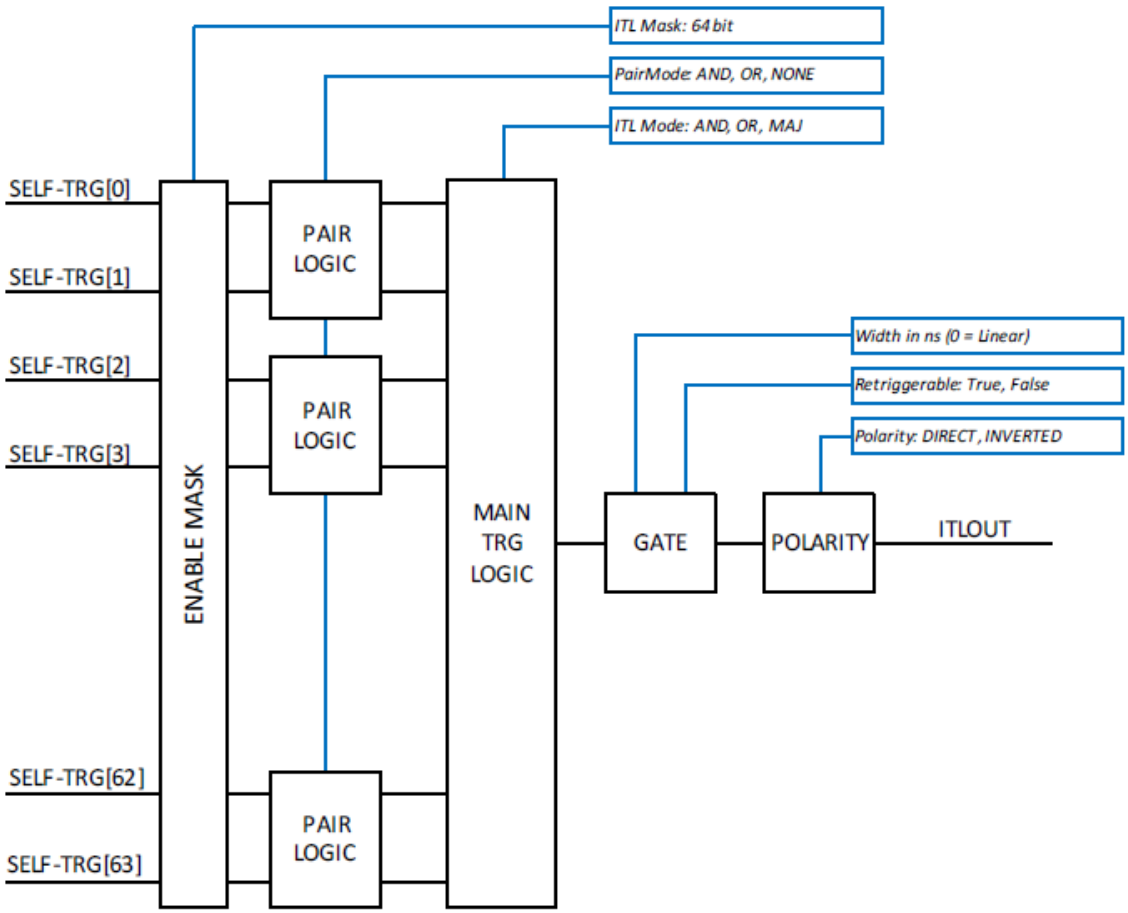
parameter VetoPolarity

Defines the polarity of the Veto

- **ActiveHigh**
  - Veto is active high. The signal acts as an "Inhibit" for the trigger
- **ActiveLow**
  - Veto is active low. The signal acts as a "Gate" the trigger

10.2.4 ITL





parameter ITLA/BMask

Enable Mask at the input of the ITLA/B.

parameter ITLA/BPairLogic

Pairs of channels can be combined with an OR or AND before feeding in the Main trigger Logic. This is typically used in the readout of tubes or scintillator bars, where the two ends are read in coincidence, for instance in position sensitive detectors (the coincidence window will be set by the SelfTriggerWidth parameter). When the AND/OR logic is applied, the two outputs of the Pair Logic blocks are identical.

Note that they are counted twice in the following Majority logic. If the Pair Logic is disabled ("NONE" option), the block is transparent, and the two outputs are just a replica of the inputs.

- **OR**
  - Both Pair Logic Outputs = OR of two consecutive self-triggers
- **AND**
  - Both Pair Logic Outputs = AND of two consecutive self-triggers
- **NONE**
  - Outputs = Inputs

parameter ITLA/BMainLogic

Each channel of the digitizer feature a digital bipolar triangular filter discriminator with programmable rise time and threshold able to self-trigger on the input pulses and generate a self-trigger signal. In DPP Mode, the channels acquire independently, so the channel self-trigger is used locally to acquire a waveform. The trigger threshold is then referred to the bipolar triangular filter, and the threshold crossing arms the event selection. The trigger fires at the zero crossing of the time filter signal. The user can see the derivative trace on the signal inspector. It is also possible to combine all the self-triggers of the board, according to a specific trigger logic. There are two independent logic blocks, ITLA and ITLB. Their output can be used separately to feed, for instance, AcqTrigger and TrgOut, or combined in a second level trigger logic to implement more complex trigger schemes. Therefore, the ITLs can either generate the local acquisition trigger, common to all the channels, for the acquisition of the waveform, or propagate the signal outside, through the TRGOUT, thus making it possible to combine triggers of multiple boards in an external trigger logic, that eventually feeds back the TRGIN of the digitizers. Each ITL is made of an input enable mask (64 bits, one per channel), an optional pairing logic that combines the self triggers of two consecutive channels (e.g. paired coincidence) and the main trigger logic that combines the 64 selftriggers with an OR, AND or Majority logic. The output can be linear (no stretching) or reshaped by a programmable gate generator, either re-triggerable or not and finally programmed for polarity (direct or inverted).

- **OR**
  - ITLOUT = masked OR of channel self-triggers
- **AND**
  - ITLOUT = masked AND of channel self-triggers
- **Majority**
  - ITLOUT = masked Majority of channel self-triggers

parameter ITLA/BMajorityLev

Defines the majority level of the Main Logic of the ITL A/B block. The majority output is calculated at every clock cycle, and it becomes TRUE when Nch >= MajLev, where Nch is the number of self-triggers active in that clock cycle and MajLev is the programmed majority level.

Note that when the Pair Logic is used to combine the self triggers two by two (AND/OR), each pair produces two identical signals that will be counted twice in the majority level.

parameter ITLA/BGateWidth

Width of the gate generator at the output of the ITLA/B block.

Unit of Measure: ns

parameter ITLA/BPolarity

Polarity of the gate generator output.

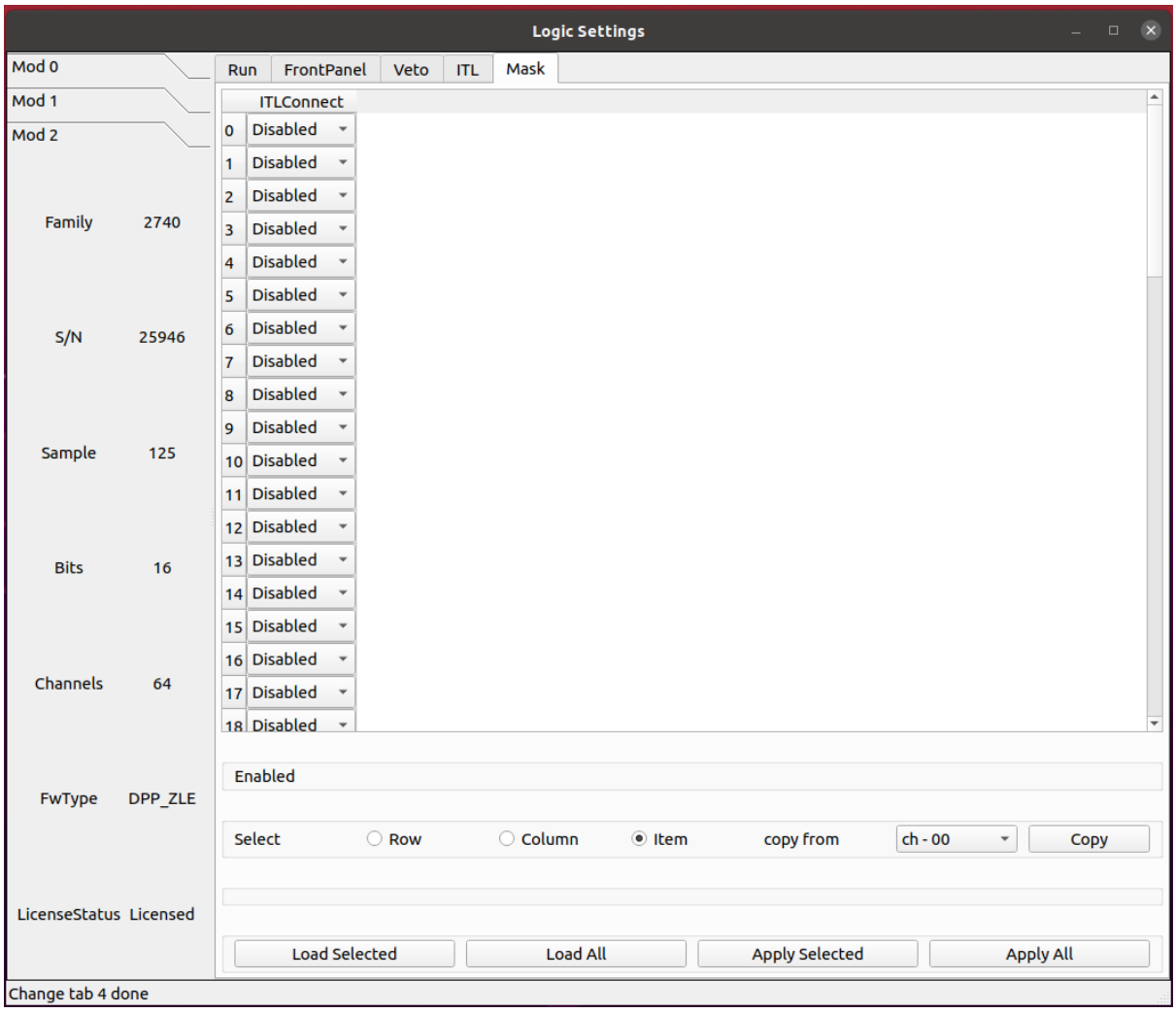
- **Direct**
  - Direct polarity
- **Inverted**
  - Inverted polarity

parameter ITLA/BEnRetrigger

Set the ITLA/B to be retriggerable.

- **True**
  - The ITLA/B is retriggerable
- **False**
  - The ITLA/B is not retriggerable

10.2.5 mask



parameter ITLConnect

Alternative to ITLAMask, ITLBMask. Determines if the channel partecipate in ITLA or ITLB

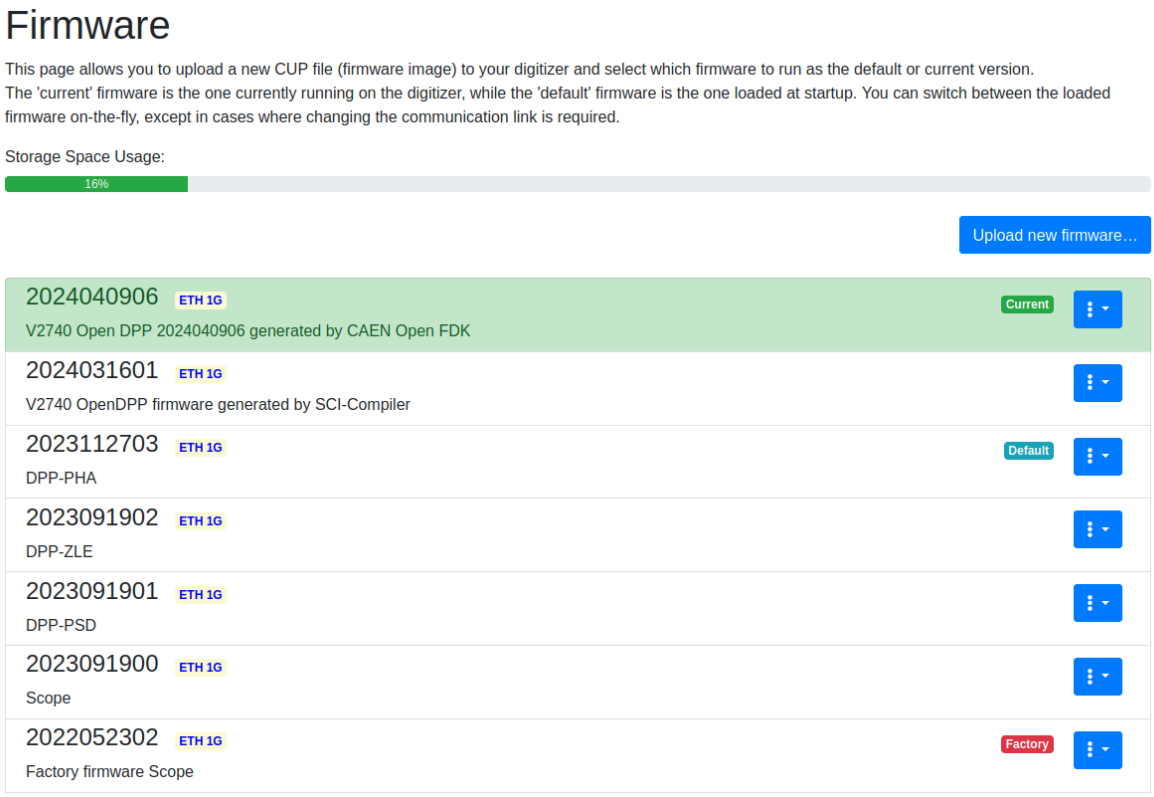
- **Disabled**
  - The channel is disabled
- **ITLA**
  - The channel participates in ITLA logic block
- **ITLB**

- The channel participates in ITLB logic block

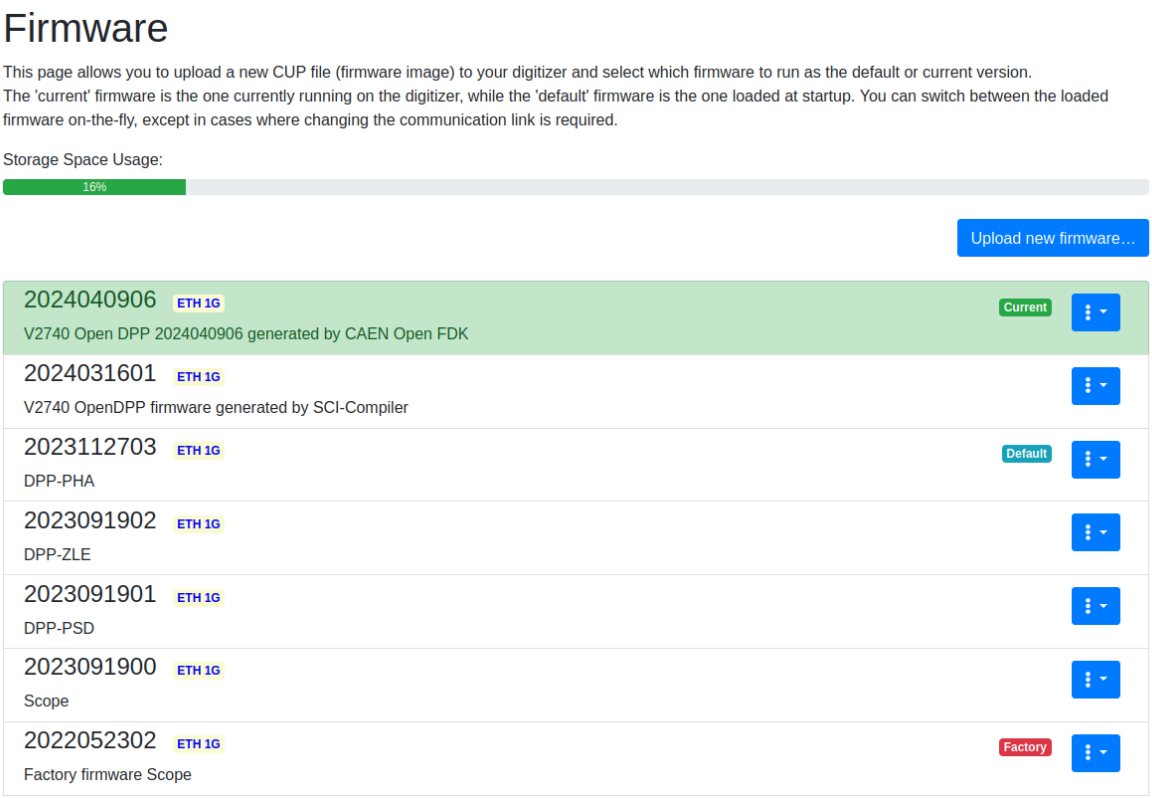
CHAPTER 11

Open FPGA

CAEN open FPGA includes two development modes, one is the graphical software SCI-Compiler, which is suitable for people without FPGA programming foundation to develop firmware; The other is a non-public Open-FDK, suitable for professional FPGA developers. The firmware developed in these two modes can be distinguished by the information of the firmware. The firmware we released is developed for Open-FDK.



The firmware developed by SCI-Compiler requires the running acquisition module to have a SCI-Compiler Runtime license, otherwise it can only run for 30 minutes. The 30 minute testing time can be regained by restarting.



The firmware developed by Open-FDK does not have any license requirements for the acquisition modules.

CHAPTER 12

UserDPP firmware

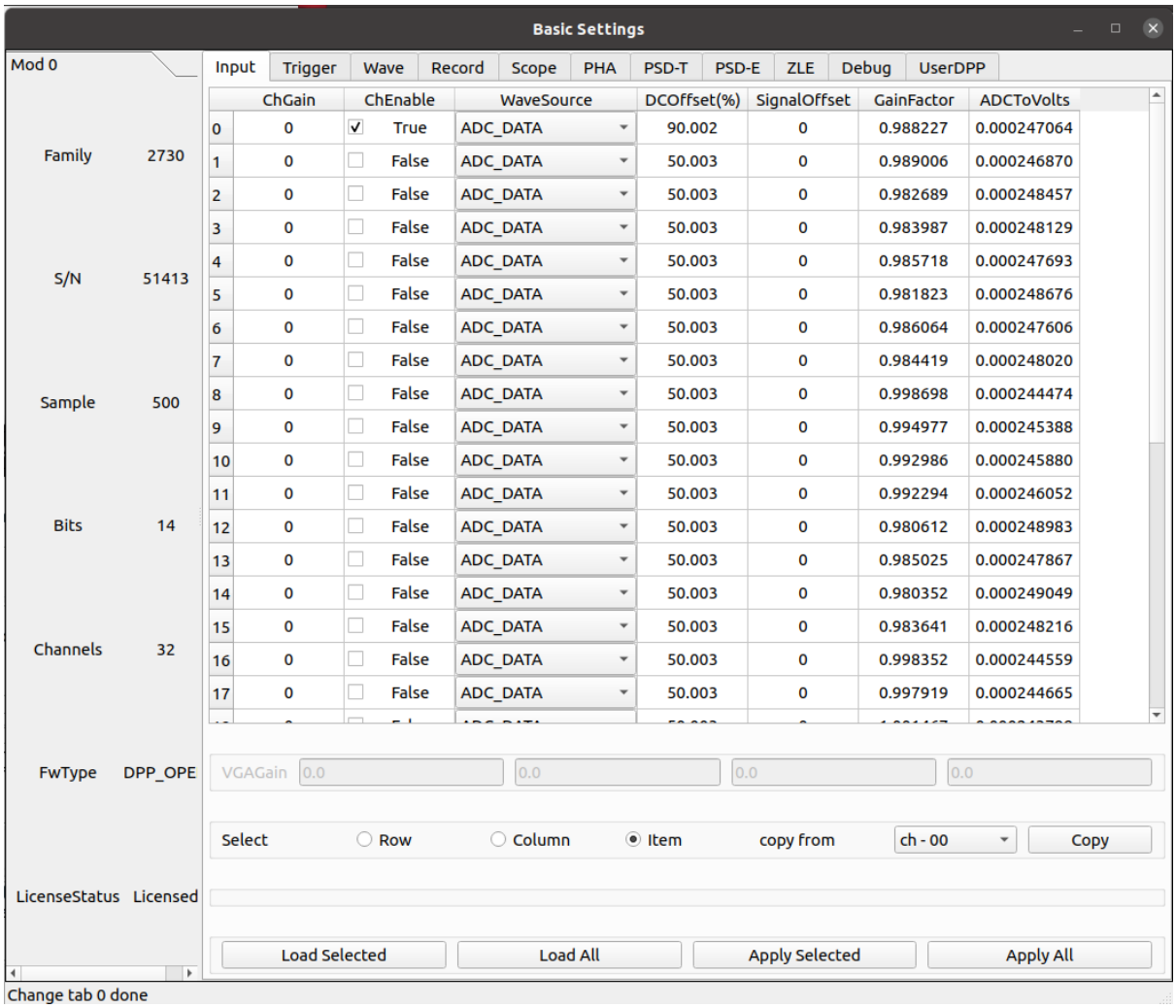
The firmware developed by Hongyi Wu will soon be updated in this section of the manual.

The firmware of 2730 has an FPGA running clock of 125 MHz. Add up the four sampling points of each clock as one sampling point for processing.

Coming soon···

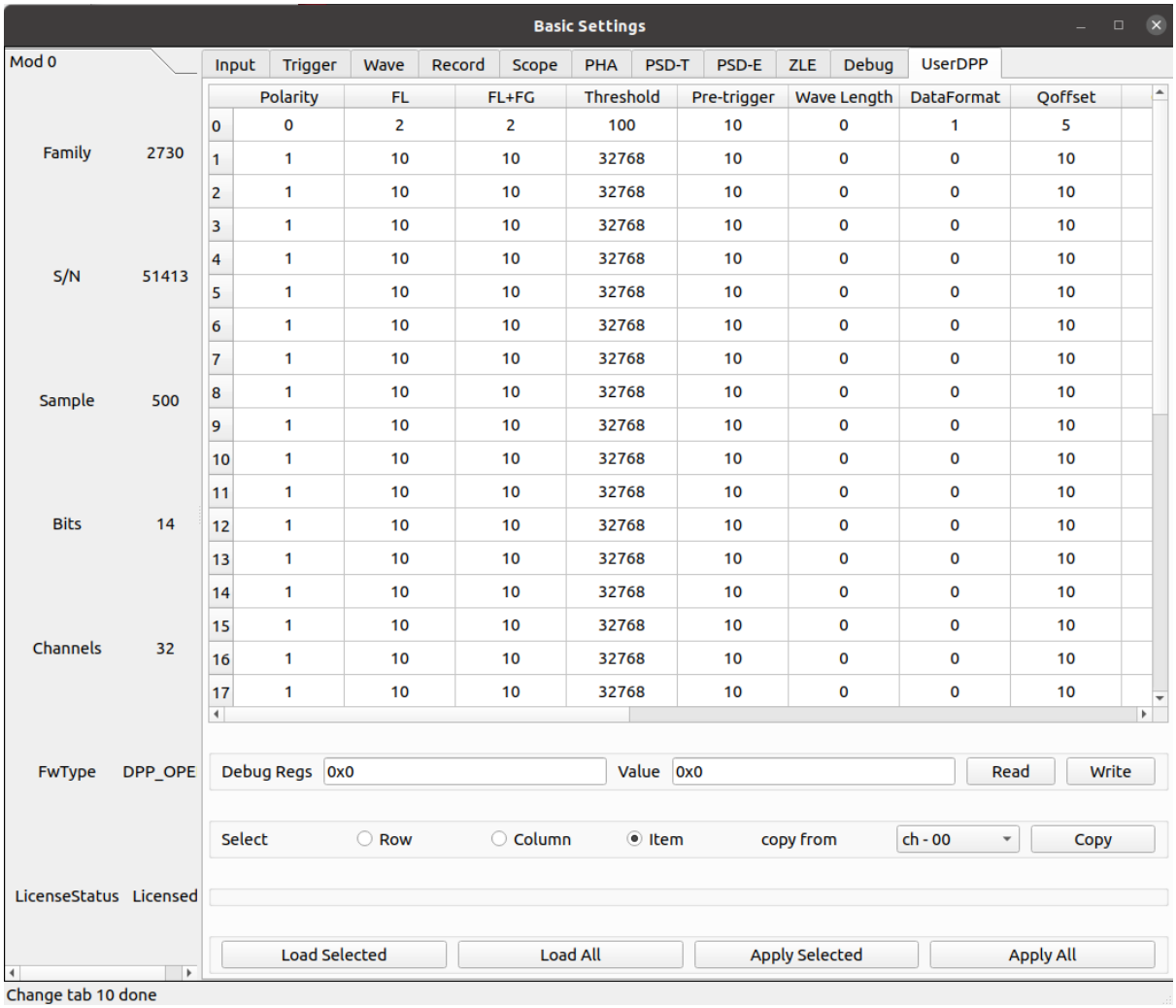
12.1 Basic

12.1.1 Input



	ChGain	ChEnable	WaveSource	DCOffset(%)	SignalOffset	GainFactor	ADCToVolts
0	0	True	ADC_DATA	90.002	0	0.988227	0.000247064
1	0	False	ADC_DATA	50.003	0	0.989006	0.000246870
2	0	False	ADC_DATA	50.003	0	0.982689	0.000248457
3	0	False	ADC_DATA	50.003	0	0.983987	0.000248129
4	0	False	ADC_DATA	50.003	0	0.985718	0.000247693
5	0	False	ADC_DATA	50.003	0	0.981823	0.000248676
6	0	False	ADC_DATA	50.003	0	0.986064	0.000247606
7	0	False	ADC_DATA	50.003	0	0.984419	0.000248020
8	0	False	ADC_DATA	50.003	0	0.998698	0.000244474
9	0	False	ADC_DATA	50.003	0	0.994977	0.000245388
10	0	False	ADC_DATA	50.003	0	0.992986	0.000245880
11	0	False	ADC_DATA	50.003	0	0.992294	0.000246052
12	0	False	ADC_DATA	50.003	0	0.980612	0.000248983
13	0	False	ADC_DATA	50.003	0	0.985025	0.000247867
14	0	False	ADC_DATA	50.003	0	0.980352	0.000249049
15	0	False	ADC_DATA	50.003	0	0.983641	0.000248216
16	0	False	ADC_DATA	50.003	0	0.998352	0.000244559
17	0	False	ADC_DATA	50.003	0	0.997919	0.000244665

12.1.2 UserDPP



12.2 Logic

CHAPTER 13

Applications

This chapter introduces some experimental application cases, test results, etc.

dB = 20lg(A/B)

- **0 dB**
  - x1
- **6 dB**
  - x2
- **9.5 dB**
  - x3
- **12 dB**
  - x4
- **14 dB**
  - x5
- **15.6 dB**
  - x6
- **16.9 dB**
  - x7
- **18 dB**
  - x8
- **19 dB**
  - x9
- **20 dB**
  - x10
- **21.6 dB**
  - x12

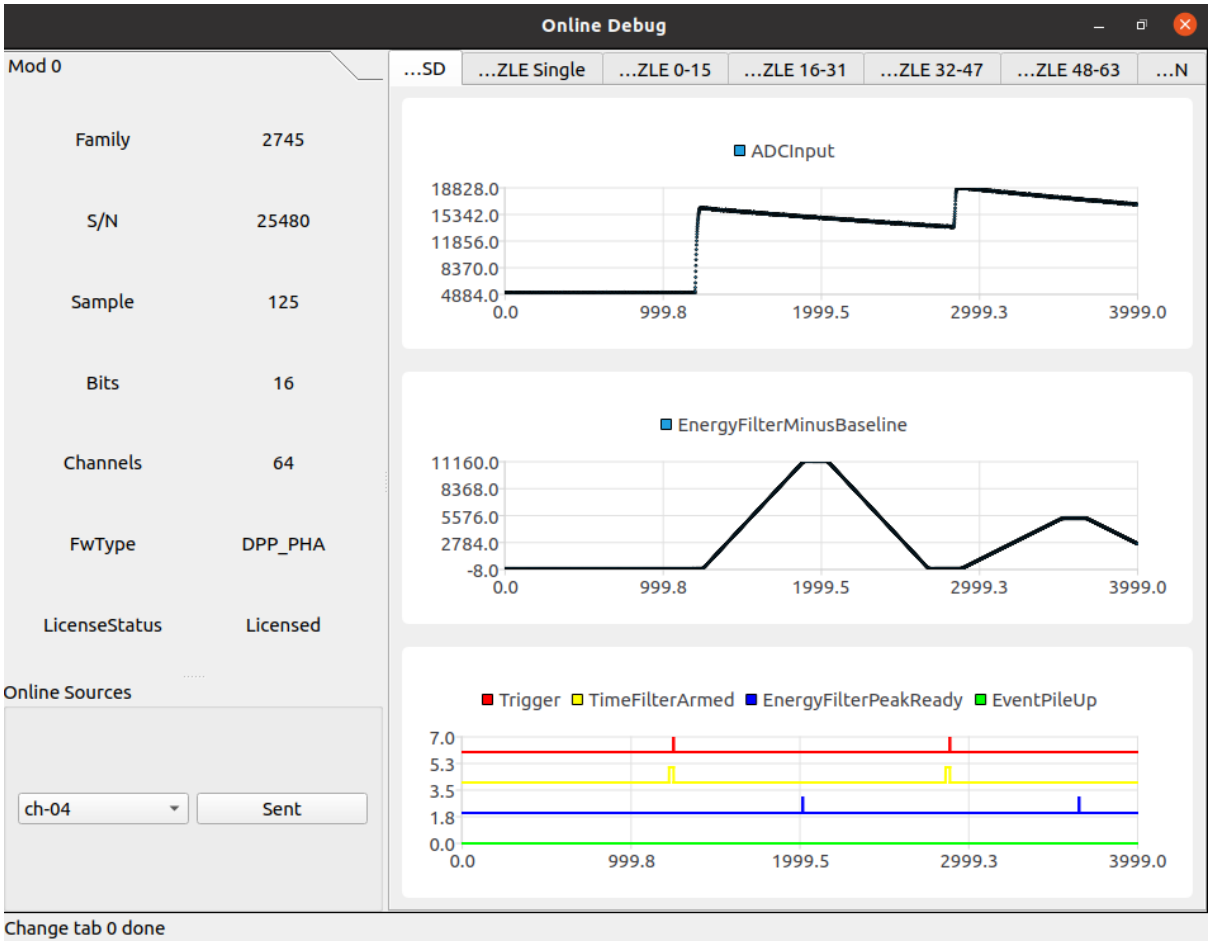
- **22.3 dB**
  - x13
- **22.9 dB**
  - x14
- **24.1 dB**
  - x15
- **25.1 dB**
  - x18
- **26 dB**
  - x20
- **29.5 dB**
  - x30
- **32 dB**
  - x40
- **34 dB**
  - x50
- **35.6 dB**
  - x60
- **36.9 dB**
  - x70
- **38 dB**
  - x80
- **39 dB**
  - x90
- **40 dB**
  - x100

CHAPTER 14

HPGe

- **VGAGain**
  - Adjust the gain to achieve a measurement range of 6 MeV
- **Thre**
  - based on the noise situation
  - about 50
- **PoleZero**
  - based on pre amp parameter
  - about 50us
- **TriggerTriangular**
  - 104 ns
- **BaselineAvg**
  - MediumHigt
- **EnergyRiseTIme**
  - 5104 ns
- **EnergyFlatTop**
  - 1200 ns
- **PeakingPosition**
  - 91%
- **PeakingAvg**
  - OneShot


Online Debug
Mod 0
...SD
...ZLE Single
...ZLE 0-15
...ZLE 16-31
...ZLE 32-47
...ZLE 48-63
...N
Family
2745
S/N
25480
Sample
125
Bits
16
Channels
64
FwType
DPP_PHA
LicenseStatus
Licensed
Online Sources
ch-04
Sent
ADCInput
9740.0
8664.0
7588.0
6512.0
5436.0
0.0
999.8
1999.5
2999.3
3999.0
TimeFilter
665.0
292.3
-80.5
-453.3
-826.0
Trigger
TimeFilterArmed
EnergyFilterPeakReady
EventPileUp
7.0
5.3
3.5
1.8
0.0
Change tab 0 done



Measurement result of x2745

- **dynamic range**
  [0-2MeV]
  - FWHM: 0.15% @ 1332 keV TriggerRate ~ 1000/s
- **dynamic range**
  [0-6MeV]
  - FWHM: 0.16% @ 1332 keV TriggerRate ~ 1000/s
  - FWHM: 0.175% @ 1332 keV TriggerRate ~ 4000/s
  - FWHM: 0.190% @ 1332 keV TriggerRate ~ 10000/s

CHAPTER 15

Si

Coming soon.

CHAPTER 16

UserDPP 应用

16.1 2745 HPGe

16.2 2740 He-3

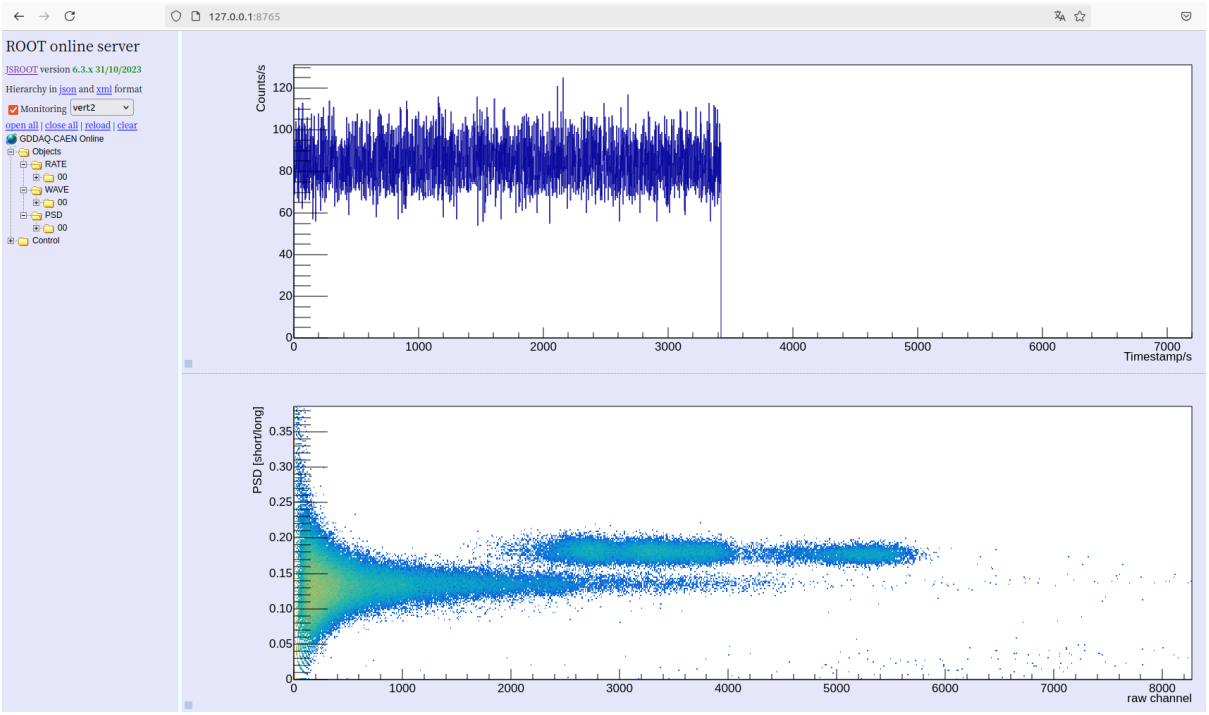
16.3 2740 Si

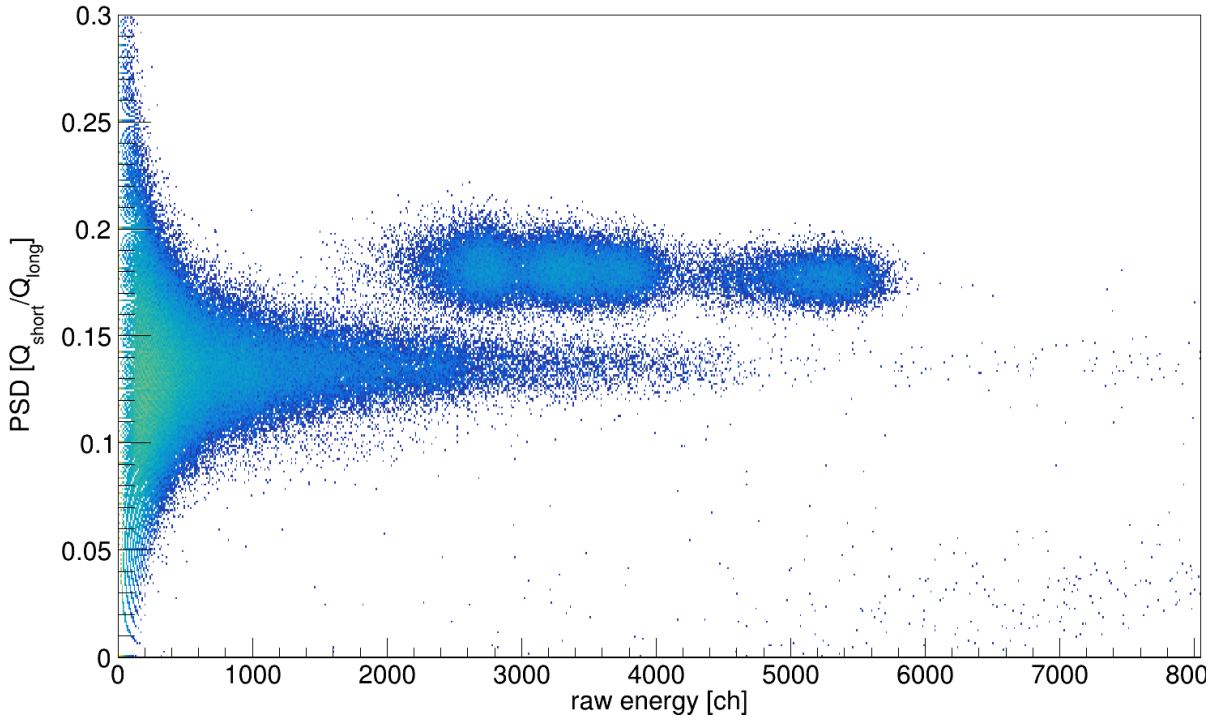
16.4 2730 BaF2

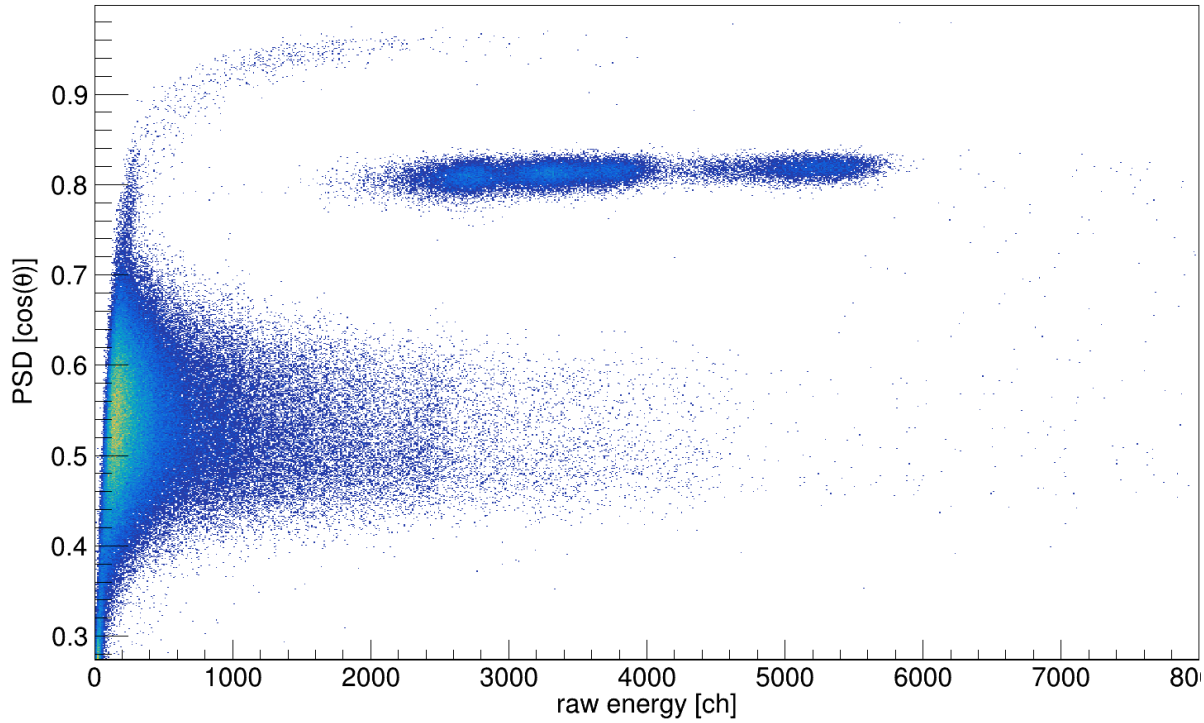
1 inch BaF2 detector

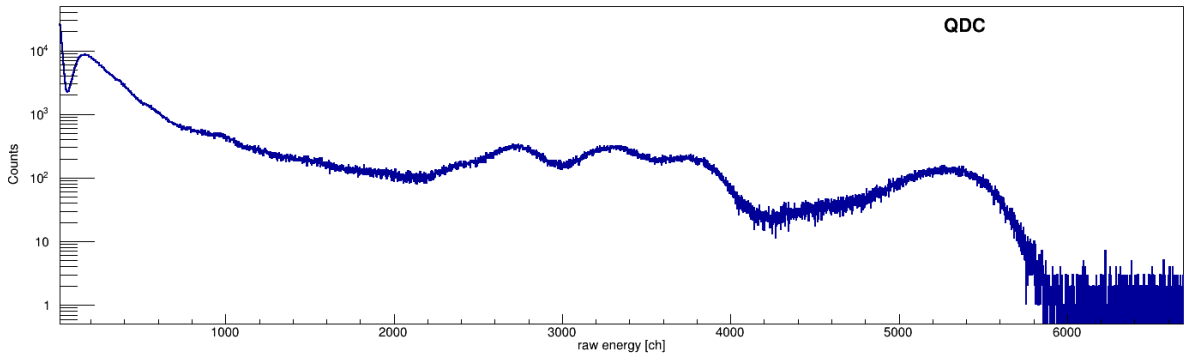
parameter

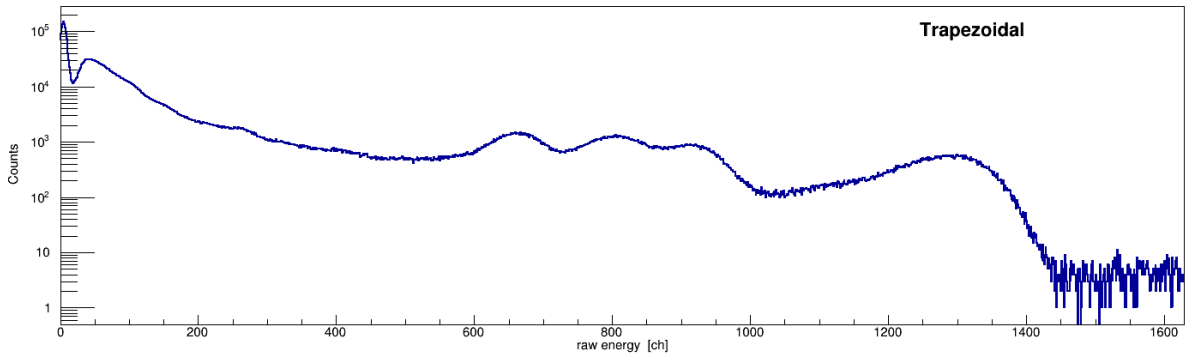
- polarity: 0
- FL: 2
- FL+FG: 2
- Threshold: 100
- Qoffset: 5
- Qshort: 10
- Qlong: 150
- QGainShift: 4
- 2^n BL: 5
- BL Hold: 2000
- RT offset: 10
- RT gate: 80
- XIAOffset: 145
- SL: 140
- SG: 10


127.0.0.1:8765
ROOT online server
JSROOT version 6.3.x 31/10/2023
Hierarchy in json and xml format
Monitoring vert2
open all | close all | reload | clear
GDDAQ-CAEN Online
Objects
RATE
00
WAVE
00
PSD
00
Control
Counts/s
Timestamp/s
PSD [short/long]
raw channel


PSD [Q_short/Q_long]
raw energy [ch]


PSD [cos(θ)]
raw energy [ch]


QDC
Counts
raw energy [ch]


Trapezoidal
Counts
raw energy [ch]

CHAPTER 17

Data analysis

The output timestamp unit of the data conversion program is **ns**

CHAPTER 18

Decoder

The DecodeAndSortAll program is used to convert data recored by different modules one run into a ROOT file. The user's physical analysis is based on the ROOT file generated by this program. The data generated by this program has been arranged in ascending order of timestamp.

The user first needs to modify the definition in the **UesrDefine.hh** file

```
// #define ONLYPHA
// #define ONLYPSD
// #define ONLYZLE
// #define ONLYSCOPE
```

If all modules in the DAQ only use one type of firmware, enable the corresponding definition. If there is no definition of a single firmware, the output data file will default to supporting all types of firmware.

```
#define ROOTFILEPATH "./"       //the path of ROOT file
#define ROOTFILENAME "data"     //the name of ROOT file
// The path and file name for ROOT files
```

```
#define RAWFILEPATH "/home/wuhongyi/"      //Path of raw data
#define RAWFILENAME "data"                 //The file name of the raw data
#define MODNUMBER 2                        //Number of modules used in the chassis
const unsigned short SamplingRate[MODNUMBER] = {500, 125};//Specify the sampling␣
↪rate of each modules separately; 125/500/1000 sampling rates; 0 to skip the␣
↪module
const unsigned short Firmware[MODNUMBER] = {2, 0};//DPP_PHA=0 DPP_ZLE=1 DPP_PSD=2␣
↪DPP_DAW=3 OPEN=4 Scope=5
// Specify the firmware type for each module. If the type is specified incorrectly,
↪ there will be issues decoding the data
```

After modification, execute the following command to compile the program:

```
make clean
make
```

After successful compilation, an executable file **decodeandsort** will be generated, and the program will run as follows:

```
./decodeandsort [RunNnumber]
```

Among them, *[RunNnumber]* is the running number of the file you want to convert.

CHAPTER 19

Event Builder

coming soon…