PRELIMINARY

# Technical Information Manual

Revision n. 7

06 February 2012

**MOD. N6742**

*16+1 CH 12 BIT*

*5 GS/S DIGITIZER*

**MANUAL REV.7**

**NPO:**

**00100/09:6742x.MUTx/07**

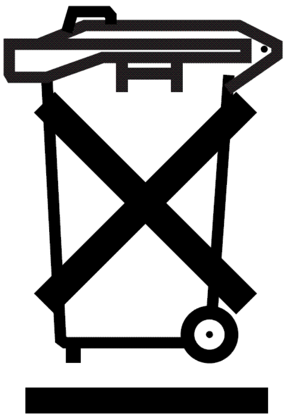
CAEN will repair or replace any product within the guarantee period if the Guarantor declares that the product is defective due to workmanship or materials and has not been caused by mishandling, negligence on behalf of the User, accident or any abnormal conditions or operations.

**CAEN declines all responsibility for damages or injuries caused by an improper use of the Modules due to negligence on behalf of the User. It is strongly recommended to read thoroughly the CAEN User's Manual before any kind of operation.**

*CAEN reserves the right to change partially or entirely the contents of this Manual at any time and without giving any notice.*

## Disposal of the Product

*The product must never be dumped in the Municipal Waste. Please check your local regulations for disposal of electronics products.*



**MADE IN ITALY** : We stress the fact that all the boards are made in Italy because in this globalized world, where getting the lowest possible price for products sometimes translates into poor pay and working conditions for the people who make them, at least you know that who made your board was reasonably paid and worked in a safe environment. (this obviously applies only to the boards marked "MADE IN ITALY", we can not attest to the manufacturing process of "third party" boards).

# TABLE OF CONTENTS

**1. GENERAL DESCRIPTION ........ 7**

1.1. OVERVIEW ........ 7

1.2. BLOCK DIAGRAM ........ 9

**2. TECHNICAL SPECIFICATIONS ........ 10**

2.1. PACKAGING AND COMPLIANCY ........ 10

2.2. POWER REQUIREMENTS ........ 10

2.3. FRONT PANEL ........ 11

2.4. EXTERNAL CONNECTORS ........ 12
*2.4.1. ANALOG INPUT connectors ........ 12*
*2.4.2. INPUT connectors ........ 12*
*2.4.3. CONTROL connectors ........ 12*
*2.4.4. ADC REFERENCE CLOCK connectors ........ 12*
*2.4.5. Optical LINK connector ........ 13*
*2.4.6. USB Port ........ 13*

2.5. OTHER COMPONENTS ........ 13
*2.5.1. Displays ........ 13*

TECHNICAL SPECIFICATIONS TABLE ........ 14

**3. FUNCTIONAL DESCRIPTION ........ 15**

3.1. ANALOG INPUT STAGE ........ 15

3.2. DOMINO RING SAMPLING ........ 15

3.3. TR0 INPUT ........ 16

3.4. CLOCK DISTRIBUTION ........ 17

3.5. DATA CORRECTION ........ 19
*3.5.1. Cell offset correction ........ 19*
*3.5.2. Index sampling correction ........ 20*
*3.5.3. Time correction ........ 21*

3.6. EVENT STRUCTURE ........ 23
*3.6.1. Memory FULL management ........ 24*

3.7. TRIGGER MANAGEMENT ........ 25
*3.7.1. Trigger distribution ........ 26*

3.8. TEST PATTERN GENERATOR ........ 26

3.9. RESET, CLEAR AND DEFAULT CONFIGURATION ........ 27
*3.9.1. Global Reset ........ 27*
*3.9.2. Memory Reset ........ 27*

3.10. DATA TRANSFER CAPABILITIES ........ 27

3.10.1.1. Single data transfer ........ 27

3.10.1.2. Block transfers ........ 27
*3.10.2. Event Polling ........ 28*

3.11. OPTICAL LINK AND USB ACCESS ....28

**4. SOFTWARE TOOLS ....29**

**5. BOARD INTERNAL REGISTERS ....32**

5.1. CONFIGURATION ROM (0XF000-0XF084; R) ....33
5.2. GROUP N CHANNEL THRESHOLD (0X1N80; R/W) ....34
5.3. GROUP N STATUS (0X1N88; R) ....34
5.4. DAUGHTER BOARD FW REVISION (0X1N8C; R) ....35
5.5. GROUP N BUFFER OCCUPANCY (0X1N94; R) ....35
5.6. GROUP N CHANNEL DC OFFSET (0X1N98; R/W) ....35
5.7. GROUP N ADC CONFIGURATION (0X1N9C; R/W) ....35
5.8. DRS4 TEMPERATURE (0X1NA0; R) ....35
5.9. CHANNEL N DAC SEL (0X1NA4; R/W) ....36
5.10. GROUP N CHANNEL TRIGGER MASK (0X1NA8; R/W) ....36
5.11. MEMORY CALIBRATION TABLES ENABLE (0X1NCC; R/W) ....36
5.12. MEMORY CALIBRATION TABLES DATA (0X1ND0; R/W) ....36
5.13. GROUP N TR THRESHOLD (0X1ND4; R/W) ....36
5.14. GROUP N TR DC OFFSET (0X1NDC; R/W) ....37
5.15. GROUP CONFIGURATION REGISTER (0X8000; R/W) ....38
5.16. GROUP CONFIGURATION BIT SET (0X8004; W) ....38
5.17. GROUP CONFIGURATION BIT CLEAR (0X8008; W) ....39
5.18. BUFFER ORGANIZATION (0X800C; R/W) ....39
5.19. CUSTOM SIZE (0X8020; R/W) ....39
5.20. INITIAL TEST WAVE VALUE (0X807C) ....39
5.21. SAMPLING FREQUENCY (0X80D8) ....39
5.22. ACQUISITION CONTROL (0X8100; R/W) ....39
5.23. ACQUISITION STATUS (0X8104; R) ....40
5.24. SOFTWARE TRIGGER (0X8108; W) ....40
5.25. TRIGGER SOURCE ENABLE MASK (0X810C; R/W) ....40
5.26. FRONT PANEL TRIGGER OUT ENABLE MASK (0X8110; R/W) ....41
5.27. POST TRIGGER SETTING (0X8114; R/W) ....41
5.28. FRONT PANEL I/O DATA (0X8118; R/W) ....41
5.29. FRONT PANEL I/O CONTROL (0X811C; R/W) ....41
5.30. GROUP ENABLE MASK (0X8120; R/W) ....41
5.31. ROC FPGA FIRMWARE REVISION (0X8124; R) ....42

5.32. EVENT STORED (0X812C; R) ....42
5.33. SET MONITOR DAC (0X8138; R/W) ....42
5.34. BOARD INFO (0X8140; R) ....42
5.35. MONITOR MODE (0X8144; R/W) ....42
5.36. EVENT SIZE (0X814C; R) ....42
5.37. CONTROL (0XEF00; R/W) ....43
5.38. STATUS (0XEF04; R) ....43
5.39. INTERRUPT STATUS ID (0XEF14; R/W) ....43
5.40. INTERRUPT EVENT NUMBER (0XEF18; R/W) ....43
5.41. BLOCK TRANSFER EVENT NUMBER (0XEF1C; R/W) ....43
5.42. SCRATCH (0XEF20; R/W) ....43
5.43. SOFTWARE RESET (0XEF24; W) ....44
5.44. SOFTWARE CLEAR (0XEF28; W) ....44
5.45. FLASH ENABLE (0XEF2C; R/W) ....44
5.46. FLASH DATA (0XEF30; R/W) ....44
5.47. CONFIGURATION RELOAD (0XEF34; W) ....44

**6. INSTALLATION ....45**

6.1. POWER ON SEQUENCE ....45
6.2. POWER ON STATUS ....45
6.3. FIRMWARE UPGRADE ....45
6.4. DRIVERS ....46

## *LIST OF FIGURES*

FIG. 1.1: MOD. N6742 BLOCK DIAGRAM ....9
FIG. 2.1: MOD. N6742 FRONT PANEL ....11
FIG. 2.2: MCX CONNECTOR ....12
FIG. 2.3: AMP CLK IN CONNECTOR ....12
FIG. 2.4: LC OPTICAL CONNECTOR ....13
FIG. 3.1: ANALOG INPUT DIAGRAM ....15
FIG. 3.2: INPUT DIAGRAM ....16
FIG. 3.3: TR0 LOGIC BLOCK DIAGRAM ....17
FIG. 3.4: CLOCK DISTRIBUTION DIAGRAM ....18
FIG. 3.5: SAMPLED WAVEFORM AND NOISE HISTOGRAM BEFORE CELL OFFSET CORRECTION ....19
FIG. 3.6: SAMPLED WAVEFORM AND NOISE HISTOGRAM AFTER CELL OFFSET CORRECTION ....19
FIG. 3.7: SAMPLED WAVEFORM AND NOISE HISTOGRAM AFTER INDEX SAMPLING CORRECTION ....20

FIG. 3.8: SAMPLED TR0 SIGNAL IN GR0 AND GR1 BEFORE TIME CORRECTION ....................21
FIG. 3.9: SAMPLED TR0 SIGNAL IN GROUP 0 AND 1 AFTER TIME CORRECTION ....................21
FIG. 3.10: INL TIME PROFILE OF DRS CHIPS 0 AND 1 BEFORE TIME CORRECTION ....................22
FIG. 3.11: INL TIME PROFILE OF DRS CHIPS 0 AND 1 AFTER TIME CORRECTION ....................22
FIG. 3.12: EVENT FORMAT ....................23
FIG. 3.13: GROUP DATA FORMAT....................24
FIG. 3.14: BLOCK DIAGRAM OF TRIGGER MANAGEMENT....................25
FIG. 3.15: FPGA TEST WAVEFORM....................26
FIG. 3.16: EXAMPLE OF BLOCK TRANSFER READOUT ....................28
FIG. 4.1: BLOCK DIAGRAM OF THE SOFTWARE LAYERS ....................29
FIG. 4.2: WAVEDUMP OUTPUT WAVEFORMS ....................30
FIG. 4.3: CAENSCOPE OSCILLOSCOPE TAB ....................30
FIG. 4.4: CAENUPGRADER GRAPHICAL USER INTERFACE....................31
FIG. 4.5: DPP CONTROL SOFTWARE GRAPHICAL USER INTERFACE AND ENERGY PLOT ....................31

# *LIST OF TABLES*

TABLE 1.1: AVAILABLE ITEMS ....................8
TABLE 2.1: POWER REQUIREMENTS....................10
TABLE 2.2: FRONT PANEL LEDS ....................13
TABLE 2.3: MOD. N6742 TECHNICAL SPECIFICATIONS....................14
TABLE 5.1: ADDRESS MAP FOR THE MOD. N6742....................32
TABLE 5.2: ROM ADDRESS MAP FOR THE MOD. N6742....................34

# 1. General description

## 1.1. Overview

The Mod. N6742 is a NIM module housing 16+1 Channels 12bit 5GS/s Switched Capacitor Digitizer based on DRS4 chip, with 1 Vpp input dynamic range on single ended MCX coaxial connectors.

The DC offset is adjustable via 16bit DAC (± 1V range) on each channel and allows tot sample either bipolar (Vin=± 0.5V) or unipolar (full positive Vin= 0 ÷ +1V or negative Vin= 0 ÷ -1V) analog input swing without losing dynamic resolution.

The analog input signals are continuously sampled into the DRS4s in a circular analog memory buffer (1024 cells); default sampling frequency is 5GS/s; 2.5GS/s and 1GS/s frequencies can be also programmed. As a trigger signal arrives, all analog memory buffers are frozen and subsequently digitized with a 12bit resolution into a digital memory buffer. The digital memory (128 events deep for each channel, where 1 event = 1024x12bit) allows to store subsequent events, even if the readout is not yet started. Moreover, since the digital memory buffers work like FIFOs, the readout activity does not affect write operations of subsequent events.

A common board trigger can be provided via either software, or by TRG-IN input.

A special fast analog trigger input TR0 (TTL/NIM levels compatible), can be used as lo-latency external trigger signal. This special input can be also sampled into the DRS4s analog memory buffers for applications where high resolution timing and time analysis with a common reference signal (like a trigger or system clock) is required. During analog to digital conversion process, the N6742 cannot handles other triggers, this is called Dead Time. This time will be increased if also TR0 channel is added to the acquisition of the analog inputs.

The module N6742 features a PLL for clock synthesis with a selectable internal or external reference clock source.

Multi-board synchronization can be done by driving a clock on CLOCK-IN input, allowing all DRS4s to run synchronously with this external reference. All analog inputs will be sampled at the same time without time drift, allows high resolution timing and time analysis across multiple N6742.

The board houses a daisy chainable Optical Link able to transfer data at 80 MB/s, thus it is possible to connect up to 8 ADC boards (256+16 ADC channels) to a single Optical Link Controller; a USB2.0 compatible port is also featured. Optical Link and USB access are internally arbitrated.

**Table 1.1: Available items**

| Code | Description |
|---|---|
| WN6742XAAAAA | N6742 – 16+1 Ch. 12 bit 5 GS/s Digitizer: 128kS/ch, EP3C16, SE |
| WN6742BXAAAA | N6742B – 16+1 Ch. 12 bit 5 GS/s Digitizer: 1024kS/ch, EP3C16, SE |
| WA654XAAAAAA | A654 - Single Channel MCX to LEMO Cable Adapter |
| WA654K4AAAAA | A654 KIT4 - 4 MCX TO LEMO Cable Adapter |
| WA2818XAAAAA | A2818 - PCI Optical Link |
| WA3818AXAAAA | A3818 - PCIe 1 Optical Link |
| WA3818BXAAAA | A3818 - PCIe 2 Optical Link |
| WA3818CXAAAA | A3818 - PCIe 4 Optical Link |
| WAI2730XAAAA | AI2730 - Optical Fibre 30 m. simplex |
| WAI2720XAAAA | AI2720 - Optical Fibre 20 m. simplex |
| WAI2705XAAAA | AI2705 - Optical Fibre 5 m. simplex |
| WAI2703XAAAA | AI2703 - Optical Fibre 30cm. simplex |
| WAY2730XAAAA | AY2730 - Optical Fibre 30 m. duplex |
| WAY2720XAAAA | AY2720 - Optical Fibre 20 m. duplex |
| WAY2705XAAAA | AY2705 - Optical Fibre 5 m. duplex |

## 1.2. Block Diagram

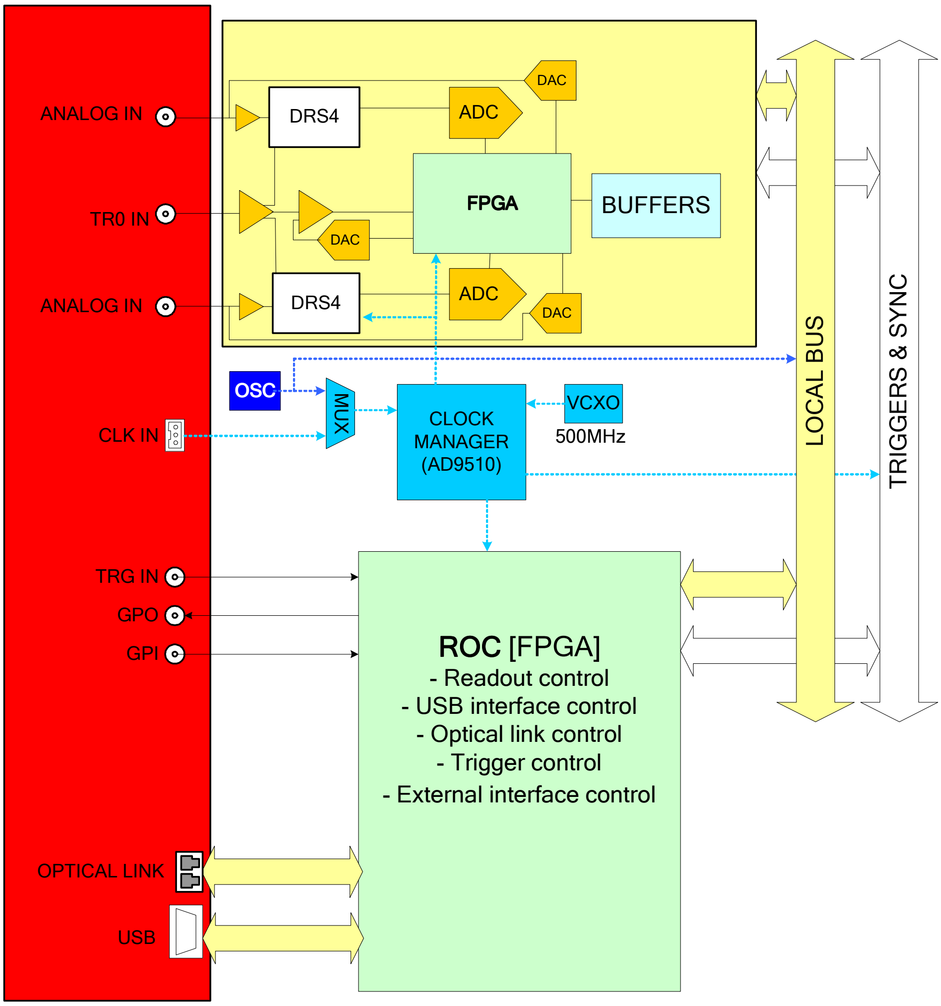



**Fig. 1.1: Mod. N6742 Block Diagram**

The function of each block will be explained in detail in the subsequent sections.

# 2. Technical specifications

## 2.1. Packaging and Compliancy

The unit is housed in a single width NIM module.

## 2.2. Power requirements

The power requirements of the module are as follows:

**Table 2.1: Power requirements**

| | |
|---|---|
| +6 V | 3.9 A |
| -6 V | 90 mA |

## 2.3. Front Panel

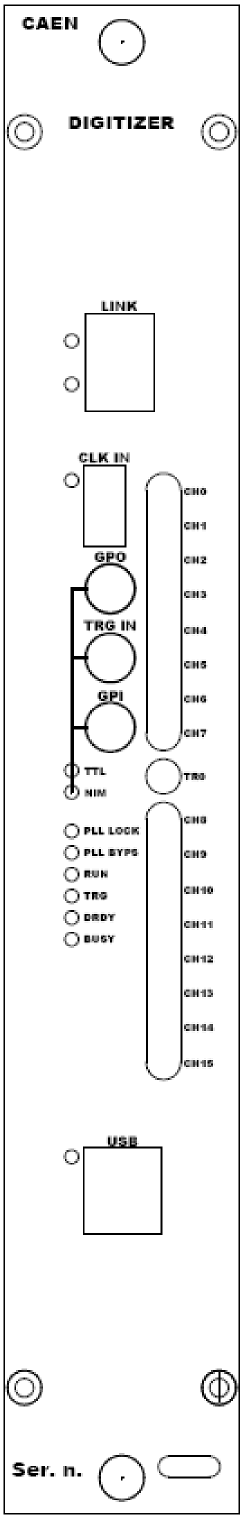



**Fig. 2.1: Mod. N6742 front panel**

## 2.4. External connectors

### 2.4.1. *ANALOG INPUT connectors*

### 2.4.2. *INPUT connectors*

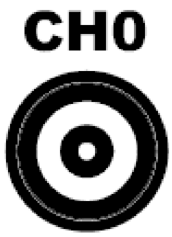



**Fig. 2.2: MCX connector**

*Function*:
Analog input, single ended, Zin=50Ω; TR[1,0] (Fast TRG) input, Zin=50Ω
*Mechanical specifications*:
MCX connector (CS 85MCX-50-0-16 SUHNER)
*Suggested plug*: MCX-50-2-16 type
*Suggested cable*: RG174 type

### 2.4.3. *CONTROL connectors*

*Function:*
- GPO: General purpose output (NIM/TTL, on Rt = 50Ω); used as output for trigger propagation
- TRG IN: External trigger input (NIM/TTL, Zin= 50Ω)
- GPI: General purpose input (NIM/TTL, Zin=50Ω)

*Mechanical specifications*: 00-type LEMO connectors

### 2.4.4. *ADC REFERENCE CLOCK connectors*

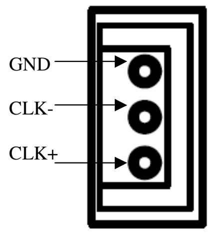



**Fig. 2.3: AMP CLK IN Connector**

*Function:*
CLK IN: External clock/Reference input, AC coupled (diff. LVDS, ECL, PECL, LVPECL, CML), Zdiff= 110Ω.
*Mechanical specifications:*
AMP 3-102203-4 AMP MODUII

### 2.4.5. Optical LINK connector

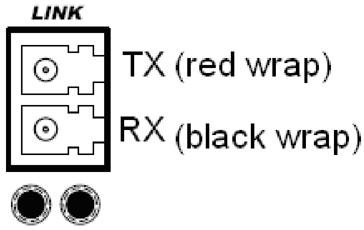



**Fig. 2.4: LC Optical Connector**

*Mechanical specifications:*
LC type connector; to be used with Multimode 62.5/125µm cable with LC connectors on both sides
*Electrical specifications:*
Optical link for data readout and slow control with transfer rate up to 80MB/s; daisy chainable.

### 2.4.6. USB Port

*Mechanical specifications:*
B type USB connector
*Electrical specifications:*
USB 2.0 and USB 1.1 compliant

## 2.5. Other components

### 2.5.1. Displays

The front panel hosts the following LEDs:

**Table 2.2: Front panel LEDs**

| Name: | Colour: | Function: |
| --- | --- | --- |
| **CLK_IN** | green | External clock enabled. |
| **NIM** | green | Standard selection for GPO, TRG IN, GPI. |
| **TTL** | green | Standard selection for GPO, TRG IN, GPI. |
| **USB** | green | Data transfer activity |
| **LINK** | green/yellow | Network present; Data transfer activity |
| **PLL _LOCK** | green | The PLL is locked to the reference clock |
| **PLL _BYPS** | green | The reference clock drives directly ADC clocks; the PLL circuit is switched off and the PLL_LOCK LED is turned off. |
| **RUN** | green | RUN bit set |
| **TRG** | green | Triggers are accepted |
| **DRDY** | green | Event/data (depending on acquisition mode) are present in the Output Buffer |
| **BUSY** | red | All the buffers are full |

# Technical specifications table

**Table 2.3: Mod. N6742 technical specifications**

| | |
|---|---|
| **Package** | 1-unit wide NIM module |
| **Analog Input** | 16 channels (MCX 50 Ohm) Single-ended<br>Input range: 1 Vpp<br>Bandwidth: >500MHz<br>Programmable DAC for Offset Adjust x ch.<br>adjustment range: ±1V |
| **TR0 Input** | MCX 50 Ohm, NIM/TTL<br>For fast local trigger and high resolution timing reference |
| **Switched Capacitor array** | Based on DRS4 chip Switched capacitor ADC<br>1024 storage cells per channels (200 ns recorded time per event)<br>Continuously sampled up to 5 GS/s simultaneously on each channels<br>After trigger analog samples are digitized by external ADC. |
| **Digital Conversion** | Resolution: 12 bit |
| **Dead Time** | 110µs Analog inputs only; 181µs Analog inputs + TR0 input |
| **ADC Sampling Clock generation** | Sampling clock generation supports two operating modes:<br>- PLL mode - internal reference (50 MHz local oscillator)<br>- PLL mode - external reference on CLK_IN (Jitter<100ppm, Frequency 50 MHz). |
| **Digital I/O** | CLK_IN (AMP Modu II):<br>- AC coupled differential input clock LVDS, ECL, PECL, LVPECL, CML (single ended NIM/TTL available on request)<br>- Jitter<100ppm<br>TRG_IN (LEMO 50 Ohm, NIM/TTL)<br>GPI (LEMO 50 Ohm, NIM/TTL) |
| **ADC and Memory controller FPGA** | 1 Altera Cyclone EP3C16 for 16+1 channels |
| **Memory Buffer** | 128 event/ch, (1024 samples per event); Multi Event Buffer with independent read and write access. |
| **Trigger** | Common Trigger<br>- TRG_IN (External signal)<br>- Software (from USB or Optical Link)<br>Fast local trigger<br>- Programmable threshold on the special input channel TR0 |
| **Optical Link** | CAEN proprietary protocol, up to 80 MB/s transfer rate,<br>Daisy chainable: it is possible to connect up to 8/32 ADC modules to a single Optical Link Controller (Mod. A2818/A3818) |
| **USB interface** | USB2.0 compliant<br>Up to 30 MB/s transfer rate |
| **Upgrade** | Firmware can be upgraded via USB/Optical Link |
| **Software** | Libraries (C and LabView), Demos and Software tools for Windows and Linux |
| **Power requirements** | 3.9 A @ +6V, 90 mA @ -6V |

# 3. Functional description

## 3.1. Analog input stage

Input dynamic is 1Vpp on single ended MCX coaxial connectors (Zin= 50 Ohm). A 16bit DAC allows to add up to ±1V DC offset in order to preserve the full dynamic range also with uni-polar positive or negative input signal. The input bandwidth ranges from DC to 500 MHz.

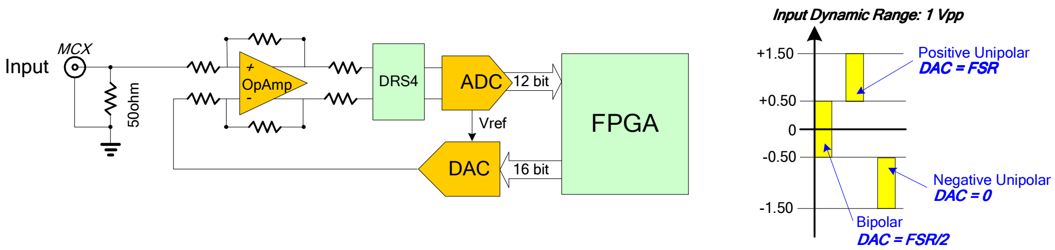



**Fig. 3.1: Analog input diagram**

## 3.2. Domino Ring Sampling

The analog input signals are continuously sampled into the DRS4s (Domino Ring Sampler), which consists of an on-chip inverter chain (domino wave circuit) generating a maximum 5GS/s sampling frequency; 2.5GS/s and 1GS/s frequencies can be also programmed (see § 5.20). No external sampling clock is required.

This signal opens write switches in all 9 sampling channels, where the differential input signals are sampled (1024 sampling capacitance cells per channel).

After being started, the domino wave runs continuously in a circular fashion (after the end of the ring, samples are over written) until decoupled from the write switches by a trigger signal, which freezes the currently stored signal in the sampling capacitance cells.

Subsequently the cells are multiplexed into the 12 bit ADCs whose output are stored by the FPGA into the Digital Memory Buffer and ready for readout in the shape of events data.

A 16bit DAC allow to add up to ±1V DC offset in order to preserve the full dynamic range also with uni-polar positive or negative input signals

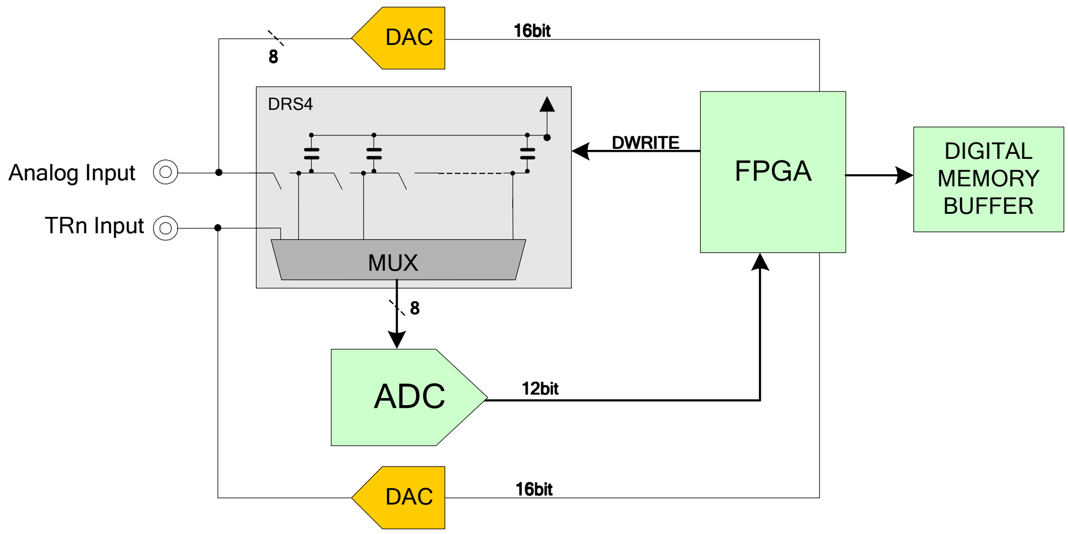



**Fig. 3.2: Input diagram**

Detailed documentation of the DRS4 is available at http://drs.web.psi.ch/

## 3.3. TR0 Input

The module feature the fast trigger input TR0 with extended level amplitude (TTL/NIM compatible); TR0 is common to all channels. TR0 can be used as external trigger (see § 3.6). Moreover it can be also sampled into the DRS4s analog memory buffers for applications where high resolution timing and time analysis with a common reference signal (like a trigger or system clock) is required; this is achieved through the Configuration Register, “Signal TR0 Readout Enable” bit setting (see § 5.15) allows to store TR0 input with samples coming both from group 0 and 1.

In order to be sampled, the TR0 signal must be compatible with the DRS4 chips input dynamics (1V), therefore on N6742 mezzanine Rev.0 PCB the TR0 signal is attenuated by a factor 3; on N6742 mezzanine Rev.1 PCB the TR0 signal is attenuated by a factor 2, therefore if TR0 signal larger than 2V is going to be sampled, an external attenuator shall be used.

To properly handle bipolar signals and also unipolar positive or negative signal, a 16 bit DAC allows to add a DC offset to TR0; offset value can be programmed via Group n TR DC Offset register (see § 5.14)

When the TR0 is used as trigger, it is processed by an internal comparator, whose threshold can be programmed via Group n TR Threshold register (see § 5.13): as the threshold is exceeded, the FPGA triggers the DRS4’s and samples digitizing takes place. The trigger signals can be sensed either on the leading or the trailing edge, depending on Configuration register setting (see § 5.15).

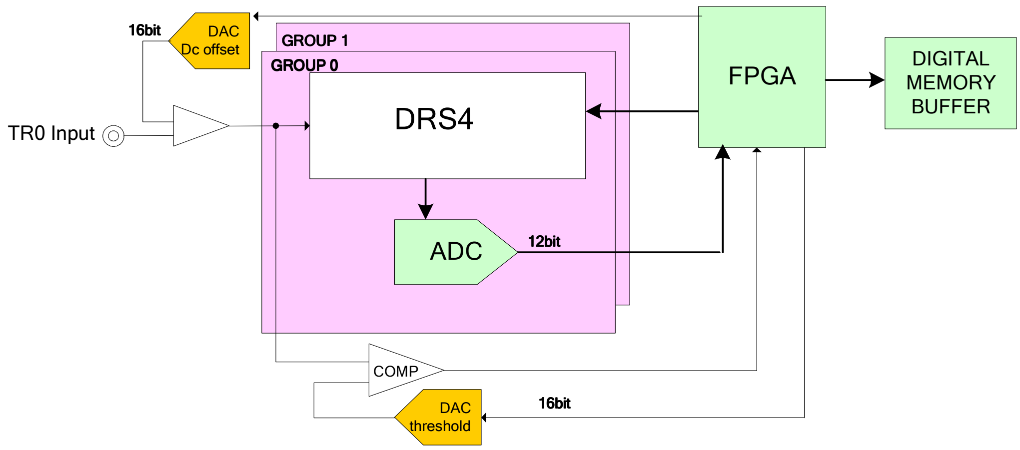



**Fig. 3.3: TR0 logic block diagram**

## 3.4. Clock Distribution

The module features a PLL for clock synthesis with a selectable internal or external reference clock source.

Multi-board synchronization can be done by driving a clock on CLOCK-IN input, allowing all DRS4s to run synchronously with this external reference. All analog inputs will be sampled at the same time without time drift, allows high resolution timing and time analysis across multiple modules.

The module clock is provided by OSC-CLK and REF-CLK.
OSC-CLK is a fixed 50MHz clock provided by an on board oscillator; it handles Optical Link, USB and Local Bus (communication between motherboard and mezzanine boards; see red traces in the figure below).
REF-CLK handles trigger logic, acquisition logic (samples storage into RAM, buffer freezing on trigger) through a clock chain. Such domain can use either an external (via front panel signal) or an internal (via local oscillator) source; in the latter case OSC-CLK and REF-CLK will be synchronous (the operation mode remains the same anyway).
REF-CLK is processed by AD9520 device, which delivers clock out signals to to ADCs, and to the trigger logic (refer to AD9520 data sheet for more details, available on http://www.analog.com/

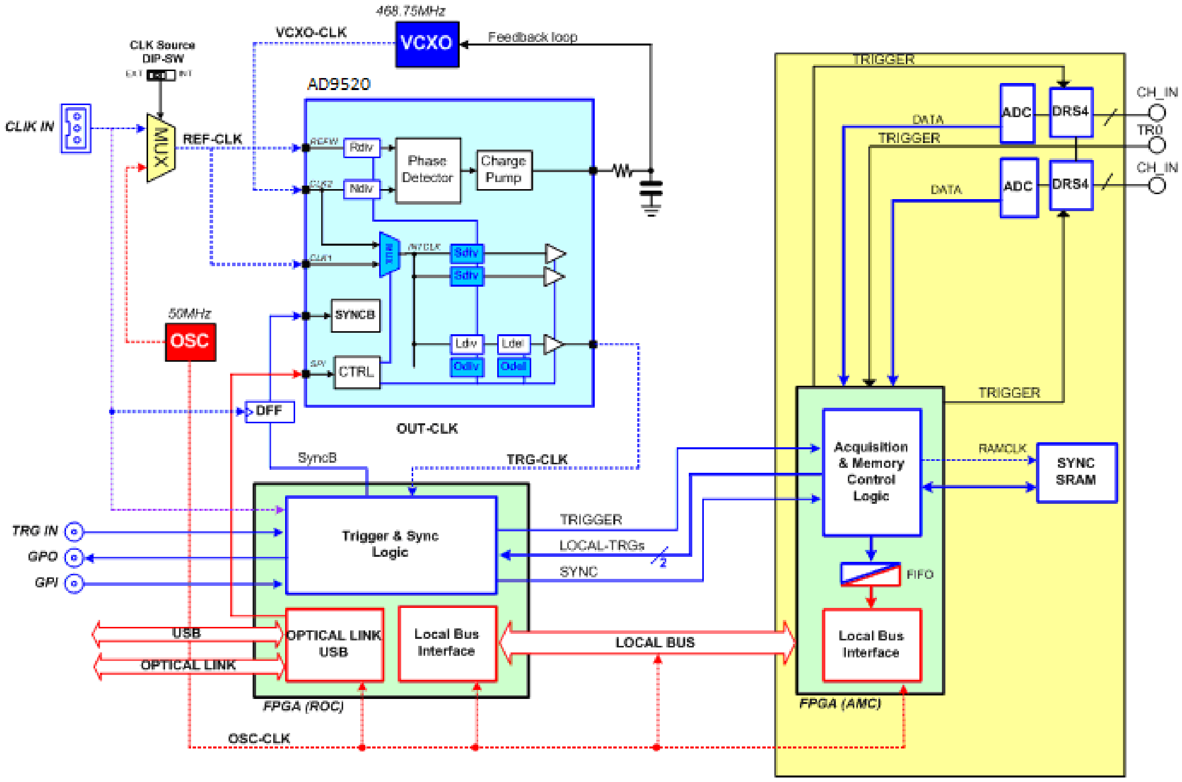



**Fig. 3.4: Clock distribution diagram**

# 3.5. Data correction

Three types of data correction are required, in order to compensate for unavoidable construction differences in the DRS4 chips. All boards are factory calibrated during production test and correction parameters are saved on board. Application software provided by CAEN recovers automatically the calibration parameters and runs them in order to correct the stored data events.

## 3.5.1. *Cell offset correction*

Unavoidable construction differences between the “analog memory cells” (see § 3.1) require an amplitude calibration algorithm.
The following images show the sampled waveform and noise histogram before and after correction:

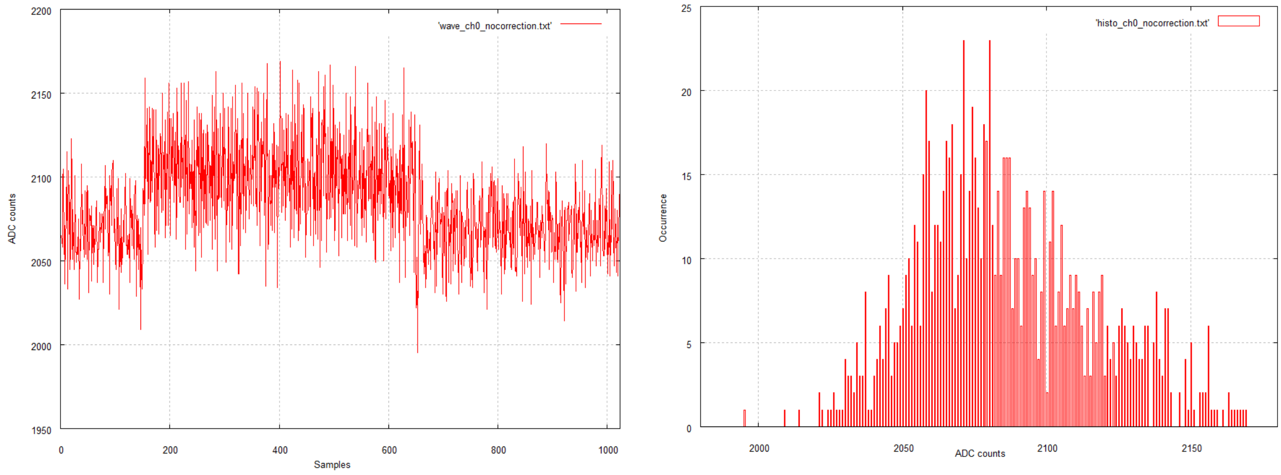



**Fig. 3.5: Sampled waveform and noise histogram before cell offset correction**

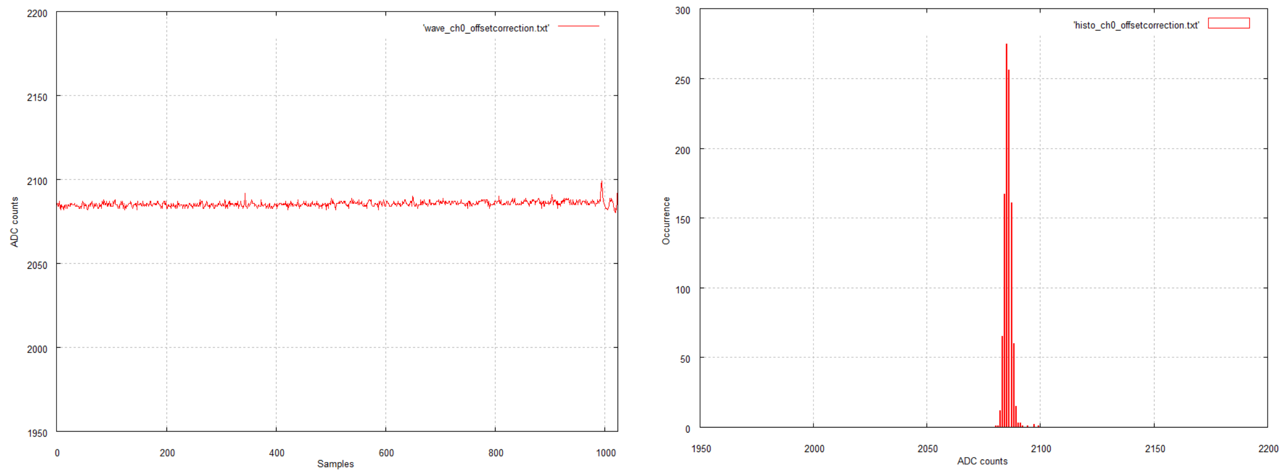



**Fig. 3.6: Sampled waveform and noise histogram after cell offset correction**

### *3.5.2. Index sampling correction*

It has been observed a fixed pattern noise, introduced by the DRS4, over the last samples (~30 samples) in a waveform, therefore the "index sampling correction" is necessary; this correction actually reduces this noise, thus, anytime the best accuracy is required, the last ~30 samples should be rejected.

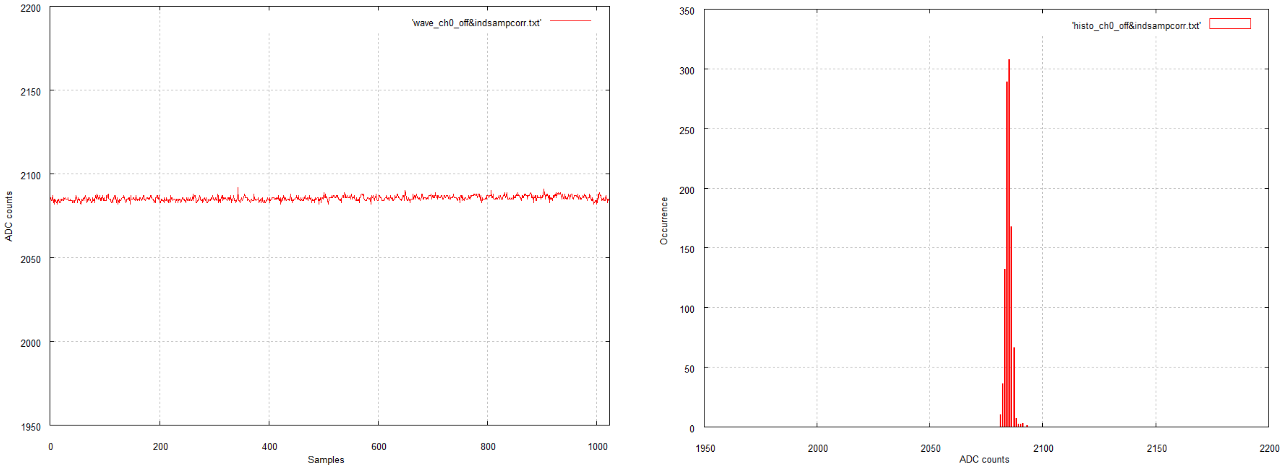



**Fig. 3.7: Sampled waveform and noise histogram after index sampling correction**

### 3.5.3. *Time correction*

The sampling sequence is handled by the DRS4 through 1024 physical delay lines; the unavoidable construction differences between such delay lines must be compensated through a time calibration. The following figures show the fast trigger signal (TR0) sampled by the DRS chip related to Group 0 and 1 and the integral non linearity (INL) time profile of DRS chips, before and after correction:

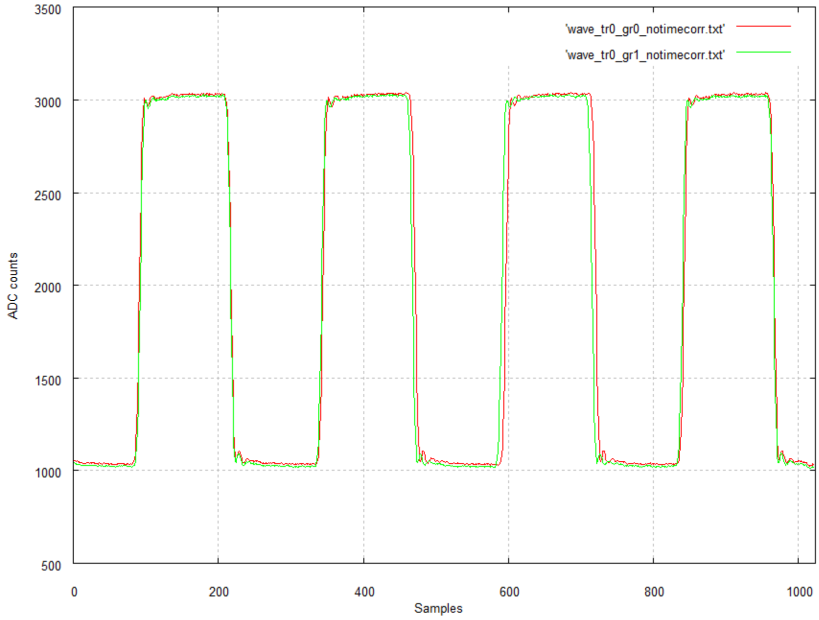



**Fig. 3.8: Sampled TR0 signal in GR0 and GR1 before time correction**

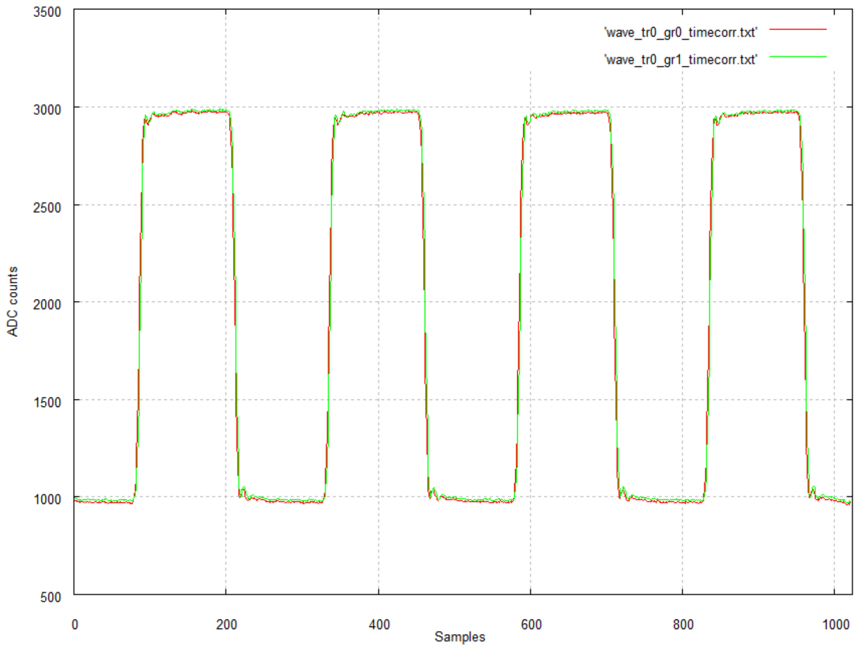



**Fig. 3.9: Sampled TR0 signal in Group 0 and 1 after time correction**

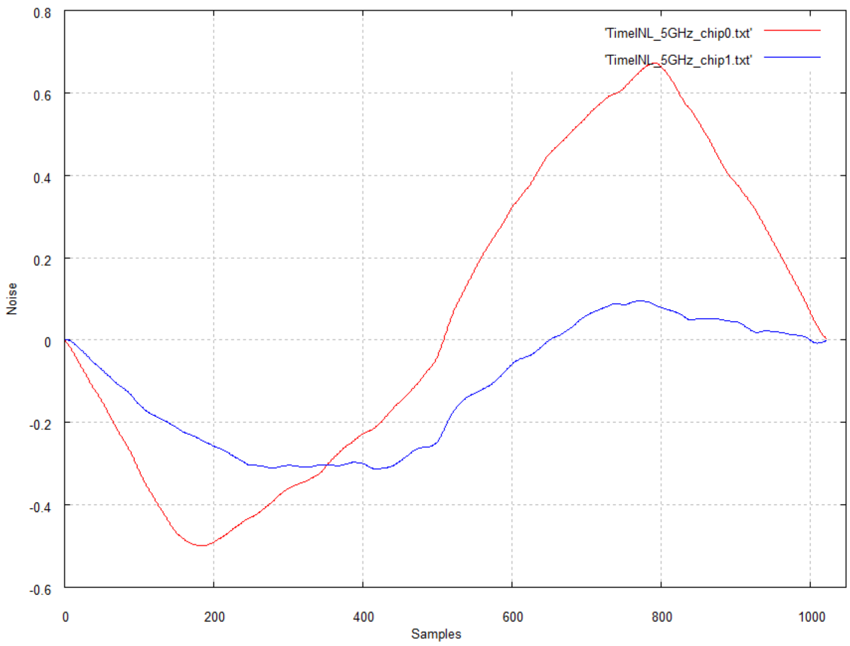



**Fig. 3.10: INL time profile of DRS chips 0 and 1 before time correction**

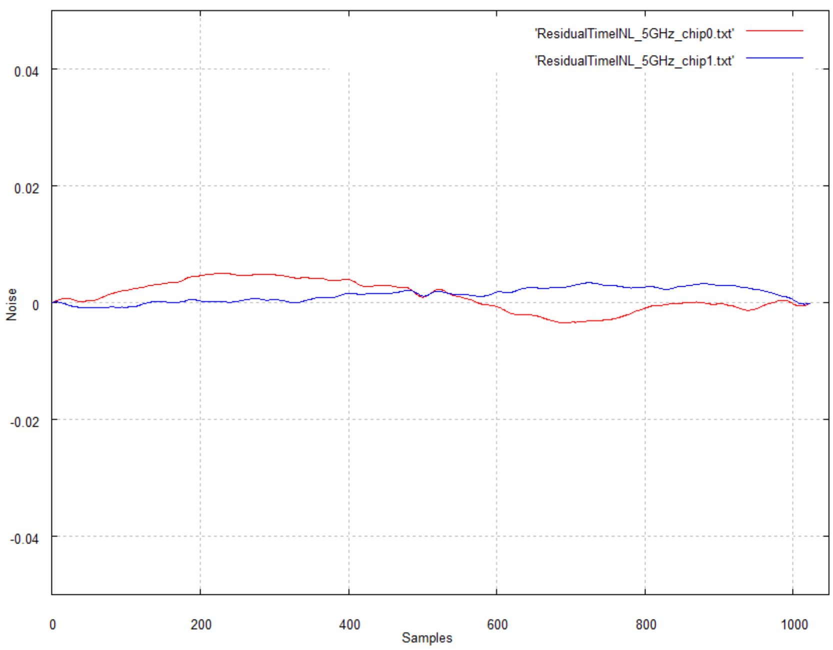



**Fig. 3.11: INL time profile of DRS chips 0 and 1 after time correction**

## 3.6. Event structure

An event is structured as follows:

- Header (four 32-bit words)
- Data (variable size and format)

The event can be readout either via USB or Optical Link; data format is 32 bit word.

<table>
<tr><th></th><th>31</th><th>30</th><th>29</th><th>28</th><th>27</th><th>26</th><th>25</th><th>24</th><th>23</th><th>22</th><th>21</th><th>20</th><th>19</th><th>18</th><th>17</th><th>16</th><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>9</th><th>8</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td rowspan="4">HEADER</td><td>1</td><td>0</td><td>1</td><td>0</td><td colspan="28">TOTAL EVENT SIZE (LWORDS)</td></tr>
<tr><td colspan="5">BOARD ID</td><td colspan="5"></td><td colspan="14">PATTERN</td><td colspan="6"></td><td colspan="2">G.MSK</td></tr>
<tr><td colspan="10"></td><td colspan="22">EVENT COUNTER</td></tr>
<tr><td colspan="32">EVENT TIME TAG</td></tr>
<tr><td rowspan="3">GROUP 0</td><td colspan="32">GROUP 0 EVENT DESCRIPTION WORD</td></tr>
<tr><td colspan="32">GROUP 0 CHANNEL DATA</td></tr>
<tr><td colspan="32">GROUP 0 TRIGGER TIME TAG</td></tr>
<tr><td rowspan="3">GROUP 1</td><td colspan="32">GROUP 1 EVENT DESCRIPTION WORD</td></tr>
<tr><td colspan="32">GROUP 1 CHANNEL DATA</td></tr>
<tr><td colspan="32">GROUP 1 TRIGGER TIME TAG</td></tr>
</table>

**Fig. 3.12: Event Format**

The Header is composed by four words, namely:

- Size of the event (number of 32 bit words)
- Board ID (GEO); 16 bit pattern, latched on the LVDS I/O as one trigger arrives; Group Mask (=1: Groups participating to event; ex GR0 and GR1 participating → Gr_Mask: 0x3, this information must be used by the software to acknowledge what Group the samples are coming from; the first event contains the samples from the Group with the lowest number)
- Event Counter: It is the trigger counter; it can count either accepted triggers only, or all triggers (see § 5.19).
- Trigger Time Tag: It is a 32 bit counter (31 bit count + 1 overflow bit), which is reset as acquisition starts and is incremented at each sampling clock hit. It is the trigger time reference.

Each group is composed by 8 analog channels (group 0 = channel 0 – 7, group 1 = channel 8 – 15) and by the special channel TR0: such signal is common to both groups; it can be used as Local Trigger or "digitized" and stored with the data for high resolution timing analysis between the ADC channels and the TR0 itself.

TR0 can trigger Group 0 and Group 1 and can be stored with data from Group 0 (therefore the stored waveform will be labelled as Tr00) and with data from Group 1 (therefore the stored waveform will be labelled as Tr01).

The part of an event related to each group has the following format (example of Group 0):

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>0</td><td>0</td><td colspan="10">START INDEX CELL</td><td>0</td><td>0</td><td colspan="2">FREQ</td><td>0</td><td>0</td><td>0</td><td>TR</td><td colspan="12">SIZE CH 0..7</td></tr>
<tr><td colspan="8">S0-CH2 (LO)</td><td colspan="12">S0-CH1</td><td colspan="12">S0-CH0</td></tr>
<tr><td colspan="4">S0-CH5(LO)</td><td colspan="12">S0-CH4</td><td colspan="12">S0-CH3</td><td colspan="4">S0-CH2(HI)</td></tr>
<tr><td colspan="12">S0-CH7</td><td colspan="12">S0-CH6</td><td colspan="8">S0-CH5(HI)</td></tr>
<tr><td colspan="8">S1-CH2 (LO)</td><td colspan="12">S1-CH1</td><td colspan="12">S1-CH0</td></tr>
<tr><td colspan="4">S1-CH5(LO)</td><td colspan="12">S1-CH4</td><td colspan="12">S1-CH3</td><td colspan="4">S1-CH2(HI)</td></tr>
<tr><td colspan="12">S1-CH7</td><td colspan="12">S1-CH6</td><td colspan="8">S1-CH5(HI)</td></tr>
<tr><td colspan="32">….</td></tr>
<tr><td colspan="12">S(N-1)-CH7</td><td colspan="12">S(N-1)-CH6</td><td colspan="8">S(N-1)-CH5(HI)</td></tr>
<tr><td colspan="8">S2-TR00 (LO)</td><td colspan="12">S1-TR00</td><td colspan="12">S0-TR00</td></tr>
<tr><td colspan="4">S5-TR00(LO)</td><td colspan="12">S4-TR00</td><td colspan="12">S3-TR00</td><td colspan="4">S2-TR00(HI)</td></tr>
<tr><td colspan="12">S7-TR00</td><td colspan="12">S6-TR00</td><td colspan="8">S5-TR00(HI)</td></tr>
<tr><td colspan="32">….</td></tr>
<tr><td colspan="12">S(N-1)- TR00</td><td colspan="12">S(N-2)- TR00</td><td colspan="8">S(N-3)- TR00(HI)</td></tr>
<tr><td>31</td><td>30</td><td colspan="30">TRIGGER TIME TAG</td></tr>
</table>

**Fig. 3.13: Group Data Format**

In the Group Event Description Word (yellow in the figure above) the following fields are shown:

- Bit [29:20] Start Index Cell of DRS4 related to this event
- Bit [17:16] (sampling frequency):
    - 00 = 5GS/s
    - 01 = 2.5GS/s
    - 10 = 1GS/s
    - 11 = not used
- Bit [12] (tr):
    - 0 = TR0 signal not present in readout
    - 1 = TR0 signal present in readout
- Bit [11:0] Size related to channel 0-7 (number of 32 bit words): when each channel has 1024 samples, “Size Ch 0-7” is 0xC00.

If readout of TR0 is disabled, data related to such channel (light blue in figure above) are not present in the event; if readout of TR0 is enabled, data size related to such channel is Size TR0 = (Size Ch 0-7)/8.

Trigger Time Tag records the Trigger arrival time; each bin has a 8.5ns width.

### *3.6.1. Memory FULL management*

Bit5 of Acquisition Control register (see § 5.22), allows to select Memory FULL management mode:

In Normal Mode the board becomes full, whenever all buffers are full; otherwise (“Always one buffer free” mode) it is possible to always keep one buffer free: board becomes full, whenever N-1buffers are full; with N = nr. of blocks.

In Normal Mode, the board waits until one buffer is filled since FULL status is exited (whether the trigger is overlapped or not). The board exits FULL status at the moment which the last datum from the last channel participating to the event is read.

In “Always one buffer free” mode, one buffer cannot be used (therefore it is NOT POSSIBLE, with this mode, to set Buffer Code to 0000; see § 5.19), but this allows to eliminate dead time when FULL status is exited.

## 3.7. Trigger management

Signal digitization can be triggered basically in two ways:

- **Common trigger**: a trigger produced via software (via USB or Optical Link) or sent via front panel TRG_IN signal (NIM/TTL signal on LEMO connector, 50 Ohm impedance.). In this case, all the channels in a board share the same trigger.
- **Low latency trigger**: a logic level fed directly into the DRS4 via the front panel TR0 signal.

As a trigger signal arrives, the analog buffers related to that trigger, are frozen and then digitized with a 12bit resolution ADC into the digital memory buffer.

During analog to digital conversion process, the module cannot handles other triggers; this “Dead Time” is larger if also TR0 input channel is sampled together with the analog inputs (see §3.3).

The TR0 is actually an analog input, but it is also TTL/NIM compatible; in order to use it as low latency external trigger signals, it is necessary to set properly the Configuration Register “Local TR0 Trigger Enable” bit (see §3.3).

Once the acquisition is triggered in one of the ways described above, digitization takes place as described in § 3.2.

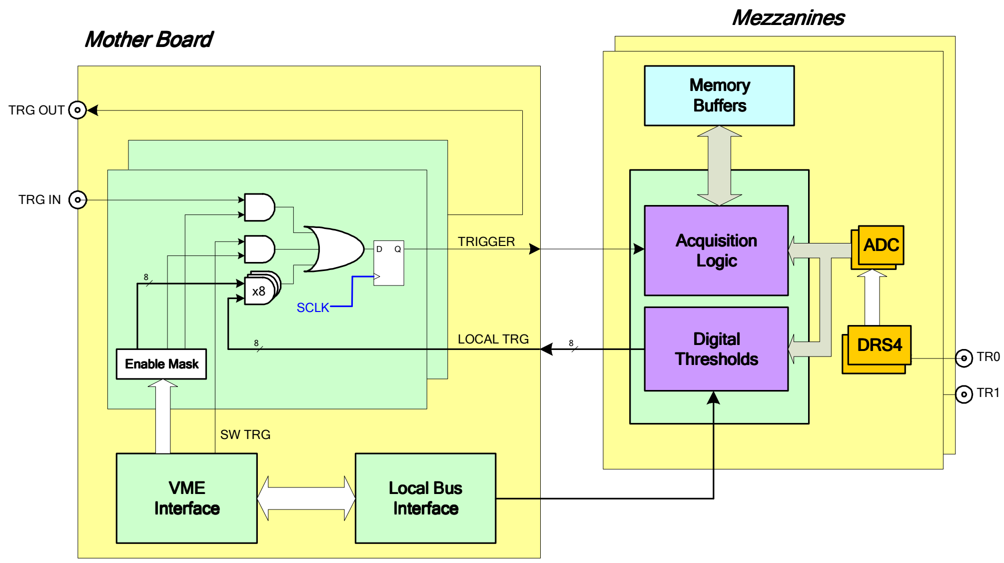



**Fig. 3.14: Block diagram of Trigger management**

### *3.7.1. Trigger distribution*

The OR of all the enabled trigger sources, after being synchronized with the internal clock, becomes the global trigger of the board and is fed in parallel to all the channels, which store an event.

A Trigger Out is also generated on the relevant front panel TRG_OUT connector (NIM or TTL), and allows to extend the trigger signal to other boards.

For example, in order to start the acquisition on all the channels in the crate, as one of the channels ramps over threshold, the Local Trigger must be enabled as Trigger Out, the Trigger Out must then be fed to a Fan Out unit; the obtained signal has to be fed to the External Trigger Input of all the boards in the crate (including the board which generated the Trigger Out signal).

## 3.8. Test pattern generator

The FPGA can emulate the ADC and write into memory a sawtooth signal for test purposes. It can be enabled via Group Configuration register, see § 5.14.

The following figure shows the test waveforms for Group 0 (Ch 0..7) and Group 1 (Ch 8..15) respectively.

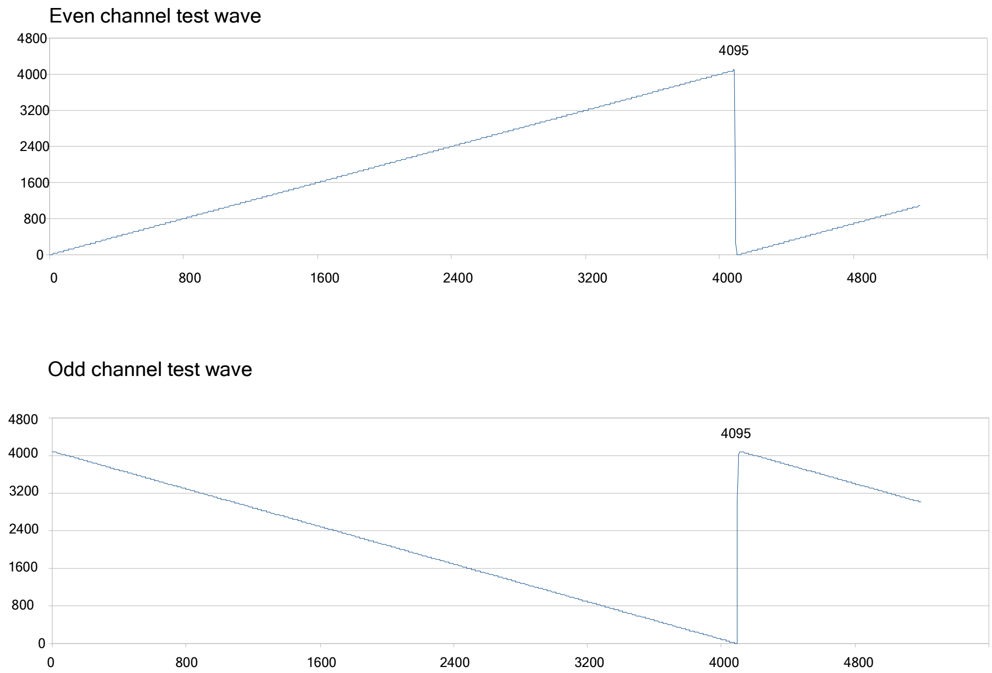



**Fig. 3.15: FPGA test waveform**

Since an event is made up of up to 1024 samples, the test event samples only a "portion" of the sawtooth; the start point of the sampling can be programmed via Initial Test Wave Value register (see § 5.20); for example if this register is set to 0x0FF then the channels in the Group 0 samples the ramp between 255 and 1278; the channels in the Group 1 instead samples the complementary value, therefore between 3840 and 2817.

# 3.9. Reset, Clear and Default Configuration

## *3.9.1. Global Reset*

Global Reset is performed at Power ON of the module or via a Software Reset, see § 5.43. It allows to clear the data off the Output Buffer, the event counter and performs a FPGAs global reset, which restores the FPGAs to the default configuration. It initializes all counters to their initial state and clears all detected error conditions.

## *3.9.2. Memory Reset*

The Memory Reset clears the data off the Output Buffer.
The Memory Reset can be forwarded via a write access to Software Clear Register (see § 5.44).

# 3.10. Data transfer capabilities

The board supports 32bit single data readout and block transfers; the events, once written in the SRAMs (Memory Event Buffers), become available for readout via USB or Optical Link. During the memory readout, the board can continue to store more events (independently from the readout) on the free buffers. The acquisition process is therefore "deadtimeless", until the memory becomes full.
Although the memories are SRAMs, addresses are taken them from a FIFO. Therefore, data are read from the memories sequentially, according to the selected Readout Logic, from a memory space mapped on 4Kbytes (0x0000÷0x0FFC).
The events are readout sequentially and completely, starting from the Header of the first available event, followed by the Trigger Time Tag, the Event Counter and all the samples of the group channels (from 0 to 7). Once an event is completed, the relevant memory buffer becomes free and ready to be written again (old data are lost). After the last word in an event, the first word (Header) of the subsequent event is readout. It is not possible to readout an event partially.

### 3.10.1.1. Single data transfer

This mode allows to readout a word per time, from the header (actually 4 words) of the first available event, followed by all the words until the end of the event, then the second event is transferred. The exact sequence of the transferred words is shown in § 3.6.
We suggest, after the 1st word is transferred, to check the Event Size information and then do as many cycles as necessary (actually Event Size -1) in order to read completely the event.

### 3.10.1.2. Block transfers

Block transfer allows, via a single channel access, to read N events in sequence, N is set via the Block Transfer Event Number register (see § 5.41).

```
[Event Size] = [2*(Group Size)] + [16 bytes]
```

Group Size depends on Custom Size setting (see § 5.19) and whether TR0 signal is stored in the event or not.

Then it is necessary to perform as many cycles as required in order to readout the programmed number of events.
We suggest to enable Bus error signal during Block transfer cycles, in order to end the cycle avoiding filler readout. The last Block transfer cycle will not be completed, it will be ended by Bus error signal after the #N event in memory is transferred (see example in the figure below).

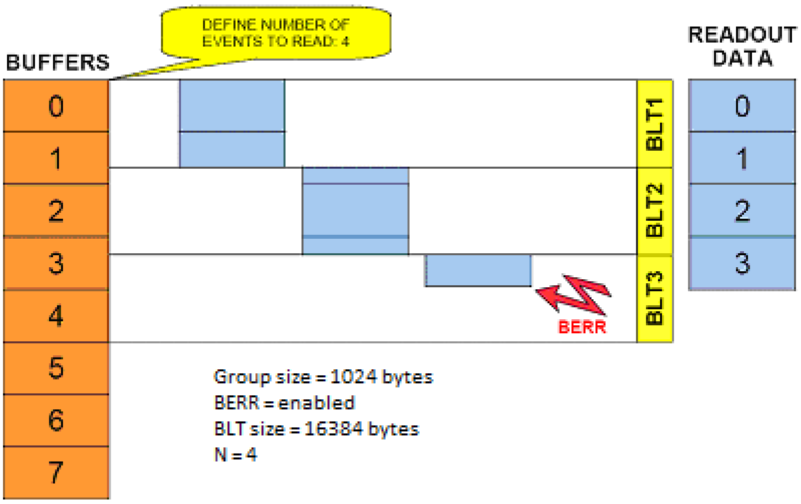



**Fig. 3.16: Example of Block transfer readout**

### *3.10.2. Event Polling*

A read access to Event Size register (see § 5.36) allows "polling" the number of 32 bit words composing the next event to be read: this permits to perform a properly sized (according to the Event Size information) Block transfer readout from the Memory Event Buffer.

## 3.11. Optical Link and USB access

The board houses a USB2.0 compliant port, providing a transfer rate up to 30 MB/s, and a daisy chainable Optical Link able to transfer data at 80 MB/s; the latter allows to connect up to eight N6742 to a single Optical Link Controller: for more information, see www.caen.it (path: Products / Front End / PCI/PCIe / Optical Controller)
The parameters for read/write accesses via optical link are Address Modifier, Base Address, data Width, etc; wrong parameter settings cause Bus Error.
Control Register bit 3 allows to enable the module to broadcast an interrupt request on the Optical Link; the enabled Optical Link Controllers propagate the interrupt on the PCI bus as a request from the Optical Link is sensed.

# 4. Software tools

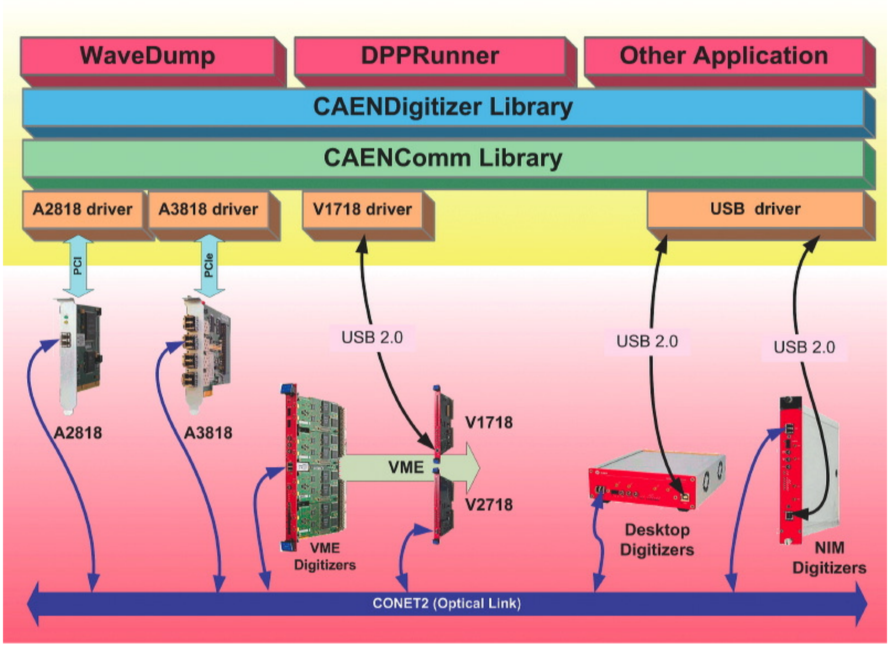



**Fig. 4.1: Block diagram of the software layers**

CAEN provides drivers for both the physical communication channels (USB and the proprietary CONET Optical Link managed by the A2818 PCI card or A3818 PCIe cards; see § 6.4), a set of C and LabView libraries, demo applications and utilities. Windows and Linux are both supported. The available software is the following:

- **CAENComm** library contains the basic functions for access to hardware; the aim of this library is to provide a unique interface to the higher layers regardless the type of physical communication channel. The CAENComm requires the CAENVMELib library to be installed even in the cases where the VME is not used.
- **CAENDigitizer** is a library of functions designed specifically for the digitizer family and it supports also the boards running special DPP (Digital Pulse Processing) firmware. The purpose of this library is to allow the user to open the digitizer, program it and manage the data acquisition in an easy way: with few lines of code the user can make a simple readout program without the necessity to know the details of the registers and the event data format. The CAENDigitizer library implements a common interface to the higher software layers, masking the details of the physical channel and its protocol, thus making the libraries and applications that rely on the CAENDigitizer independent from the physical layer. The library is based on the CAENComm library that manages the communication at low level (read and write access). CAENVMELib and CAENComm libraries must be already installed on the host PC before installing the CAENDigitizer; however, both CAENVMELib and CAENComm libraries are completely transparent to the user.
- **WaveDump** is a Console application that allows to program the digitizer (according to a text configuration file that contains a list of parameters and instructions), to start the acquisition, read the data, display the readout and trigger rate, apply some post processing (such as FFT and amplitude histogram), save data to a file and also plot the waveforms using the external plotting tool “gnuplot”, available on internet for free. This program is quite basic and has no graphics but it is an excellent example of C code that demonstrates the use of libraries and methods for an efficient readout and

data analysis. **NOTE:** WaveDump does not work with digitizers running DPP firmware. The users who intend to write the software on their own are suggested to start with this demo and modify it according to their needs. For more details please see the WaveDump User Manual and Quick Start Guide (Doc nr.: UM2091, GD2084 ).

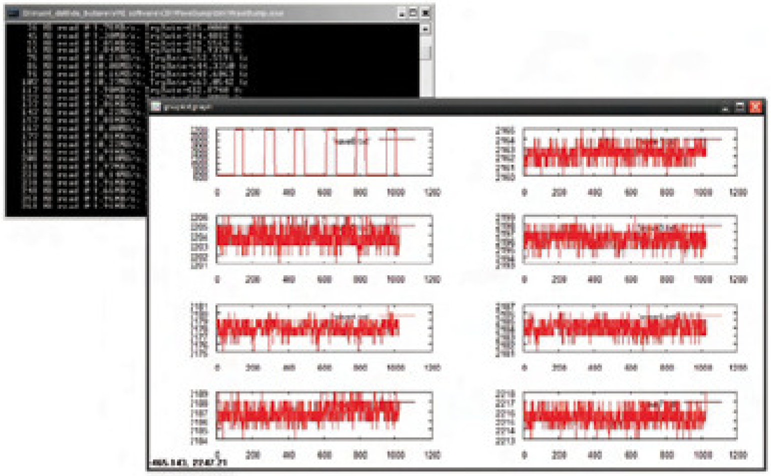


**Fig. 4.2: WaveDump output waveforms**

- **CAENScope** is a fully graphical program that implements a simple oscilloscope: it allows to see the waveforms, set the trigger thresholds, change the scales of time and amplitude, perform simple mathematical operations between the channels, save data to file and other operations. CAENscope is provided as an executable file; the source codes are not distributed. **NOTE:** CAENScope does not work with digitizers running DPP firmware and it is not compliant with x742 digitizer family. For more details please see the CAENScope Quick Start Guide GD2484.

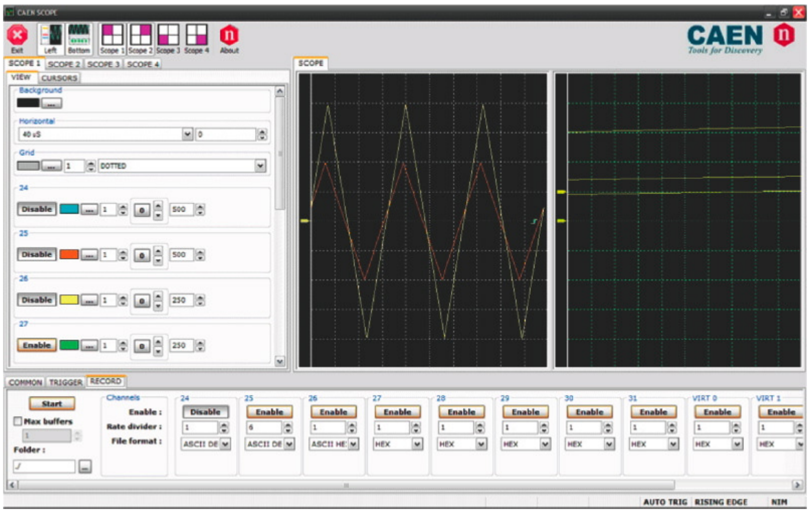



**Fig. 4.3: CAENScope oscilloscope tab**

- **CAENUpgrader** is a software composed of command line tools together with a Java Graphical User Interface (for Windows and Linux OS). CAENUpgrader allows in few easy steps to upload different firmware versions on CAEN boards, to upgrade the VME digitizers PLL, to get board information and to manage the firmware license. CAENUpgrader requires the installation of 2 CAEN libraries (CAENComm, CAENVMELib) and Java SE6 (or later). CAENComm allows CAENUpgrader to access target boards via USB or via CAEN proprietary CONET optical link.

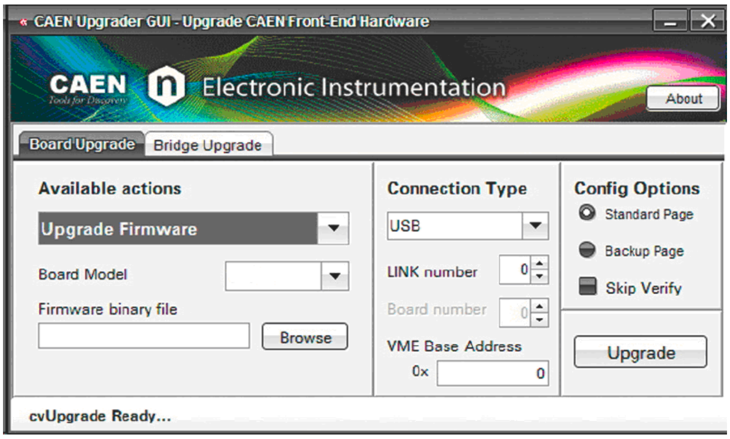



**Fig. 4.4: CAENUpgrader Graphical User Interface**

- **DPP Control Software** is an application that manages the acquisition in the digitizers which have DPP firmware installed on it. The program is made of different parts: there is a GUI whose purpose is to set all the parameters for the DPP and for the acquisition; the GUI generates a textual configuration file that contains all the parameters. This file is read by the Acquisition Engine (**DPPrunner)**, which is a C console application that programs the digitizer according to the parameters, starts the acquisition and manage the data readout. The data, that can be waveforms, time stamps, energies or other quantities of interest, can be saved to output files or plotted using gnuplot as an external plotting tool, exactly like in WaveDump. NOTE: so far DPP Control Software is developed for Mod. x724 and Mod. x720 digitizer series.

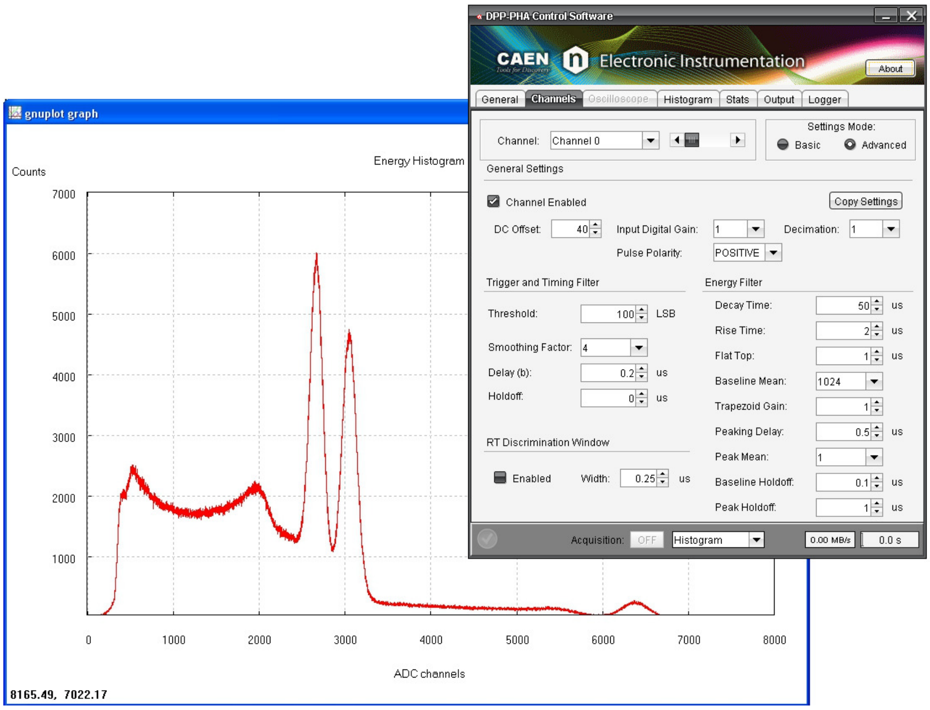



**Fig. 4.5: DPP Control Software Graphical User Interface and Energy plot**

# 5. Board internal registers

The following sections will describe in detail the registers (accessible via software in 32bit r/w mode) content.

Registers whose name begins with "Group n", are referred to channel groups, with index "n" in the address is either 0 or 1; each group is composed by eight subsequent channels

⚠ N.B.: bit fields that are not described in the register bit map are reserved and must not be over written by the User.

**Table 5.1: Address Map for the Mod. N6742**

| REGISTER NAME | ADDRESS | MODE | H_RES | S_RES | CLR |
|---|---|---|---|---|---|
| EVENT READOUT BUFFER | 0x0000-0x0FFC | R | X | X | X |
| Group n Channel THRESHOLD | 0x1n80 | R/W | X | X | |
| Group n STATUS | 0x1n88 | R | X | X | |
| Daughter board FW revision | 0x1n8C | R | | | |
| Group n BUFFER OCCUPANCY | 0x1n94 | R | X | X | X |
| Group n Channel DC offset | 0x1n98 | R/W | X | X | |
| Group n DAC SEL | 0x1nA4 | R / W | X | X | |
| DRS4 n Temperature | 0x1nA0 | R | X | X | |
| Group n CH. TRIG ENABLE MASK | 0x1nA8 | R/W | X | X | |
| Memory Calib. Tab. ENABLE | 0x1nCC | R/W | X | X | |
| Memory Calib. Tab. DATA | 0x1nD0 | R/W | X | X | |
| Group n TR THRESHOLD | 0x1nD4 | R / W | X | X | |
| Group n TR DC offset | 0x1nDC | R / W | X | X | |
| Group CONFIGURATION | 0x8000 | R/W | X | X | |
| Group CONFIG. BIT SET | 0x8004 | W | X | X | |
| Group CONFIG. BIT CLEAR | 0x8008 | W | X | X | |
| BUFFER ORGANIZATION | 0x800C | R/W | X | X | |
| CUSTOM SIZE | 0x8020 | R/W | X | X | |
| INITIAL TEST WAVE | 0x807C | R/W | X | X | |
| SAMPLING FREQUENCY | 0x80D8 | R/W | X | X | |
| ACQUISITION CONTROL | 0x8100 | R/W | X | X | |
| ACQUISITION STATUS | 0x8104 | R | X | X | |
| SW TRIGGER | 0x8108 | W | | | |
| TRIG. SOURCE ENABLE MASK | 0x810C | R/W | X | X | |
| FRONT PAN. TRIG_OUT EN. MASK | 0x8110 | R/W | X | X | |
| POST TRIGGER SETTING | 0x8114 | R/W | X | X | |
| FRONT PANEL I/O DATA | 0x8118 | R/W | X | X | |
| FRONT PANEL I/O CONTROL | 0x811C | R/W | X | X | |
| Group ENABLE MASK | 0x8120 | R/W | X | X | |
| ROC FPGA FIRMWARE REVISION | 0x8124 | R | | | |
| EVENT STORED | 0x812C | R | X | X | X |
| SET MONITOR DAC | 0x8138 | R/W | X | X | |
| BOARD INFO | 0x8140 | R | | | |
| MONITOR MODE | 0x8144 | R/W | X | X | |

| REGISTER NAME | ADDRESS | MODE | H_RES | S_RES | CLR |
|---|---|---|---|---|---|
| EVENT SIZE | 0x814C | R | X | X | X |
| CONTROL | 0xEF00 | R/W | X | | |
| STATUS | 0xEF04 | R | | | |
| INTERRUPT STATUS ID | 0xEF14 | R/W | X | | |
| INTERRUPT EVENT NUMBER | 0xEF18 | R/W | X | X | |
| BLOCK TR. EVENT NUMBER | 0xEF1C | R/W | X | X | |
| SCRATCH | 0xEF20 | R/W | X | X | |
| SW RESET | 0xEF24 | W | | | |
| SW CLEAR | 0xEF28 | W | | | |
| FLASH ENABLE | 0xEF2C | R/W | X | | |
| FLASH DATA | 0xEF30 | R/W | X | | |
| CONFIGURATION RELOAD | 0xEF34 | W | | | |
| CONFIGURATION ROM | 0xF000-0xF3FC | R | | | |

## 5.1. Configuration ROM (0xF000-0xF084; r)

The following registers contain some module’s information, they are 32bit accessible (read only):

- **OUI**: manufacturer identifier (IEEE OUI)
- **Version**: purchased version
- **Board ID**: Board identifier
- **Revision**: hardware revision identifier
- **Serial MSB**: serial number (MSB)
- **Serial LSB**: serial number (LSB)

**Table 5.2: ROM Address Map for the Mod. N6742**

| Description | Address | Content |
|---|---|---|
| checksum | 0xF000 | *0xA4* |
| checksum_length2 | 0xF004 | 0x00 |
| checksum_length1 | 0xF008 | 0x00 |
| checksum_length0 | 0xF00C | 0x20 |
| constant2 | 0xF010 | 0x83 |
| constant1 | 0xF014 | 0x84 |
| constant0 | 0xF018 | 0x01 |
| c_code | 0xF01C | 0x43 |
| r_code | 0xF020 | 0x52 |
| oui2 | 0xF024 | 0x00 |
| oui1 | 0xF028 | 0x40 |
| oui0 | 0xF02C | 0xE6 |
| vers | 0xF030 | 0x70 |
| board2 | 0xF034 | 0x03 |
| board1 | 0xF038 | 0x1A |
| board0 | 0xF03C | 0x56 |
| revis3 | 0xF040 | *0x00* |
| revis2 | 0xF044 | *0x00* |
| revis1 | 0xF048 | *0x00* |
| revis0 | 0xF04C | *0x01* |
| sernum1 | 0xF080 | *0x00* |
| sernum0 | 0xF084 | *0x16* |

These data are written into one Flash page; at Power ON the Flash content is loaded into the Configuration RAM, where it is available for readout.

## 5.2. Group n Channel Threshold (0x1n80; r/w)

| Bit | Function |
|---|---|
| [31:0] | *reserved* |

## 5.3. Group n Status (0x1n88; r)

| Bit | Function |
|---|---|
| [9] | Mezzanine PCB rev: 1 = 1<br>0 = 0 |
| [8] | DRS Chips Busy |
| [7] | Group Odd PLL Lock |
| [6] | Group Even PLL Lock |
| [5] | *reserved* |
| [4] | Group Odd Enable |
| [3] | Group Even Enable |
| [2] | SPI Bus Busy: 1 = Busy<br>0 = SPI ready |
| [1] | Memory empty |
| [0] | Memory full |

## 5.4. Daughter board FW revision (0x1n8C; r)

| Bit | Function |
|---|---|
| [31:16] | Revision date in Y/M/DD format |
| [15:8] | Firmware Revision (X) |
| [7:0] | Firmware Revision (Y) |

Bits [31:16] contain the Revision date in Y/M/DD format.
Bits [15:0] contain the firmware revision number coded on 16 bit (X.Y format).
Example: revision 1.3 of 12th June 2010 is: 0xA6120103

## 5.5. Group n Buffer Occupancy (0x1n94; r)

| Bit | Function |
|---|---|
| [10:0] | Occupied buffers (0..1024) |

## 5.6. Group n Channel DC offset (0x1n98; r/w)

| Bit | Function |
|---|---|
| [19:16] | Channel index from 0x0 to 0x7 (only one DAC ch.) or 0xF (all DAC ch.) |
| [15:0] | DAC Data |

The input DC offset can be adjusted group per group and channel per channel by means of a programmable 16bit DAC; there is a DAC serving each group (8 channels).
Default value 0x8F00 (about 0mV, for input bipolar signals.)
The channel index field (bits [19:16] ) is used only in write access. In read access, channel index must be set on CH DAC SEL register (see Channel DAC Select register).
For example, in order to set the DAC Value 0x6C00 for channel 15 (channel 7 of group 1) a write access to address 0x1198 with value 0x76C00 must be performed.
In order to readout the channel 15 DAC Value, a write access to address 0x11A4 with value 0x7 must be performed, and then a read access to address 0x1198.

⚠ **WARNING:** Before writing this register it is necessary to check that the "SPI Bus Busy" flag in the Status Register (§ 5.3) is set to "0", otherwise the process of writing will not give error messages out but it will not run properly!

## 5.7. Group n ADC Configuration (0x1n9C; r/w)

| Bit | Function |
|---|---|
| [31:0] | *reserved* |

## 5.8. DRS4 temperature (0x1nA0; r)

| Bit | Function |
|---|---|
| [7:0] | DRS4 temperature from 0°C to 127°C |

## 5.9. Channel n DAC SEL (0x1nA4; r/w)

| Bit | Function |
|---|---|
| [3:0] | DAC Channel index for readout, from 0x0 to 0x7. |

For example, in order to read the channel 15 DAC Value, a write access to address 0x11A4 with value 0x7 (channel 15 is channel 7 of group 1) must be performed, and then a read access to address 0x1198.

## 5.10. Group n Channel Trigger Mask (0x1nA8; r/w)

| Bit | Function |
|---|---|
| [31:0] | *reserved* |

## 5.11. Memory Calibration Tables Enable (0x1nCC; r/w)

| Bit | Function |
|---|---|
| [0] | 1 = Memory Calibration Tables ENABLED<br>0 = Memory Calibration Tables DISABLED |

This register allows to access the memory location where calibration data are stored (see § 3.5). CAUTION: before writing this register it is necessary to verify that SPI Bus Busy Flag in the Status register (§ 5.3) is 0 and, in any case, its use is reserved to experienced Users, since a wrong value written in the Memory Calibration Data will erase the module's calibration pattern.

## 5.12. Memory Calibration Tables Data (0x1nD0; r/w)

| Bit | Function |
|---|---|
| [7:0] | Data to be serialized or read from Memory Tables calibration |

This register allows to access the memory location where calibration data are stored (see § 3.5). CAUTION: before writing this register it is necessary to verify that SPI Bus Busy Flag in the Status register (§ 5.3) is 0 and, in any case, its use is reserved to experienced Users, since a wrong value written in the Memory Calibration Data will erase the module's calibration pattern.

## 5.13. Group n TR Threshold (0x1nD4; r/w)

| Bit | Function |
|---|---|
| [15:0] | Threshold |

The threshold on TR0 for local trigger generation can be set by a programmable 16bit DAC. One TR0 signal is common to two groups, therefore, for example write access to either 0x10D4 or 0x11D4 leads to the same setting for TR0 input.
For TR0 Threshold setting example, see the paragraph below.

⚠ **WARNING:** Before writing this register it is necessary to check that the "SPI Bus Busy" flag in the Status Register (§ 5.3) is set to "0", otherwise the process of writing will not give error messages out but it will not run properly!

## 5.14. Group n TR DC offset (0x1nDC; r/w)

| Bit | Function |
|---|---|
| [15:0] | DC Offset |

The TR0 signal offset can be set by a programmable 16bit DAC.
TR0 signal is common to both groups, therefore, for example write access to either 0x10DC or 0x11DC leads to the same setting for TR0 input.
If you are sending bipolar or negative signal to TR0, you have to set the TR0 Offset to 0x1000 (default setting); if you are sending positive signal to TR0, you have to set the TR0 Offset to 0x4000.

TR0 setting examples (trigger level on half voltage swing):

| **Mezzanine PCB Rev.1** |
|---|
| **ECL signal on TR0** |
| TR0 DC Offset = 0x55A0 (write 0x55A0 at address 0x10DC or 0x11DC)<br>TR0 Threshold = 0x6666 (write 0x6666 at address 0x10D4 or 0x11D4) |
| **NIM signal on TR0** |
| TR0 DC Offset = 0x8000 (write 0x8000 at address 0x10DC or 0x11DC)<br>TR0 Threshold = 0x51C6 (write 0x51C6 at address 0x10D4 or 0x11D4) |
| **Negative signal on TR0: V= 0 ÷ -400mV** |
| TR0 DC Offset = 0x8000 (write 0x8000 at address 0x10DC or 0x11DC)<br>TR0 Threshold = 0x5C16 (write 0x5C16 at address 0x10D4 or 0x11D4) |
| **Negative signal on TR0: V= 0 ÷ -200mV** |
| TR0 DC Offset = 0x8000 (write 0x8000 at address 0x10DC or 0x11DC)<br>TR0 Threshold = 0x613E (write 0x613E at address 0x10D4 or 0x11D4) |
| **Bipolar signal on TR0** |
| TR0 DC Offset = 0x8000 (write 0x8000 at address 0x10DC or 0x11DC)<br>TR0 Threshold = 0x6666 (write 0x6666 at address 0x10D4 or 0x11D4) |
| **TTL on TR0 or Positive signal on TR0: V= 0 ÷ ≥2V** |
| TR0 DC Offset = 0xA800 (write 0xA800 at address 0x10DC or 0x11DC)<br>TR0 Threshold = 0x6666 (write 0x6666 at address 0x10D4 or 0x11D4) |
| **Positive on TR0: V= 0 ÷ 2V** |
| TR0 DC Offset = 0x91A7 (write 0x91A7 at address 0x10DC or 0x11DC)<br>TR0 Threshold = = 0x6666 (write 0x6666 at address 0x10D4 or 0x11D4) |

| **Mezzanine PCB Rev.0** |
|---|
| **NIM signal on TR0** |
| TR0 DC Offset = 0x1000 (write 0x1000 at address 0x10DC or 0x11DC)<br>TR0 Threshold = 0x717D (write 0x717D at address 0x10D4 or 0x11D4) |
| **Negative signal on TR0: V= 0 ÷ -400mV** |
| TR0 DC Offset = 0x1000 (write 0x1000 at address 0x10DC or 0x11DC)<br>TR0 Threshold = 0x6E72 (write 0x6E72 at address 0x10D4 or 0x11D4) |
| **Bipolar signal on TR0** |
| TR0 DC Offset = 0x1000 (write 0x1000 at address 0x10DC or 0x11DC)<br>TR0 Threshold = 0x6C80 (write 0x6C80 at address 0x10D4 or 0x11D4) |
| **TTL on TR0 or Positive signal on TR0: V= 0 ÷ ≥2V** |
| TR0 DC Offset = 0x4000 (write 0x4000 at address 0x10DC or 0x11DC)<br>TR0 Threshold = 0x7158 (write 0x7158 at address 0x10D4 or 0x11D4) |

On boards with Mezzanines PCB Rev.1 it is suggested to use an external attenuator, in order to handle TR0 positive signals larger than +2V.

⚠ **WARNING:** Before writing this register it is necessary to check that the "SPI Bus Busy" flag in the Status Register (§ 5.3) is set to "0", otherwise the process of writing will not give error messages out but it will not run properly!

## 5.15. Group Configuration Register (0x8000; r/w)

| Bit | Function |
|---|---|
| [31:28] | Select monitor signal from daughter board<br>0000= no signal<br>0001= all fast trigger<br>0010= accepted fast trigger<br>0011= busy |
| [27:13] | *reserved* (MUST ALWAYS BE SET TO 0) |
| [12] | TR0 Trigger Enable: when this bit is 1, TR0 signal is used as local trigger:<br>0= TR0 Local Trigger disabled (Default)<br>1= TR0 Local Trigger enabled |
| [11] | Signal TR0 Readout Enable: when this bit is 1, signal TR0 is present in data readout:<br>0= Signal TR0 Readout disabled (Default)<br>1= Signal TR0 Readout enabled |
| [10:9] | *reserved* (MUST ALWAYS BE SET TO 0) |
| [8] | Individual Trigger: must be 1 |
| [7] | *reserved* (MUST ALWAYS BE SET TO 0) |
| [6] | TR0 Trigger polarity:<br>0= Rising Edge (Default)<br>1= Falling Edge. |
| [5] | *reserved* (MUST ALWAYS BE SET TO 0) |
| [4] | *reserved* (MUST ALWAYS BE SET TO 1) |
| [3] | Test Mode: when this bit is 1, the ADC samples are replaced by a sawtooth generated by the FPGA<br>0= Normal mode (data from the DRS4 and ADC, Default)<br>1= Test Mode (emulated data: from the sawtooth generator) |
| [2:0] | *reserved* (MUST ALWAYS BE SET TO 0) |

There are three ways to write the content of the Configuration Register:

- Normal Write (at address 0x8000): the content of the register is fully overwritten by the new data.
- Bit Set Mode (at address 0x8004): writing '1' in one bit, will set that bit; writing '0' leaves the bit unchanged.
- Bit Clear Mode (at address 0x8008): writing '1' in one bit, will clear that bit; writing '0' leaves the bit unchanged.

The use of the Bit Set/Clear modes are recommended when concurrent processes can access the register; this prevents a process to operate on the content of the register while another process has already changed it. The read access to the Control Register can be done at 0x8000 address.

## 5.16. Group Configuration Bit Set (0x8004; w)

| Bit | Function |
|---|---|
| [31:0] | Bits set to 1 means that the corresponding bits in the Group Configuration register are set to 1. |

## 5.17. Group Configuration Bit Clear (0x8008; w)

| Bit | Function |
|---|---|
| [31:0] | Bits set to 1 means that the corresponding bits in the Group Configuration register are set to 0. |

## 5.18. Buffer Organization (0x800C; r/w)

| Bit | Function |
|---|---|
| [31:0] | *reserved* (always set to 0) |

## 5.19. Custom Size (0x8020; r/w)

| Bit | Function |
|---|---|
| [1:0] | 00 = 1024 sample/ch<br>01 = 520 sample/ch<br>10 = 256 sample/ch<br>11 = 136 sample/ch |

This register must not be written while acquisition is running.

## 5.20. Initial test wave value (0x807C)

| Bit | Function |
|---|---|
| [11:0] | Test wave start value |

## 5.21. Sampling Frequency (0x80D8)

| Bit | Function |
|---|---|
| [1:0] | 00 = 5 GS/s<br>01 = 2.5 GS/s<br>10 = 1 GS/s<br>11 = *reserved* (do not use) |

This register must not be written while acquisition is running.

## 5.22. Acquisition Control (0x8100; r/w)

| Bit | Function |
|---|---|
| [5] | 0 = Normal Mode (default): board becomes full, whenever all buffers are full<br>1 = Always keep one buffer free: board becomes full, whenever N-1 buffers are full; N = 1024 |
| [4] | *reserved* |
| [3] | 0 = COUNT ACCEPTED TRIGGERS<br>1 = COUNT ALL TRIGGERS<br>allows to reject overlapping triggers (see § 3.5) |
| [2] | 0 = Acquisition STOP<br>1 = Acquisition RUN<br>allows to RUN/STOP Acquisition |

| | |
|---|---|
| [1:0] | *reserved* |

Bit [2] allows to Run and Stop data acquisition; when such bit is set to 1 the board enters Run mode and a Memory Reset (see § 3.9.2) is automatically performed. When bit [2] is reset to 0 the stored data are kept available for readout. In Stop Mode all triggers are neglected.

## 5.23. Acquisition Status (0x8104; r)

| Bit | Function |
|---|---|
| [8] | Board ready for acquisition (PLL and ADCs are synchronized correctly)<br>0 = not ready<br>1 = ready<br>This bit should be checked after software reset to ensure that the board will enter immediately run mode after RUN mode setting; otherwise a latency between RUN mode setting and Acquisition start might occur. |
| [7] | PLL Status Flag (see § 2.5.1):<br>0 = PLL loss of lock<br>1 = no PLL loss of lock<br>NOTE: flag can be restored to 1 via read access to Status Register (see § 5.38) |
| [6] | PLL Bypass mode (see § 2.5.1 ):<br>0 = No bypass mode<br>1 = Bypass mode |
| [5] | Clock source (see § 3.4):<br>0 = Internal<br>1 = External |
| [4] | EVENT FULL: it is set to 1 as the maximum nr. of events to be read is reached |
| [3] | EVENT READY: it is set to 1 as at least one event is available to readout |
| [2] | 0 = RUN off<br>1 = RUN on |
| [1:0] | *reserved* |

## 5.24. Software Trigger (0x8108; w)

| Bit | Function |
|---|---|
| [31:0] | A write access to this location generates a trigger via software |

## 5.25. Trigger Source Enable Mask (0x810C; r/w)

| Bit | Function |
|---|---|
| [31] | 0 = Software Trigger Disabled<br>1 = Software Trigger Enabled |
| [30] | 0 = External Trigger Disabled<br>1 = External Trigger Enabled |
| [29:0] | *reserved* |

EXTERNAL TRIGGER ENABLE (bit30) enables the board to accept the TRG_IN
SW TRIGGER ENABLE (bit 31) enables the board to accept the software trigger (see § 5.24).

## 5.26. Front Panel Trigger Out Enable Mask (0x8110; r/w)

| Bit | Function |
|---|---|
| [31] | 0 = Software Trigger Disabled<br>1 = Software Trigger Enabled |
| [30] | 0 = External Trigger Disabled<br>1 = External Trigger Enabled |
| [29:2] | *reserved* |
| [1] | 0 = Group 1 trigger disabled<br>1 = Group 1 trigger enabled |
| [0] | 0 = Group 0 trigger disabled<br>1 = Group 0 trigger enabled |

This register bits[3:0] enable the groups to generate a trigger as the TR0 signal exceeds the set threshold (see § 5.11). EXTERNAL TRIGGER ENABLE (bit30) enables the board to generate the TRG_OUT. SW TRIGGER ENABLE (bit 31) enables the board to broadcast a software trigger (see § 5.24).

## 5.27. Post Trigger Setting (0x8114; r/w)

| Bit | Function |
|---|---|
| [9:0] | Size of the post trigger window |

The register value sets the size of the post trigger window (expressed in steps of about 8.5ns); the maximum value for the post trigger is 0x3FF.

NOTE: This setting is valid from Firmware revision 0.3 on. For previous revisions, the register is over 7 bits and the maximum value for the post trigger is 0x7F

## 5.28. Front Panel I/O Data (0x8118; r/w)

| Bit | Function |
|---|---|
| [15:0] | Front Panel I/O Data |

Allows to Readout the logic level of LVDS I/Os and set the logic level of LVDS Outputs.

## 5.29. Front Panel I/O Control (0x811C; r/w)

| Bit | Function |
|---|---|
| [1] | 0= panel output signals (GPO) enabled<br>1= panel output signals (GPO) enabled in high impedance |
| [0] | 0 = TRG/CLK are NIM I/O Levels<br>1 = TRG/CLK are TTL I/O Levels |

## 5.30. Group Enable Mask (0x8120; r/w)

| Bit | Function |
|---|---|
| [1] | 0 = Group 1 disabled<br>1 = Group 1 enabled |
| [0] | 0 = Group 0 disabled |

| | |
|---|---|
| | 1 = Group 0 enabled |

Enabled groups provide the samples which are stored into the events (and not erased). The mask cannot be changed while acquisition is running.

## 5.31. ROC FPGA Firmware Revision (0x8124; r)

| Bit | Function |
|---|---|
| [31:16] | Revision date in Y/M/DD format |
| [15:8] | Firmware Revision (X) |
| [7:0] | Firmware Revision (Y) |

Bits [31:16] contain the Revision date in Y/M/DD format.
Bits [15:0] contain the firmware revision number coded on 16 bit (X.Y format).

## 5.32. Event Stored (0x812C; r)

| Bit | Function |
|---|---|
| [31:0] | This register contains the number of events currently stored in the Output Buffer |

This register value cannot exceed the maximum number of available buffers according to setting of buffer size register.

## 5.33. Set Monitor DAC (0x8138; r/w)

| Bit | Function |
|---|---|
| [31:0] | *reserved* |

## 5.34. Board Info (0x8140; r)

| Bit | Function |
|---|---|
| [15:8] | Memory size (Mbyte/Group) |
| [7:0] | Board Type: 0x06 |

## 5.35. Monitor Mode (0x8144; r/w)

| Bit | Function |
|---|---|
| [31:0] | *reserved* |

## 5.36. Event Size (0x814C; r)

| Bit | Function |
|---|---|
| [31:0] | Nr. of 32 bit words in the next event |

## 5.37. Control (0xEF00; r/w)

| Bit | Function |
|---|---|
| [4] | *reserved, must be set to 1* |
| [3] | 0 = Optical Link interrupt disabled<br>1 = Optical Link interrupt enabled |
| [2 :0] | *reserved, must be set to 0* |

Interrupt request can be removed by accessing this register and disabling the active interrupt level

## 5.38. Status (0xEF04; r)

| Bit | Function |
|---|---|
| [2] | 0 = Slave Terminated Transfer Flag: no terminated transfer<br>1 = Slave Terminated Transfer Flag: one transfer has been terminated by N6742 (unsupported register access or block transfer prematurely terminated in event aligned reaout) |
| [1] | *reserved* |
| [0] | 0 = No Data Ready<br>1 = Event Ready |

## 5.39. Interrupt Status ID (0xEF14; r/w)

| Bit | Function |
|---|---|
| [31:0] | This register contains the STATUS/ID that the module places on the data stream during the Interrupt Acknowledge cycle |

## 5.40. Interrupt Event Number (0xEF18; r/w)

| Bit | Function |
|---|---|
| [9:0] | INTERRUPT EVENT NUMBER |

If interrupts are enabled, the module generates a request whenever it has stored in memory a Number of events > INTERRUPT EVENT NUMBER

## 5.41. Block Transfer Event Number (0xEF1C; r/w)

| Bit | Function |
|---|---|
| [7:0] | This register defines the maximum number of events that can be transferred in a Block Transfer Cycle, after which the board asserts the Bus Error to stop the transfer. Allowed setting is between 0 (meaning no limit) and 255. |

## 5.42. Scratch (0xEF20; r/w)

| Bit | Function |
|---|---|
| [31:0] | Scratch (*to be used to write/read words for test purposes*) |

## 5.43. Software Reset (0xEF24; w)

| Bit | Function |
|---|---|
| [31:0] | A write access to this register causes a board reset (the acquisition is stopped, all the registers are set to the default settings and all data are cleared). |

## 5.44. Software Clear (0xEF28; w)

| Bit | Function |
|---|---|
| [31:0] | A write access to this register causes a data clear (the registers setting is not modified). |

## 5.45. Flash Enable (0xEF2C; r/w)

| Bit | Function |
|---|---|
| [0] | 0 = Flash write ENABLED<br>1 = Flash write DISABLED |

This register is handled by the Firmware upgrade tool.

## 5.46. Flash Data (0xEF30; r/w)

| Bit | Function |
|---|---|
| [7:0] | Data to be serialized towards the SPI On board Flash |

This register is handled by the Firmware upgrade tool.

## 5.47. Configuration Reload (0xEF34; w)

| Bit | Function |
|---|---|
| [31:0] | A write access to this register causes a software reset (see § 3.9), a reload of Configuration ROM parameters and a PLL reconfiguration. |

# 6. Installation

- The Module fits mechanically into all NIM crates
- Use only crates with forced cooling air flow
- Turn the crate OFF before board insertion/removal
- Remove all cables connected to the front panel before board insertion/removal

**CAUTION**
**USE ONLY CRATES WITH FORCED COOLING AIR FLOW SINCE OVERHEAT MAY DAMAGE THE MODULE!**

**CAUTION**
**ALL CABLES MUST BE REMOVED FROM THE FRONT PANEL BEFORE EXTRACTING THE BOARD FROM THE CRATE!**

## 6.1. Power ON sequence

To power ON the board follow this procedure:

1. insert the module into the crate
2. power up the crate

## 6.2. Power ON status

At power ON the module is in the following status:

- the Output Buffer is cleared;
- registers are set to their default configuration

## 6.3. Firmware upgrade

The N6742 firmware is stored onto on-board non-volatile memory. CAEN provides a firmware upgrade tool (see § 4) that can be used with either USB or optical link paths. Firmware updates are available in the Digitizer web page, while the software package, application notes and user manual are available in the CAENUpgrader web page at www.caen.it; follow the instructions for installation and usage.

WARNING: in case of programming failures, the board hosts a backup image of factory firmware.
Please contact CAEN at support.frontend@caen.it for instuctions in order to restore the backup image.
Once the board is successfully powered with backup firmware, the standard firmware image can be reprogrammed.

## 6.4. Drivers

N6742 needs CAEN USB driver to be installed in order to use the USB communication channel:

- **Download** the driver package compliant to your Operating System (Windows or Linux) on CAEN web site in the 'Software/Firmware' area at the digitizer page.
- **Uncompress** the package to your host.
- **For Windows users:**
  - **Installer option:** with the hardware not connected, run the single installer file and complete the installation Wizard. Then, connect the hardware and the driver will be automatically find by the OS.
  - **Driver files option:** connect the hardware; then, perform the driver installation by pointing Windows to the folder where driver files have been extracted.
- **For Linux users:** follow the installation instructions inside the README file in the package.

Concerning the OPTICAL LINK communication channel, please refer to A2818 PCI card or A3818 PCIe cards User Manual for driver installation.