

PRELIMINARY

Technical Information Manual

Revision n. 2
13 February 2007

MOD. V1724
8 CHANNEL 14 BIT
100 MS/S DIGITIZER
MANUAL REV.2

NPO:
00103/05:V1724x.MUTx/02

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.



TABLE OF CONTENTS

| | |
|---|-----------|
| 1. GENERAL DESCRIPTION..... | 7 |
| 1.1. OVERVIEW | 7 |
| 1.2. BLOCK DIAGRAM | 8 |
| 2. TECHNICAL SPECIFICATIONS..... | 9 |
| 2.1. PACKAGING..... | 9 |
| 2.2. POWER REQUIREMENTS | 9 |
| 2.3. FRONT PANEL..... | 10 |
| 2.4. EXTERNAL CONNECTORS..... | 11 |
| 2.4.1. ANALOG INPUT connectors..... | 11 |
| 2.4.2. CONTROL connectors..... | 11 |
| 2.4.3. ADC REFERENCE CLOCK connectors | 11 |
| 2.4.4. Digital I/O connectors..... | 12 |
| 2.4.5. Optical LINK connector | 12 |
| 2.5. OTHER FRONT PANEL COMPONENTS | 12 |
| 2.5.1. Displays | 12 |
| 2.6. INTERNAL COMPONENTS | 12 |
| 2.7. TECHNICAL SPECIFICATIONS TABLE | 14 |
| 3. FUNCTIONAL DESCRIPTION..... | 15 |
| 3.1. ANALOG INPUT..... | 15 |
| 3.1.1. Single ended input | 15 |
| 3.1.2. Differential input | 15 |
| 3.2. CLOCK MANAGEMENT | 16 |
| 3.2.1. Internal clock..... | 16 |
| 3.2.2. External clock and multiboard synchronisation..... | 16 |
| 3.3. ACQUISITION MODES | 17 |
| 3.3.1. Acquisition run/stop..... | 17 |
| 3.3.2. Gate and Sample mode acquisition | 17 |
| 3.3.3. Acquisition Triggering: Samples and Events..... | 18 |
| 3.3.4. Event structure..... | 19 |
| 3.4. TRIGGER MANAGEMENT | 20 |
| 3.4.1. External trigger | 21 |
| 3.4.2. Software trigger..... | 21 |
| 3.4.3. Local channel auto-trigger..... | 21 |
| 3.4.4. Trigger distribution | 21 |
| 3.5. FRONT PANEL I/OS..... | 22 |
| 3.6. ANALOG INSPECTION AND MAJORITY | 22 |
| 3.6.1. Analog monitor..... | 22 |
| 3.6.2. Majority..... | 23 |
| 3.7. RESET, CLEAR AND DEFAULT CONFIGURATION..... | 23 |
| 3.7.1. Global Reset | 23 |
| 3.7.2. Memory Reset | 23 |

| | | |
|-----------|---|-----------|
| 3.7.3. | Timer Reset..... | 23 |
| 3.8. | VMEBUS INTERFACE | 23 |
| 3.8.1. | Addressing capabilities..... | 24 |
| 3.9. | DATA TRANSFER CAPABILITIES | 25 |
| 3.10. | EVENTS READOUT | 25 |
| 3.10.1. | Sequential readout..... | 25 |
| 3.10.2. | Random readout (to be implemented)..... | 27 |
| 3.11. | OPTICAL LINK | 27 |
| 4. | VME INTERFACE | 28 |
| 4.1. | REGISTERS ADDRESS MAP | 28 |
| 4.2. | CONFIGURATION ROM (0xF000-0xF084; R) | 29 |
| 4.3. | THRESHOLD REGISTER (0x1N80; R)..... | 30 |
| 4.4. | OVER/UNDER THRESHOLD REGISTER (0x1N84; R/W) | 30 |
| 4.5. | DAC REGISTER (0x1N98; R/W)..... | 30 |
| 4.6. | ADC CONFIGURATION REGISTER (0x1N9C; R/W) | 31 |
| 4.7. | CHANNEL CONFIGURATION REGISTER (0x8000; R/W) | 31 |
| 4.8. | CHANNEL CONFIGURATION SET (0x8004; W) | 31 |
| 4.9. | CHANNEL CONFIGURATION RESET (0x8008; W) | 31 |
| 4.10. | BUFFER SIZE REGISTER (0x800C; R/W) | 32 |
| 4.11. | BUFFER CLEAR (0x8010; R/W)..... | 32 |
| 4.12. | RANDOM ACCESS CONFIGURATION (0x8014; R/W)..... | 32 |
| 4.13. | ACQUISITION CONTROL (0x8100; R/W)..... | 32 |
| 4.14. | ACQUISITION STATUS (0x8104; R)..... | 33 |
| 4.15. | SOFTWARE TRIGGER REGISTER (0x8108; W) | 33 |
| 4.16. | TRIGGER SOURCE ENABLE MASK (0x810C; R/W)..... | 34 |
| 4.17. | FRONT PANEL TRIGGER OUT ENABLE MASK (0x8110; R/W) | 34 |
| 4.18. | POST TRIGGER SETTING (0x8114; R/W) | 35 |
| 4.19. | FRONT PANEL I/O DATA (0x8118; R/W) | 35 |
| 4.20. | FRONT PANEL I/O CONTROL (0x811C; R/W) | 35 |
| 4.21. | CHANNEL ENABLE MASK (0x8120; R/W)..... | 36 |
| 4.22. | FIRMWARE REVISION (0x8124; R) | 36 |
| 4.23. | DOWNSAMPLE FACTOR (0x8128; R/W) | 36 |
| 4.24. | EVENT STORED REGISTER (0x812C; R)..... | 36 |
| 4.25. | VME CONTROL REGISTER (0xEF00; R/W) | 37 |
| 4.26. | VME STATUS REGISTER (0xEF04; R)..... | 37 |
| 4.27. | BOARD ID (0xEF08; R/W) | 37 |
| 4.28. | MCST BASE ADDRESS AND CONTROL REGISTER (0xEF0C; R/W)..... | 38 |

| | | |
|-----------|---|-----------|
| 4.29. | RELOCATION ADDRESS (0xEF10; r/w)..... | 38 |
| 4.30. | INTERRUPT STATUS ID (0xEF14; r/w) | 38 |
| 4.31. | INTERRUPT EVENT NUMBER (0xEF18; r/w) | 38 |
| 4.32. | BLT EVENT NUMBER REGISTER (0xEF1C; r/w) | 38 |
| 4.33. | SCRATCH (0xEF20; r/w) | 38 |
| 4.34. | SOFTWARE RESET REGISTER (0xEF24; w) | 39 |
| 4.35. | SOFTWARE CLEAR REGISTER (0xEF28; w) | 39 |
| 4.36. | FLASH ENABLE (0xEF2C; r/w) | 39 |
| 4.37. | FLASH DATA (0xEF30; r/w)..... | 39 |
| 5. | INSTALLATION | 40 |
| 5.1. | POWER ON SEQUENCE | 40 |
| 5.2. | POWER ON STATUS | 40 |
| 5.3. | FIRMWARE UPGRADE..... | 40 |

LIST OF FIGURES

| | | |
|------------|---|----|
| FIG. 1.1: | MOD. V1724 BLOCK DIAGRAM | 8 |
| FIG. 2.1: | MOD. V1724 FRONT PANEL | 10 |
| FIG. 2.2: | AMP CLK IN/OUT CONNECTOR | 11 |
| FIG. 2.3: | ROTARY AND DIP SWITCHES LOCATION | 13 |
| FIG. 3.1: | SINGLE ENDED INPUT DIAGRAM..... | 15 |
| FIG. 3.2: | DIFFERENTIAL INPUT DIAGRAM | 15 |
| FIG. 3.3: | CLOCK DISTRIBUTION DIAGRAM | 16 |
| FIG. 3.4: | DATA STORAGE IN GATE MODE..... | 17 |
| FIG. 3.5: | DATA STORAGE IN SAMPLE MODE..... | 18 |
| FIG. 3.6: | TRIGGER OVERLAP | 19 |
| FIG. 3.7: | BLOCK DIAGRAM OF TRIGGER MANAGEMENT | 21 |
| FIG. 3.8: | A24 ADDRESSING | 24 |
| FIG. 3.9: | A32 ADDRESSING | 24 |
| FIG. 3.10: | CR/CSR ADDRESSING | 24 |
| FIG. 3.11: | SOFTWARE RELOCATION OF BASE ADDRESS..... | 25 |
| FIG. 3.12: | EXAMPLE OF BLT READOUT | 26 |
| FIG. 3.13: | EXAMPLE OF RANDOM READOUT | 27 |

LIST OF TABLES

| | | |
|------------|---------------------------|---|
| TABLE 1.1: | MOD. V1724 VERSIONS | 7 |
|------------|---------------------------|---|

| | |
|---|----|
| TABLE 2.1: MODEL V1724 POWER REQUIREMENTS | 9 |
| TABLE 2.2 : FRONT PANEL LEDs | 12 |
| TABLE 2.3 : MOD. V1724 TECHNICAL SPECIFICATIONS | 14 |
| TABLE 3.1: BUFFER ORGANISATION | 18 |
| TABLE 3.2 : EVENT ORGANISATION..... | 20 |
| TABLE 3.3 : FRONT PANEL I/OS DEFAULT SETTING..... | 22 |
| TABLE 4.1: ADDRESS MAP FOR THE MODEL V1724 | 28 |
| TABLE 4.2: ROM ADDRESS MAP FOR THE MODEL V1724 | 30 |
| TABLE 4.3: OUTPUT BUFFER MEMORY BLOCK DIVISION..... | 32 |

1. General description

1.1. Overview

The Mod. V1724 is a 1-unit wide VME 6U module housing a 8 Channel 14 bit 100 MS/s Flash ADC Waveform Digitizer with threshold Auto-Trigger capabilities.

The board is available with different input range, memory and connector configuration, as summarised by the following table:

Table 1.1: Mod. V1724 versions

| Model | Input type | SRAM Memory | Optical link | Interface |
|---------|--------------|--------------|--------------|-----------|
| V1724LC | Single ended | 512 Ksamples | No | VME64 |
| V1724 | Single ended | 512 Ksamples | Yes | VME64 |
| V1724B | Single ended | 4 Msamples | Yes | VME64 |
| V1724C | Differential | 512 Ksamples | Yes | VME64 |
| V1724D | Differential | 4 Msamples | Yes | VME64 |
| V1724E | Single ended | 4 Msamples | Yes | VME64 |
| V1724F | Differential | 4 Msamples | Yes | VME64 |
| VX1724 | Single ended | 512 Ksamples | Yes | VME64X |
| VX1724B | Single ended | 4 Msamples | Yes | VME64X |
| VX1724C | Differential | 512 Ksamples | Yes | VME64X |
| VX1724D | Differential | 4 Msamples | Yes | VME64X |
| VX1724E | Single ended | 4 Msamples | Yes | VME64X |
| VX1724F | Differential | 4 Msamples | Yes | VME64X |

Single ended input versions, optionally, are available with ± 5 V dynamic range.

The DC offset of the signal can be adjusted channel per channel by means of a programmable 16bit DAC.

The board features a front panel clock/reference In/Out and a PLL for clock synthesis from internal/external references. This allows multi board phase synchronizations to an external standard or to a V1724 clock master board.

The data stream is continuously written in a circular memory buffer; when the trigger occurs the FPGA writes further N samples for the post trigger and freezes the buffer that then can be read either via VME or via Optical Link; the acquisition can continue without dead-time in a new buffer. Each channel has a SRAM memory, divided in buffers of programmable size.

The trigger signal can be provided via the front panel input as well as via the VMEbus, but it can also be generated internally, as soon as a programmable voltage threshold is reached. The individual Auto-Trigger of one channel can be propagated to the other channels and onto the front panel Trigger Output.

The VME interface is VME64X compliant and the data readout can be performed in Single Data Transfer (D32), 32/64 bit Block Transfer (BLT/MBLT) and Chained Block Transfer (CBLT); 2eVME and 2eSST modes are also supported.

The board houses a daisy chainable Optical Link able to transfer data at 125 MB/s, thus it is possible to connect up to eight V1724 (64 ADC channels) to a single Optical Link Controller (Mod. A2818, see Accessories/Controller).

The V1724 can be controlled and readout through the Optical Link in parallel to the VME interface.

The Mod. V1724LC is also available, a simplified version of the Mod. V1724, without Optical Link and Analog Monitor features.

1.2. Block Diagram

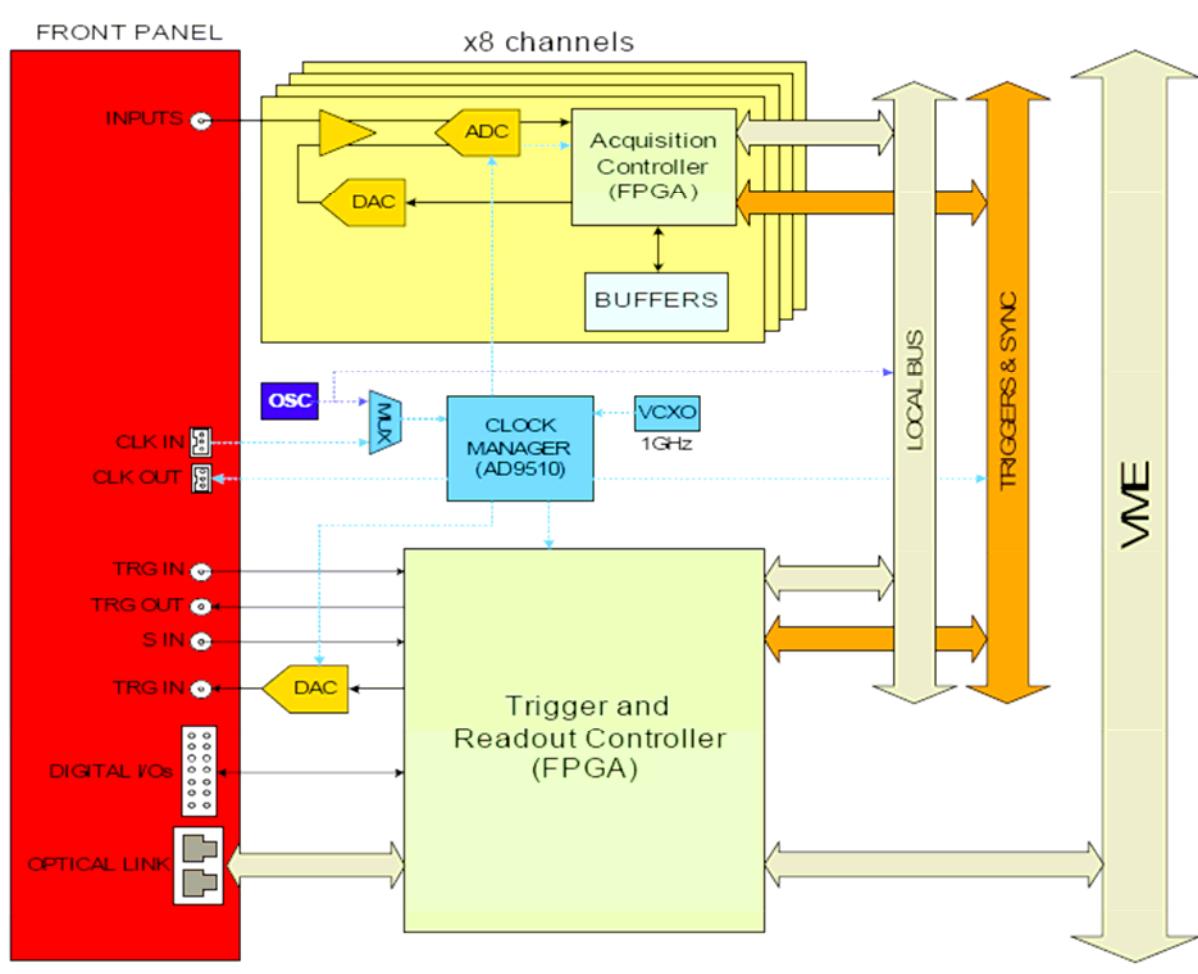


Fig. 1.1: Mod. V1724 Block Diagram

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

2. Technical specifications

2.1. Packaging

The module is housed in a 6U-high, 1U-wide VME unit. The board hosts the VME P1, and P2 connectors and fits into both VME standard and V430 backplanes.

2.2. Power requirements

The power requirements of the module are as follows:

Table 2.1: Model V1724 power requirements

| | |
|--------------|---------------|
| +5 V | 4.50 A |
| +12 V | 0.2 A |
| -12 V | 0.2 A |

2.3. Front Panel

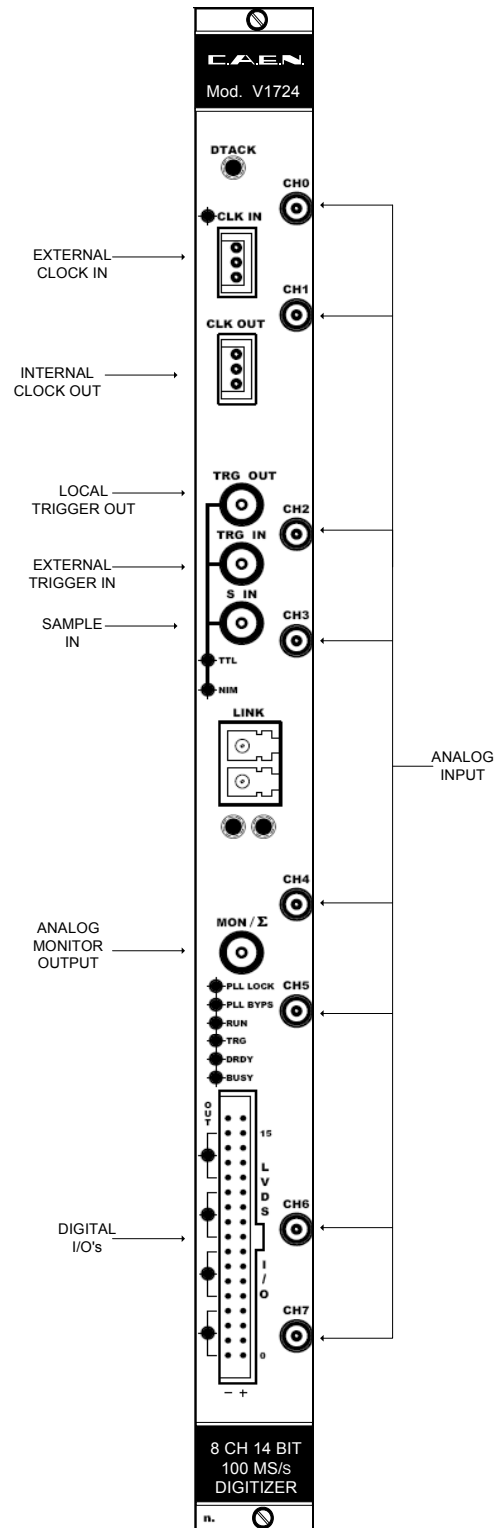


Fig. 2.1: Mod. V1724 front panel

2.4. External connectors

2.4.1. ANALOG INPUT connectors

Single ended version (see options in § 1.1):

Function:

Analog input, single ended, input dynamics: 2.25Vpp $Z_{in}=50\Omega$ (on request: 10Vpp $Z_{in}=1K\Omega$)

Mechanical specifications:

MCX connector (CS 85MCX-50-0-16 SUHNER)

Differential version (see options in § 1.1):

Function:

Analog input, differential, input dynamics: 2.25Vpp $Z_{in}=50\Omega$ or 10Vpp $Z_{in}=1K\Omega$

Mechanical specifications:

Tyco MODU II

2.4.2. CONTROL connectors

Function:

- TRG OUT: Local trigger output (NIM/TTL, high impedance)
- TRG IN: External trigger input (NIM/TTL, 50 Ohm)
- S IN: Sample front panel input (NIM/TTL, 50 Ohm)
- MON/ Σ : DAC output (not available on Mod. V1724LC)

Mechanical specifications:

00-type LEMO connectors

2.4.3. ADC REFERENCE CLOCK connectors

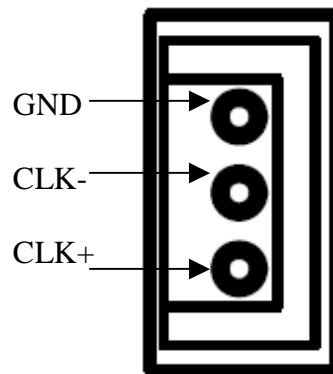


Fig. 2.2: AMP CLK IN/OUT Connector

Function:

CLK IN: External clock AC coupled (diff. LVDS, ECL, PECL, LVPECL, CML), $Z_{diff}= 110\Omega$.

Mechanical specifications:

AMP 3-102203-4 connector

Function:

CLOCK OUT: External clock AC coupled (diff. LVDS), $Z_{diff}= 110\Omega$.

Mechanical specifications:

AMP 3-102203-4 connector

2.4.4. Digital I/O connectors

Function: N.16 programmable differential LVDS I/O signals, Zdiff= 110 Ohm. Four Independent signal group 0÷3, 4÷7, 8÷11, 12÷15, In / Out direction control; see also § 3.5.

Mechanical specifications:

3M-7634-5002- 34 pin Header Connector

2.4.5. Optical LINK connector

Mechanical specifications:

LC type connector; to be used with Multimode 62.5/125µm cable with LC connectors on both sides; not featured on Mod. V1724LC

Electrical specifications:

Optical link for data readout and slow control with transfer rate up to 125MB/s; daisy chainable.

2.5. Other front panel components

2.5.1. Displays

The front panel hosts the following LEDs:

Table 2.2 : Front panel LEDs

| Name: | Colour: | Function: |
|----------|--------------|---|
| DTACK | green | VME read/write access to the board |
| CLK_IN | green | External clock enabled. |
| NIM | green | Standard selection for CLK I/O (V1724LC Rev.0), TRG OUT, TRG IN, S IN. |
| TTL | green | Standard selection for CLK I/O (V1724LC Rev.0), TRG OUT, TRG IN, S IN. |
| LINK | green/yellow | Network present; Data transfer activity |
| PLL_LOCK | green | |
| PLL_BYPS | green | |
| RUN | green | RUN bit set (see § 4.14) |
| TRG | green | |
| DRDY | green | Event/data (depending on acquisition mode) are present in the Output Buffer |
| BUSY | red | All the buffers are full |
| OUT_LVDS | green | Signal group OUT direction enabled. |

2.6. Internal components

ROTARY SWITCHES (SW2..5): *Type:* 4 rotary switches "Base Address [31:16]".
Function: Set the VME base address of the module.

CLOCK SWITCH (SW1): *Function:* Select clock source (External or Internal)

FW JUMPER (JP2): *Function:* Select the use of "Standard" or the "Backup" firmware revision at power on.

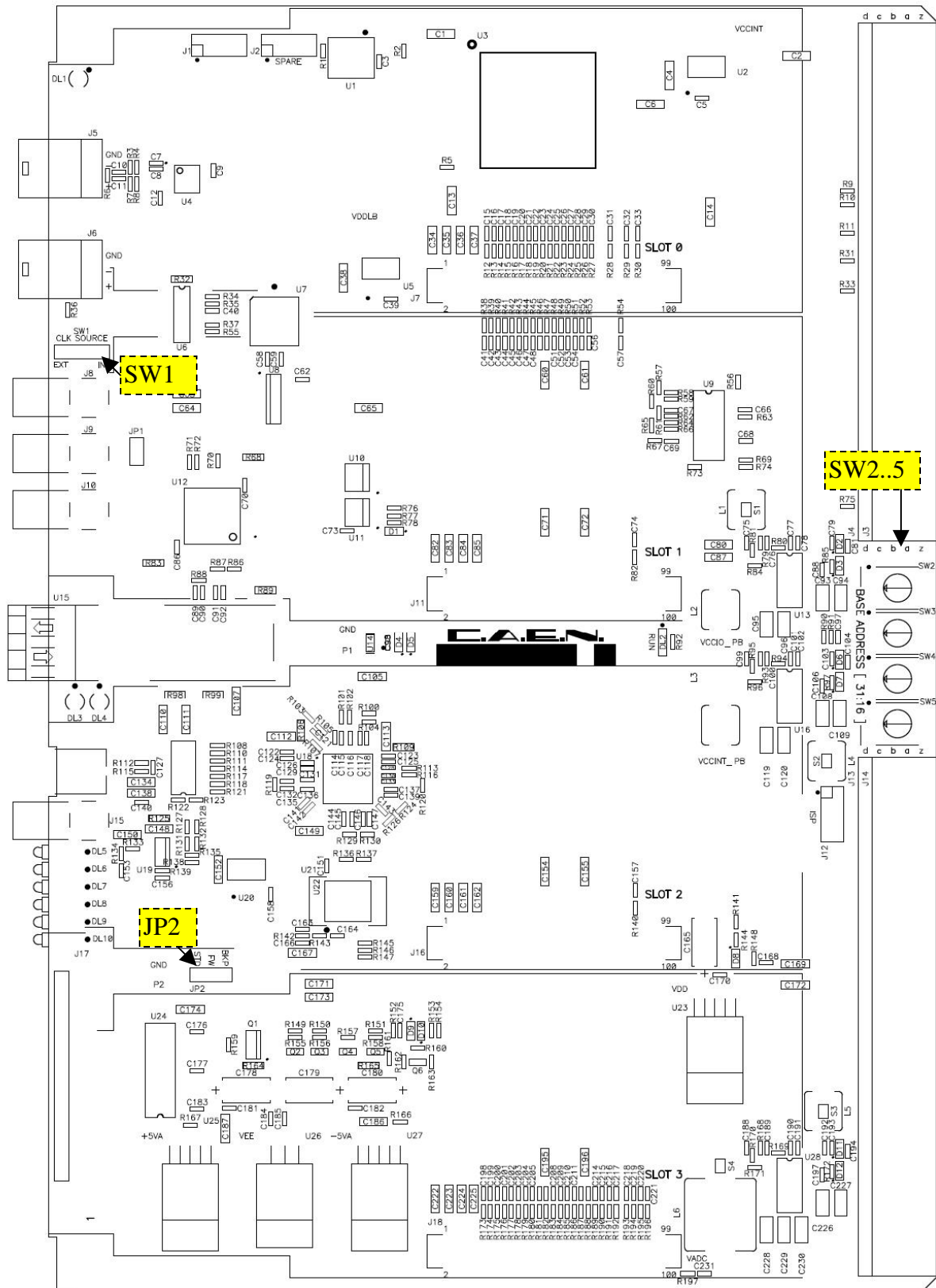


Fig. 2.3: Rotary and dip switches location

2.7. Technical specifications table

Table 2.3 : Mod. V1724 technical specifications

| | |
|--|--|
| Package | 1-unit wide VME 6U module |
| Analog Input | 8 channels, single-ended or differential (depending on version); 2.25Vpp (10Vpp Single-ended on request) input range, positive or negative; 40MHz Bandwidth; programmable DAC for Offset Adjust (Single-ended versions only) |
| Resolution | 14 bit |
| Sampling Clock | AC coupled, external differential clock input LVDS, ECL, PECL, LVPECL, CML (single ended NIM / TTL is also possible via special CAEN cable) in 10 ÷ 100MHz frequency range. Output clock (TTL / NIM). Internal 100MHz oscillator. Direct ADC feed or clock synthesis with low jitter programmable PLL (not available on Mod. V1724LC). Multi board synchronization (one board can act as clock master) External Clock Gate (NIM / TTL) for burst or single sampling mode |
| Memory | From 512K sample/ch to 4M sample/ch (see § 1.1); Multi Event Buffer with independent read and write access. Programmable event size and pre-post trigger. |
| Trigger | Common External TRG-IN (NIM or TTL) and VME Command Individual channel autotrigger (time over/under threshold) TRG-OUT (NIM or TTL) for the trigger propagation to other V1724 boards 32 bit Trigger Time Stamp |
| Trigger Time Stamp | 32bit – 10ns (43s range) Sync input for Time Stamp logging |
| FPGA signal process <i>(to be implemented)</i> | One Altera Cyclone EP1C4 or EP1C20 per channel Fully programmable digital filters (Deconvolution Moving Window, Trapezoidal FIR Filter, Pole-Zero Cancellation, Energy and Time extraction) (not available on Mod. V1724LC). Firmware downloadable via VME by the user |
| Optical Link | Data readout and slow control with transfer rate up to 125MB/s Daisy chainable: one A2818 PCI card can control and read eight V1724 boards in a chain (not available on Mod. V1724LC). |
| VME interface | VME64X compliant D32, BLT32, MBLT64, CBLT32/64, 2eVME, 2eSST, Multi Cast Cycles Transfer rate: 60MB/s (MBLT64), 100MB/s (2eVME), 160MB/s (2eSST) Sequential and random access to the data of the Multi Event Buffer The Chained readout allows to read one event from all the channels in a VME crate with a BLT access |
| Upgrade | V1724 firmware can be upgraded via VME |
| Software | General purpose C and LabView Libraries and Demo Programs |
| Analog Monitor | 12bit / 100MHz DAC output able to reproduce the waveform at the different stages of the filters Majority current sum (1mA per triggered channel, chainable between boards) (not available on Mod. V1724LC). |
| LVDS I/O | 16 general purpose LVDS I/O controlled by the FPGA Busy, Data Ready, Memory full, Individual Trig-Out and other function can be programmed An Input Pattern from the LVDS I/O can be associated to each trigger as an event marker |

3.2. Clock Management

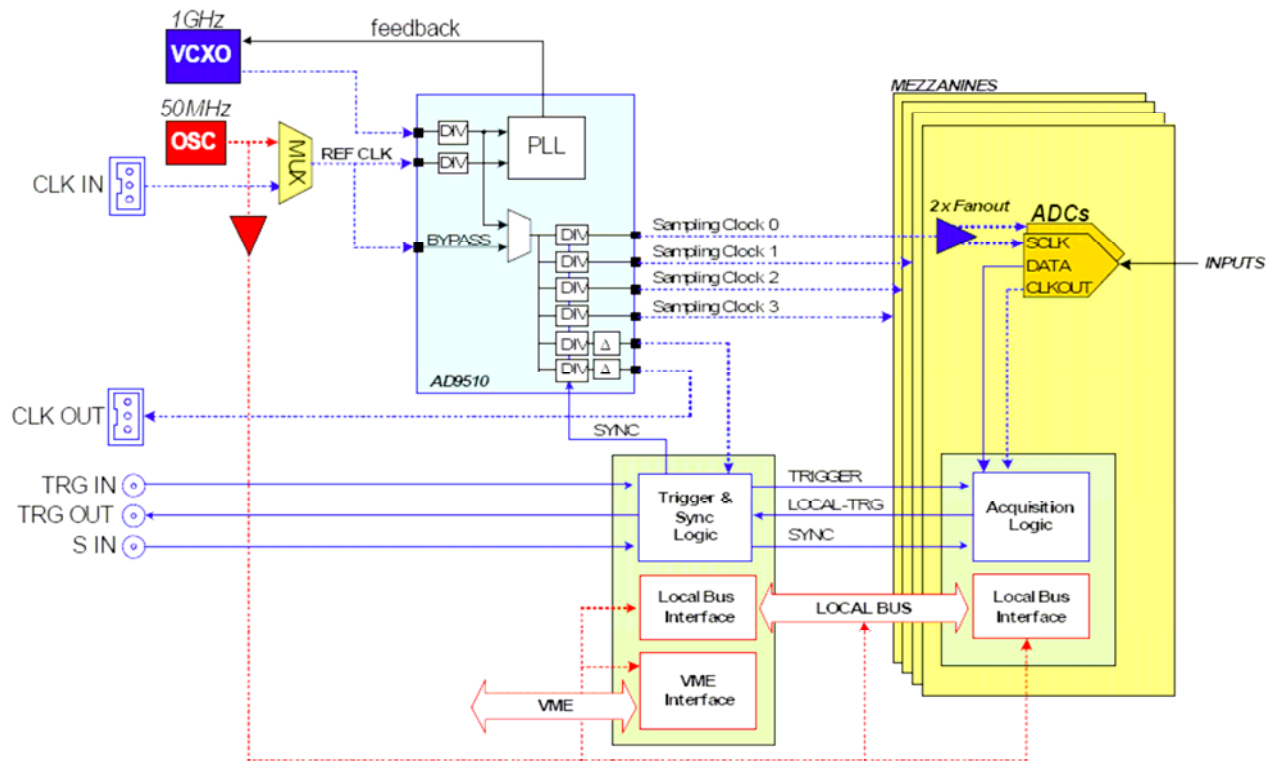


Fig. 3.3: Clock distribution diagram

3.2.1. Internal clock

The board can work using the internal 50 MHz Local Oscillator as clock source. This source is processed by the clock distribution electronics (AD9510 chip), which delivers the 100 MHz Sampling Clock to the ADCs. The 100 MHz Sampling Clock is fed also to the FPGA ROC (in order to allow trigger synchronisation) and to CLK_OUT (processed by a programmable divider and phase adjustment). Clock source (internal or external) is selected via internal switch SW1 (see § 2.6).

3.2.2. External clock and multiboard synchronisation

The board can work providing an external clock source (10÷100 MHz range). This source is processed by the clock distribution electronics (AD9510 chip), which delivers the 100 MHz Sampling Clock to the ADCs. The 100 MHz Sampling Clock is fed also to the FPGA ROC (in order to allow trigger synchronisation) and to CLK_OUT (processed by a programmable divider and phase adjustment). It is possible to adjust frequency and phase of CLK_OUT and use such signal as clock source for another board, and so on, in order to have the same clock source for all the boards.

Synchronisation is achieved, besides having the same clock source for all the boards, with all the boards sharing the same trigger signal (TRG_IN), and a common synchronisation (S_IN) signal: the latter is necessary in order to avoid a one Sampling Clock period jitter.

3.3. Acquisition Modes

3.3.1. Acquisition run/stop

The acquisition can be started in two ways, according to Acquisition Control register Bits [1:0] setting (see § 4.13):

- setting the RUN/STOP bit (bit[2]) in the Acquisition Control register (bits [1:0] of Acquisition Control must be set to REGISTER-CONTROLLED RUN MODE or S-IN CONTROLLED RUN MODE)
- driving S_IN signal high (bits [1:0] of Acquisition Control must be set to 01)

Subsequently acquisition is stopped either:

- resetting the RUN/STOP bit (bit[2]) in the Acquisition Control register (bits [1:0] of Acquisition Control must be set to REGISTER-CONTROLLED RUN MODE or S-IN CONTROLLED RUN MODE)
- driving S_IN signal low (bits [1:0] of Acquisition Control set to 01)

3.3.2. Gate and Sample mode acquisition

It is possible to use the S_IN signal (see § 2.4.2) as “gate” to enable samples storage. The samples produced by the 100 MHz ADC are stored in memory only if they are validated by the S_IN signal, otherwise they are rejected. Two operating modes are foreseen, as described in the following.

3.3.2.1. Gate mode

In Gate mode all the values sampled as the S-IN signal is active (high) are stored; for this purpose it is necessary to:

- Set bits [1:0] of Acquisition Control register to S-IN GATE MODE
- Set bit [0] of Channel Configuration Register (see § 4.7) to 0

All the values sampled as the S-IN signal is active (high) are stored.

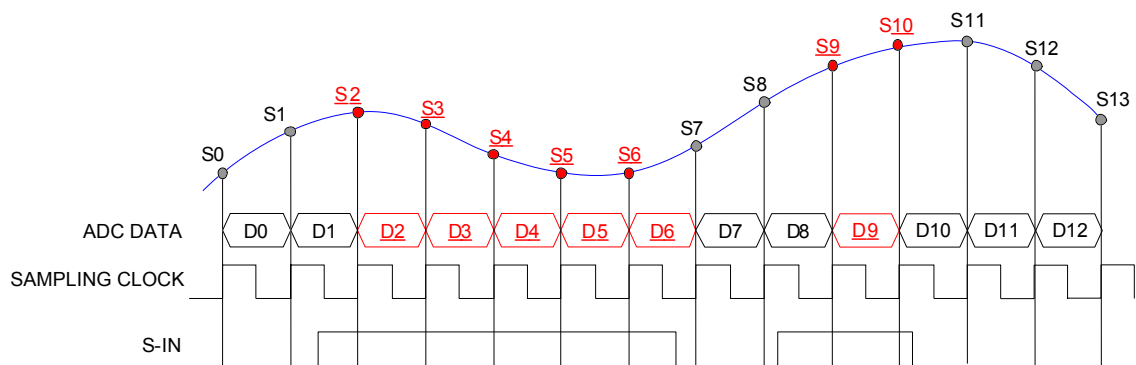


Fig. 3.4: Data storage¹ in Gate Mode

3.3.2.2. Sample mode

In Sample mode only the first value sampled after the S-IN signal leading edge is stored; for this purpose it is necessary to:

¹ underscored = stored

- Set bits [1:0] of Acquisition Control register to S-IN GATE MODE
- Set bit [0] of Channel Configuration Register (see § 4.7) to 1

Note that, if the S-IN signal is not synchronised with the sampling clock, then a 1 clock period jitter occurs between the S-IN leading edge and the actual sampling time.

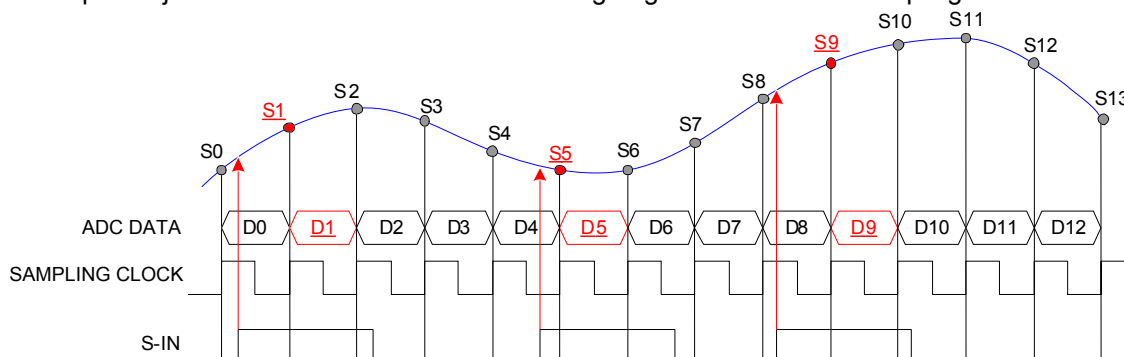


Fig. 3.5: Data storage² in Sample Mode

3.3.3. Acquisition Triggering: Samples and Events

When the acquisition is running, a trigger signal allows to:

- store a Trigger Time Tag (TTT): the value of a 32 bit counter which steps on with the sampling clock and represents a time reference
- increment the EVENT COUNTER (see § 4.24)
- fill the active buffer with the pre/post-trigger samples, whose number is programmable (Acquisition window width, § 4.18), freezing then the buffer for readout purposes, while acquisition continues on another buffer

Table 3.1: Buffer Organisation

| REGISTER (see § 4.10) | BUFFER NUMBER | SIZE of one BUFFER (samples) | |
|--------------------------|---------------|------------------------------|-------------------|
| | | SRAM 1MB/ch (512KS) | SRAM 8MB/ch (4MS) |
| 0x00 | 1 | 512K | 4M |
| 0x01 | 2 | 256K | 2M |
| 0x02 | 4 | 128K | 1M |
| 0x03 | 8 | 64K | 512K |
| 0x04 | 16 | 32K | 256K |
| 0x05 | 32 | 16K | 128K |
| 0x06 | 64 | 8K | 64K |
| 0x07 | 128 | 4K | 32K |
| 0x08 | 256 | 2K | 16K |
| 0x09 | 512 | 1K | 8K |
| 0x0A | 1024 | 512 | 4K |

An event is therefore composed by the trigger time tag, pre- and post-trigger samples and the event counter.

Overlap between “acquisition windows” may occur (a new trigger occurs while the board is still storing the samples related to the previous trigger); this overlap can be either rejected or accepted (programmable via VME).

² underscored = stored

If the board is programmed to accept the overlapped triggers, as the “overlapping” trigger arrives, the current active buffer is filled up, then the samples storage continues on the subsequent one.

In this case events will not have all the same size (see figure below).

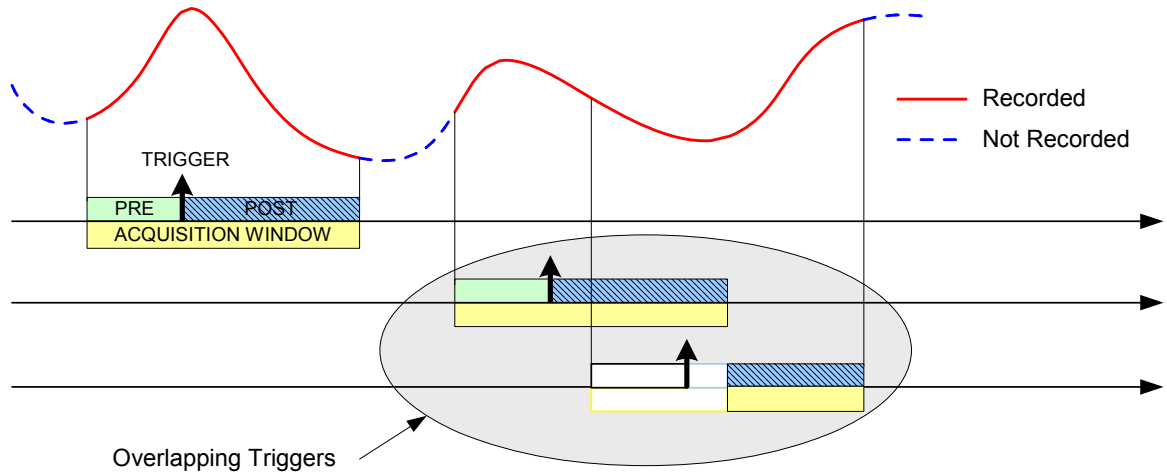


Fig. 3.6: Trigger Overlap

A trigger can be refused for the following causes:

- acquisition is not active
- memory is FULL and therefore there are no available buffers
- the required number of samples for building the pre-trigger of the event is not reached yet; this happens typically as the trigger occurs too early either with respect to the RUN_ACQUISITION command (see § 3.3.1) or with respect to a buffer emptying after a MEMORY_FULL status
- the trigger overlaps the previous one and the board is not enabled for accepting overlapped triggers

As a trigger is refused, the current buffer is not frozen and the acquisition continues writing on it. The Event Counter can be programmed in order to be either incremented or not. If this function is enabled, the Event Counter value identifies the number of the triggers sent (but the event number sequence is lost); if the function is not enabled, the Event Counter value coincides with the sequence of buffers saved and readout.

3.3.4. Event structure

An event is structured as follows:

- identifier (Trigger Time Tag, Event Counter)
- samples caught in the acquisition windows

The event is stored in the board memories and can be readout via VME; data format is 32 bit long word, therefore each long_word contains 2 samples.

The event format is the following:

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|----------------------|----|----|----|------------|----|----|----|---------------|----|----|----|----|----|----|----|----------------------|----|----|----|----|----|--------------|---|---|---|---|---|---|---|---|---|
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 1 | 0 | 1 | 0 | EVENT SIZE | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| BOARD-ID | | | | reserved | | | | | | | | | | | | | | | | | | CHANNEL MASK | | | | | | | | | |
| reserved | | | | | | | | EVENT COUNTER | | | | | | | | | | | | | | | | | | | | | | | |
| TRIGGER TIME TAG | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| SAMPLE [1] – CH[0] | | | | | | | | | | | | | | | | SAMPLE [0] – CH[0] | | | | | | | | | | | | | | | |
| SAMPLE [3] – CH[0] | | | | | | | | | | | | | | | | SAMPLE [2] – CH[0] | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| SAMPLE [N-1] – CH[0] | | | | | | | | | | | | | | | | SAMPLE [N-2] – CH[0] | | | | | | | | | | | | | | | |
| SAMPLE [1] – CH[1] | | | | | | | | | | | | | | | | SAMPLE [0] – CH[1] | | | | | | | | | | | | | | | |
| SAMPLE [3] – CH[1] | | | | | | | | | | | | | | | | SAMPLE [2] – CH[1] | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| SAMPLE [N-1] – CH[1] | | | | | | | | | | | | | | | | SAMPLE [N-2] – CH[1] | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| SAMPLE [1] – CH[7] | | | | | | | | | | | | | | | | SAMPLE [0] – CH[7] | | | | | | | | | | | | | | | |
| SAMPLE [3] – CH[7] | | | | | | | | | | | | | | | | SAMPLE [2] – CH[7] | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| SAMPLE [N-1] – CH[7] | | | | | | | | | | | | | | | | SAMPLE [N-2] – CH[7] | | | | | | | | | | | | | | | |

| |
|----------|
| HEADER |
| DATA CH0 |
| DATA CH1 |
| ... |
| DATA CH7 |

Table 3.2 : Event organisation

Header

It is composed by four words, namely:

- Size of the event (number of words)
- Board ID/Channel Mask (=1: enabled channels)
- Event Counter: It is the trigger counter; it can count either accepted triggers only, or all triggers (see § 4.13). Optionally, the trigger time tag can be replaced by a 16 bit pattern, latched on the LVDS I/O as one trigger arrives (see § 4.20) .
- Trigger Time Tag: It is a 32 bit counter (31 bit count + 1 overflow bit), which is reset either as acquisition starts or via front panel Reset signal (see § 3.7), and is incremented at each sampling clock hit. It is the trigger time reference.

Samples

Stored samples; data from masked channels are not read.

3.4. Trigger management

All the channels in a board share the same trigger: this means that all the channels store an event at the same time and in the same way (same number of samples and same position with respect to the trigger); several trigger sources are available.

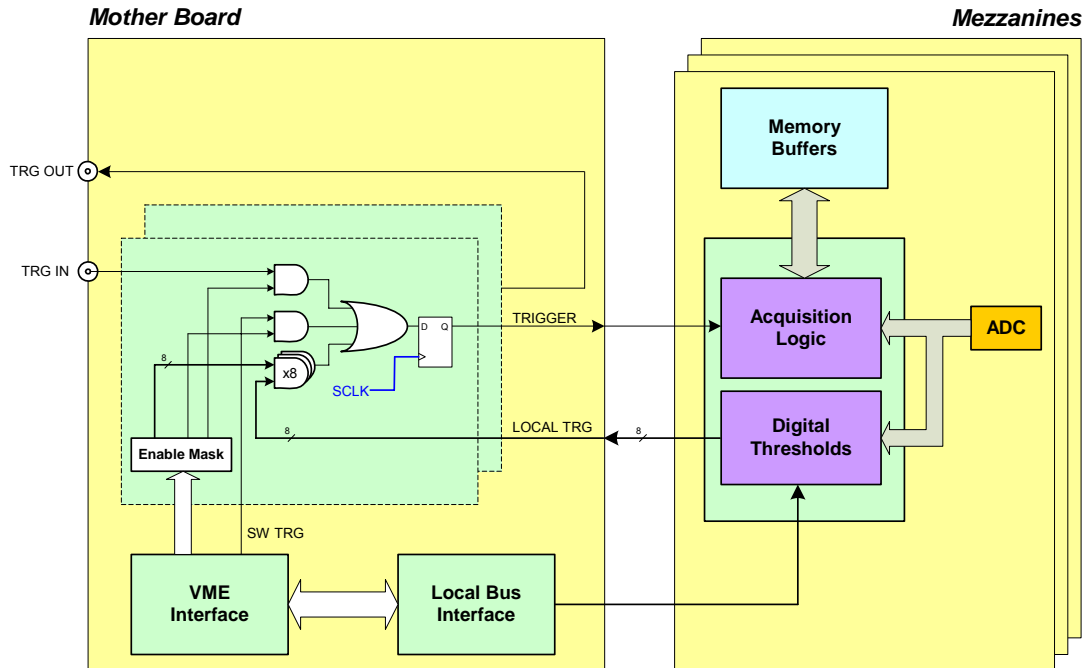


Fig. 3.7: Block diagram of Trigger management

3.4.1. External trigger

External trigger can be NIM/TTL signal on LEMO front panel connector, 50 Ohm impedance. The external trigger is synchronised with the internal clock (see § 0); if External trigger is not synchronised with the internal clock, a one clock period jitter occurs.

3.4.2. Software trigger

Software trigger are generated via VME bus (write access in the relevant register, see § 4.15).

3.4.3. Local channel auto-trigger

Each channel can generate a local trigger as the digitised signal exceeds the V_{th} threshold (ramping up or down, depending on VME settings), and remains under or over threshold for N_{th} samples at least (N_{th} is programmable via VME). The V_{th} digital threshold, the edge type, and the minimum number N_{th} of samples are programmable via VME register accesses.

N.B.: the local trigger signal does not start directly the event acquisition on the relevant channel; such signal is propagated to the central logic which produces the global trigger, which is distributed to all channels (see § 0).

3.4.4. Trigger distribution

The OR of all the enabled trigger sources, after being synchronised 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).

3.5. Front Panel I/Os

The V1724 is provided with 16 programmable general purpose LVDS I/O signals. Signals can be programmed via VME (see § 4.19 and § 4.20).

Default configuration is:

Table 3.3 : Front Panel I/Os default setting

| Nr. | Direction | Description |
|-----|-----------|-------------------------------------|
| 0 | out | Ch 0 Trigger Request |
| 1 | out | Ch 1 Trigger Request |
| 2 | out | Ch 2 Trigger Request |
| 3 | out | Ch 3 Trigger Request |
| 4 | out | Ch 4 Trigger Request |
| 5 | out | Ch 5 Trigger Request |
| 6 | out | Ch 6 Trigger Request |
| 7 | out | Ch 7 Trigger Request |
| 8 | out | Memory Full |
| 9 | out | Event Data Ready |
| 10 | out | Channels Trigger |
| 11 | out | RUN Status |
| 12 | in | Trigger Time Tag Reset (active low) |
| 13 | in | Memory Clear (active low) |
| 14 | -- | RESERVED |
| 15 | -- | RESERVED |

3.6. Analog inspection and Majority

3.6.1. Analog monitor

The board houses a 10bit (100MHz) DAC, whose input is controlled by the FPGA on the motherboard, which can provide the data stream from one channel's ADC (bypassing the trigger and data storage logic). The DAC output, available on MON/ Σ front panel connector, allows to view on an oscilloscope the signal digitised by the channel for debug purposes. Moreover, if the channel FPGA executes numerical filterings on the signal, it is possible to view "run time" the filtered signal, in order to test the filter and programmed parameters operation.

3.6.2. *Majority*

It is also possible to generate a Majority signal with the DAC: a current signal whose amplitude is instantaneously proportional to the number of channels which produced a local trigger. In this way, via an external discriminator, it is possible to produce a global trigger signal, as the number of triggering channels has exceeded a particular threshold. The majority signal can be daisy chained with other boards in the crate thus allowing to obtain a majority of all the channels of a set of boards. **N.B.: this feature is not available on the Mod. V1724LC**

3.7. Reset, Clear and Default Configuration

3.7.1. *Global Reset*

Global Reset is performed at Power ON of the module or via a VME RESET (SYS_RES), see § 4.34. 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 initialises all counters to their initial state and clears all detected error conditions.

3.7.2. *Memory Reset*

The Memory Reset clears the data off the Output Buffer.

The Memory Reset can be forwarded via either a write access to Software Clear Register (see § 4.35) or with a pulse sent to the front panel Memory Clear input (see § 3.5).

3.7.3. *Timer Reset*

The Timer Reset allows to initialize the timer which allows to tag an event. The Timer Reset can be forwarded with a pulse sent to Trigger Time Tag Reset input (see § 3.5).

3.8. VMEBus interface

The module is provided with a fully compliant VME64/VME64X interface (see § 1.1), whose main features are:

- EUROCARD 9U Format
- J1/P1 and J2/P2 with either 160 pins (5 rows) or 96 (3 rows) connectors
- A24, A32 and CR-CSR address modes
- D32, BLT/MBLT, 2eVME, 2eSST data modes
- MCST write capability

3.8.1. Addressing capabilities

3.8.1.1. Base address

The module works in A24/A32 mode. The Base Address of the module can be fixed through four rotary switches (see § 2.6) and is written into a word of 24 or 32 bit. The Base Address can be selected in the range:

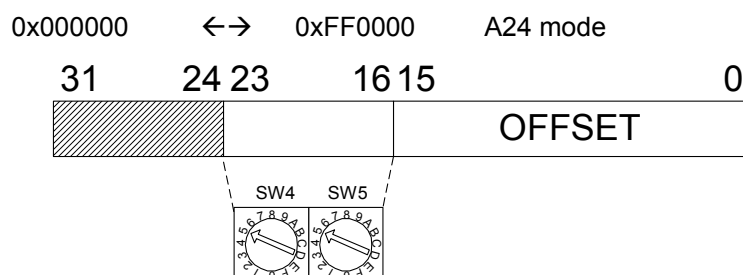


Fig. 3.8: A24 addressing

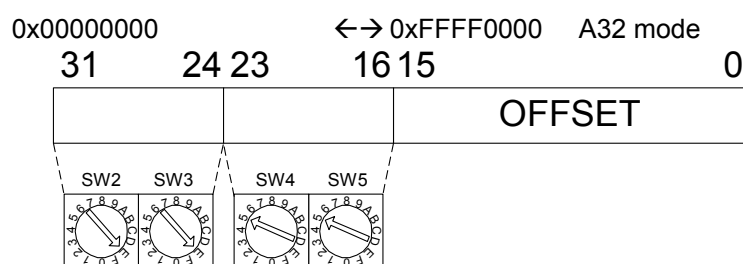


Fig. 3.9: A32 addressing

The Base Address of the module is selected through four rotary switches (see § 2.6), then it is validated only with either a Power ON cycle or a System Reset (see § 3.7).

3.8.1.2. CR/CSR address

GEO address is picked up from relevant backplane lines and written onto bit 23..19 of CR/CSR space, indicating the slot number in the crate; the recognised Address Modifier for this cycle is 2F. *This feature is implemented only on versions with 160pin connectors.*

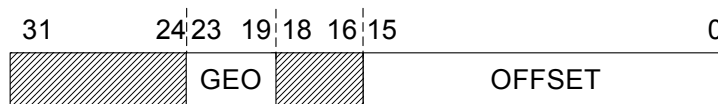


Fig. 3.10: CR/CSR addressing

3.8.1.3. Address relocation

Relocation Address register (see § 4.29) allows to set via software the board Base Address (valid values $\neq 0$). Such register allows to overwrite the rotary switches settings; its setting is enabled via VME Control Register (see § 4.25). The used addresses are:

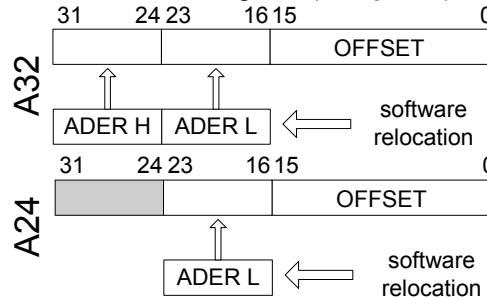


Fig. 3.11: Software relocation of base address

3.9. Data transfer capabilities

The board supports D32 single data readout, Block Transfer BLT32 and MBLT64, 2eVME and 2eSST cycles. Sustained readout rate is up to 60 MB/s with MBLT64, up to 100 MB/s with 2eVME and up to 160 MB/s with 2eSST.

3.10. Events readout

3.10.1. Sequential readout

The events, once written in the SRAMs (Memory Event Buffers), become available for readout via VME. 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, VMEBus does not handle directly the addresses, but takes 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 channels (from 0 to 7). Once an event is completed, the relevant memory buffer is emptied and ready to be written again (old data are erased). 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 (see also § 3.3.4).

3.10.1.1. SINGLE D32

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 § 3.3.4.

We suggest, after the 1st word is transferred, to check the Event Size information and then do as many D32 cycles as necessary (actually Event Size -1) in order to read completely the event and to avoid handling "filler" data.

3.10.1.2. BLOCK TRANSFER D32/D64, 2eVME

BLT32 allows, via a single channel access, to read N events in sequence, N is set via the BLT Event Number register (see § 4.32).

The event size depends on the Buffer Size Register setting (§ 4.10); namely:

$$[\text{Event Size}] = [8 * (\text{Block Size})] + [16 \text{ bytes}]$$

Then it is necessary to perform as many cycles as required in order to readout the programmed number of events.

We suggest to enable BERR signal during BLT32 cycles, in order to end the cycle avoiding filler readout. The last BLT32 cycle will not be completed, it will be ended by BERR after the #N event in memory is transferred (see example in the figure below).

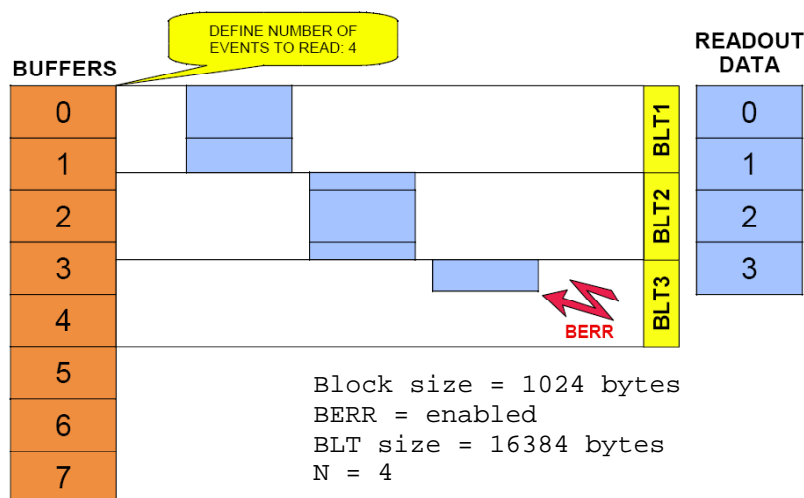


Fig. 3.12: Example of BLT readout

Since some 64 bit CPU's cut off the last 32 bit word of a transferred block, if the number of words composing such block is odd, it is necessary to add a dummy word (which has then to be removed via software) in order to avoid data loss. This can be achieved by setting the ALIGN64 bit in the VME Control register (see § 4.25).

MBLT64 cycle is similar to the BLT32 cycle, except that the address and data lines are multiplexed to form 64 bit address and data buses.

The 2eVME allows to achieve higher transfer rates thanks to the requirement of only two edges of the two control signals (DS and DTACK) to complete a data cycle.

3.10.1.3. CHAINED BLOCK TRANSFER D32/D64

The V1724 allows to readout events from more daisy chained boards (Chained Block Transfer mode).

The technique which handles the CBLT is based on the passing of a token between the boards; it is necessary to verify that the used VME crate supports such cycles.

Several contiguous boards, in order to be daisy chained, must be configured as "first", "intermediate" or "last" via MCST Base Address and Control Register (see § 4.28). A common Base Address is then defined via the same register; when a BLT cycle is executed at the address CBLT_Base + 0x0000 ÷ 0x0FFC, the "first" board starts to transfer its data, driving DTACK properly; once the transfer is completed, the token is passed to the second board via the IACKIN-IACKOUT lines of the crate, and so on until the "last" board, which completes the data transfer and asserts BERR (which has to be

enabled): the Master then ends the cycle and the slave boards are rearmed for a new acquisition.

If the size of the BLT cycle is smaller than the events size, the board which has the token waits for another BLT cycle to begin (from the point where the previous cycle has ended).

3.10.2. Random readout (to be implemented)

Events can be readout partially (not necessarily starting from the first available) and are not erased from the memories, unless a command is performed. In order to perform the random readout it is necessary to execute an **Event Block Request**, setting properly the Random Access Configuration register (see § 4.12).

Indicating the event to be read (page number = 12 bit datum), the offset of the first word to be read inside the event (12 bit datum) and the number of words to be read (size = 10 bit datum). At this point the data space can be read, starting from the header (which reports the required size, not the actual one, of the event), the Trigger Time Tag, the Event Counter and the part of the event required on the channel addressed in the Event Block Request.

After data readout, in order to perform a new random readout, it is necessary a new Event Block Request, otherwise Bus Error is signalled. In order to empty the buffers, it is necessary a write access to the Buffer Flush register (see § 4.11): the datum written is the number of buffers in sequence to be emptied.

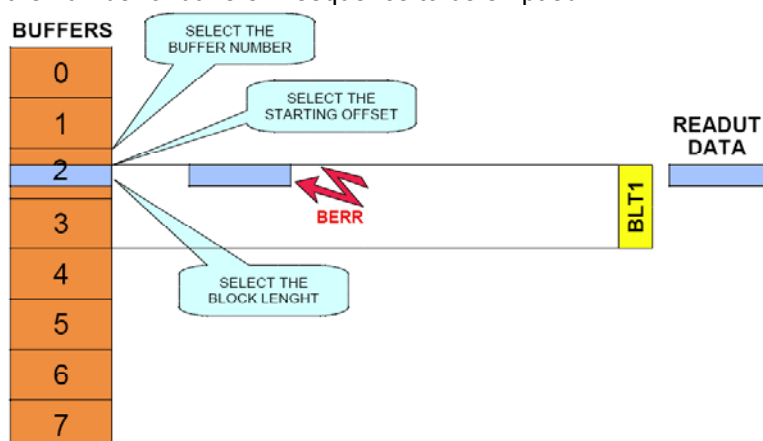


Fig. 3.13: Example of random readout

3.11. Optical Link

The board houses a daisy chainable Optical Link able to transfer data at 125 MB/s, therefore it is possible to connect up to eight V1724 (64 ADC channels) to a single Optical Link Controller (CAEN Mod. A2818).

The V1724 can be controlled and readout through the Optical Link in parallel to the VME interface.

The connection between the V1724 and the A2818 takes place through an optical fiber cable (Mod. AY2705, AY2720, AI2705, AI2720). The Optical Link allows to perform all the cycles foreseen by the VME64X.

N.B.: this feature is not available on the Mod. V1724LC

4. VME Interface

The following sections will describe in detail the board's VME-accessible registers content.

4.1. Registers address map

Table 4.1: Address Map for the Model V1724

| REGISTER NAME | ADDRESS | ASIZE | DSIZE | MODE | H_RES | S_RES | CLR |
|-------------------------------------|---------------|-------------|-------|------|-------|-------|-----|
| EVENT READOUT BUFFER | 0x0000-0x0FFC | A24/A32/A64 | D32 | R | | | |
| Channel n THRESHOLD | 0x1n80 | A24/A32 | D32 | R/W | | | |
| Channel n TIME OVER/UNDER THRESHOLD | 0x1n84 | A24/A32 | D32 | R/W | | | |
| Channel n STATUS | 0x1n88 | A24/A32 | D32 | R | | | |
| Channel n FIRMWARE REVISION | 0x1n8C | A24/A32 | D32 | R | | | |
| Channel n RESERVED | 0x1n90 | A24/A32 | D32 | R | | | |
| Channel n BUFFER OCCUPANCY | 0x1n94 | A24/A32 | D32 | R | | | |
| Channel n DAC | 0x1n98 | A24/A32 | D32 | R/W | | | |
| Channel n ADC CONFIGURATION | 0x1n9C | A24/A32 | D32 | R/W | | | |
| Channel n RESERVED | 0x1nA0 | A24/A32 | D32 | R | | | |
| CHANNEL CONFIGURATION | 0x8000 | A24/A32 | D32 | R/W | | | |
| CHANNEL CONFIGURATION SET | 0x8004 | A24/A32 | D32 | W | | | |
| CHANNEL CONFIGURATION RESET | 0x8008 | A24/A32 | D32 | W | | | |
| BUFFER SIZE | 0x800C | A24/A32 | D32 | R/W | | | |
| BUFFER CLEAR | 0x8010 | A24/A32 | D32 | R/W | | | |
| RANDOM ACCESS CONFIGURATION | 0x8014 | A24/A32 | D32 | R/W | | | |
| ACQUISITION CONTROL | 0x8100 | A24/A32 | D32 | R/W | | | |
| ACQUISITION STATUS | 0x8104 | A24/A32 | D32 | R | | | |
| SW TRIGGER | 0x8108 | A24/A32 | D32 | W | | | |
| TRIGGER SOURCE ENABLE MASK | 0x810C | A24/A32 | D32 | R/W | | | |
| FRONT PANEL TRIGGER OUT ENABLE MASK | 0x8110 | A24/A32 | D32 | R/W | | | |
| POST TRIGGER SETTING | 0x8114 | A24/A32 | D32 | R/W | | | |
| FRONT PANEL I/O DATA | 0x8118 | A24/A32 | D32 | R/W | | | |
| FRONT PANEL I/O CONTROL | 0x811C | A24/A32 | D32 | R/W | | | |
| CHANNEL ENABLE MASK | 0x8120 | A24/A32 | D32 | R/W | | | |

| REGISTER NAME | ADDRESS | ASIZE | DSIZE | MODE | H_RES | S_RES | CLR |
|----------------------------------|---------------|---------|-------|------|-------|-------|-----|
| FIRMWARE REVISION | 0x8124 | A24/A32 | D32 | R | | | |
| DOWNSAMPLE FACTOR | 0x8128 | A24/A32 | D32 | R/W | | | |
| EVENT STORED | 0x812C | A24/A32 | D32 | R | | | |
| VME CONTROL | 0xEF00 | A24/A32 | D32 | R/W | | | |
| VME STATUS | 0xEF04 | A24/A32 | D32 | R | | | |
| BOARD ID | 0xEF08 | A24/A32 | D32 | R/W | | | |
| MULTICAST BASE ADDRESS & CONTROL | 0xEF0C | A24/A32 | D32 | R/W | | | |
| RELOCATION ADDRESS | 0xEF10 | A24/A32 | D32 | R/W | | | |
| INTERRUPT STATUS ID | 0xEF14 | A24/A32 | D32 | R/W | | | |
| INTERRUPT EVENT NUMBER | 0xEF18 | A24/A32 | D32 | R/W | | | |
| BLT EVENT NUMBER | 0xEF1C | A24/A32 | D32 | R/W | | | |
| SCRATCH | 0xEF20 | A24/A32 | D32 | R/W | | | |
| SW RESET | 0xEF24 | A24/A32 | D32 | W | | | |
| SW CLEAR | 0xEF28 | A24/A32 | D32 | W | | | |
| FLASH ENABLE | 0xEF2C | A24/A32 | D32 | R/W | | | |
| FLASH DATA | 0xEF30 | A24/A32 | D32 | R/W | | | |
| CONFIGURATION RELOAD | 0xEF34 | A24/A32 | D32 | W | | | |
| CONFIGURATION ROM | 0xF000-0xF3FC | A24/A32 | D32 | R | | | |

4.2. Configuration ROM (0xF000-0xF084; r)

The following registers contain some module's information, they are D32 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 4.2: ROM Address Map for the Model V1724

| 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 | 0x00 |
| board2 | 0xF034 | 0x00 |
| board1 | 0xF038 | 0x06 |
| board0 | 0xF03C | 0xBC |
| 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.

4.3. Threshold Register (0x1n80; r)

| Bit | Function |
|--------|--|
| [13:0] | Threshold Value for Trigger Generation |

Each channel can generate a local trigger as the digitised signal exceeds the V_{th} threshold, and remains under or over threshold for N_{th} samples at least; this register allows to set V_{th} (LSB=); see also § 3.4.3.

4.4. Over/Under Threshold Register (0x1n84; r/w)

| Bit | Function |
|--------|-------------------------------------|
| [11:0] | Number of Data under/over Threshold |

Each channel can generate a local trigger as the digitised signal exceeds the V_{th} threshold, and remains under or over threshold for N_{th} samples at least; this register allows to set N_{th} ; see also § 3.4.3.

4.5. DAC Register (0x1n98; r/w)

| Bit | Function |
|---------|-----------------------|
| [23:16] | DAC configuration bit |
| [15:0] | DAC Data |

Bits [15:0] allow to define a DC offset to be added the input signal in the $-1.25V \div +1.25V$ range (low range) or in the $-1V \div +8V$ range (high range). Bits [23:16] allows to select

which group of channels is to be added with such offset (0x00=no offset, 0x10 even channels, 0x24 odd channels, 0x34 all channels). See also § 3.1.1

4.6. ADC Configuration Register (0x1n9C; r/w)

| Bit | Function |
|-----|---|
| [2] | ADC to Randomize the Output Binary Code |
| [1] | ADC Mode of Output Binary Code |
| [0] | ADC Dither |

This register allows to pilot the relevant ADC signals. See the LTC2208-14 14-Bit, 130Msps ADC data sheet for details.

4.7. Channel Configuration Register (0x8000; r/w)

| Bit | Function |
|-----|---|
| [7] | 0 = Trigger Output Generation Disabled 1 = Trigger Output Generation Enabled allows the channel to generate local trigger |
| [6] | 0 = Trigger Output on Input Over Threshold 1 = Trigger Output on Input Under Threshold allows to generate local trigger either on channel over or under threshold (see § 4.3 and § 4.4) |
| [5] | 0 = Trigger Input Disable 1 = Trigger Input Enable allows the channel to sense External Trigger Input |
| [4] | 0 = Memory Random Access 1 = Memory Sequential Access |
| [3] | 0 = Test Waveform Generation Disabled 1 = Test Waveform Generation Enabled |
| [1] | 0 = Trigger Overlapping Not Enabled 1 = Trigger Overlapping Enabled Allows to handle trigger overlap (see § 3.3.3) |
| [0] | 0 = "Window" Gate 1 = "Single Shot" Gate Allows to handle samples validation (see § 3.3.1) |

This register allows to perform settings which apply to all channels.

4.8. Channel Configuration Set (0x8004; w)

| Bit | Function |
|--------|---|
| [7..0] | Bits set to 1 means that the corresponding bits in the Channel Configuration register are set to 1. |

4.9. Channel Configuration Reset (0x8008; w)

| Bit | Function |
|--------|---|
| [7..0] | Bits set to 1 means that the corresponding bits in the Channel Configuration register are set to 0. |

4.10. Buffer Size Register (0x800C; r/w)

| Bit | Function |
|-------|-----------|
| [3:0] | SIZE CODE |

The SIZE CODE allows to divide the available Output Buffer Memory into a certain number of blocks, according to the following table:

Table 4.3: Output Buffer Memory block division

| SIZE CODE | Nr. of blocks | Memory locations | Block_size | Samples/block |
|-----------|---------------|------------------|------------|---------------|
| 0000 | 1 | 262144 | 1024K | 512K |
| 0001 | 2 | 131072 | 512K | 256K |
| 0010 | 4 | 65536 | 256K | 128K |
| 0011 | 8 | 32768 | 128K | 64K |
| 0100 | 16 | 16384 | 64K | 32K |
| 0101 | 32 | 8192 | 32K | 16K |
| 0110 | 64 | 4096 | 16K | 8K |
| 0111 | 128 | 2048 | 8K | 4K |
| 1000 | 256 | 1024 | 4K | 2K |
| 1001 | 512 | 512 | 2K | 1K |
| 1010 | 1024 | 256 | 1K | 512 |

4.11. Buffer Clear (0x8010; r/w)

| Bit | Function |
|--------|--|
| [11:0] | N = Clears the first N Output Buffer Memory Blocks, see § 4.10 |

4.12. Random Access Configuration (0x8014; r/w)

| Bit | Function |
|---------|--|
| [31:22] | OFFSET of the first word to be read inside the event |
| [21:10] | PAGE NUMBER OF SAMPLES TO READ |
| [0:9] | ADDRESS OF BUFFER TO READ |

This register allows to perform settings to be used during Random readout, see § 3.10.2

4.13. Acquisition Control (0x8100; r/w)

| Bit | Function |
|-------|---|
| [4] | 0 = DOWNSAMPLE DISABLED 1 = DOWNSAMPLE ENABLED allows to enable/disable downsampling, whose factor is set via Downsample Factor register (see § 4.23) |
| [3] | 0 = COUNT ACCEPTED TRIGGERS 1 = COUNT ALL TRIGGERS allows to reject overlapping triggers (see § 3.3.3) |
| [2] | 0 = Acquisition STOP 1 = Acquisition RUN allows to RUN/STOP Acquisition |
| [1:0] | 00 = REGISTER-CONTROLLED RUN MODE 01 = S-IN CONTROLLED RUN MODE 10 = S-IN GATE MODE 11 = MULTI-BOARD SYNC MODE |

Bits [1:0] description:

00 = REGISTER-CONTROLLED RUN MODE: multiboard synchronisation via S_IN front panel signal

- RUN control: start/stop via set/clear of bit[2]
- GATE always active (Continuous Gate Mode) or Downsample Mode
- Continuous Gate Mode can be used only if Channel gate mode (see § 4.7) is set in Window Mode
- Downsample Mode can be used prior DOWNSAMPLE FACTOR register (see § 4.23) valid setting (≠0)

01 = S-IN CONTROLLED RUN MODE: Multiboard synchronisation via S-IN front panel signal

- S-IN works both as SYNC and RUN_START command
- GATE always active (Continuous Gate Mode) or Downsample mode:
 - Continuous Gate Mode: Gate always active; to be used only if Channel Gate Mode (CHANNEL Configuration Register) is set to Window Mode
 - Downsample Mode: it is set via DOWNSAMPLE ENABLE and a value ≠0 at DOWNSAMPLE FACTOR register

10 = S-IN GATE MODE

- Multiboard synchronisation is disabled
- S-IN works as Gate signal set/clear of RUN/STOP bit

11 = MULTI-BOARD SYNC MODE

- Used only for Multiboard synchronisation

4.14. Acquisition Status (0x8104; r)

| Bit | Function |
|-----|---|
| [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 1 = RUN on |

4.15. Software Trigger Register (0x8108; w)

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

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

| Bit | Function |
|------|---|
| [31] | 0 = Software Trigger Disabled 1 = Software Trigger Enabled |
| [30] | 0 = External Trigger Disabled 1 = External Trigger Enabled |
| [7] | 0 = Channel 7 trigger disabled 1 = Channel 7 trigger enabled |
| [6] | 0 = Channel 6 trigger disabled 1 = Channel 6 trigger enabled |
| [5] | 0 = Channel 5 trigger disabled 1 = Channel 5 trigger enabled |
| [4] | 0 = Channel 4 trigger disabled 1 = Channel 4 trigger enabled |
| [3] | 0 = Channel 3 trigger disabled 1 = Channel 3 trigger enabled |
| [2] | 0 = Channel 2 trigger disabled 1 = Channel 2 trigger enabled |
| [1] | 0 = Channel 1 trigger disabled 1 = Channel 1 trigger enabled |
| [0] | 0 = Channel 0 trigger disabled 1 = Channel 0 trigger enabled |

This register bits[0,7] enable the channels to generate a local trigger as the digitised signal exceeds the Vth threshold (see § 3.4.3).

Bit0 enables Ch0 to generate the trigger, bit1 enables Ch1 to generate the trigger and so on.

EXTERNAL TRIGGER ENABLE (bit30) enables the board to sense TRG-IN signals

SW TRIGGER ENABLE (bit 31) enables the board to sense software trigger (see § 4.15).

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

| Bit | Function |
|------|---|
| [31] | 0 = Software Trigger Disabled 1 = Software Trigger Enabled |
| [30] | 0 = External Trigger Disabled 1 = External Trigger Enabled |
| [7] | 0 = Channel 7 trigger disabled 1 = Channel 7 trigger enabled |
| [6] | 0 = Channel 6 trigger disabled 1 = Channel 6 trigger enabled |
| [5] | 0 = Channel 5 trigger disabled 1 = Channel 5 trigger enabled |
| [4] | 0 = Channel 4 trigger disabled 1 = Channel 4 trigger enabled |
| [3] | 0 = Channel 3 trigger disabled 1 = Channel 3 trigger enabled |
| [2] | 0 = Channel 2 trigger disabled 1 = Channel 2 trigger enabled |
| [1] | 0 = Channel 1 trigger disabled 1 = Channel 1 trigger enabled |
| [0] | 0 = Channel 0 trigger disabled 1 = Channel 0 trigger enabled |

This register bits[0,7] enable the channels to generate a TRG_OUT front panel signal as the digitised signal exceeds the Vth threshold (see § 3.4.3).

Bit0 enables Ch0 to generate the TRG_OUT, bit1 enables Ch1 to generate the TRG_OUT and so on.

EXTERNAL TRIGGER ENABLE (bit30) enables the board to generate the TRG_OUT
SW TRIGGER ENABLE (bit 31) enables the board to generate the TRG_OUT (see § 4.15).

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

| Bit | Function |
|--------|--------------------|
| [31:0] | Post trigger value |

The number written in this register corresponds to $N_{\text{samples}} \times 2 + \text{Offset}$ (t.b.d.), with N_{samples} = number of post-trigger samples.

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

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

This register allows to:

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

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

| Bit | Function |
|-------|---|
| [15] | 0 = I/O Normal operations 1 = I/O Test Mode |
| [14] | 1 = TRIGGER OUT Value set to test mode |
| [7:6] | 00 = General Purpose I/O 01 = Programmed I/O 10 = Pattern mode: LVDS signals are input and their value is written into header TTT field (see § 3.3.4) |
| [5] | 0 = LVDS I/O 15..12 are inputs 1 = LVDS I/O 15..12 are outputs |
| [4] | 0 = LVDS I/O 11..8 are inputs 1 = LVDS I/O 11..8 are outputs |
| [3] | 0 = LVDS I/O 7..4 are inputs 1 = LVDS I/O 7..4 are outputs |
| [2] | 0 = LVDS I/O 3..0 are inputs 1 = LVDS I/O 3..0 are outputs |
| [1] | 0 = panel output signals (TRG-OUT/CLKOUT) enabled 1 = panel output signals (TRG-OUT/CLKOUT) enabled in high impedance |
| [0] | 0 = TRG/CLK are NIM I/O Levels 1 = TRG/CLK are TTL I/O Levels |

4.21. Channel Enable Mask (0x8120; r/w)

| Bit | Function |
|-----|---|
| [7] | 0 = Channel 7 disabled 1 = Channel 7 enabled |
| [6] | 0 = Channel 6 disabled 1 = Channel 6 enabled |
| [5] | 0 = Channel 5 disabled 1 = Channel 5 enabled |
| [4] | 0 = Channel 4 disabled 1 = Channel 4 enabled |
| [3] | 0 = Channel 3 disabled 1 = Channel 3 enabled |
| [2] | 0 = Channel 2 disabled 1 = Channel 2 enabled |
| [1] | 0 = Channel 1 disabled 1 = Channel 1 enabled |

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

4.22. Firmware Revision (0x8124; r)

| Bit | Function |
|--------|-----------------------|
| [15:8] | Firmware Revision (X) |
| [7:0] | Firmware Revision (Y) |

This register contains the firmware revision number coded on 16 bit. For example the REV. 1.2: 0000000100000010

4.23. Downsample Factor (0x8128; r/w)

| Bit | Function |
|--------|---|
| [31:0] | This register allows to set N : sampling frequency will be divided by $N+1$. Downsampling is enabled via Acquisition Control register; see § 4.13 |

4.24. Event Stored register (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.

4.25. VME Control Register (0xEF00; r/w)

| Bit | Function |
|--------|---|
| [6] | 0 = RELOC Disabled (BA is selected via Rotary Switch; see § 2.6) 1 = RELOC Enabled (BA is selected via RELOC register; see § 4.29) |
| [5] | 0 = ALIGN64 Disabled 1 = ALIGN64 Enabled (see § 3.10.1.2) |
| [4] | 0 = BERR Not Enabled; the module sends a DTACK signal until the CPU inquires the module 1 = BERR Enabled; the module is enabled either to generate a Bus error to finish a block transfer or during the empty buffer read out in D32 |
| [2 :0] | Interrupt level (0= interrupt disabled) |

4.26. VME Status Register (0xEF04; r)

| Bit | Function |
|-----|--|
| [3] | 0 = board not PURGED; 1 = board PURGED: the board has no data, otherwise if a CBLT is taking place, the board has already transferred all its data: in this case, the board leaves purged status as soon as the CBLT ends and the last board in the chain asserts BERR) |
| [2] | 0 = BERR FLAG: no Bus Error has occurred 1 = BERR FLAG: a Bus Error has occurred (this bit is re-set after a status register read out) |
| [1] | 0 = The Output Buffer is not FULL; 1 = The Output Buffer is FULL. |
| [0] | 0 = No Data Ready; 1 = Event Ready |

4.27. Board ID (0xEF08; r/w)

| Bit | Function |
|--------|---|
| [4 :0] | GEO (VME64X versions only; 0 on other versions) |

This register allows to write the correct GEO address of the module in this register before CBLT operation so that the GEO address will be contained in the EVENT HEADER Board ID field (see § 3.3.4).

4.28. MCST Base Address and Control Register (0xEF0C; r/w)

| Bit | Function |
|-------|---|
| [7:0] | These bits contain the most significant bits of the MCST/CBLT address of the module set via VME, i.e. the address used in MCST/CBLT operations. |
| [9:8] | Allows to set up the board for daisy chaining: 00 = disabled board 01 = last board 10 = first board 11 = intermediate |

4.29. Relocation Address (0xEF10; r/w)

| Bit | Function |
|---------|--|
| [15..0] | These bits contains the A31...A16 bits of the address of the module: it can be set via VME for a relocation of the Base Address of the module. |

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

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

4.31. 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

4.32. BLT Event Number Register (0xEF1C; r/w)

| Bit | Function |
|-------|---|
| [7:0] | This register contains the number of complete events which has to be transferred via BLT/CBLT (see § 3.10.1.2). |

4.33. Scratch (0xEF20; r/w)

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

4.34. Software Reset Register (0xEF24; w)

| Bit | Function |
|--------|--|
| [31:0] | A write access to this location allows to perform a software reset |

4.35. Software Clear Register (0xEF28; w)

| Bit | Function |
|--------|---|
| [31:0] | A write access to this location clears all the memories |

4.36. Flash Enable (0xEF2C; r/w)

| Bit | Function |
|-----|---|
| 0 | 0 = Flash write ENABLED 1 = Flash write DISABLED |

This register is handled by the Firmware upgrade tool.

4.37. 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. Installation

- The Mod. V1724 fits into all 6U VME crates.
- Turn the crate OFF before board insertion/removal
- Remove all cables connected to the front panel before board insertion/removal



CAUTION

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

5.1. Power ON sequence

To power ON the board follow this procedure:

1. insert the V1724 board into the crate
2. power up the crate

5.2. 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 (see § 4)

5.3. Firmware upgrade

The board can store two firmware versions, called STD and BKP respectively; at Power On, a microcontroller reads the Flash Memory and programs the module with the firmware version selected via the JP2 jumper (see § 2.6), which can be placed either on the STD position (left), or in the BKP position (right). It is possible to upgrade the board firmware via VME, by writing the Flash; for this purpose, download the software package available at:

<http://www.caen.it/nuclear/product.php?mod=V1724>

The package includes the new firmware release program file and a text file with C program examples which will guide the User through the development of the software necessary in order to update the Flash Memory.

N.B.: it is strongly suggested to upgrade ONLY one of the stored firmware revisions (generally the STD one): if both revision are simultaneously updated, and a failure occurs, it will not be possible to upload the firmware via VME again!