PRELIMINARY

Technical Information Manual

Revision n. 1
10 November 2006

MOD. V1724

8 CHANNEL 14 BIT
100 MS/S DIGITIZER

MANUAL REV.1

NPO:
00103/05:V1724x.MUTx/01

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.

CE

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.

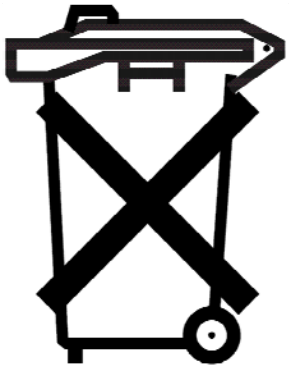

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

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 single ended analog input signal has a dynamic range of ±1.125 V or ±5 V (specify by order).
Two versions are available: V1724 and V1724LC; the two versions share all the same features except as otherwise indicated.
The DC offset of the signal can be adjusted channel per channel by means of a programmable 16bit DAC for bipolar or unipolar signals.
The ADC sampling clock can be provided either externally (10 to 100MHz range) or 100MHz internally via external special cable connection beetween CLK I/O out and CLK IN.
This allows multi board phase synchronizations to an external standard or to a V1724 clock master board. It is also possible to bypass the PLL and directly drive ADCs clock via external signal in 10 ÷ 100MHz range.
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 512K samples (2M samples specify by order) of memory, divided in buffers of programmable size from 500 sample x 1024 buffers (5 µs window) to 512K samples x 1 buffers (5.12 ms window).
The trigger signal can be provided externally 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).
The board houses a daisy chainable Optical Link (not available on Mod. V1724LC) 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 (Mod. A2818, see Accessories/Controller).
The V1724 can be controlled and readout through the Optical Link in parallel to the VME interface.

1.2. Block Diagram

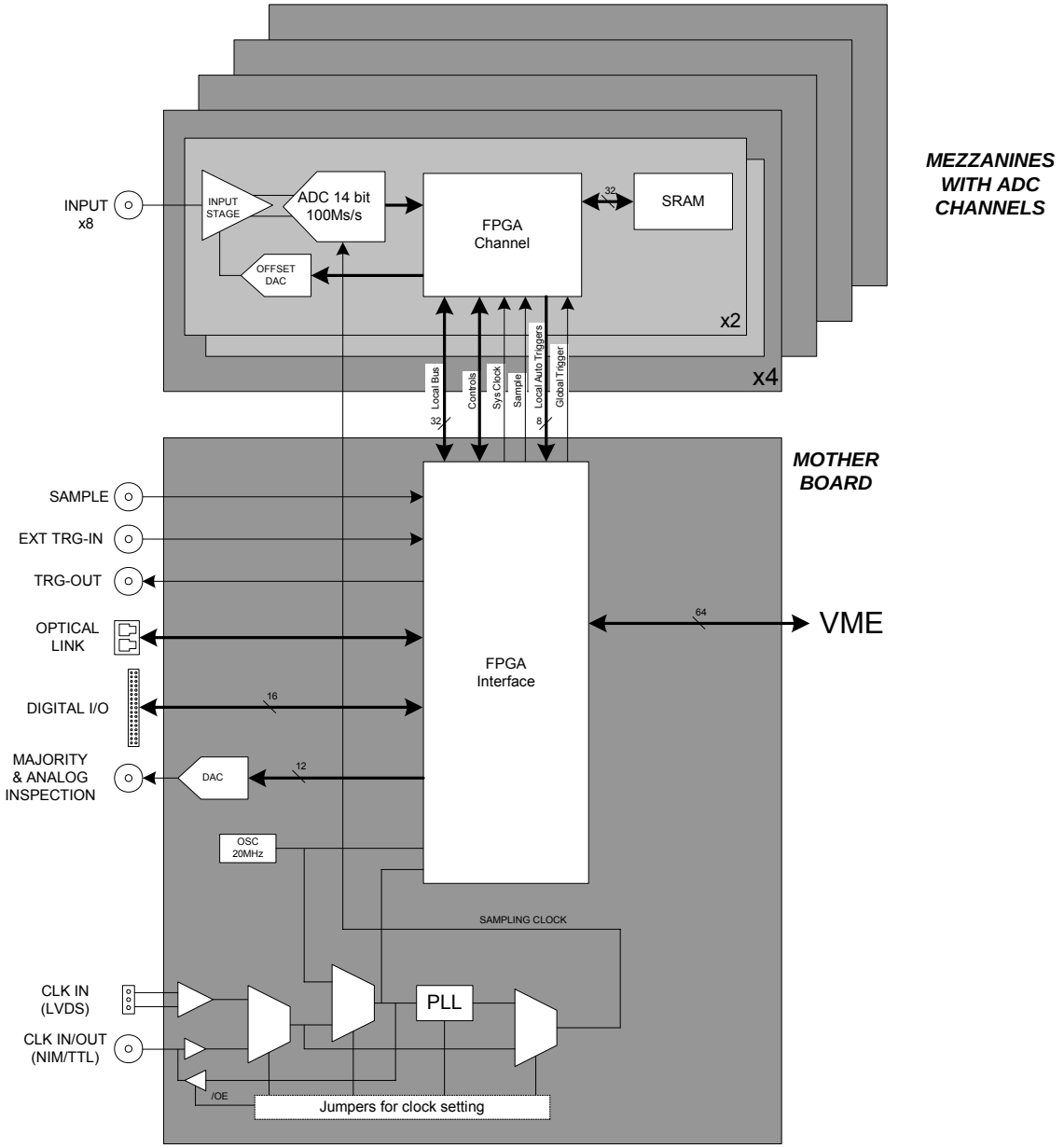

Fig. 1.1: Mod. V1724 Block Diagram

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

2. Functional description

2.1. Analog Input

The signal from the input connector can be either single ended (on MCX connector) or differential (on Tyco MODU II 3-pin connector). Input dynamics can be ±1.125V (Zin= 50 Ω) or ±5V (Zin= 1KΩ).
In single ended input configuration, a 16bit DAC allows to adjust a DC offset to the signal in the ±1.25V range (low range) or in the ±5V range (high range).
It is therefore possible to digitize bipolar signals with zero offset or unipolar signals either in the 0 to –2.25V, 0 to +2.25V, range (low range) or in the 0 to –10V, 0V to +10 V range (high range). Intermediate ranges are also available by free DACs value setting.
The input bandwidth ranges from DC to 40 MHz (with 2th order linear phase anti-aliasing filter).

2.2. Acquisition Modes

The Mod. V1724 ADC has a 0 ÷ 2 V input stage and executes the conversion at a typical 100 MHz frequency.
The acquisition begins via the START_ACQUISITION command: the samples coming from the ADC at the sampling frequency are continuously written into the memories organised as circular buffers with programmable size. Acquisition is stopped via STOP_ACQUISITION command.
When the acquisition is running, a trigger signal allows to:

- store a Trigger Time Tag: the value of a 32 bit counter which steps on with the sampling clock and represents a time reference
- increment the event counter
- fill the active buffer with the post-trigger samples (whose number is programmable), freezing then the buffer for readout purposes, while acquisition continues on another buffer

An event is therefore composed by pre- and post-trigger samples, the trigger time tag 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, via VME.
If the board is programmed to accept the overlapped triggers, then, as the “overlapping” trigger arrives, the current active buffer is “frozen” and the samples are stored on the subsequent one.
The previous event will be then “incomplete” (i.e. it has a number of samples smaller than the programmed one). Anyway the missing samples will be included in the subsequent event. The overlapped trigger is the one case where events have not all the same size.
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 START_ACQUISITION command 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 either incremented or not. If this function is enabled, the Event Counter value identifies the number of the trigger sent (but the event number sequence is lost); if the function is enabled, the Event Counter value coincides with the sequence of buffers saved and readout.
The sampling clock can be either external (10 to 100MHz range) or 100MHz via external cable connection beetween CLK I/O out and CLK IN. It can be also internally generated by a PLL (**not available for Mod. V1724LC**).
The sampling clock can be either external or internally generated by a PLL. In some cases the signal sampling must be performed at low frequency (lower than the ADC allowed range) or even be triggered by an asynchronous sampling signal. In such cases it is possible to use the Sample front panel input (NIM/TTL, 50 Ohm on LEMO connector), by enabling the External Sample Mode via VME bus.

The samples produced by the 100 MHz ADC are stored in memory only if they are validated by the Sample signal, otherwise they are rejected. The Sample signal can work either in Gate mode or in Edge mode. In Gate mode all the values sampled as the Sample signal is active (high) are stored, in Edge mode only the first value sampled after the Sample signal leading edge is stored.
Note that, if the Sample signal is not synchronised with the sampling clock, then a 1 clock period jitter occurs between the Sample leading edge and the actual sampling time.

In some cases it is necessary either to store the samples at a frequency slower than the sampling frequency (decimation) or at the arrival of an asynchronous sampling signal.
In both cases it is necessary to enable ***Sample Mode***: the “sample validation” via a proper signal. In other words, only some of the ADC samples are actually written in memory, the other are discarded. In its turn the ***Sample Mode*** foresees two cases:

– ***External Sample Signal (bit.)***: the **S-IN** front panel input (NIM or TTL) enables the sampling: the samples validated by the rising edge of S-IN (the first sample after the edge), are written in memory.
  ▪ If S-IN is not synchronised with the sampling clock, a one clock period jitter occurs.
– ***Sampling clock downscale (bit):*** it is possible to program the samples decimation, thus keeping only one sample every 2^N samples, where N is set via VME register.

In both cases, the relationship between the trigger and the samples (pre- and post-trigger of the acquistion window) is based on the number of samples, not on the timing. In other words, being fixed the number of samples, the duration of the acquisition window changes according to the timing of the validation signal.

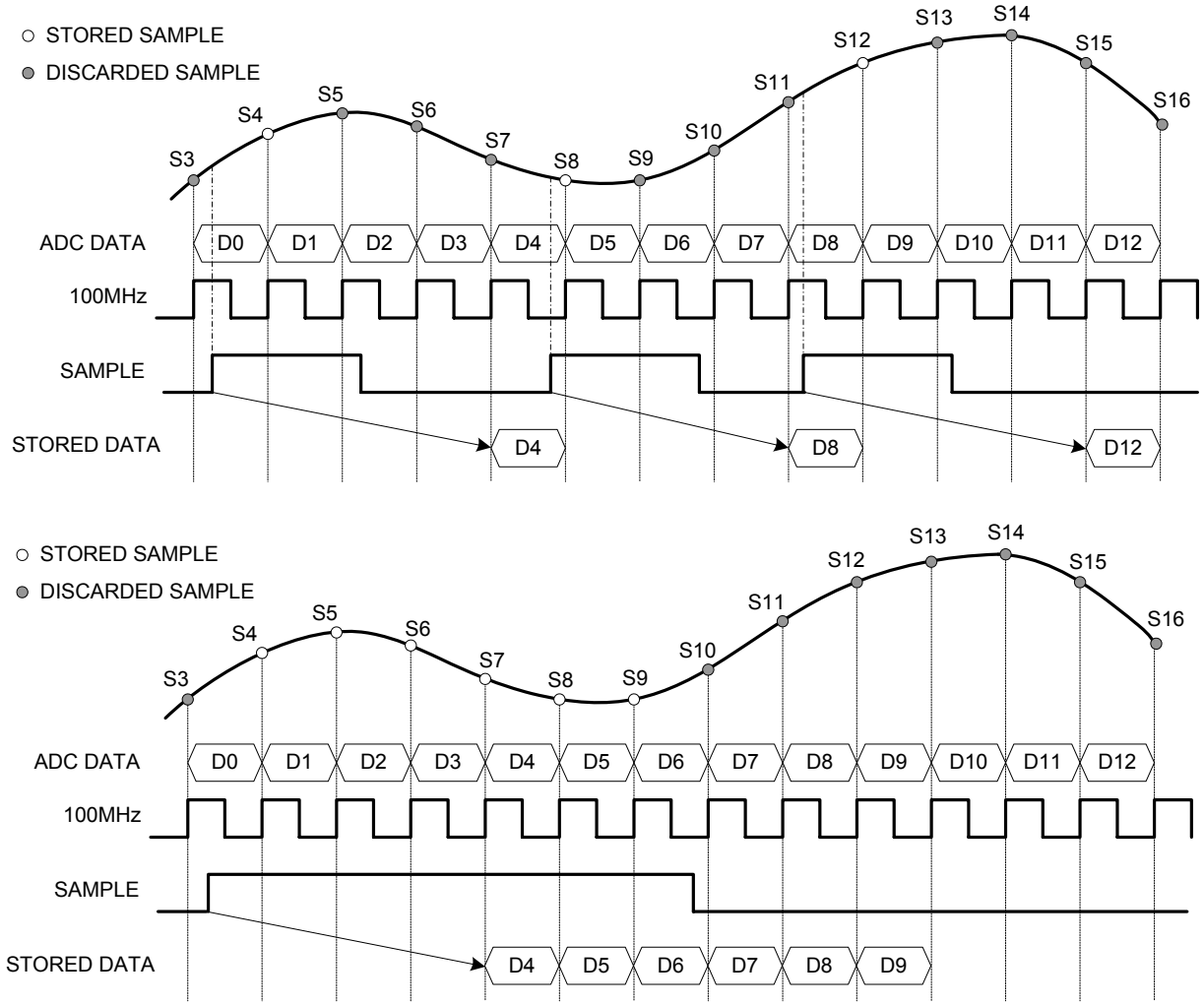

Fig. 2.1: Data storage in Gate Mode and Edge Mode

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

2.3. ADC Sampling Clock

The board works with typical sampling clock of 100MHz, 10 ÷ 100MHz range can be used. The sampling clock is fed externally on the ***CLK IN*** connector (AMP Modu II, differential LVDS/PECL, Zdiff= 110Ω).
Single ended NIM or TTL signals can be also received by using special CAEN cable.
CLK I/O (LEMO, NIM/TTL) output provides a 100 MHz clock, produced by an internal oscillator. In order to use such clock reference, CLK I/O out and CLK IN must be connected (via special CAEN cable). If several boards must be synchronised, it is possible to fed CLK I/O out to a Fan Out/Translator unit which provides the clock signal propagation to the other boards to be synchronised (including the board which generates the clock itself).

2.4. Trigger

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 tha same way (same number of samples and same position with respect to the trigger).
Several trigger sources are available:
external trigger: NIM/TTL signal on LEMO front panel connector, 50 Ohm impedance. The external is “merged” with the internal clock; if it is not synchronised, a one clock period jitter occurs.
Software trigger: signal generated via VME bus (write access in the relevant register).
Local channel auto-trigger: each channel can generate a local trigger as the digitised signal exceeds the Vth threshold (ramping up or down), and remains under or over threshold fo Nth samples at least. The Vth digital threshold, the edge type, and the minimum number Nth 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. It is not possible to store an event on a board single channel.

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 Lemo 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 towards 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.

2.5. 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, therefore each long word contains 2 samples.
The event format is the following:

Event Stucture	
31	**0**
HEADER	
EVENT COUNTER	
TIME TAG	
SAMPLE[1][CH0]	SAMPLE[0][CH0]
SAMPLE[3][CH0]	SAMPLE[2][CH0]
…	
SAMPLE[N-1][CH0]	SAMPLE[N-2][CH0]
SAMPLE[1][CH1]	SAMPLE[0][CH1]
SAMPLE[3][CH1]	SAMPLE[2][CH1]
…	
SAMPLE[N-1][CH1]	SAMPLE[N-2][CH1]
....	.....
....	.....
SAMPLE[1][CH7]	SAMPLE[0][CH7]
SAMPLE[3][CH7]	SAMPLE[2][CH7]
…	
SAMPLE[N-1][CH7]	SAMPLE[N-2][CH7]

Header content:
GEO Address of the board
Size of the event (number of words)

Trigger Time Tag
It is a 32 bit counter (31 bit count + 1 overflow bit), which is reset either as acquisition starts or via Reset front panel signal, and is incremented at each sampling clock hit. It is the trigger time reference.

Event Counter
It is the trigger counter; it can count either accepted triggers only either all triggers. Optionally, the trigger time tag can be replaced by a 16 bit pattern, latched on the LVDS I/O as one trigger arrives.

2.5.1. Readout

The events, once written in the SRAMs, become available for readout via VME. During the memory readout, the board can continue to store new 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, the User is not allowed to handle directly the addresses, but data are handled as if they were readout from a FIFO. In other words, a logical VME address corresponds to each channel, and data are read from the memories sequentially, according to the selected Readout Logic, from a memory space mapped on 4Kbytes (0x0000÷0x0FFC). Two Readout modes are possible:

Sequential Readout: 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 the 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 partially an event.

Random Readout: events can be readout partially (not necessarily starting from the first available) and are not erased from the memories, unless a comand is performed. In order to perform the random readout it is necessary to execute an ***Event Block Request***: a write access to the register (??)
Indicating the event to be read (12 bit = page number), the offset of the first word to be read inside the event (12 bit) and the number of words to be read (10 bit = size). 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 ***Free_Buffers*** register: the datum written is the number of buffers in sequence to be emptied.
In both modes it is possible to define a “Logic readout block” (LRB): a data block which is readout sequentially, interrupted as the last word is read. A LRB does not coincide necessarily with an event. With the Sequential Readout, a LRB coincide with a programmable NE number of events.
Readout modes are described in the following subsections.

2.5.2. SINGLE D32

This mode allows to readout a word per time. In case of Sequential Readout, it starts from the header of the first available event, followed by all the words until the end of the event, then the second event is transferred. Trigger Time Tag, the Event Counter and all the samples of the channels (from 0 to 7). If the memory is empty (no integer events available), a read access is either terminated by a BERR from the board, or with ”filler” datum (0xFFFFFFFF); it is important to notice that valid 0xFFFFFFFF words may also exist; therefore readout should be performed only if data are present (Data Ready bit set).
In case of Random Readout, the event sector can be read in D32, a word per time. After the last datum in the sector, the board works as if memory were empty: it is therefore necessary to perform another Event Block Request before reading other data.

2.5.3. BLOCK TRANSFER D32/D64

BLT32 and MBLT (D64) allow, via a single channel access, to read N subsequent data. The BLT size does not necessarily coincide with the size of the “Logic readout block” which is going to be read. Anyway, if during the BLT the end of the LRB is reached, after

the last word is transferred, the board interrups the cycle with a BERR; if BERR is not enabled, the cycle si completed with filler data. If the BLT cycle ends before the end of the LRB is reached, then the subsequent BLT starts from the end of the previous one, thus allowing to complete the LRB. If BERR is enabled and the end of the BLT coincides with the end of the LRB, it is necessary to execute another BLT cycle, which is terminated with a BERR (0 transferred words).

2.5.4. CHAINED BLOCK TRANSFER D32/D64

This mode allows to readout either one or more events from all the channels in a board, or from more daisy chained boards. In this case the LRB is composed by NE events of the first board, followed by NE events of the second and so on, until the last board.
The technique which handles the CBLT is based on the passing of a token between the boards.
Several boards, in order to be daisy chained, must be configured as “first”, “intermediate” or “last” via register (?). A common Base Address is then defined via VME; when a BLT cycles 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: 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 LBR size, the board which has the token waits for another BLT cycle to begin (from the point where the previous cycle has ended). The CBLT cycle is terminated when the end of the LBR is reached and BERR is asserted, regardless the number of performed BLT’s.

2.5.5. Optical Link Interface, Digital I/Os, Analog Inspection and Majority

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 LEMO 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.
Otherwise it is 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.

3. Technical specifications

3.1. Packaging

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

3.2. Power requirements

The power requirements of the modules' versions are as follows:

Table 3.1: Model V1724 power requirements

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

3.3. Front Panel

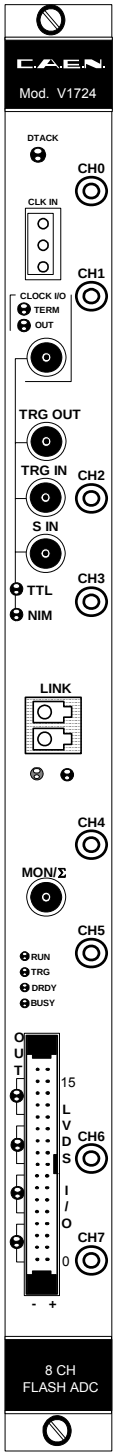

Fig. 3.1: Model V1724 front panel

3.4. External connectors

3.4.1. ANALOG INPUT connectors

Function:
Analog input, single ended, input dynamics: 2.25Vpp Zin=50Ω or 10Vpp Zin=1KΩ (choose by order)
Mechanical specifications:
MCX connector (CS 85MCX-50-0-16 SUHNER)

3.4.2. CONTROL connectors

Function:
- TRG OUT: Local trigger output (NIM/TTL, 50 Ohm)
- 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

3.4.3. ADC REFERENCE CLOCK connectors

Function:
CLK IN: External clock AC coupled (diff. LVDS or PECL), Zdiff= 110Ω.
Mechanical specifications:
AMP 3-102203-4 connector
Function:
CLOCK I/O: External clock (NIM/TTL, 50Ω)
Mechanical specifications:
00-type LEMO connectors

3.4.4. Digital I/O connectors

Function: N.16 differential LVDS I/O signals, Zdiff= 110ohm. Four Indipendent singnal group 0÷3, 4÷7, 8÷11, 12÷15, In / Out direction control.
Mechanical specifications:
3M-7634-5002- 34 pin Header Connector

3.4.5. EXTERNAL TRIGGER connectors

Mechanical specifications:
two 00-type LEMO connectors (bridged).
Electrical specifications:
Rising-edge active, NIM Zin= 50ohm; min. width 25 ns, double Trigger resolution: 75 ns.
This signal is internally or-wired with the TRG signal on the Control Connector (see § 3.4.1)

3.5. Other front panel components

3.5.1. Displays

The front panel (refer to Fig. 3.1) hosts the following LEDs:

Name:	Colour:	Function:
DTACK	green	Lights on VME read/write access to the board
TERM	red	CLK I/O 50Ω termination enabled.
OUT	red	CLK I/O OUT direction
NIM	green	Standard select for CLK I/O, TRG OUT, TRG IN, S IN.
TTL	green	Standard select for CLK I/O, TRG OUT, TRG IN, S IN.
LINK	green yellow	Network present. Data transfert activity.
RUN	green	
TRG	green	
DRDY	green	Lights when event/data (depending on acquisition mode) is present in the Output Buffer
BUSY	red	
OUT_LVDS	green	Lights on signal group OUT direction enabled.

3.6. Internal components

ROTARY SWITCHES: *Type:* 4 rotary switches “Base Address [31:16]”. *Function:* Set the VME base address of the module. See Figure below for their location.

TERM SWITCH: *Function:* Insert (ON) or remove (OFF) 50Ω termination on CLK I/O signal.

OUT SWITCH: *Function: Enable (ON) or Disable (OFF) Output drive on CLK I/O signal.*

FW JUMPER: Select the use of “Standard” or the “Backup” firmware type at power on.

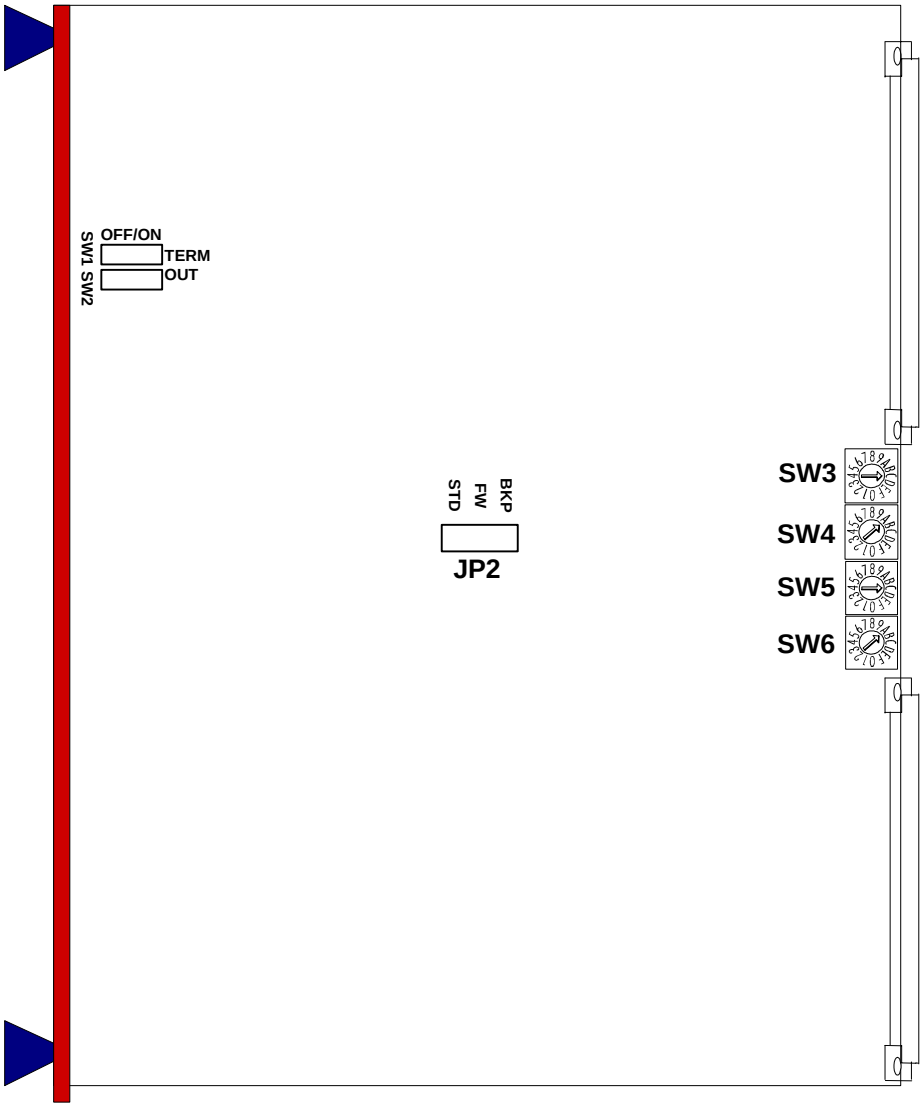

Fig. 3.2: Rotary and dip switches location

3.7. Technical specifications table

Table 3.2 : Model V1724 technical specifications

Package	1-unit wide VME 6U module
Analog Input	8 channels, 40MHz Bandwidth, 2,25Vpp or 10Vpp input range, positive or negative, programmable DAC for Offset Adjust. Single-ended (MCX - 50Ohm)
Resolution	14 bit
Sampling Clock	AC coupled, external differential clock input LVDS / PECL (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	512K sample/ch or 2M sample/ch, Multi Event Buffer with independent read and write access. Programmable event size and pre-post trigger. Up to 2048 events. 64 Ksample max record lenght (corresponding to 640us digitization)
Trigger	Common External TRG-IN (NIM or TTL) and VME Command Individual channel autotrigger (time over/under threshold, energy trigger or other programmable logic) TRG-OUT (NIM or TTL) for the trigger propagation to other V1724 boards 32 bit Trigger Time Stamp. Sync Signal to synchronize the internal time counter to an external reference
Trigger Time Stamp	32bit – 10ns (43s range) Sync input for Time Stamp logging
FPGA signal process	One 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 transer rate up to 80MB/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 and Multi Cast Cycles Transfer rate: 60MB/s 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

4. Operating modes

4.1. Installation

The Mod. V1724 fits into both V430 VME 6U (JAUX dataway) and standard 6U VME crates. The board supports live insertion/extraction into/from the crate, i.e. it is possible to insert or extract them from the crate without turning the crate off.
CAUTION: all cables connected to the front panel of the board must be removed before extracting/inserting the board from/into the crate.

CAUTION

ALL CABLES MUST BE REMOVED FROM THE FRONT PANEL

BEFORE EXTRACTING THE BOARD FROM THE CRATE!

4.2. Power ON sequence

To power ON the board follow this procedure:

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

4.3. 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 § 5);

4.4. Addressing capability

The module can be addressed in 2 different ways, specifically:

1. via Base Address;
2. via Multicast/Chained Block Transfer addressing mode.

4.4.1. Addressing via Base Address

The module works in A24/A32 mode. This implies that the module's address must be specified in a word of 24 or 32 bit. The Address Modifier codes recognised by the module are summarised in Table 4.1.

Table 4.1: Module recognised Address Modifier

A.M.	Description
0x3F	A24 supervisory block transfer
0x3E	A24 supervisory program access
0x3D	A24 supervisory data access
0x3C	A24 supervisory 64 bit block transfer
0x3B	A24 non privileged block transfer
0x3A	A24 non privileged program access
0x39	A24 non privileged User data access
0x38	A24 non privileged 64 bit block transfer
0x0F	A32 supervisory block transfer
0x0E	A32 supervisory program access
0x0D	A32 supervisory data access
0x0C	A32 supervisory 64 bit block transfer)
0x0B	A32 non privileged block transfer
0x0A	A32 non privileged program access
0x09	A32 non privileged data access
0x08	A32 non privileged 64 bit block transfer

The Base Address can be selected in the range:

0x000000	←→	0xFF0000	A24 mode
0x00000000	←→	0xFFFF0000	A32 mode

The Base Address of the module can be fixed through four rotary switches (see § 3.6)

4.4.2. Base addressing examples

The following is an example of Base Addressing for two boards inserted in a VME crate.

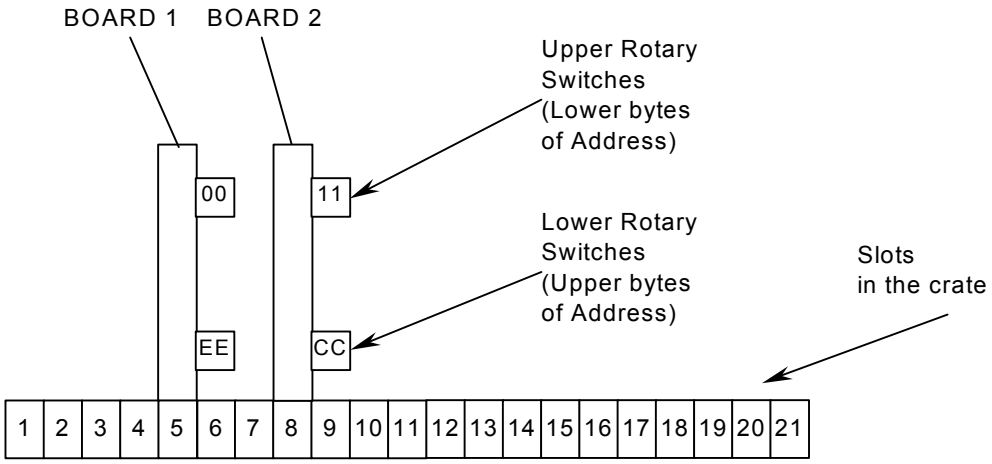

Fig. 4.1: Base Addressing: Example 1

If the board 1 and board 2 are respectively with the rotary switches for VME Base Addressing set as shown in the figure, the complete address of the registers of the two boards will be as follows:

Board 1:
Base addressing A32: 0xEE000000 + offset
Base addressing A24: 0x000000 + offset

Board 2:
Base addressing A32: 0xCC110000 + offset
Base addressing A24: 0x110000 + offset

4.4.3. MCST/CBLT addressing

When the Multicast/Chained Block Transfer addressing mode is adopted, the module works in A32 mode only. The Address Modifiers codes recognised by the module are:

MCST:	0x0E	A32 supervisory block transfer
	0x0D	A32 supervisory 64 bit block transfer
	0x0A	A32 non privileged block transfer
	0x09	A32 non privileged 64 bit block transfer
CBLT:	0x0F	A32 supervisory block transfer
	0x0C	A32 supervisory 64 bit block transfer
	0x0B	A32 non privileged block transfer
	0x08	A32 non privileged 64 bit block transfer

The boards can be accessed in Multicast Commands mode (MCST mode), that allows to write in the registers of several boards at the same time by accessing the MCST Base Address in A32 only once.

The boards can be accessed in Chained Block Transfer mode (CBLT mode) that allows to readout sequentially a certain number of contiguous boards in a VME crate. This access is allowed in BLT32 and BLT64 modes only to the MCST Base Address.

N.B.: The Base Address used for MCST and CBLT operations is the same, i.e. throughout this User's Manual the "MCST Base Address" identifies the same Address, used both for MCST commands (in Write only, for the allowed registers) and the CBLT Readout (in Read only, for the Output Buffer only).

The MCST Base Address must be set in a different way from the ordinary Base Address. Its most significant byte (i.e. bits 31 through 24 of base address) must be written in the MCST/CBLT Address Register and must be set in common to all boards belonging to the MCST/CBLT chain (i.e. all boards must have the same setting of the MCST/CBLT Base Address on bits 31 through 24). The default setting is 0xAA.
In CBLT/ MCST operations, the IACKIN/ IACKOUT daisy chain is used to pass a token from one board to the following one. The board which has received the token stores/sends the data from/to the master via MCST / CBLT access. No empty slots must therefore be left between the boards or, in alternative, empty slots can be left only in case VME crates with automatic IACKIN/IACKOUT short-circuiting are used.
Once the addresses have been set, the first and last board in a chain must have, respectively, only the FIRST_BOARD (F_B) and only the LAST_BOARD (L_B) bit set to 1 in the MCST Control Register. On the contrary, all intermediate boards must have both the FIRST_BOARD and the LAST_BOARD bits set to 1 (active, intermediate) or both the FIRST_BOARD and the LAST_BOARD bits set to 0 (inactive). By default these bits are set to 0 (the board is inactive).

Board status	Board position in the chain	F_B bit	L_B bit
inactive	-	0	0
active	last	0	1
active	first	1	0
active	intermediate	1	1

Please note that in a chain there must be one (and only one) *first board* (i.e. a board with F_B bit set to 1 and the L_B bit set to 0) and one (and only one) *last board* (i.e. a board with F_B bit set to 0 and the L_B bit set to 1).

The complete address in A32 mode is:

A [31:24] MCST/CBLT Address
A [23:16] 00
A [15:0] offset

In MCST/CBLT operation it is possible to define more chains in the same crate, but each chain must have an address different from the other.

N.B.: In CBLT operation the data coming from different boards are tagged with the HEADER and with the TRAILER words containing the GEO address in the 5 MSB (see § 4.4.2). It is up to the User to write the GEO address in the GEO register before executing the CBLT operation, since the V1724 are not enabled to pick the GEO address from the VME Bus. If the GEO address is not written in the relevant register before performing the CBLT operation, it will not be possible to identify the module which the data are coming from.

4.4.4. MCST/CBLT addressing examples

The following is an example of MCST and CBLT addressing for four V1724 boards plugged into a VME crate. To access the boards the steps to be performed are as follows:

1. Set the MCST address for all boards via VME Base Address;
2. Set the bits F_B and L_B of the MCST Control Register according to the operational status (active or inactive) of each board and to its position in the chain (first, intermediate or last);
3. Write or read the boards via MCST/CBLT addressing.

An example of User procedures which can be used to perform a write access is:

vme_write *(address, data, addr_mode, data_mode)*

which contain the following parameters:

Address:	the complete address, i.e. Base Address + offset;
Data:	the data to be either written;
Addr_mode:	the addressing mode (only A32 allowed in MCST);
Data_mode:	the data mode.

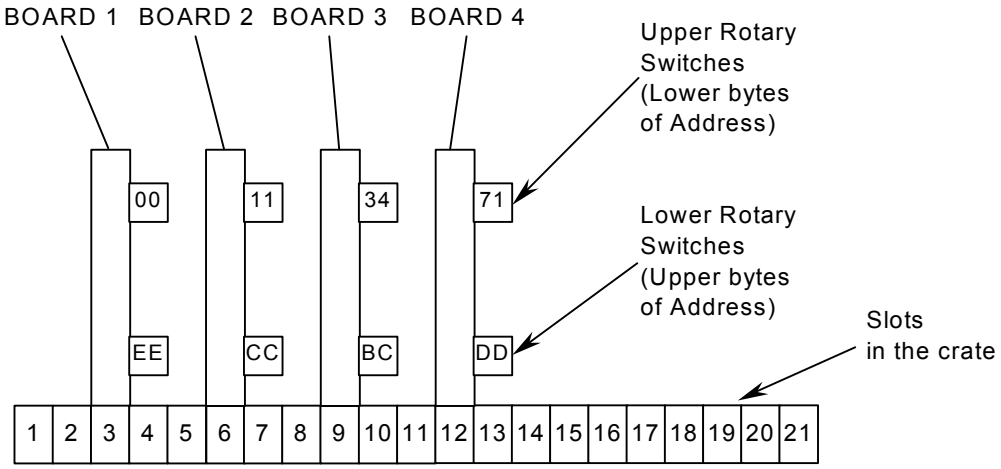

Fig. 4.2: MCST/CBLT Addressing Example

In the following, a software example, using the above mentioned procedures, is listed:

Example of Access via Base Address

```
vme_write (0xEE001010, 0xAA, A32, D16)   /* set MCST Address=0xAA for board 1 */
vme_write (0xCC111010, 0xAA, A32, D16)   /* set MCST Address=0xAA for board 2 */
vme_write (0xBC341010, 0xAA, A32, D16)   /* set MCST Address=0xAA for board 3 */
vme_write (0xDD711010, 0xAA, A32, D16)   /* set MCST Address=0xAA for board 4 */
vme_write (0xEE001012, 0x02, A32, D16)   /* set  board 1 = First */
vme_write (0xCC111012, 0x03, A32, D16)   /* set  board 2 = Active */
vme_write (0xBC341012, 0x00, A32, D16)   /* set  board 3 = Inactive */
vme_write (0xDD711012, 0x01, A32, D16)   /* set  board 4 = Last */
```

```
vme_write (0xAA001014, 0x00, A32, D16)   /* send SOFTWARE RESET to all the boards */
```

N.B.: there must be always one (and only one) FIRST BOARD and one (and only one) LAST BOARD.

4.5. Interrupter capability

The Mod. V1724 house a VME INTERRUPTER. The modules respond to D16 Interrupt Acknowledge cycles providing: a word whose 8 LSB are the STATUS/ID.

4.5.1. Interrupt Status/ID

The interrupt STATUS/ID is 8-bit wide, and it is contained in the 8 LSB of the Interrupt Vector Register. The register is available at the VME address: Base Address + 0x100C.

4.5.2. Interrupt Level

The module's interrupter produces its request on one of the 7 IRQ lines. The interrupt level is programmable via VME.

4.5.3. Interrupt Generation

An Interrupt is generated when the number of words stored in the memory equals the value written in the Almost Full Level Register at the VME address: Base Address + 0x1022. If the value in Interrupt Level Register is set to 0 the interrupt is disabled (default setting).

4.5.4. Interrupt Request Release

The INTERRUPTER removes its Interrupt request when either a Read Access is performed to the Output Buffer so that the number of events stored in the memory decreases and becomes less than the value written in the Almost Full Level Register or when a module's clear is performed.

4.6. Data transfer

4.6.1. Readout

The events, once written in the SRAMs, become available for readout via VME. During the memory readout, the board can continue to store new 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, the User is not allowed to handle directly the addresses, but data are handled as if they were readout from a FIFO. In other words, a logical VME address corresponds to each channel, and data are read from the memories sequentially, according to the selected Readout Logic, from a memory space mapped on 4Kbytes (0x0000÷0x0FFC). Two Readout modes are possible:

Sequential Readout: 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 the 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 partially an event.

Random Readout: events can be readout partially (not necessarily starting from the first available) and are not erased from the memories, unless a comand is performed. In order to perform the random readout it is necessary to execute an ***Event Block Request***: a write access to the register (??)
Indicating the event to be read (12 bit = page number), the offset of the first word to be read inside the event (12 bit) and the number of words to be read (10 bit = size). 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 ***Free_Buffers*** register: the datum written is the number of buffers in sequence to be emptied.
In both modes it is possible to define a “Logic readout block” (LRB): a data block which is readout sequentially, interrupted as the last word is read. A LRB does not coincide necessarily with an event. With the Sequential Readout, a LRB coincide with a programmable NE number of events.
Readout modes are described in the following subsections.

4.6.2. SINGLE D32

This mode allows to readout a word per time. In case of Sequential Readout, it starts from the header of the first available event, followed by all the words until the end of the event, then the second event is transferred. Trigger Time Tag, the Event Counter and all the samples of the channels (from 0 to 7). If the memory is empty (no integer events available), a read access is either terminated by a BERR from the board, or with ”filler” datum (0xFFFFFFFF); it is important to notice that valid 0xFFFFFFFF words may also

exist; therefore readout should be performed only if data are present (Data Ready bit set).
In case of Random Readout, the event sector can be read in D32, a word per time. After the last datum in the sector, the board works as if memory were empty: it is therefore necessary to perform another Event Block Request before reading other data.

4.6.3. BLOCK TRANSFER D32/D64

BLT32 and MBLT (D64) allow, via a single channel access, to read N subsequent data. The BLT size does not necessarily coincide with the size of the “Logic readout block” which is going to be read. Anyway, if during the BLT the end of the LRB is reached, after the last word is transferred, the board interrups the cycle with a BERR; if BERR is not enabled, the cycle si completed with filler data. If the BLT cycle ends before the end of the LRB is reached, then the subsequent BLT starts from the end of the previous one, thus allowing to complete the LRB. If BERR is enabled and the end of the BLT coincides with the end of the LRB, it is necessary to execute another BLT cycle, which is terminated with a BERR (0 transferred words).

4.6.4. CHAINED BLOCK TRANSFER D32/D64

This mode allows to readout either one or more events from all the channels in a board, or from more daisy chained boards. In this case the LRB is composed by NE events of the first board, followed by NE events of the second and so on, until the last board.
The technique which handles the CBLT is based on the passing of a token between the boards.
Several boards, in order to be daisy chained, must be configured as “first”, “intermediate” or “last” via register (?). A common Base Address is then defined via register ???; when a BLT cycles 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: 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 LBR size, the board which has the token waits for another BLT cycle to begin (from the point where the previous cycle has ended).
The CBLT cycle is terminated when the end of the LBR is reached and BERR is asserted, regardless the number of performed BLT’s.

4.6.5. Optical Link Interface, Digital I/Os, Analog Inspection and Majority

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 LEMO 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.
Otherwise it is 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

4.7. 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, which can be placed either on the STD position, or in the BKP position. 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 simoultaneously updated, and a failure occurs, it will not be possible to upload the firmware via VME again!

5. VME Interface

5.1. Register address map

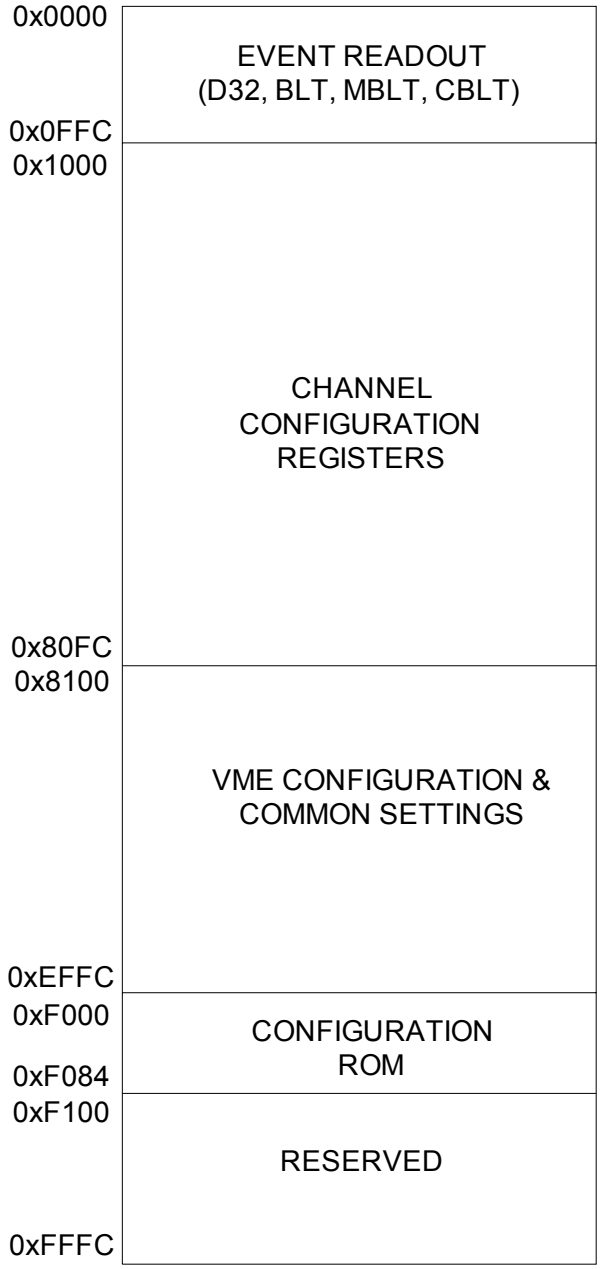

Fig. 5.1: Memory map

The following table shows the addresses of the VME configuration registers and Common Settings registers. They occupy VME memory space between 0x8100 and 0xEFFC (see Memory map).

Table 5.1: Address Map for the Model V1724

Register	Register Address	ADDR SIZING	DATA SIZING	Mode
VME Control	0x8104	A24/A32	D32	R/W
VME Status	0x8106	A24/A32	D16	R
Soft Intterrupt Level	0x8108	A24/A32	D16	R/W
Soft Interrupt Vector	0x810A	A24/A32	D16	R/W
Soft GEO address	0x810C	A24/A32	D16	R
MultiCast Base Address	0x810E	A24/A32	D16	R/W
Multicast Control	0x8110	A24/A32	D16	R/W
SW Reset	0x8112	A24/A32	D16	W
SW CLear	0x8114	A24/A32	D16	W
SW Trigger	0x8116	A24/A32	D16	W
Trigger Enable	0x811C	A24/A32	D16	R/W
BLT Event Number	0x811E	A24/A32	D16	R/W
VME FPGAirmware Revision	0x8120	A24/A32	D32	R
Reserved	0x8124	A24/A32	D32	R/W
Reserved	0x812C	A24/A32	D16	R/W
Reserved	0x812E	A24/A32	D16	R/W
Reserved	0x8130	A24/A32	D16	R/W
Reserved	0x8132	A24/A32	D32	R/W
Post Trigger Setting	0x8136	A24/A32	D32	R/W
Front Panel I/O Data	0x813C	A24/A32	D16	R/W
Front Panel I/O Control	0x813E	A24/A32	D16	R/W
Reserved	0x8140	A24/A32	D16	R/W

The following table shows the addresses of the channel FPGA registers.
One register complete address (32 bit) is composed by a Channel Address (8 bit) and an Internal Register Address.
The Channel Address MSB allows to select between broadcast addressing (=1) to all FPGAs and to individual channel (=0); the 3 LSBs of Channel Address allows to set the individual channel number.

Table 5.2: Address Map of the Channel Configuration Registers

Register	Register Address									Mode
	Channel Address								Internal Register Address	
Control	1	0	0	0	0	0	0	0	0x00	W
Control	0	0	0	1	0	a[2]	a[1]	a[0]	0x00	R
Control (Set)	1	0	0	0	0	0	0	0	0x04	W
Control (Reset)	1	0	0	0	0	0	0	0	0x08	W
Buffer Size	1	0	0	0	0	0	0	0	0x0C	W
Buffer Size	0	0	0	1	0	a[2]	a[1]	a[0]	0x0C	R
Page Memory Removed	1	0	0	0	0	0	0	0	0x10	W
Page Memory Removed	0	0	0	1	0	a[2]	a[1]	a[0]	0x10	R
Read Configuration	1	0	0	0	0	0	0	0	0x14	W
Read Configuration	0	0	0	1	0	a[2]	a[1]	a[0]	0x14	R
Threshold	1	0	0	0	0	0	0	0	0x80	W
Threshold	0	0	0	1	0	a[2]	a[1]	a[0]	0x80	R
Under/Over Threshold	1	0	0	0	0	0	0	0	0x84	W
Under/Over Threshold	0	0	0	1	0	a[2]	a[1]	a[0]	0x84	R
Status	0	0	0	1	0	a[2]	a[1]	a[0]	0x88	R
Revision	0	0	0	1	0	a[2]	a[1]	a[0]	0x8C	R
Download Event	0	0	0	1	0	a[2]	a[1]	a[0]	0x90	R
Page Memory Used	0	0	0	1	0	a[2]	a[1]	a[0]	0x94	R
DAC	0	0	0	1	0	a[2]	a[1]	a[0]	0x98	W/R
ADC Configuration	0	0	0	1	0	a[2]	a[1]	a[0]	0x9C	W/R

5.1.1. Configuration ROM

The following registers contain some module's information according to the Table 3.2, they are D16 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 5.3: 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 ROM.

5.2. VME Control Register

Bit	Function
[15:2]	Reserved
[1]	0 = Select Low Switching Clock Frequency 1 = Select High Switching Clock Frequency
[0]	0 = Switching Clock Disabled 1 = Switching Clock Enabled

5.3. VME Status Rgister

Bit	Function
[15:0]	Reserved

5.4. VME Interrupt Level

Bit	Function
[1:0]	Interrupt Level

5.5. VME Interrupt Vector

Bit	Function
[7:0]	Interrupt Vector Address

5.6. Multicast Base Address

Bit	Function
[3:0]	Multicast Base Address

5.7. Multicast/CBLT Control

Bit	Function
[1:0]	Multicast/CBLT settings: “00” : Disabled “01” : Active First “10” : Active Last “11” : Active Intermediate

5.8. Front Panel I/O Data

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

5.9. Front Panel I/O Control

Bit	Function
[15:6]	Reserved
[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 = Outputs Enabled 1 = Outputs Disabled
[0]	0 = NIM I/O Levels 1 = TTL I/O Levels

5.10. Post Trigger setting

Bit	Function
[31:0]	Post trigger value

5.11. VME FPGA Firmware Revision

Bit	Function
[31:0]	Revision number

5.12. BLT Event Number

Bit	Function
[7:0]	Number of events in event-aligned BLT readout

5.13. Trigger Enable

Bit	Function
[15]	0 = Software Trigger Disabled 1 = Software Trigger Enabled
[14]	0 = External Trigger sensing disabled 1 = External Trigger sensing enabled
[13]	0 = Global Trigger generation disabled 1 = Global Trigger generation enabled
[12]	Reserved
[11]	Reserved
[10]	0 = Channel 7 trigger disabled 1 = Channel 7 trigger enabled
[9]	0 = Channel 6 trigger disabled 1 = Channel 6 trigger enabled
[8]	Reserved
[7]	0 = Channel 5 trigger disabled 1 = Channel 5 trigger enabled
[6]	0 = Channel 4 trigger disabled 1 = Channel 4 trigger enabled
[5]	Reserved
[4]	0 = Channel 3 trigger disabled 1 = Channel 3 trigger enabled
[3]	0 = Channel 2 trigger disabled 1 = Channel 2 trigger enabled
[2]	Reserved
[1]	0 = Channel 1 trigger disabled 1 = Channel 1 trigger enabled
[0]	0 = Channel 0 trigger disabled 1 = Channel 0 trigger enabled

5.14. Control Register

Bit	Function
[9]	0 = Memory Read Test Disabled 1 = memory Read test Enabled
[8]	0 = Memory BIST Test and Continuous Write Disabled 1 = Memory BIST Test and Continuous Write Enabled
[7]	0 = Trigger Output Disabled 1 = Trigger Output Enabled
[6]	0 = Trigger Output on Data Input Over Threshold 1 = Trigger Output on Data Input Under Threshold
[5]	0 = Trigger Input Disabled 1 = Trigger Input Enabled
[4]	0 = Memory Random Access 1 = Memory Sequential Access
[3]	0 = Test Memory Disabled 1 = Test memory Enabled
[2]	0 = Memory Disabled 1 = Memory Enabled
[1]	0 = Trigger Overlapped Not Enabled 1 = Trigger Overlapped Enabled
[0]	0 = Window Acquisition 1 = Sample Acquisition

5.15. Status Register

Bit	Function
[5]	0 = Block Remove Command Successes 1 = Block remove Command Failed
[4]	0 = Memory BIST Test KO 1 = Memory BIST Test OK
[3]	0 = Memory BIST Test not Finished 1 = Memory BIST Test Finished
[2]	0 = DAC Programming Finished 1 = DAC Programming Running
[1]	0 = FIFO Descriptor not Empty 1 = FIFO Descriptor Empty
[0]	0 = FIFO Descriptor not Full 1 = FIFO Descriptor Full

5.16. Buffer Size Register

Bit	Function
[3:0]	0000 = 512K Samples 0001 = 256K Samples 0010 = 128K Samples 0011 = 64K Samples 0100 = 32K Samples 0101 = 16K Samples 0110 = 8K Samples 0111 = 4K Samples 1000 = 2K Samples 1001 = 1K Samples 1010 = 512Samples

5.17. Revision Register (Rev. X.Y)

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

5.18. Download Event Register

Bit	Function
[31:0]	Any Data

5.19. Page Memory Removed Register

Bit	Function
[11:0]	The Number of Page Memory to Be Removed

5.20. Page Memory Used Register

Bit	Function
[11:0]	The Number of Page Memory Used

5.21. Read Configuration Register

Bit	Function
[31:0]	[31:22] = Address of Memory Page To Read [21:12] = Number of samples To Read [11:0] = Offset of Address

5.22. DAC Register

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

5.23. Threshold Register

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

5.24. Over/Under Threshold Register

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

5.25. ADC Configuration Register

Bit	Function
[2]	0 = ADC Output Not Randomised 1 = ADC Output Randomised
[1]	
[0]	0 = ADC Dither Disabled 1 = ADC Dither Enabled