PRELIMINARY

Technical Information Manual

Revision n. 0
17 July 2002

MOD. V890

128 CHANNEL
MULTIHIT TDC

NPO:
00113/00:V890x.MUTx/00

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.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. General description

1.1. Overview

The Model V890 is a 1-unit wide VME 6U module that houses 128 independent Multi-Hit/Multi-Event Time to Digital Conversion channels. The unit houses 4 TDC chips, developed by CERN/ECP-MIC Division [1], thus called from now on the CERN/ECP-MIC TDCs [2]. Resolution can be set at 100 ps (19 bit dynamics), 200 ps (19 bit dynamics) or 800 ps (17 bit dynamics).

The CERN/ECP-MIC TDC is a General Purpose time-to-digital converter, with 32 channels per chip. The chips can be enabled to the rising/falling edges and width detection; for each channel there is a digital adjust for the zero-ing of any offsets and pedestals.

Two different versions are foreseen: the **Mod. V890**[1] and the **Mod. V890 B**. The Mod. V890 is provided with the P1, P2 VME connectors and the JAUX connector, thus requiring the CERN V430 VMEbus crate [5], while the Mod. V890 B has only the P1 and P2 VME connectors and fits into both standard and V430 VMEbus crates.

The module features ECL inputs (LVDS inputs are available on request)

The data acquisition can be programmed in "EVENTS" (“TRIGGER MATCHING MODE” with a programmable time window) or in "CONTINUOUS STORAGE MODE".

The board houses a 32 kwords deep Output Buffer, that can be readout via VME (as single data, Block Transfer and Chained Block Transfer) in a completely independent way from the acquisition itself.

The module programming is performed via a microcontroller that implements a high-level interface towards the User in order to mask the board and the TDCs' hardware.

The unit accepts the following CONTROL signals (ECL differential, 110 Ω) in common to all channels:
- TRIGGER: a common TRIGGER input;
- RST: allows the TDCs’ Bunch Reset;
- CLOCK: allows to provide an external Clock to the board.
- CLEAR: allows to clear the Output Buffer;

The TRIGGER can be also sent as NIM signal on a double (bridged) LEMO 00 connector. A ECL output, OUT_PROG, whose function can be programmed (see § 6.20), is also available on the CONTROL Bus.

Six front panel LEDs show the status of the unit:
- DTACK lights up each time the module generates the VME signal DTACK;
- PWR lights up when the module is inserted in the crate
- TERM ON lights up when all the lines of the CONTROL bus are terminated;

[1] Mod. **V890** available exclusively on request.

- FULL lights up when the memory is full;
- ERROR lights up when a TDC global error occurs;
- DATA READY lights up when the Data Ready condition occurs (see § 6.8).

The module houses a VME RORA INTERRUPTER [4]: the interrupt is generated on the occurrence (User programmable) of one of the following conditions (see also § 4.5):

- at least one complete event is present in the buffer;
- the buffer is not empty (at least one word is present in the buffer);
- the buffer is almost full (programmable).

The module works in A24/A32 mode.
The internal registers are available in D16 mode only, while the data buffer is available in D32, BLT32 or MBLT64. The module supports also the Chained Block Transfer mechanism (CBLT) and the Multicast commands (MCST). Geographical address is also available (see § 4 for the module's addressing and data transfer details).

1.2. Block Diagram

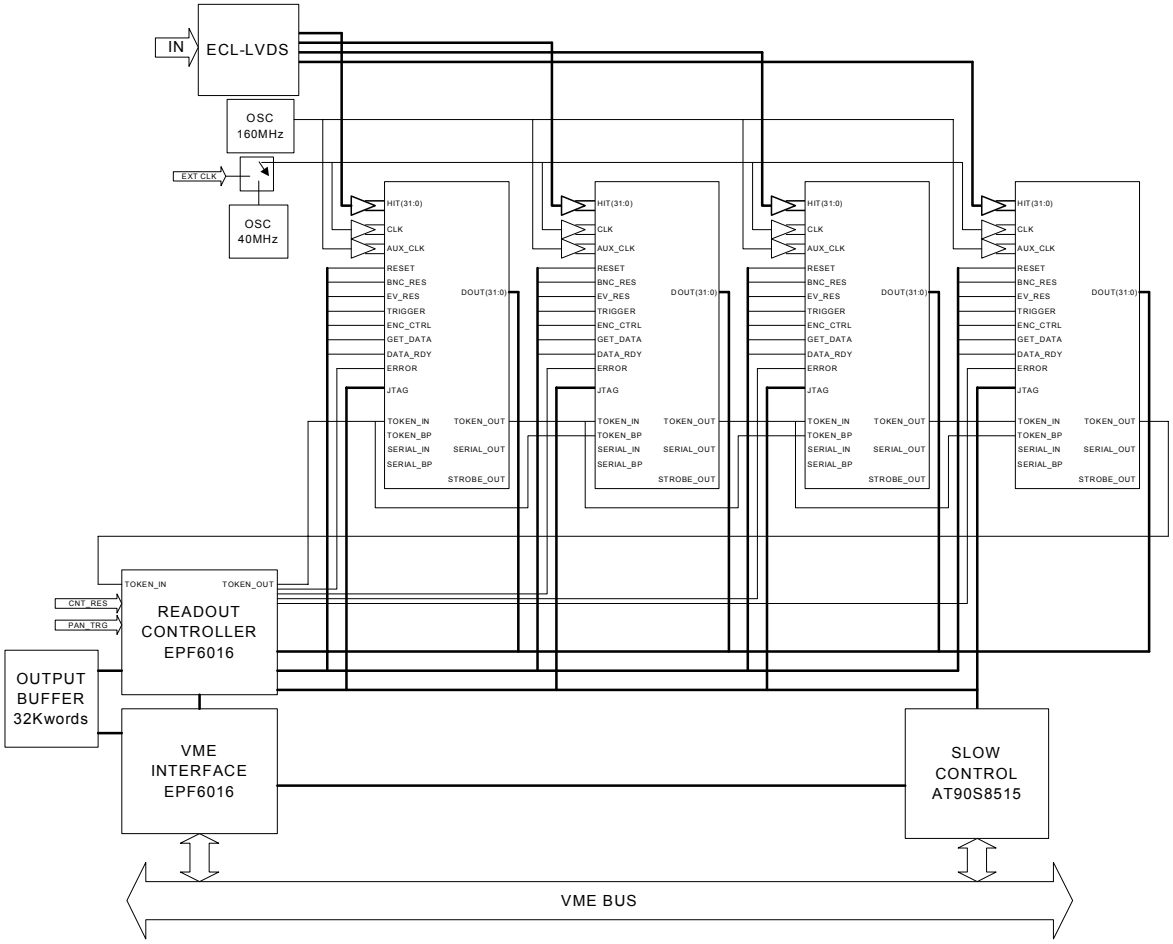


Fig. 1.1: Mod. V890 Block Diagram

2. Functional description

The V890 operating principles are based on the functionality of the CERN/ECP-MIC TDC chip [2]. Two different module setups are selectable via software for different acquisition scenarios, namely:

- Trigger Matching Mode;
- Continuous Storage Mode.

It is possible to switch from one operation setup to the other by simply resetting and reprogramming the module (with this operation the data in memory will be lost). See § 6.2 for a full description of the data format.

2.1. Operating mode selection

The operating mode can be selected via the 00xx and 01xx OPCODES (see § 5.2).

2.2. Trigger Matching Mode

In this acquisition mode the time measure is performed loading the enabled counters' status into the **Level 1 Buffer** (L1), each time a **hit** occurs. The **hits** occurred within the **Match Window** are stored into the **Output Buffer**, common to all channels. Both the duration and the relative position, "offset", (with respect to the **Trigger**) of the Match Window are programmable (see Fig. 2.3, refer to § 5.3.1 and § 5.3.2 for duration/offset programming). The *trigger latency* (i.e. the gap between two subsequent triggers, see Fig. 2.1) must be wider than the Match Window in order to have all the hits occurred whitin the Match Window already loaded into the L1 Buffer when the data storage into the Output Buffer begins. Actually this process must be performed within a time window (**Search Window**), slightly wider than the Match Window in order to avoid data loss (see § 5.3.3 for the extra-search margin programming). Older hits are removed from the L1 buffer by an automatic reject-function, in order to avoid the memory overflow. The *reject latency* is greater than the trigger latency, in order to avoid valid data loss.

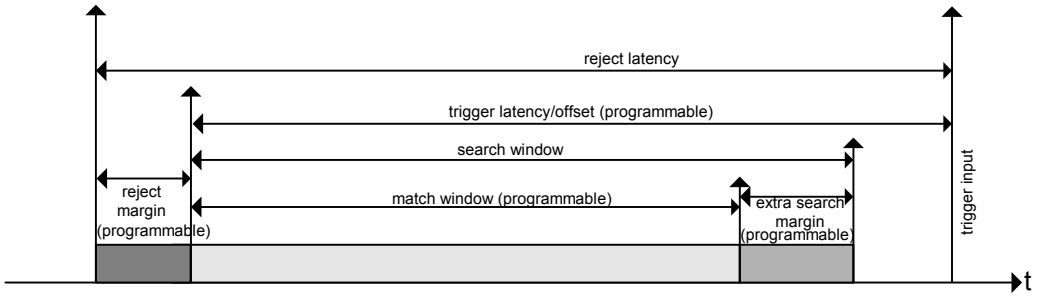


Fig. 2.1: Mod. V890 Trigger Matching Mode timing diagram

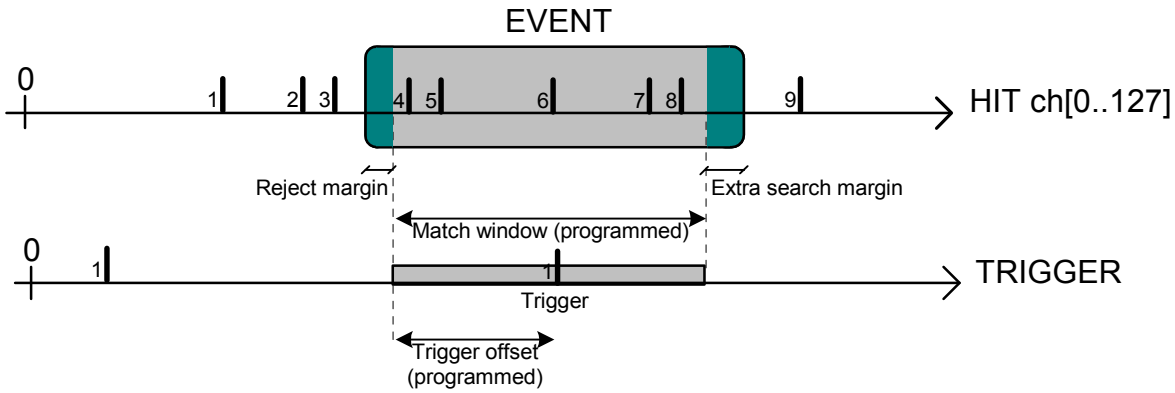


Fig. 2.2: Mod. V890 Trigger Matching Mode timing diagram

An event is thus composed by the hits received by the enabled channels whithin the Match Window. Each event consists of the following *words*:

- the **Global Header** (32 bit word), that contains the geographical address and the trigger counter (up to 24 bit);
- the **data** collected by the four TDC chips, each of them composed as follows:
  - a **TDC Header** (if enabled);
  - the **TDC Measurements**;
  - the **TDC Error** (if enabled);
  - the **TDC End Of Block** (if enabled);
- the global **Global End Of Block** (GEOB), which contains the geographical address, the event words counter (up to 16 bit), and a "status" (3 bit), whose value is 1 if an error has occurred, 2 if the Output Buffer is in *overflow* and 0 in any other case (see § 6.2.1 for a full description of the events' structure).

The stored data can represent an *absolute* time measurement (where *time zero* is represented by the latest module's Bunch RESET) or a *relative* (to the Match window beginning) one; this setting is described in § 5.3.5 and § 5.3.6. The Fig. 2.2 shows the hits received by one TDC chip; the event loaded in the module's Output Buffer has the following structure:

TDC EVENT DATA
TDC HEADER event n. 1 (if enabled)
DATUM #1 HIT time(4)
DATUM #2 HIT time(5)
DATUM #3 HIT time(6)
DATUM #4 HIT time(7)
DATUM #5 HIT time(8)
TDC EOB number of readout data = 5 (if enabled)

The number of words composing an event has no upper limit. If the hits' rate is higher than the TDC to Output Buffer data transfer speed (or higher than the VME Buffer readout speed), OVERFLOW status will be signalled by the relevant bit in the *Status Register* (see § 6.8), as a consequence, data loss will occur.

2.2.1. Timing constraints

The trigger related parameters (see § 5.3) must be programmed according to the following constraints:

If the User whishes to store into the Output Buffer hits occurred before the trigger, the constraint is:

Match window width < |Trigger offset| < 102375 ns

Note that in this case the Trigger offset is negative, see Fig. 2.3.

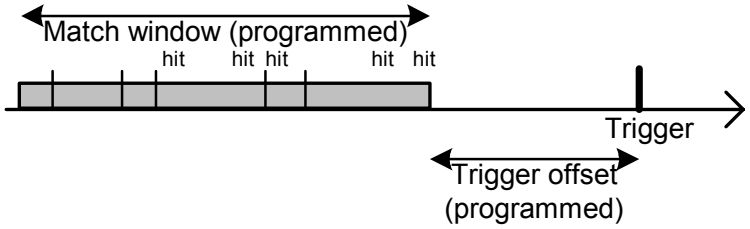


Fig. 2.3: Storage of hits occurred prior triggering

If the User whishes to store into the Output Buffer hits which may or may not have occurred when the trigger is sent, the constraint is:

Match window width + trigger offset < 1000 ns

Note that in this case the Trigger offset can be either negative or positive, see Fig. 2.4.

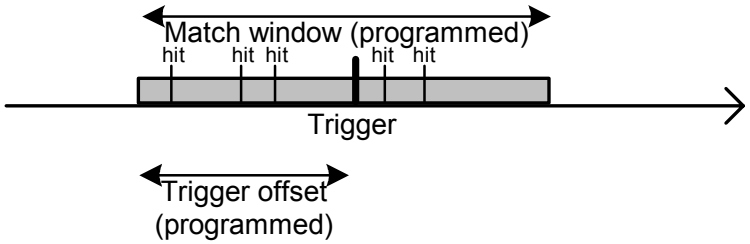


Fig. 2.4: Storage of hits which may occur after triggering

2.3. Continuous Storage Mode

In this readout mode the data loaded into the L1 Buffer are straightly forwarded into the Output Buffer. The trigger signal is not used and thus no group of data is stored as "event". All the hits received by the enabled channels are stored as valid data into the Output Buffer. The Time Origin (Time Zero) is represented by the latest Bunch RESET, sent as front panel signal (see § 3.4.2).

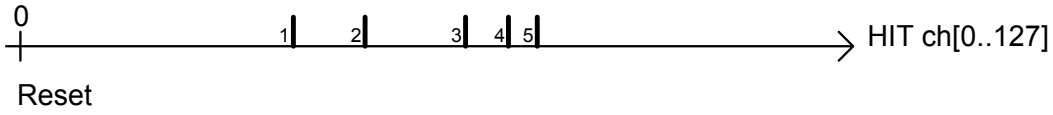


Fig. 2.5: Continuous storage

The storage of data in the Output Buffer never foresees the writing of the control words (HEADER and EOB). The data are written in sequential order, reflecting the time evolution of the external signals:

DATUM #1	HIT time(1)
DATUM #2	HIT time(2)
DATUM #3	HIT time(3)
DATUM #4	HIT time(4)
DATUM #5	HIT time(5)
DATUM #6	HIT time(6)

If the total rate of RESET and of HIT signals is higher than the transfer rate of the data from the TDCs to the local buffer (or from the local buffer to the VME) an overflow condition will be flagged by the ERROR bits in the Status Register (see § 6.8). In this case there will be a data loss.

3. Technical specifications

3.1. Packaging

The Model V890 is housed in a 6U-high, 1U-wide VME unit. The board hosts the VME P1, P2 connectors and, depending on the version, the PAUX connector.
The version equipped with the PAUX connector (**V890**) requires the VME V430 backplane.

3.2. Power requirements

The power requirements of the Mod. V890 versions are as follows:

Table 3.1: Model V890 power requirements

Power supply	Mod. V890	Mod. V890 B
+5 V	TBD	TBD
-5 V	TBD	-

3.3. Front Panel

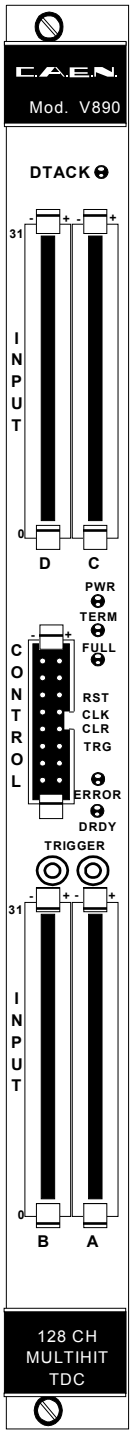


Fig. 3.1: Model V890 front panel

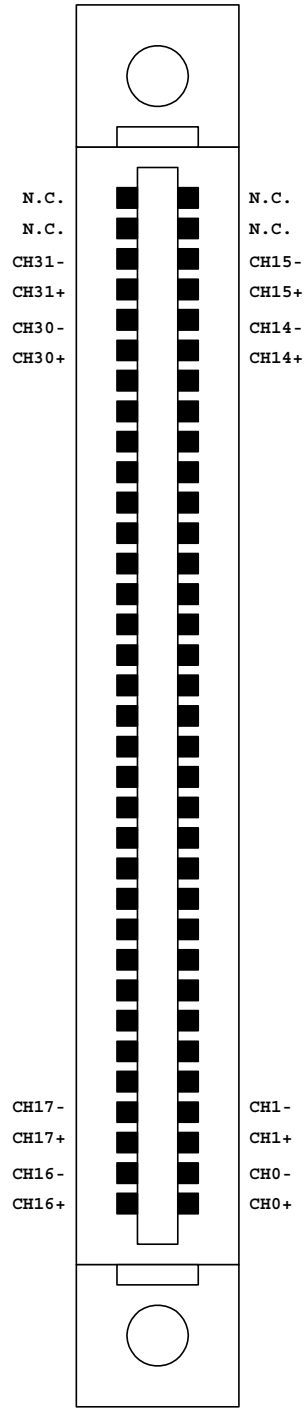


Fig. 3.2: INPUT connector pin assignment

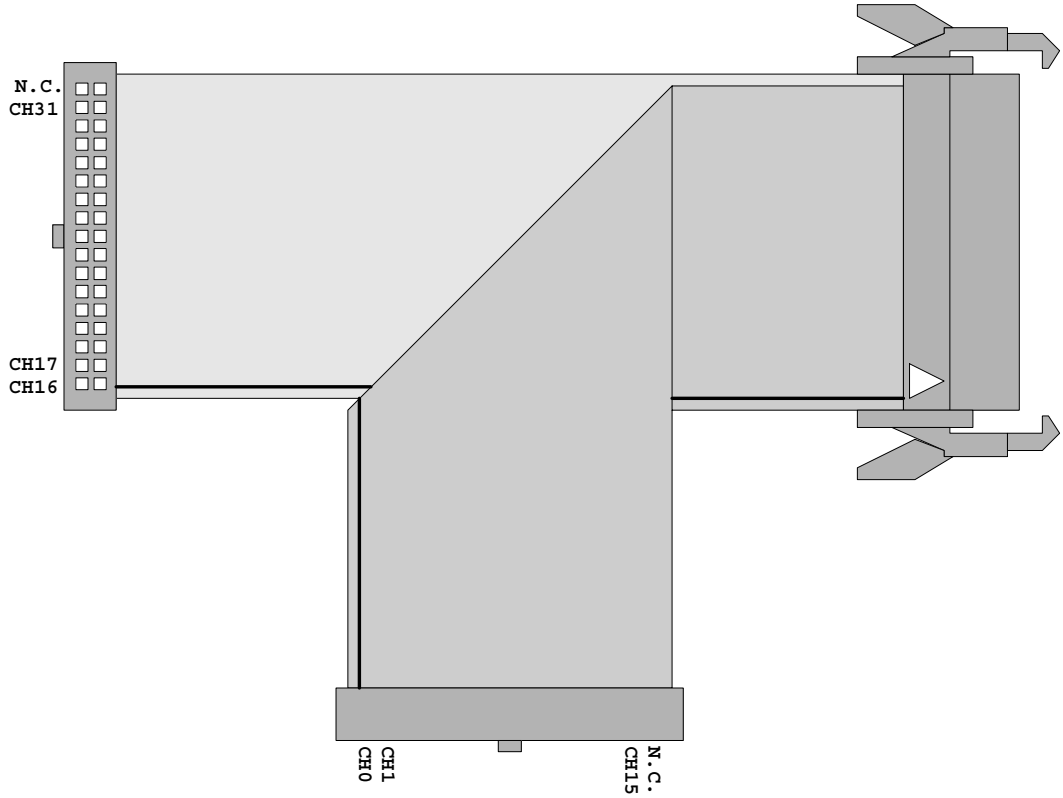


Fig. 3.3: INPUT connector cabling

3.4. External connectors

The location of the connectors is shown in Fig. 3.1.Their function and electro-mechanical specifications are listed in the following subsections.

3.4.1. INPUT connectors

Mechanical specifications:
N. 4 High Density connectors[2], Robinson Nugent P50E-068-P1-SR1-TG type, (34+34) pins; for the 128 single channel inputs.
Connector A refers to Channels 0 to 31.
Connector B refers to Channels 32 to 63.
Connector C refers to Channels 64 to 95.
Connector D refers to Channels 96 to 127.

[2] The CAEN Mod. A967 cable adapter, which allows to adapt one High Density connector into two 1" 17+17-pin Header-type connectors, is available.

Electrical specifications:
ECL input signals, 110 Ω impedance. The 17th higher pair of pins of each connector's side is not connected.

3.4.2. CONTROL connector

Mechanical specifications:
two 8+8-pin, 3M 3408-5202 Header-type connectors.

- CLOCK: Rising-edge active, differential ECL level, 110 Ω.
- TRIGGER: Rising-edge active, differential ECL level, 110 Ω; min. width 25 ns
- CLR: Active high, differential ECL level, 110 Ω impedance; min. width 25 ns.
- OUT_PROG: Rising-edge active, differential ECL level, 110 Ω impedance.
- Bunch RESET: Rising-edge active, differential ECL level, 110 Ω; min. width 25 ns.

Pin assignment is shown in Fig. 3.4.

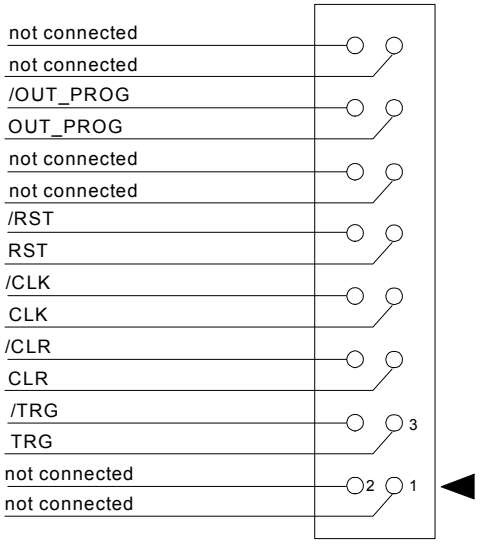


Fig. 3.4: CONTROL connector pin assignment

3.4.3. EXTERNAL TRIGGER connectors

Mechanical specifications:
two 00-type LEMO connectors (bridged).
Electrical specifications:
Rising-edge active, NIM, high impedance; min. width 25 ns

3.5. Other front panel components

3.5.1. Displays

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

DTACK: *Colour:* green.
Function: it lights up green whenever a VME read/write access to the board is performed.

PWR: *Colour:* green/red.
Function: it lights up green when the board is inserted into the crate and the crate is powered up; when it is red, it indicates that there is an over-current status: in this case, remove the overload source, switch the module off and then switch it on again.

TERM: *Colour:* green/orange.
Function: it lights up green when all the lines of the control bus are terminated, it also lights up orange for a while at power ON to indicate that the board is configuring.

FULL: *Colour:* red.
Function: it lights up when the Multi-Event Buffer is full; it also lights up for a while at power ON to indicate that the board is configuring.

ERROR: *Colour:* red.
Function: it lights up to signal a TDC global error; it also lights up for a while at power ON to indicate that the board is configuring

DRDY: *Colour:* yellow.
Function: it lights up when at least one event is present in the output buffer; it also lights up for a while at power ON to indicate that the board is configuring.

3.6. Internal hardware components

The V890 module is constituted by a motherboard with a piggy-back board plugged into it. In the following some hardware setting components, located on the boards, are listed. See Fig. 3.5 for their exact location on the PCB and their settings.

3.6.1. Switches

ROTARY SWITCHES: *Type:* 4 rotary switches.
Function: they allow to select the VME address of the module. See Fig. 3.5 for their location.

SW1: *Type:* DIP switch.
Function: it allows the hardware termination (if enabled) of the Control Bus on 110 Ω (see also § 6.9)

Right position (dot visible): Control Bus not terminated;

Left position (dot not visible): Control Bus terminated.

SW2: *Type:* DIP switch.
Function: it allows to switch from/to internal/external clock

Right position (dot visible): internal clock;

Left position (dot not visible): external clock.

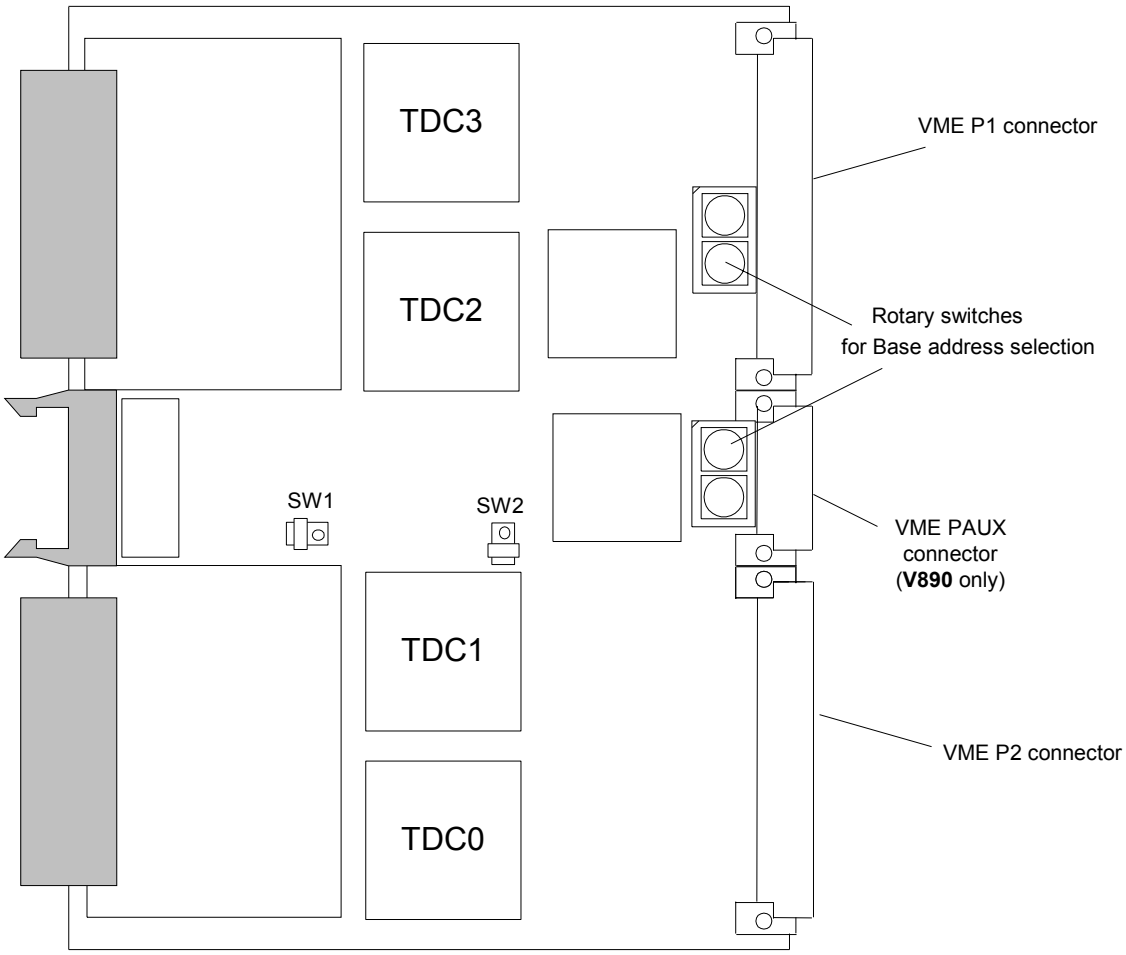


Fig. 3.5: Component Location (component side)

3.7. Technical specifications table

Table 3.2 : Model V890 technical specifications

Packaging	6U-high, 1U-wide VME unit (V430 backplane version available)
Power requirements	Refer to Table 3.1
Inputs	128 ECL (LVDS also available) inputs, 110 Ω impedance
Double hit resolution	5 ns
Acquisition modes	Trigger Matching Mode; Continuous Storage Mode
Trigger window	Programmable (see § 5.3)
Built-in memory	32 kwords deep Output Buffer
VME readout rate	10 MHz
Dynamic Range[3]	100 µs
LSB	VME programmable from 100 to 800 ps
RMS resolution	15 to 250 ps (depending on resolution mode)
Integral non linearity	TBD
Differential non linearity	TBD
Interchannel Isolation	<100 ps
Offset spread	<1 ns
Power rejection	TBD
EXT TRIGGER input	Two LEMO 00 bridged connectors, NIM signal, 110 Ω
Clock source	Internal (40 MHz) or External (on *Control* connector), dip switch selectable
Control inputs	active-high, differential ECL input signals: RST: resets Output Buffer, Status and Control registers. CLR: FAST CLEAR of TAC sections rising-edge active, differential ECL input signals: CLK: external clock TRG: trigger for the TDC latching
Control outputs	differential ECL output signal: OUT_PROG: control output signal, programmable via the *out prog control* register
Displays	DTACK: green LED; lights up at each VME access. PWR: green/red LED; green: power ON, red: failure status. TERM: green LED; control bus termination ON. FULL: red LED; memory full. ERROR: red LED; TDC global error. DRDY: yellow LED; at least one datum in the output buffer
VME	*Addressing modes*: A24, A32, MCST *Data transfer modes*: D16, D32, BLT32, BLT64, CBLT *Readout rate*: 10 MHz

[3] if the Sliding Scale is used, dynamic range is reduced by approximately 6%

4. Operating modes

4.1. Installation

If the purchased version does not feature the PAUX connector (Mod. V890 B), it can be inserted into either a standard VME 6U crate or in a V430 VME 6U crate. The Mod. V890 board requires the V430 VME 6U crate (JAUX dataway). Refer to § 1.1 for details on the various versions. Please note that some versions of the board support 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 V890 board into the crate: as the board is inserted, the PWR green LED (see § 3.5.1) lights up indicating that the board is powered;
2. if the board supports "live insertion", after insertion, the TERM LED lights up orange, the FULL LED and the ERROR LED become red and the DRDY LED becomes yellow; this indicates that the board is turned on and is configuring;
3. after a short time the FULL, the ERROR and DRDY LEDs will light off and the TERM LED will become either green or off, according to the status of the terminations on the PCB of the board: this indicates that the board is ready to acquire data.

N.B.: if the PWR LED becomes red instead of being green, there is an overload and the over-current protection is now running. In order to acquire data, it is necessary to remove the overload source, then turn the board off and switch it on again. Sometimes, it may happen that the PWR LED is red as soon as the board is inserted in the crate: this is due to the fact that the board has been just misplaced into the crate. In this case, extract the board and insert it again into the crate.

4.3. Power ON status

At power ON the module is in the following status:

- the Event Counter is set to 0;
- the Output buffer is cleared;
- the Interrupt Level is set to 0x0 (in this case interrupt generation is disabled) and the Interrupt Vector is set to 0x0;
- the MCST/CBLT address is set to 0xAA.

At power ON or after a hardware reset the module must thus be initialised.

4.4. Addressing capability

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

1. via Base Address;
2. via GEOgraphical address;
3. 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 (BLT)
0x3D	A24 supervisory data access
0x3C	A24 supervisory 64 bit block transfer (MBLT)
0x3B	A24 non privileged block transfer (BLT)
0x39	A24 non privileged User data access
0x38	A24 non privileged 64 bit block transfer (MBLT)
0x2F	Configuration Rom/Control & Status Register (CR/CSR)
0x0F	A32 supervisory block transfer (BLT)
0x0D	A32 supervisory data access
0x0C	A32 supervisory 64 bit block transfer (MBLT)
0x0B	A32 non privileged block transfer (BLT)
0x09	A32 non privileged data access
0x08	A32 non privileged 64 bit block transfer (MBLT)

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 in two ways:

- by four rotary switches;
- by writing the Base Address in the ADER_32 and ADER_24 registers

The 4 rotary switches for Base Address selection are housed on two piggy-back boards plugged into the main printed circuit board (see § 3.6.1)

To use this addressing mode the bit 0 of the Enable ADER Register (see § 6.10) must be set to 0. This is also the default setting.

The module Base Address can be also fixed by using the ADER_32 and ADER_24 Registers. These two registers set respectively the A[31:24] and the A[23:16] VME address bits (see § 6.11 and 6.12).
To use this addressing mode bit 0 of the Enable Ader Register (see § 6.10) must be set to 1.

4.4.2. Addressing via geographical address

The module works in A24 mode only. The Address Modifiers codes recognised by the module are:

AM=0x2F: A24 GEO access

All registers except for the Output Buffer (i.e. the CR/CSR area) can be accessed via geographical addressing.
The geographical address is automatically picked up at each RESET from the SN5..SN1 lines of the PAUX connector. Each slot of the VME crate is identified by the status of the SN5...SN1 lines: for example, the slot #5 will have these lines respectively at 00101 and consequently the module inserted in the slot #5 will have a GEO address set to 00101 (see Fig. 4.1).
The complete address in A24 mode for geographical addressing is:

A[31:24]	**don't care**
A[23:19]	**GEO**
A[18:16]	**0**
A[15:0]	**offset**

The following two figures show the binary and the hexadecimal representation of, respectively, the board Address and a Register Address (Status Register) in GEO addressing mode.

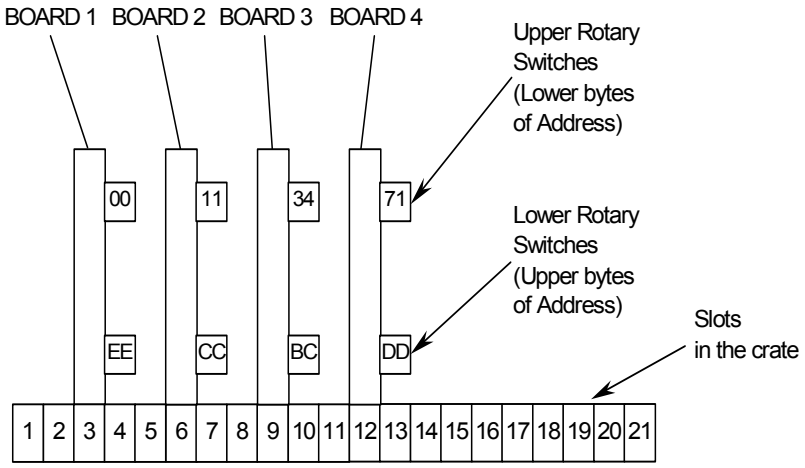


Fig. 4.1: Binary-Hexadecimal representation of the board Address in GEO mode

23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Binary representation
2				8				0				0				0				0				Hexadecimal representation

Fig. 4.2: Binary-Hexadecimal representation of Status Register Address in GEO mode

It is suggested to perform module settings in geographical addressing mode

N.B.: In the case of versions where the SN5…SN1 lines are not available (i.e. the versions without the PAUX connector), addressing via geographical address is not possible.
Although in these versions it is possible to perform a write access to the GEO Address Register (see § 6.3) for data identification during CBLT operation (see § 4.4.4), it is incorrect to use the GEO Address Register for addressing purposes when there is no PAUX.

4.4.3. Base/GEO addressing examples

The following is an example of Base/GEO Addressing for two V890 boards inserted in a VME crate.

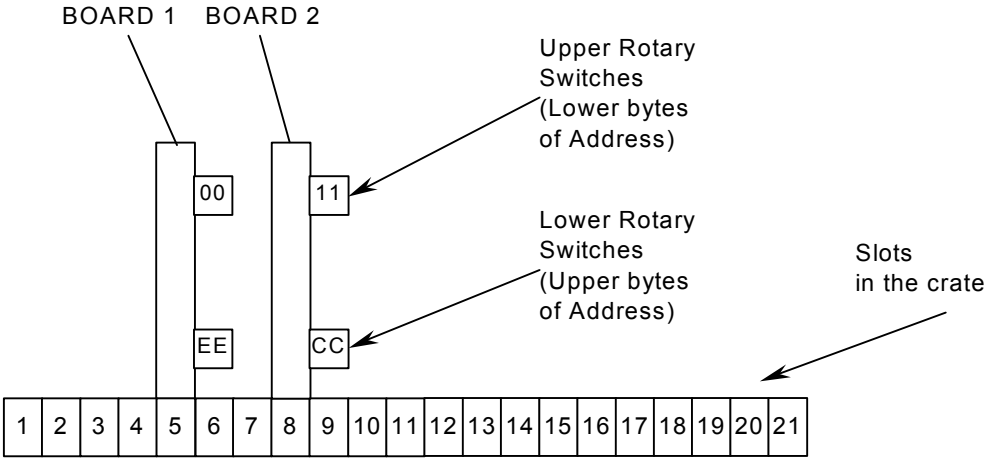


Fig. 4.3: Base/GEO Addressing: Example 1

If the board 1 and board 2 are respectively inserted in the slots 5 and 8 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
GEO addressing A24: 0x280000 + offset (Output Buffer excluded).

Board 2:
Base addressing A32: 0xCC110000 + offset
Base addressing A24: 0x110000 + offset
GEO addressing A24: 0x400000 + offset (Output Buffer excluded).

4.4.4. 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:

AM=0x0F: A32 supervisory block transfer (CBLT)
AM=0x0D: A32 supervisory data access (MCST)
AM=0x0B: A32 User block transfer (CBLT)
AM=0x09: A32 User data access (MCST)

The boards can be accessed in Multicast Commands mode (MCST mode, see [4]), 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, see [4]) 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) 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 (see § 6.4) 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 CBLT/ MCST access. No empty slots must thus 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 (see § 6.5). 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 EOB words containing the GEO address in the 5 MSB (see § 6.2). In the versions without the PAUX connector it is up to the User to write the GEO address in the GEO register (this operation is allowed only if the PAUX is not present) before executing the CBLT operation. 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.

4.4.5. MCST/CBLT addressing examples

The following is an example of MCST and CBLT addressing for four V890 boards plugged into a VME crate. To access the boards the steps to be performed are as follows:

1. Set the MCST address (see § 6.4) for all boards via VME Base Address or geographical addressing (if available);
2. Set the bits F_B and L_B of the MCST Control Register (see § 6.5) 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 or read;
Addr_mode:	the addressing mode (A32);
Data_mode:	the data mode (D16, D32 or D64).

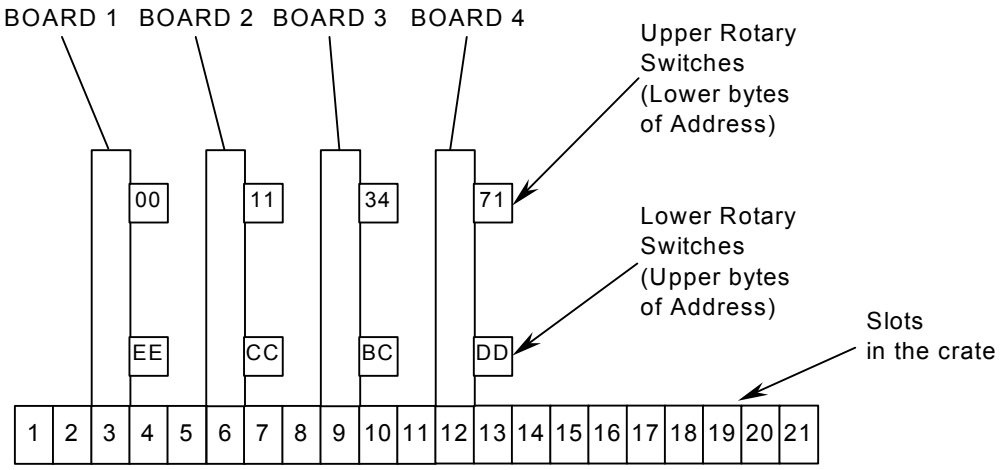


Fig. 4.4: MCST/CBLT Addressing Example

In the following two software examples using the above mentioned procedures are 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)   /* set RESET MODE for all the boards */
```

Example of Access via geographical address

```
vme_write (0x18100E, 0xAA, A24, D16)     /* set MCST Address=0xAA for board 1 */
vme_write (0x30100E, 0xAA, A24, D16)     /* set MCST Address=0xAA for board 2 */
vme_write (0x48100E, 0xAA, A24, D16)     /* set MCST Address=0xAA for board 3 */
vme_write (0x51100E, 0xAA, A24, D16)     /* set MCST Address=0xAA for board 4 */
vme_write (0x181012, 0x02, A24, D16)     /* set board 1 = First */
vme_write (0x301012, 0x03, A24, D16)     /* set board 2 = Active */
vme_write (0x481012, 0x00, A24, D16)     /* set board 3 = Inactive */
vme_write (0x511012, 0x01, A24, D16)     /* set board 4 = Last */
vme_write (0xAA001014, 0x00, A32, D16)   /* set RESET MODE for 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. V890 houses a RORA-type VME INTERRUPTER. The INTERRUPTER responds to 8 bit, 16 bit and 32 bit Interrupt Acknowledge cycles providing an 8-bit STATUS/ID on the VME data lines D00..D07.

4.5.1. Interrupt Status/ID

The interrupt STATUS/ID is 8-bit wide, and it is contained in the 8LSB of the Interrupt Vector Register (see § 6.7). The register is available at the VME address: Base Address + 0x100C.

4.5.2. Interrupt Level

The interrupt level corresponds to the value stored in the 3LSB of the Interrupt Level Register (see § 6.6). The register is available at the VME address: Base Address + 0x100A. If the 3LSB of this register are set to 0, the Interrupt generation is disabled.

4.5.3. Interrupt Generation

An Interrupt is generated when the number of events stored in the memory equals the value written in the Almost Full Level Register at the VME address: Base Address + 0x1022 (see § 6.15). If the value in Event Trigger Register is set to 0 the interrupt is disabled (default setting).

4.5.4. Interrupt Request Release

The INTERRUPTER removes its Interrupt request when 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.

4.6. Trigger Matching Mode data transfer

The Mod. V890 Output Buffer can be readout via VME through D32, BLT32, BLT64, CBLT32 and CBLT64 modes. The module has to be programmed in order to work in the required mode (see § 5.2.1), and some parameters have to be set in order to keep data aligned (i.e. to transfer an integer number of events per cycle). In the following each readout mode will be described in detail.
Nw is the number of words which compose an event. **It's up to the user to verify the presence of valid data in the Output Buffer before read out**; this can be performed via the Status Register read out (*Data Ready Bit*, see § 6.8). When a trigger signal occurs, an event is loaded into the Output Buffer (see § 6.2).

4.6.1. D32 OUTPUT BUFFER readout (Trigger Matching)

This readout mode can be performed in two ways:

- Wait for DATA_READY signal then readout the **Nw** words until the *global end of block* (EOB).
- Start readout without waiting for DATA_READY signal: *fillers* will be readout until a *global header* is met; then **Nw** words and global EOB will be readout.

4.6.2. BLT32/64 OUTPUT BUFFER readout with Bus Error and Event Aligned BLT disabled

BLT32/64 readout with BERR and Event Aligned BLT mode disabled does not permit to collect one or more complete events since the number of bytes **Nb** which are transferred in a BLT cycle has usually a "typical" value (256, 2048…) which rarely coincide with a multiple of the number of bytes **NB** composing an event, i.e. NB=Nw·4 (see § 6.2). It is more than likely that in a BLT cycle an event could be partially read out and the remaining part would be read out in the subsequent cycle. If the Output Buffer contains a number **N** of words smaller than Nb/4 (the number of words readout in one BLT cycle), the module transfers its memory content, completed with (Nb/4)-N *fillers*.

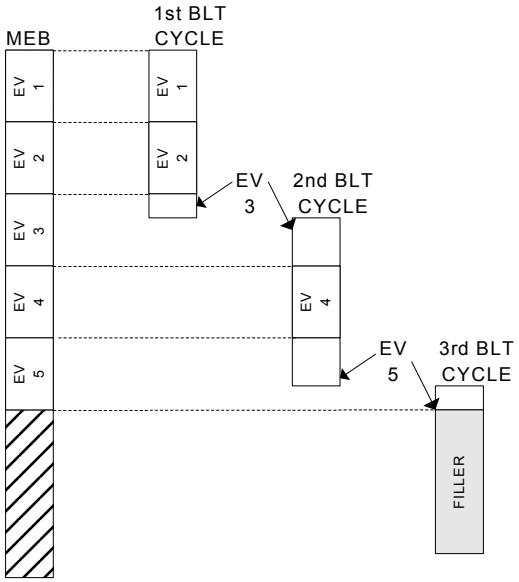


Fig. 4.5 Example of BLT data transfer with BERR and Event Aligned BLT disabled

4.6.3. BLT32/64 OUTPUT BUFFER readout with Bus Error enabled

In this mode the module produces a BERR signal (see § 6.9) once the last data in the Output Buffer has been transferred and no filler is added.

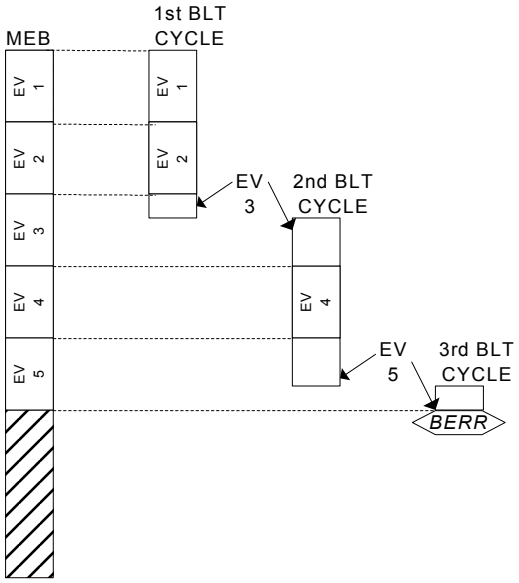


Fig. 4.6 Example of BLT data cycle with BERR enabled and Event Aligned BLT disabled

4.6.4. BLT32/64 OUTPUT BUFFER readout with Event Aligned BLT enabled

This mode allows to transfer an integer number **Ne** of events, set via the BLT Event Number Register (see § 6.16). The default setting value of this register is 0, so this mode disabled. In order to enable the Event Aligned BLT the desired number of events Ne must be written in the BLT event number register then the header must be enabled. This mode allows the module to transfer a maximum number Ne of events; three cases may occur:

- The memory has a number of events smaller than Ne: the module transfers its memory content and completes the cycle adding fillers (filler =0).
- The memory has a number of events larger than Ne and Nb/4>Ne (where Nb/4 is the number of words transferred in one BLT cycle): the module transfers Ne events and completes the cycle adding fillers.
- The memory has a number of events larger than Ne and Nb/4<Ne: the transfer requires more than one cycle to be completed, the last cycle is completed with fillers.

Anytime a value is written in the BLT Event Number Register a module's *clear* is performed.

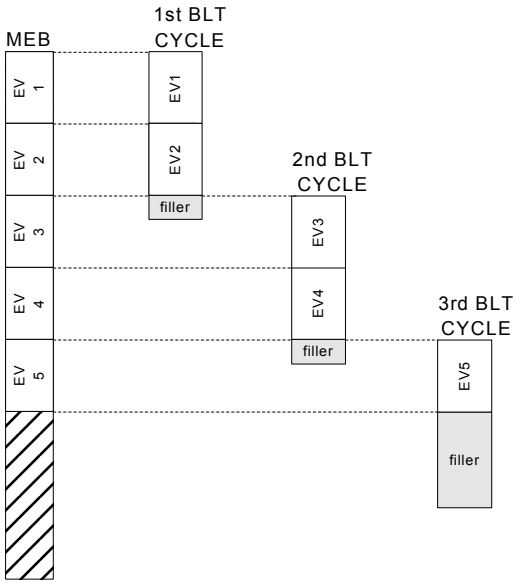


Fig. 4.7 Example of BLT data cycle with BERR disabled and Event Aligned BLT enabled

4.6.5. BLT32/64 OUTPUT BUFFER readout with both Event Aligned BLT and Bus Error enabled

This mode allows the module to transfer a maximum number Ne of events; three cases may occur:

- The memory has a number of events smaller than Ne: the module transfers its memory content then asserts BERR, thus terminating the cycle.

- The memory has a number of events larger than Ne and Nb/4>Ne: the module transfers Ne events then asserts BERR thus terminating the cycle.

- The memory has a number of events larger than Ne and Nb/4<Ne: the transfer requires more than one cycle to be completed.

N.B.: Enabling the BERR is useful in order to save time, avoiding fillers' transfer.

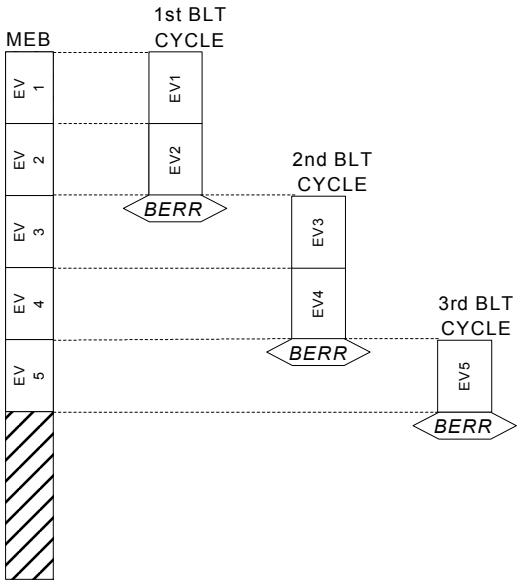


Fig. 4.8 Example of BLT data transfer with BERR and Event Aligned BLT enabled

4.6.6. CBLT32/64 OUTPUT BUFFER readout

The CBLT mode allows data readout from a set of modules in the crate. The readout starts from the first module of the set: it transfers its first event, then passes the *token* to the subsequent module via the IACKIN/IACKOUT daisy chain lines. If one module's Output Buffer is empty or *Inactive* (see § 6.5), the token is passed with no data transfer. If the data transfer is not completed by the first CBLT cycle, a second one may be attached to the first and so on until the last module in the set has transferred its last data: then it asserts BERR, which is automatically enabled when the CBLT is performed, thus completing the cycle.

CBLT64: if an odd number of words is transferred, the CPU completes the cycle adding a word composed by fillers.

4.7. Countinuous Storage Mode data transfer

The Mod. V890 Output Buffer can be readout via VME through either D32 or BLT32/BLT64 modes. The module has to be programmed in order to work in the required mode (see § 5.2.2).
When a trigger signal occurs all the accumulated hits are written into the Output Buffer; the DATA_READY bit in the Status Register (see § 6.8) is set to one when at least one datum is present in the buffer.

4.7.1. D32 OUTPUT BUFFER readout (Continuous Storage)

This readout mode can be performed in two ways:

- Set the Almost Full Level to N (see § 6.15); wait until the Almost Full bit in the Status Register is set to one then readout the N recorded words in D32 mode.

- Readout continuously from the buffer, removing the “fillers” via software.

4.7.2. BLT32/64 OUTPUT BUFFER readout (Continuous Storage)

- If BERR is disabled and the Output Buffer contains a number **N** of words smaller than Nb/4 (the number of words readout in one BLT cycle, being Nb the number of bytes per cycle), the module transfers its memory content, completed with (Nb/4)-N *fillers*.

- If BERR is enabled the module produces a BERR signal (see § 6.9) once the last data in the Output Buffer has been transferred and no filler is added.

5. Operating codes

5.1. Programming capability

The module programming is performed by means of an on-board microcontroller. The User sends and receives instructions and data to/from the microcontroller via 16-bit OPCODE setup words. The handshake is the following:

Write Operation:
- the VME (master) tests the WRITE_OK bit in the Micro Handshake Register (see § 6.22); if the WO bit is set to 1, the VME can write a datum;
- the WO bit is automatically reset after the datum is written and is set back to 1 when the datum has been acquired by the server;
- when the WO bit is set back to 1, the VME can write another datum.

Read Operation:
- a valid datum can be read via VME only if the READ_OK (RO) bit in the Micro Handshake Register (see § 6.22) is set to 1;
- the RO bit is automatically reset after the datum is read out and is set back to 1 when a new datum is ready to be read out;
- when the RO bit is set back to 1, the VME can read another datum.

The OPCODE setup words have the following format:

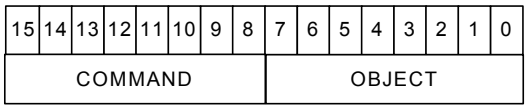


Fig. 5.1: Micro Register

The COMMAND field specifies the operation to perform, while the OBJECT field (when required) specifies the object over which the operation must be performed (e.g. the channel number). If the object refers to the channel number (OBJ = nn in Table 5.1), it can vary from 0 to 7F; if it refers to the TDC number (OBJ = 0n in Table 5.1), it can vary from 0 to 3.
When the operation does not foresee an object, the OBJECT field is meaningless.

The communication with the microcontroller begins always by sending an OPCODE; if no operands are foreseen (nR = 0 and nW = 0) the cycle ends, otherwise the microcontroller remains in a wait status until the User has read or written all the foreseen operands.

The following Table 3.1 contains, for each OPCODE, the symbolic name, the performed operation, the number of written and read operands and the number of significant bits.

Table 5.1: Operating codes list

COM	OBJ	CODE	SYMBOLIC	OPERATION	nW	nR	nbit
ACQUISITION MODE							
00	-	00xx	TRG_MATCH	set trigger matching	-	-	-
01	-	01xx	CONT_STOR	set continuos storage	-	-	-
02	-	02xx	READ_ACQ_MOD	read acquisition mode	-	1	1
03	-	03xx	SET_KEEP_TOKEN	set keep token	-	-	-
04	-	04xx	CLEAR_KEEP_TOKEN	clear keep token	-	-	-
05	-	05xx	LOAD_DEF_CONFIG	load default configuration	-	-	-
06	-	06xx	SAVE_USER_CONFIG	save User configuration	-	-	-
07	-	07xx	LOAD_USER_CONFIG	load User configuration	-	-	-
08	-	08xx	AUTOLOAD_USER_CONFIG	set auto load User configuration	-	-	-
09	-	09xx	AUTOLOAD_DEF_CONFIG	set auto load default configuration	-	-	-
TRIGGER							
10	-	10xx	SET_WIN_WIDTH	set window width	1	-	12
11	-	11xx	SET_WIN_OFFS	set window offset	1	-	12
12	-	12xx	SET_SW_MARGIN	set extra search margin	1	-	12
13	-	13xx	SET_REJ_MARGIN	set reject margin	1	-	12
14	-	14xx	EN_SUB_TRG	enable subtraction of trigger time	-	-	-
15	-	15xx	DIS_SUB_TRG	disable subtraction of trigger time	-	-	-
16	-	16xx	READ_TRG_CONF	read trigger configuration	-	5	49
TDC EDGE DETECTION & RESOLUTION							
20	-	20xx	EN_TRAILING	enable trailing edge (on all channels)	-	-	-
21	-	21xx	EN_LEADING	enable leading edge (on all channels)	-	-	-
22	-	22xx	EN_PAIR	enable paired meas. leading/ trailing edge	-	-	-
23	-	23xx	READ_DETECTION	read edge detection configuration	-	1	2
24	-	24xx	SET_TR_LEAD_LSB	set LSB of leading/trailing edge	1	-	2
25	-	25xx	SET_PAIR_RES	set leading time and width res. when pair	1	-	16
26	-	26xx	READ_RES	read resolution	-	1	16
28	-	28xx	SET_DEAD_TIME	set channel dead time between hits	1	-	2
29	-	29xx	READ_DEAD_TIME	read channel dead time between hits	-	1	2
TDC READOUT							
30	-	30xx	EN_HEAD/EOB	enable TDC header and EOB in readout	-	-	-
31	-	31xx	DIS_HEAD/EOB	disable TDC header and EOB in readout	-	-	-
32	-	32xx	READ_HEAD/EOB	read status TDC header and EOB	-	1	1
33	-	33xx	SET_EVENT_SIZE	set maximum number of hits per event	1	-	4
34	-	34xx	READ_EVENT_SIZE	read maximum number of hits per event	-	1	4
35	-	35xx	EN_ERROR_MARK	enable TDC error mark	-	-	-

36	-	36xx	DIS_ERROR_MARK	disable TDC error mark	-	-	-
37	-	37xx	EN_ERROR_BYPASS	enable bypass TDC if error	-	-	-
38	-	38xx	DIS_ERROR_BYPASS	disable bypass TDC if error	-	-	-
39	-	39xx	SET_ERROR_TYPES	set TDC internal error type	1	-	11
3A	-	3Axx	READ_ERROR_TYPES	read TDC internal error type	-	1	11
3B	-	3Bxx	SET_FIFO_SIZE	set effective size of readout FIFO	1	-	3
3C	-	3Cxx	READ_FIFO_SIZE	read effective size of readout FIFO	-	1	3
CHANNEL ENABLE							
40	nn	40nn	EN_CHANNEL	enable channel **nn**	-	-	-
41	nn	41nn	DIS_CHANNEL	disable channel **nn**	-	-	-
42	-	42xx	EN_ALL_CH	enable all channels	-	-	-
43	-	43xx	DIS_ALL_CH	disable all channels	-	-	-
44	-	44xx	WRITE_EN_PATTERN	write enable pattern for channels	8	-	128
45	-	45xx	READ_EN_PATTERN	read enable pattern for channels	-	8	128
ADJUST							
50	nn	50nn	SET_ADJUST_CH	set channel **nn** adjust	1	-	8
51	nn	51nn	READ_ADJUST_CH	read channel **nn** adjust	-	1	8
52	-	52xx	SET_GLOB_OFFS	set global offset	2	-	17
53	-	53xx	READ_GLOB_OFFS	read global offset	-	2	17
MISCELLANEOUS							
60	0n	600n	READ_TDC_ID	read programmed ID of TDC **0n**	-	2	32
61	-	61xx	READ_MICRO_REV	read firmware revision of microcontroller	-	1	8
62	-	62xx	RESET_DLL_PLL	reset DLL and PLL	-	-	-
ADVANCED							
70	nn	70nn	WRITE_SETUP_REG	write word **nn** into the scan path setup	1	-	16
71	nn	71nn	READ_SETUP_REG	read word **nn** into the scan path setup	-	1	16
72	-	72xx	UPDATE_SETUP_REG	load the scan path setup	-	-	-
73	-	73xx	DEFAULT_SETUP_REG	reload the default scan path setup	-	-	-
74	nn	74nn	READ_ERROR_STATUS	read errors in the TDC **nn** status	-	1	11
75	nn	75nn	READ_DLL_LOCK	read the DLL LOCK bit of the TDC **nn**	-	1	1
76	nn	76nn	READ_STATUS_STREAM	read the TDC **nn** status	-	4	62

5.2. Acquisition mode opcodes

The following OPCODEs allow to select the acquisition mode (see § 2.1).

5.2.1. Set Trigger Matching Mode (CODE 00xx)

It allows to select the Trigger Matching Mode, described in § 2.2. After this operation the TDCs must be reset and the Output Buffer cleared (see § 6.24 and § 6.25).

5.2.2. Set Continuous Storage Mode (CODE 01xx)

It allows to select the Continuous Storage Mode, described in § 2.2. After this operation the TDCs must be reset and the Output Buffer cleared (see § 6.24 and § 6.25).

5.2.3. Read acquisition mode (CODE 02xx)

It allows to read the status of the Acquisition Mode OPCODE settings. After this OPCODE a word must be read at the same location of the OPCODE itself. The microcontroller will remain in a wait status until a 16-bit word is read out. The least significant bit of this word is related to the Acquisition Mode according to the following:

- LSB = 1 ⇒ Trigger Matching Mode;
- LSB = 0 ⇒ Continuous Storage Mode.

5.2.4. Set keep_token (CODE 03xx)

It allows to program the module to load the data into the buffer until the end of the event or until no more data are present. It is automatically enabled when using trigger matching.

5.2.5. Clear keep_token (CODE 04xx)

It allows to program the module to load the data into the buffer one word at a time.

5.2.6. Load default configuration (CODE 05xx)

It allows to load the Default Configuration: if AUTO_LOAD is disabled (code 09xx) at Power-ON the module will be programmed in the default operation mode:

- Trigger Matching = disabled;
- Window Width = 500 ns;
- Window Offset = -1 µs;
- Reject margin = 100 ns;
- Extra search margin = 200 ns;
- All channels enabled.

5.2.7. Save User configuration (CODE 06xx)

It allows to save a User Configuration that can be recalled at any time by the User. This Configuration concerns:
- Acquisition Mode;
- Window Width;
- Window Offset;
- Reject margin;
- Extra search margin;
- Enabled channels pattern.

5.2.8. Load User configuration (CODE 07xx)

It allows to load a User Configuration previously saved by the User (see § 5.2.7).

5.2.9. Set auto load User configuration (CODE 08xx)

It allows to load automatically the User Configuration either at the next Power-ON or at the next General RESET

5.2.10. Set auto load default configuration (CODE 09xx)

It allows to load automatically the Default Configuration either at the next Power-ON or at the next General RESET

5.3. Trigger opcodes

The following OPCODEs allow to set or read some Trigger matching mode related parameters.

5.3.1. Set window width (CODE 10xx)

It allows to set the width of the trigger window. After this OPCODE has been sent, a 16-bit word must be written at the same location of the OPCODE itself. The microcontroller will remain in a wait status until this 16-bit word is written. The value of the word can be set in a range from 1 to 2047 (significant bits: [0;11]), or hex 7FF: the relevant window width is the word value times the clock period (25 ns).

5.3.2. Set window offset (CODE 11xx)

It allows to set the offset of the trigger window with respect to the trigger itself, i.e. the time difference (expressed in clock cycles, 1 cycle = 25 ns) between the trigger time and the start of the trigger window. After this OPCODE has been sent, a 16-bit word must be written at the same location of the OPCODE itself. The microcontroller will remain in a

wait status until a 16-bit word is written (bits [12;15] are meaningless). The window offset value must be greater than -32000, or hex 8300. The offset and width value must be set according to the constraints described in § 2.2.1. The window offset is synchronised with the clock cycle, thus there could be a jitter of one clock cycle in the actual offset position.

5.3.3. Set extra search margin (CODE 12xx)

After this OPCODE has been sent, a 16-bit word must be written at the same location of the OPCODE itself. The microcontroller will remain in a wait status until a 16-bit word is written. The margin value (clock cycles) can be any 12 bit value (bits [12;15] are meaningless), though reasonable values are not greater than 50.

5.3.4. Set reject margin (CODE 13xx)

It allows to set the reject margin of the trigger, expressed in clock cycles. After this OPCODE has been sent, a 16-bit word must be written at the same location of the OPCODE itself. The microcontroller will remain in a wait status until a 16-bit word is written. The margin value can be any 12 bit value (bits [12;15] are meaningless), 0 sets the margin at the beginning of the trigger window.

5.3.5. Enable subtraction of trigger time (CODE 14xx)

It allows to enable the trigger time tag subtraction: in this operating mode the time measurements are referred to the trigger time tag, i.e. to the beggining of the trigger window.

5.3.6. Disable subtraction of trigger time (CODE 15xx)

It allows to disable the trigger time tag subtraction: in this operating mode the time measurements are referred to the latest Bunch reset.

5.3.7. Read trigger configuration (CODE 16xx)

It allows to redout the trigger configuration. After this OPCODE has been sent, five 16-bit words must be read at the MICRO register address. The microcontroller will remain in a wait status until five 16-bit word are read. The first word represents the Match Window Width, the second the Trigger Counter Offset, the third the Search Window Width and the fourth the Reject Counter Offset. The fifth word's LSB has the following meaning:
LSB = 0 → trigger time subtraction disabled;
LSB = 1 → trigger time subtraction enabled.

5.4. TDC edge detection and resolution OPCODEs

These OPCODEs allow to set the measurement type on the TDC channels and their resolution.

5.4.1. Enable trailing edge (CODE 20xx)

It allows to enable the TDCs to detect the HIT signals' trailing edge on all the channels.

5.4.2. Enable leading edge (CODE 21xx)

It allows to enable the TDCs to detect the HIT signals' leading edge on all the channels.

5.4.3. Enable paired measurement (CODE 22xx)

It allows to enable the TDCs to detect the HIT signals' both leading edge and width on all the channels.

5.4.4. Read edge detection configuration (CODE 23xx)

It allows to readout the TDCs' edge detection configuration. After this OPCODE has been sent, a 16-bit word must be read at the MICRO register address. The microcontroller will remain in a wait status until a 16 bit word is read. This word's two LSBs have the following meaning:

00 → meaningless;
01 → trailing edge;
10 → leading edge;
11 → paired measurement.

5.4.5. Set LSB of leading/trailing edge (CODE 24xx)

It allows to set the LSB value in leading/trailing edge detection configuration. After this OPCODE has been sent, a 16-bit word must be written at the MICRO register address. The microcontroller will remain in a wait status until a 16 bit word is written. This word's bit 0 and bit 1 have the following meaning:

00 → 800 ps;
01 → 200 ps;
10 → 100 ps.

5.4.6. Set leading time and width resolution when pair (CODE 25xx)

It allows to set the resolution of the leading edge arrival time measurement and of the pulse width in pair detection configuration. After this OPCODE has been sent, a 16-bit word must be written at the MICRO register address. The microcontroller will remain in a wait status until a 16 bit word is written. This word's bits [0;2] and bits [8;11] have the following meaning:

bits [0;2]:
000 → 100ps
001 → 200ps
010 → 400ps
011 → 800ps
100 → 1.6ns
101 → 3.12ns
110 → 6.25ns
111 → 12.5ns

bits [8;11]:
0000 → 100ps
0001 → 200ps
0010 → 400ps
0011 → 800s
0100 → 1.6ns
0101 → 3.2ns
0110 → 6.25ns
0111 → 12.5ns
1000 → 25ns
1001 → 50ns
1010 → 100ns
1011 → 200ns
1100 → 400ns
1101 → 800ns
1110 → not valid
1111 → not valid

5.4.7. Read resolution (CODE 26xx)

It allows to readout the TDCs' resolution.
In leading/trailing edge detection configuration, after this OPCODE has been sent, a 16-bit word must be read at the MICRO register address. The microcontroller will remain in a wait status until a 16 bit word is read. This word's bit 0 and bit 1 have the following meaning:

00 → 800 ps;
01 → 200 ps;
10 → 100 ps.

In pair detection configuration, after this OPCODE has been sent, a 16-bit word must be read at the MICRO register address. The microcontroller will remain in a wait status until a 16 bit word is read. This word's bits [0;2] and bits [8;11] have the following meaning:

bits [0;2]:
000 → 100ps
001 → 200ps
010 → 400ps
011 → 800ps
100 → 1.6ns
101 → 3.12ns
110 → 6.25ns
111 → 12.5ns

bits [8;11]:
0000 → 100ps
0001 → 200ps
0010 → 400ps
0011 → 800s
0100 → 1.6ns
0101 → 3.2ns
0110 → 6.25ns
0111 → 12.5ns
1000 → 25ns
1001 → 50ns
1010 → 100ns
1011 → 200ns
1100 → 400ns
1101 → 800ns
1110 → not valid
1111 → not valid

5.4.8. Set channel dead time between hits (CODE 28xx)

It allows to set the double hit resolution (i.e. the dead time between two subsequent hits). After this OPCODE has been sent, a 16-bit word must be written at the MICRO register address. The microcontroller will remain in a wait status until this 16-bit word is written. This word's two LSBs allow the following settings:

00 → ~5ns;
01 → ~10ns;
10 → ~30ns;
11 → ~100ns.

5.4.9. Read channel dead time between hits (CODE 29xx)

It allows to readout the double hit resolution (i.e. the dead time between two subsequent hits). After this OPCODE has been sent, a 16-bit word must be read at the MICRO register address. The microcontroller will remain in a wait status until this 16-bit word is read. This word's two LSBs have the following meaning:

00 → ~5ns;
01 → ~10ns;
10 → ~30ns;
11 → ~100ns.

5.5. TDC Readout

The following OPCODEs allow some settings operating with the Trigger Matching Mode (TDCs' Header and EOB, events' composition).

5.5.1. Enable TDC Header and EOB in readout (CODE 30xx)

It allows to enable the TDCs' Header and EOB during data readout.

5.5.2. Disable TDC Header and EOB in readout (CODE 31xx)

It allows to disable the TDCs' Header and EOB during data readout.

5.5.3. Read TDC Header and EOB status (CODE 32xx)

It allows to readout whether the TDCs' Header and EOB are enabled or not. After this OPCODE has been sent, a 16-bit word must be read at the MICRO register address. The microcontroller will remain in a wait status until this 16-bit word is read. This word's LSB has the following meaning:

LSB = 0 → TDC header/EOB disabled;
LSB = 1 → TDC header/EOB enabled.

5.5.4. Set maximum number of hits per event (CODE 33xx)

It allows to set the maximum number of hits for each TDCs' event (event size). After this OPCODE has been sent, a 16-bit word must be written at the MICRO register address. The microcontroller will remain in a wait status until this 16-bit word is written. This word's four LSBs allow the following settings:

0000 → 0;
0001 → 1;
0010 → 2;

0011 → 4;
0000 → 8;
0101 → 16;
0110 → 32;
0111 → 64;
1000 → 128;
1001 → no limit;
1010÷1111 → meaningless.

the actual number of words is given by: the number of hits on each channel, the event's Header and EOB, the TDCs' Header and EOB, if enabled.

5.5.5. Read maximum number of hits per event (CODE 34xx)

It allows to readout the maximum number of hits for each TDCs' event (event size). After this OPCODE has been sent, a 16-bit word must be read at the MICRO register address. The microcontroller will remain in a wait status until this 16-bit word is read. This word's four LSBs have the following meaning:

0000 → 0;
0001 → 1;
0010 → 2;
0011 → 4;
0000 → 8;
0101 → 16;
0110 → 32;
0111 → 64;
1000 → 128;
1001 → no limit;
1010÷1111 → meaningless.

5.5.6. Enable TDC error mark (CODE 35xx)

It allows the TDCs to put an error mark into the events when a global error occurs (default).

5.5.7. Disable TDC error mark (CODE 36xx)

Does not allow the TDCs to put an error mark into the events when a global error occurs.

5.5.8. Enable bypass TDC if error (CODE 37xx)

Enables the TDCs' bypass when a global error occurs.

5.5.9. Disable bypass TDC if error (CODE 38xx)

Disables the TDCs' bypass when a global error occurs.

5.5.10. Set TDC internal error type (CODE 39xx)

It allows to enable the TDCs' internal error types for the production of the global error signal. After this OPCODE has been sent, a 16-bit word must be written at the MICRO register address. The microcontroller will remain in a wait status until this 16-bit word is written. This word's 11 LSB allow (each one independently) to enable one type of internal error (bit set to one: error enabled). Refer to [6], page 30, for errors description.

5.5.11. Read TDC internal error type (CODE 3Axx)

It allows to readout the TDCs' enabled internal error types for the production of the global error signal. After this OPCODE has been sent, a 16-bit word must be read at the MICRO register address. The microcontroller will remain in a wait status until this 16-bit word is read. This word's 11 LSB allow (each one independently) to enable one type of internal error (bit set to one: error enabled). Refer to [6], page 30, for errors description.

5.5.12. Set effective size of readout FIFO (CODE3Bxx)

It allows to set the actual size of the L1 buffer (see § 2.2). After this OPCODE has been sent, a 16-bit word must be written at the MICRO register address. The microcontroller will remain in a wait status until a 16 bit word is written. This word's bits [0;2] have the following meaning:

000 → 32 words
001 → 64 words
010 → 96 words
011 → 128 words
100 → 160 words
101 → 192 words
110 → 224 words
111 → 256 words

5.5.13. Read effective size of readout FIFO (CODE3Cxx)

It allows to read the actual size of the L1 buffer (see § 2.2). After this OPCODE has been sent, a 16-bit word must be read at the MICRO register address. The microcontroller will remain in a wait status until a 16 bit word is read. This word's bits [0;2] have the following meaning:

000 → 32 words
001 → 64 words
010 → 96 words
011 → 128 words
100 → 160 words
101 → 192 words

110 → 224 words
111 → 256 words (default)

5.6. Channel enable OPCODES

The following OPCODEs allow either to enable or to disable individually the TDCs' channels. These operations can be performed *run time* (i.e. even during TDCs' operation).

5.6.1. Enable channel nn (CODE 40nn)

It allows to enable the channel **nn** (in the range %00÷%7F, corresponding to 0÷127).

5.6.2. Disable channel nn (CODE 41nn)

It allows to disable the channel **nn** (in the range %00÷%7F, corresponding to 0÷127).

5.6.3. Enable all channels (CODE 42xx)

It allows to enable all the channels.

5.6.4. Disable all channels (CODE 43xx)

It allows to disable all the channels.

5.6.5. Write enable pattern (CODE 44xx)

It allows to write a pattern for enabling a set of channels. After this OPCODE has been sent, eight 16-bit words must be written at the MICRO register address. The microcontroller will remain in a wait status until these eight 16-bit word are written. These words' bits allows the following settings:

word 0: bit 0 ⇒ channel 0
word 0: bit 1 ⇒ channel 1
:
word 0: bit 15 ⇒ channel 15

word 1: bit 0 ⇒ channel 16
word 1: bit 1 ⇒ channel 17
:
word 1: bit 15 ⇒ channel 31
:
word 7: bit 0 ⇒ channel 112
word 7: bit 1 ⇒ channel 113

:
word 7: bit 15 ⇒ channel 127

- bit *n* = 0 ⇒ channel *n* disabled;
- bit *n* = 1 ⇒ channel *n* enabled.

5.6.6. Read enable pattern (CODE 45xx)

It allows to readout the enable pattern. After this OPCODE has been sent, eight 16-bit words must be read at the MICRO register address. The microcontroller will remain in a wait status until these eight 16-bit word are read. These words' bits have the following meaning:

word 0: bit 0 ⇒ channel 0
word 0: bit 1 ⇒ channel 1
:
word 0: bit 15 ⇒ channel 15

word 1: bit 0 ⇒ channel 16
word 1: bit 1 ⇒ channel 17
:
word 1: bit 15 ⇒ channel 31
:
word 7: bit 0 ⇒ channel 112
word 7: bit 1 ⇒ channel 113
:
word 7: bit 15 ⇒ channel 127
- bit *n* = 0 ⇒ channel *n* disabled;
- bit *n* = 1 ⇒ channel *n* enabled.

5.7. Adjust OPCODES

The following OPCODEs allow to set or readout some adjustment offsets (fine counters) and global offsets (coarse counters) of the TDCs' channels.

5.7.1. Set channel nn adjust (CODE 50nn)

It allows to add a positive offset to the channel **nn** (nn: %00÷%7F). After this OPCODE has been sent, a 16-bit word must be written at the MICRO register address. The microcontroller will remain in a wait status until this 16-bit word is written. The written value ranges from 0 to 255 clock cycles (bits 8 to 15 are meaningless).

5.7.2. Read channel nn adjust (CODE 51nn)

It allows to read out the positive offset of the channel **nn** (nn: %00÷%7F). After this OPCODE has been sent, a 16-bit word must be read at the MICRO register address. The microcontroller will remain in a wait status until this 16-bit word is read. The read value ranges from 0 to 255 clock cycles (bits 8 to 15 are meaningless).

5.7.3. Set global offset (CODE 52xx)

It allows to add a global offset to the fine (vernier) and coarse counters. After this OPCODE has been sent, two 16-it words must be written at the MICRO register address. The microcontroller will remain in a wait status until these two 16-bit word are written. The first word carries the coarse counter offset (11 significant bits), the second one carries the fine counter offset (5 significant bits).

5.7.4. Read global offset (CODE 53xx)

It allows to readout the global offset of the fine (vernier) and coarse counters. After this OPCODE has been sent, two 16-bit words must be read at the MICRO register address. The microcontroller will remain in a wait status until these two 16-bit word are read. The first word carries the coarse counter offset (11 significant bits), the second one carries the fine counter offset (5 significant bits).

5.8. Miscellaneous

*5.8.1. Read programmed ID of TDC n (CODE 60*0*n)*

It allows to readout the TDC ID programmed for the TDC **n** (n: %0÷%3). After this OPCODE has been sent, a 16-bit word must be read at the MICRO register address. The microcontroller will remain in a wait status until this 16-bit word is read.

5.8.2. Read firmaware rev. of microcontroller (CODE 61xx)

It allows to readout the firmware revision of the microcontroller. After this OPCODE has been sent, a 16-bit word must be read at the MICRO register address. The microcontroller will remain in a wait status until this 16-bit word is read.

5.8.3. Reset PLL and DLL (CODE 62xx)

It allows to reset the TDCs' PLL (Phase Locked Loop) and DLL (Delay Locked Loop), refer to [2].

5.9. Advanced

These OPCODEs refer to some tests and advanced modes. For further information, please refer to the TDCs' handbook ("*High performance general purpose TDC specification*", J. Christiansen CERN/EP-MIC, [6]).

5.9.1. Write word *nn* into the Scan Path Setup (CODE 70nn)

It allows to write the word **nn** (nn: %0÷%40) into the Scan Path Setup. After this OPCODE has been sent, a 16-bit word must be written at the MICRO register address. The microcontroller will remain in a wait status until this 16-bit word is written.

5.9.2. Read word *nn* into the Scan Path Setup (CODE 71nn)

It allows to read the word **nn** (nn: %0÷%40) from the Scan Path Setup. After this OPCODE has been sent, a 16-bit word must be read at the MICRO register address. The microcontroller will remain in a wait status until this 16-bit word is read.

5.9.3. Load the Scan Path Setup (CODE 72xx)

It allows to load into the TDC the current Scan Path Setup.

5.9.4. Reload the default Scan Path Setup (CODE 73xx)

It allows to load into the TDC the default Scan Path Setup.

5.9.5. Read errors in the TDC *n* status (CODE 74*0n*)

It allows to read errors in the TDC **nn** (nn: %0÷%3). After this OPCODE has been sent, a 16-bit word must be read at the MICRO register address. The microcontroller will remain

in a wait status until this 16-bit word is read. This word's 11 LSB are set to one if the corresponding internal error (see § 5.5.10) has occurred, zero otherwise.

*5.9.6. Read the DLL LOCK bit of the TDC n (CODE 75*0*n)*

It allows to read DLL LOCK bit status in the TDC **n** (nn: %0÷%3). After this OPCODE has been sent, a 16-bit word must be read at the MICRO register address. The microcontroller will remain in a wait status until this 16-bit word is read. This word's LSB indicates the DLL LOCK status:
LSB = 0 → DLL LOCK enabled;
LSB = 1 → DLL LOCK disabled.

*5.9.7. Read the TDC n status (CODE 76*0*n)*

It allows to read the TDC **n** (nn: %0÷%3) status. After this OPCODE has been sent, 4 16-bit words must be read at the MICRO register address. The microcontroller will remain in a wait status until four 16-bit words are read.

6. VME Interface

6.1. Register address map

The Address map for the Model V890 is listed in Table 6.1. All register addresses are referred to the Base Address of the board, i.e. the addresses reported in the Tables are the offsets to be added to the board Base Address.

Table 6.1: Address Map for the Model V890

Register content	Address	Type	Access mode
Output Buffer (**)	0x0000÷0x0FFC	*Read only*	D32
Control Register (*) (***)	0x1000	*Read/Write*	D16
Status Register	0x1002	*Read only*	D16
ADER_32	0x1004	*Read/Write*	D16
ADER_24	0x1006	*Read/Write*	D16
Enable ADER	0x1008	*Read/Write*	D16
Interrupt Level (*)	0x100A	*Read/Write*	D16
Interrupt Vector (*)	0x100C	*Read/Write*	D16
GEO Address (***)	0x100E	*Read only*	D16
MCST Base Address (***)	0x1010	*Read/Write*	D16
MCST/CBLT Ctrl (***)	0x1012	*Read/Write*	D16
Module Reset (*)	0x1014	*Write only*	D16
Count Reset (*)	0x1016	*Write only*	D16
Software Clear (*)	0x1018	*Write only*	D16
Software Trigger (*)	0x101A	*Write only*	D16
Event Counter	0x101C	*Read only*	D32
Event Stored	0x1020	*Read only*	D16
Almost Full Level (*) (***)	0x1022	*Read/Write*	D16
BLT Event Number(*)(***)	0x1024	*Read/Write*	D16
Firmware Revision	0x1026	*Read only*	D16
Testreg (***)	0x1028	*Read/Write*	D32
Out Prog Control (*) (***)	0x102C	*Read/Write*	D16
Micro (*) (***)	0x102E	*Read/Write*	D16
Micro Handshake (*)	0x1030	*Read/Write*	D16
TDC0	0x1100	*Read/Write*	D16
TDC1	0x1104	*Read/Write*	D16
TDC2	0x1108	*Read/Write*	D16
TDC3	0x110C	*Read/Write*	D16
Dummy 32 (*)	0x1200	*Read/Write*	D32
Dummy 16 (*)	0x1204	*Read/Write*	D16

(*) MSCT access allowed.

(**) BLT/CBLT access allowed.

(***) A write access to these registers causes a module's CLEAR.

The following registers contain some factory settings of the boards:

Table 6.2: Configuration ROM Addresses

Register content	Address	Content
Manufacturer's ID	0x1026÷0x102E	00-40-E6
Board ID	0x1032÷0x103E	00-00-03-34
Revision ID	0x104E	0
Slave Char. Parameters	0x10E2	5
Interrupter Capabilities	0x10F6	FE
Function 0 Data Access Width	0x1102	85
Function 1 Data Access Width	0x1106	85
Function 0 AM Code Mask	0x1122÷0x123E	FF-00-00-00-00-FF-00
Function 1 AM Code Mask	0x1142÷0x115E	FF-00-00-00-00-FF-00
Function 0 Address Decoder Mask	0x1622÷0x162E	FF-FF-00-02
Function 1 Address Decoder Mask	0x1632÷0x163E	00-FF-00-00
Module Serial Number	0x10F2, 0x10F6	Serial Number

6.2. Output Buffer Register

(Base Address + 0x0000 ÷ 0x0FFC, read only)

Settings upon the acquisition modes (i.e. the module's programming) can be performed as described in § 5.2.

6.2.1. Trigger Matching Mode

Each time a trigger signal occurs, the hits accumulated by the enabled channels are loaded into the output buffer where they are organised as follows.
The data in the buffer are organised in events.
Each event consists of:

- the **Global Header** (32 bit), that contains the geographical address and the trigger counter (up to 24 bit);
- the **data** collected by the four TDC chips, each of them composed as follows:
  - a **TDC Header** (if enabled);
  - the **TDC Measurements**;
  - the **TDC Errors** (if enabled);
  - the **TDC End Of Block** (if enabled);
- the global **Global End Of Block** (GEOB), which contains the geographical address, the event words counter (up to 16 bit), and a "status" (3 bit), whose value is 1 if an error has occurred, 2 if the Output Buffer is in *overflow* and 0 in any other case.

Bits [31,30] identify the word type:
01→Global Header, Global EOB
00→TDC Data (TDC Header, TDC Measurement, TDC Error, TDC EOB)

Bits [29÷27] identify the TDC Data type:
001→TDC Header
000→TDC Measurement
100→TDC Error
011→TDC EOB

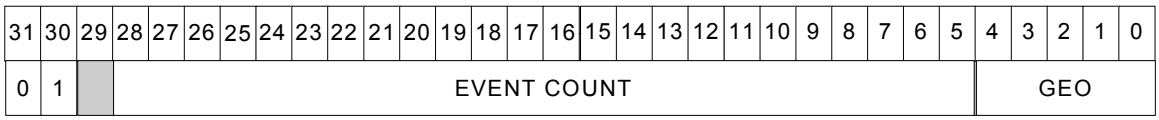


Fig. 6.1: Output buffer: the Global Header

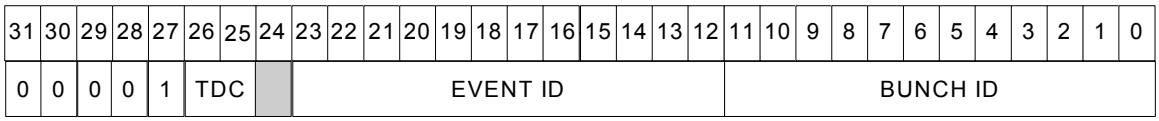


Fig. 6.2: Output buffer: the TDC Header

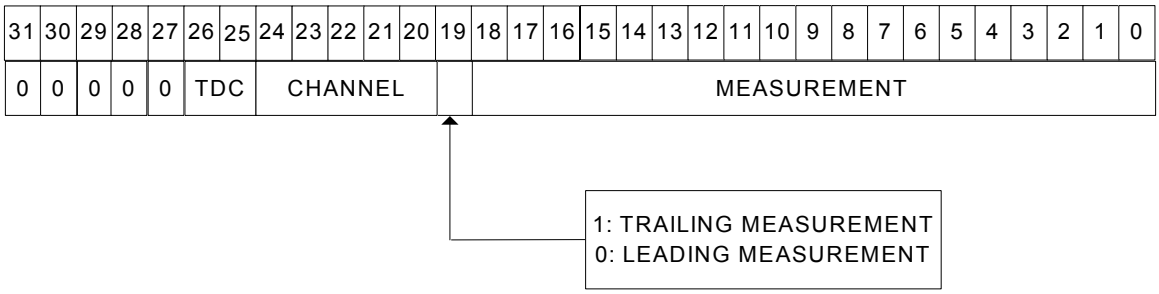


Fig. 6.3: Output buffer: the TDC Measurement (see § 6.2.1.1)

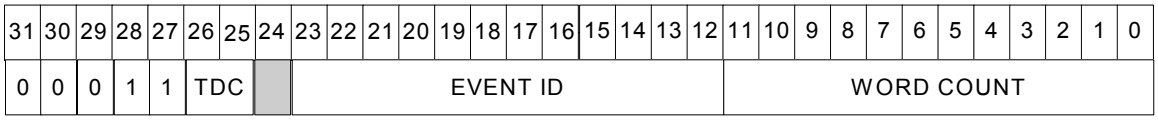


Fig. 6.4: Output buffer: the TDC End Of Block

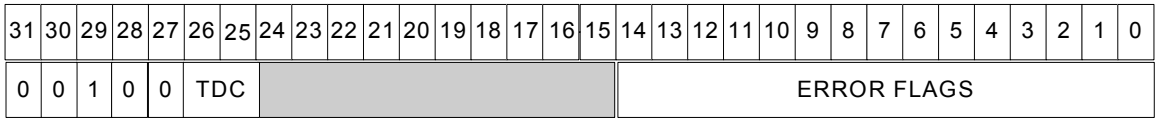


Fig. 6.5: Output buffer: the TDC Error

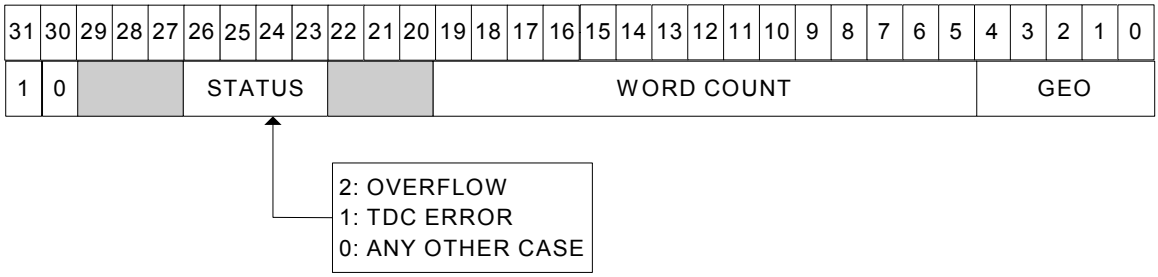
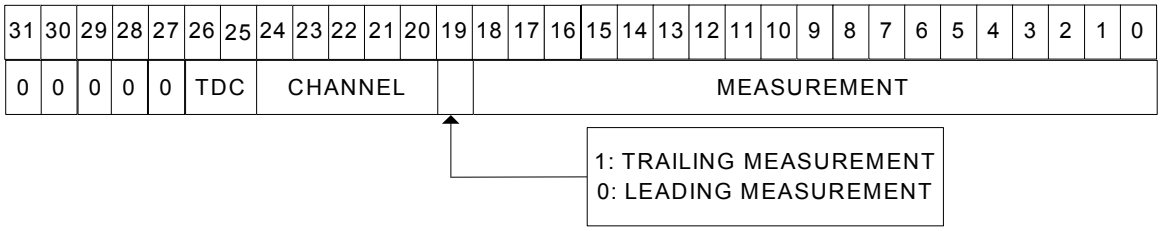
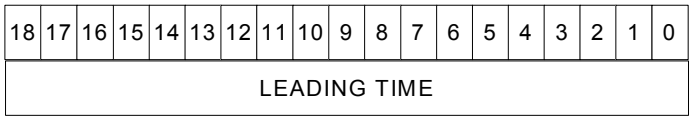


Fig. 6.6: Output buffer: the Global End Of Block

6.2.1.1. TDC Measurement

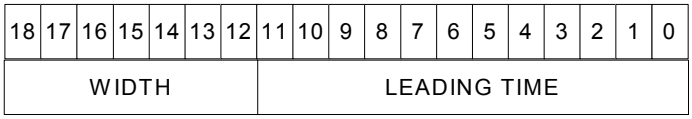


LEADING MEASUREMENT:
Single edge:



Leading time→the Leading Edge is measured with the programmed resolution

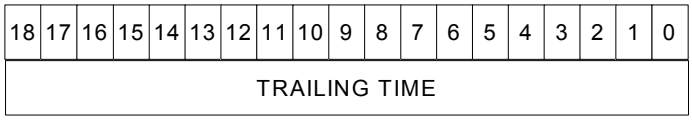
Leading/Trailing edge (mixed):



Leading time→the Leading Edge is measured with the programmed resolution

Width→the pulse width is measured with the programmed resolution

TRAILING MEASUREMENT:



Trailing time→the Trailing Edge is measured with the programmed resolution

6.2.2. Continuous Storage Mode

The digits of the output data have the following structure:

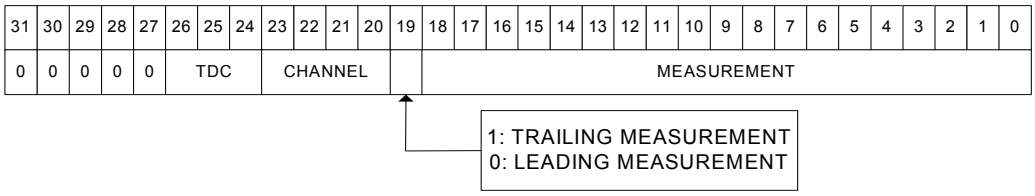


Fig. 6.7: Output buffer: CSM data format

6.2.3. Filler

The Filler Data, used to complete BLT transfers (see § 4.6), are identified by Bit [30,31]→11

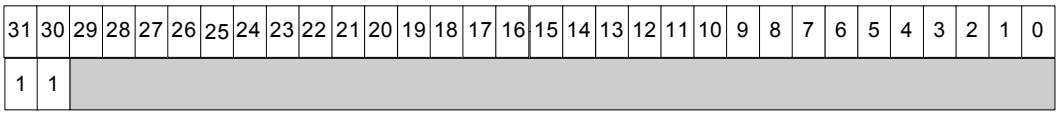


Fig. 6.8: Output buffer: the Filler

6.3. GEO Address Register

(Base Address + 0x100E, read/write; write cycles are allowed only for the versions without PAUX connector)

This register contains the geographical address of the module, i.e. the slot number picked up from the JAUX connector on the VME backplane. The register is filled up upon arrival of a RESET. The register content is the following:

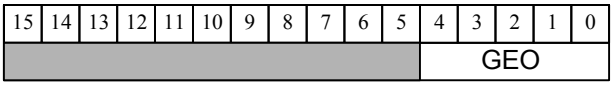


Fig. 6.9: Geographical address register

GEO [4...0] corresponds to A23...A19 in the address space of the CR/CSR area: each slot has a relevant number whose binary encoding consists of the GEO ADDR 4 to 0.

In the versions without the PAUX connector this register can be also written. The bits of the GEO Address register are set to 0 by default. In CBLT operation it is up to the User to write the correct GEO address of the module in this register before operating so that the GEO address will be contained in the HEADER and the EOB words for data identification.

If a write access to the GEO register is performed in the versions with the PAUX connector, the module does not respond and the bus will go in timeout.

N.B.: In the case of versions where the SN5…SN1 lines are not available (i.e. the versions without the PAUX connector), addressing via geographical address is not available.
Although in these versions it is possible to perform a write access to the GEO Address Register for data identification during CBLT operation (see § 4.4.4), avoid to use the GEO Address Register for addressing purposes when there is no PAUX.

N.B.: after a write access to the GEO Address register, it is necessary to perform a reset to make the change active.

6.4. MCST Base Address Register

(Base Address + 0x1010, read/write)

This register contains the most significant bits of the MCST/CBLT address of the module set via VME, i.e. the address used in MCST/CBLT operations. Refer to § 4.4.4 for details about MCST/CBLT addressing mode.

The register content is the following:

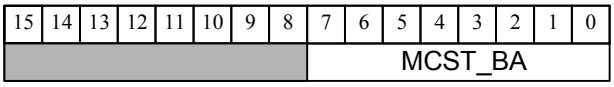


Fig. 6.10: MCST/CBLT address register

Default setting (i.e. at power ON or after hardware reset) is 0xAA.

6.5. MCST/CBLT Control Register

(Base Address + 0x1012, read/write)

This register allows performing the MCST/CBLT settings of the module: the status of the boards according to the bits' value is the following:

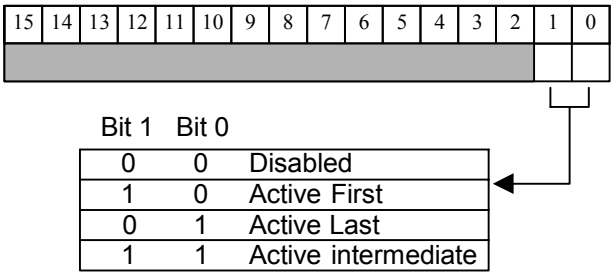


Fig. 6.11: MCST/CBLT Control Register

6.6. Interrupt Level Register

(Base Address + 0x100A, read/write)

The 3 LSB of this register contain the value of the interrupt level (Bits 3 to 15 are meaningless). Default setting is 0x0. In this case interrupt generation is disabled.

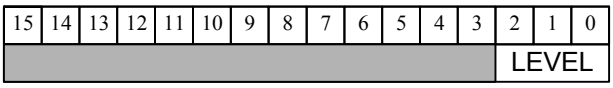


Fig. 6.12: Interrupt Level Register

6.7. Interrupt Vector Register

(Base Address + 0x100C, read/write)

This register contains the STATUS/ID that the V890 INTERRUPTER places on the VME data bus during the Interrupt Acknowledge cycle (Bits 8 to 15 are meaningless). Default setting is 0x00.

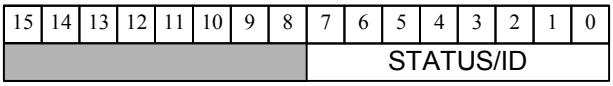


Fig. 6.13: Interrupt Vector Register

6.8. Status Register

(Base + 0x1002, read only)

This register contains information on the status of the module. TERM ON and TERM OFF refer to the terminations of the CONTROL bus lines: the last module in a chain controlled via the front panel CONTROL connector must have these terminations ON, while all the others must have them OFF. The insertion or removal of the terminations is performed either via hardware or via software (see § 3.6.1and § 6.9).

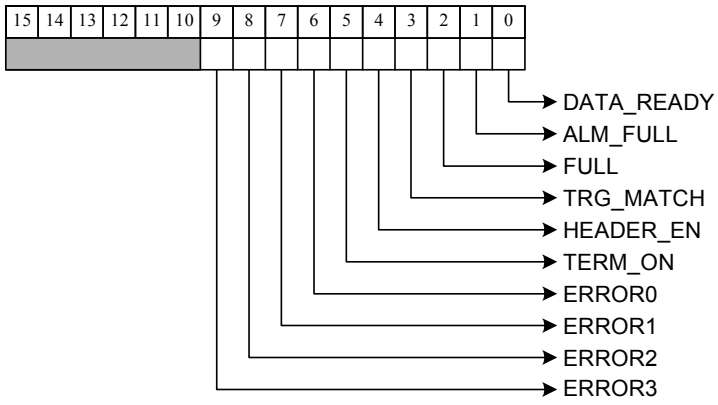


Fig. 6.14: Status Register

DREADY: Indicates that there are data (at least 1 event) in the Output Buffer.
= 0 No Data Ready;
= 1 Data Ready.

ALMOST FULL: Indicates whether the Almost Full Level has been met or not
= 0 Almost Full Level not met yet;
= 1 Almost Full Level met.

FULL: Indicates whether the Output Buffer is FULL (i.e. it contains a number of words equal to SIZE-4) or not (see § 5.5.4)
= 0 Output Buffer not FULL;
= 1 Output Buffer FULL.

TRG MATCH: Indicates the selected operating mode
= 0 trigger matching mode;
= 1 continuous storage mode.

HEADER EN: Indicates the enabling of the TDCs' Header and EOB
= 0 TDCs' Header and EOB disabled;
= 1 TDCs' Header and EOB enabled.

TERM ON: Termination ON/OFF bit
= 0 all Control Bus Terminations are OFF;
= 1 all Control Bus Terminations are ON.

ERROR i: Indicates an error in the TDCi (i = 0÷3)
= 0 TDC i operated properly.
= 1 TDC i error has occurred.

6.9. Control Register

(Base Address + 0x1000, read/write)

This register allows performing some general settings of the module.

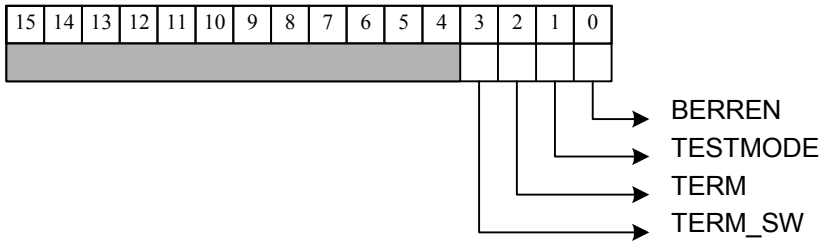


Fig. 6.15: Control Register

BERR EN: Bus Error enable bit. Used in Block Transfer mode only.
= 0 the module sends a DTACK signal until the CPU inquires the module (default);
= 1 the module is enabled to generate a Bus error to finish a block transfer.

TESTMODE: Test mode bit.
= 0 Test mode OFF
= 1 Test mode ON

TERM: Set the software termination status
= 0 termination OFF;
= 1 termination ON.

TERM SW: Allows to select the termination mode
= 0 termination via dip-switch;
= 1 termination via software.

6.10. Enable Ader Register

(Base Address + 0x1008, write only)

This Register allows to activate the Base Address relocation. The GEO address is not influenced by the relocation.

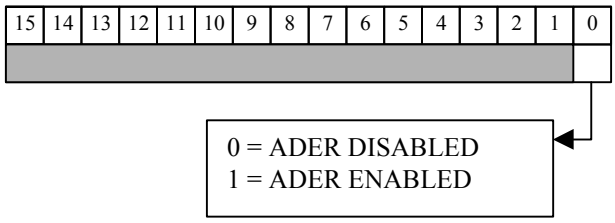


Fig. 6.16: Enable ADER Register

ENABLE ADDRESS: Select Address bit.
= 0 base address is selected via Rotary Switch (default);
= 1 base address is selected via High and Low ADER registers.

N.B.: this register determines the base address variation of the module. Any VME accesses to the previous board's base address does not work.

6.11. ADER 32 Register

(Base Address + 0x1004, read/write)

This register contains the A31...A24 bits of the address of the module: it can be set via VME for a relocation of the Base Address of the module. The register content is the following:

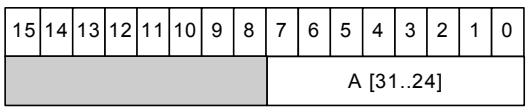


Fig. 6.17: ADER 32 Register

6.12. ADER 24 Register

(Base Address + 0x1006 read/write)

This register contains the A23...A16 bits of the address of the module: it can be set via VME for a relocation of the Base Address of the module. The register content is the following:

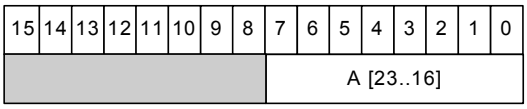


Fig. 6.18: ADER 24 Register

6.13. Event Counter Register

(Base Address + 0x101C, read only)

This register contains the number of events written into the Output Buffer since the latest module's reset.

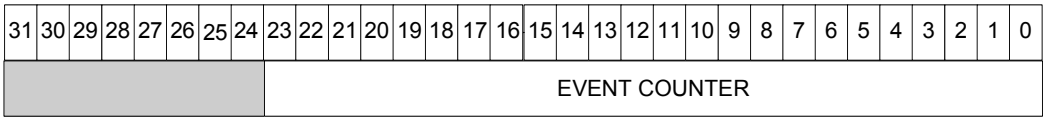


Fig. 6.19: Event Counter Register

6.14. Event Stored register

(Base Address + 0x1020, read only)

This register contains the number of events currently stored in the Output Buffer.

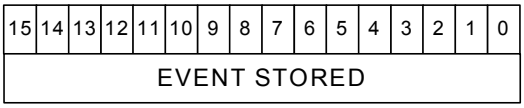


Fig. 6.20: Event Stored Register

6.15. Almost Full Level Register

(Base Address + 0x1022, read/write)

This Register allows the User to set the Almost Full Level of the Output Buffer. When the Output Buffer contains a number of words at least equal to the Almost Full Level, then an Interrupt Request (IRQ) is generated (if enabled) and the corresponding bit in the Status Register is set.

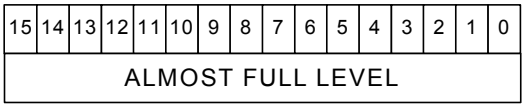


Fig. 6.21: Almost Full Level Register

The default Almost Full Level is 64 words. This Register can be accessed in D16 mode. MCST access is also allowed.
N.B.: A write access to this register causes the Event Counter Register and the Output Buffer to be cleared.

6.16. BLT Event Number Register

(Base Address + 0x1024, read/write)

This register contains the number **Ne** of complete events which is desirable to transfer via BLT. The number of events must be written in a 8 bit word. The Register's default setting is 0, which means that the Event Aligned BLT is disabled. This Register must be accessed in D16 mode. MCST access is also allowed.

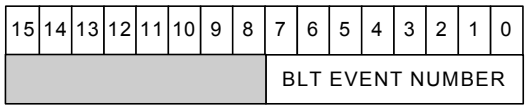


Fig. 6.22: BLT Event Number Register

N.B.: A write access to this register causes the Counters Registers, the Output Buffer, the Trigger Counter and the Output Buffer Event Number Register to be cleared.

6.17. Dummy32 Register

(Base Address + 0x1200, D32, read/write)

This register allows to perform 32 bit test accesses.

6.18. Dummy16 Register

(Base Address + 0x1204, D16, read/write)

This register allows to perform 16 bit test accesses.

6.19. Testreg Register

(Base Address + 0x1028, write only)

If TESTMODE is enabled (see § 6.9), whenever a 32 bit word is written into this register, the Output Buffer receives four test pattern (one from each TDC), programmed via the microcontroller, each of them followed by the 32 bit word written into the TESTREG

register. This allows to test the connection buses between the microcontroller and the TDCs and between the TDCs, the Readout Controller and the Output Buffer.

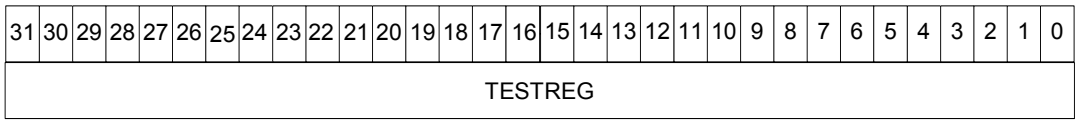


Fig. 6.23: Testreg Register

6.20. OUT_PROG Control Register

(Base Address + 0x102C, read/write)

This register allows to set the function of the OUT_PROG ECL output on the control connector in the following way:

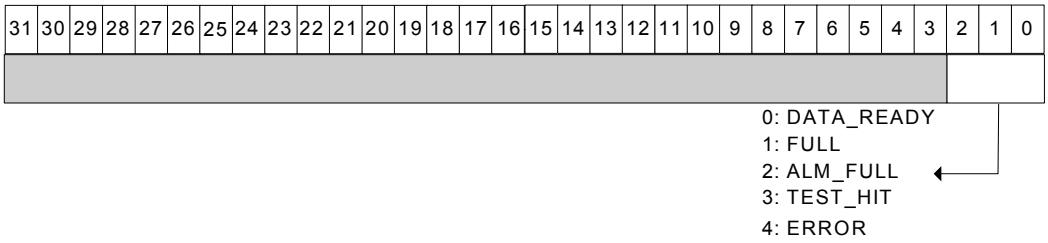


Fig. 6.24: Out_Prog Register

See § 6.8 for details about DATA_READY, FULL, ALM_FULL and ERROR. The TEST_HIT output could be used for a TDC calibration.

6.21. Micro Register

(Base Address + 0x102E, read/write)

This register is used to send instructions to the microcontroller via 16-bit OPCODE setup words. The usage of this register is fully described in § 5.1.

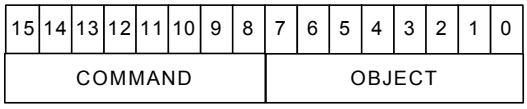


Fig. 6.25: Micro Register

6.22. Micro Handshake Register

(Base Address + 0x1030, read/write)

The Micro Handshake Register is used for the Handshake Protocol between the VME and the microcontroller. It uses only 2 bits: READ OK and WRITE OK.
All read and write operations with the Micro Register can be performed, respectively, when the bit RO or WO is set (see also § 5.1).

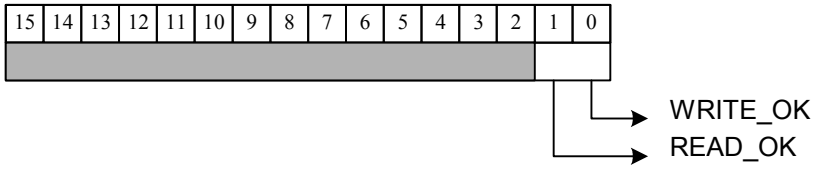


Fig. 6.26: Micro Handshake register

6.23. TDC0-TDC1-TDC2-TDC3 Registers

(Base Address + 0x1100; + 0x1104; + 0x1108; + 0x110C read/write)

32 bit registers to be used for test purposes only.

6.24. Module Reset Register

(Base Address + 0x1014 write only)

A dummy access to this register allows to generate a single shot RESET of the module. Once issued, the Output Buffer, Status and Control registers Front End are reset.

6.25. Software Clear Register

(Base Address + 0x1018 write only)

A write access to this location causes the following:

1. the TDCs are cleared;
2. the output buffer is cleared;
3. the readout controller is reset;
4. the Event counter is set to 0.

6.26. Software Trigger Register

(Base Address + 0x101A write only)

A write access to this location generates a trigger via software.

6.27. Event Counter Reset Register

(Base Address + 0x1016 write only)

A write access to this dummy register clears the Event Counter.

6.28. Firmware Revision Register

(Base Address + 0x1026, read only)

This register contains a 16 bit word identifying the featured firmware revision:
Bits [15÷8] identify the VME interface revision
Bits [7÷0] identify the Readout Controller revision

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VME Interface Revision								Readout Controller Revision							

Fig. 6.32: Firmware Revision Register

APPENDIX A

VME interface timing

A.1 VME Cycle timing in D16/D32 mode

The figure below reports the Data Select (DS0 or DS1) - Data Acknowledge (DTACK) VME cycle in D16 mode and relative timing.

The theoretical minimum duration of the VME cycle in D16/D32 mode is 120 + 60 ns.

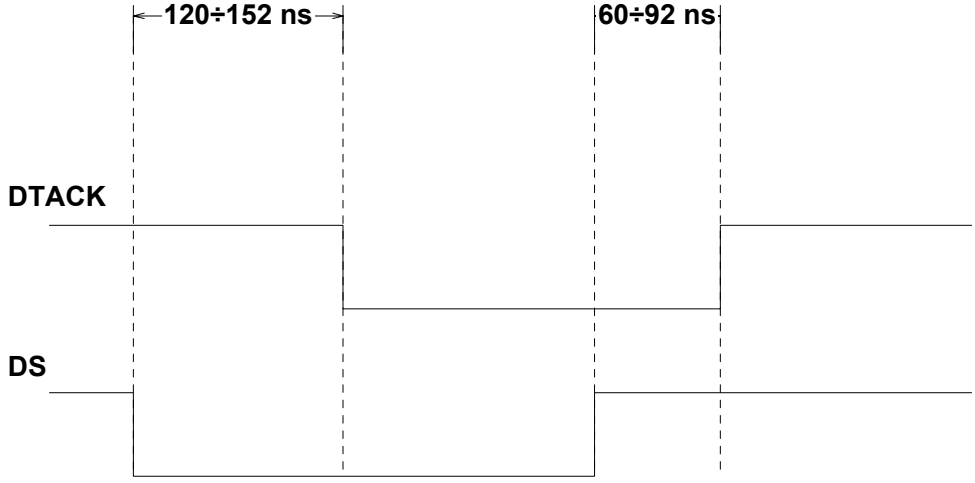


Fig.A.1: VME cycle timing in D16 mode

A.2 VME Cycle timing in BLT / CBLT mode

The figure below reports the Data Select (DS0 or DS1) - Data Acknowledge (DTACK) VME cycle in BLT / CBLT mode and relative timing.

The theoretical minimum duration of the VME cycle in BLT/CBLT mode is 60 + 15 ns.

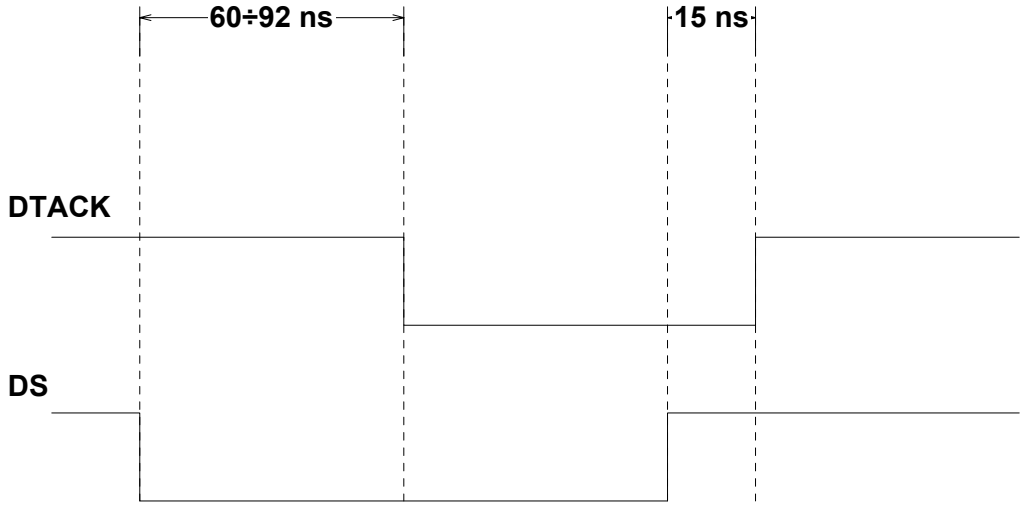


Fig.A.2: VME cycle timing in BLT/CBLT mode

A.3 VME Cycle timing in MBLT / CBLT64 mode

The figure below reports the Data Select (DS) - Data Acknowledge (DTACK) VME cycle in MBLT / CBLT64 mode and relative timing.

The theoretical minimum duration of the VME cycle in MBLT/CBLT64 mode is 120 + 15 ns.

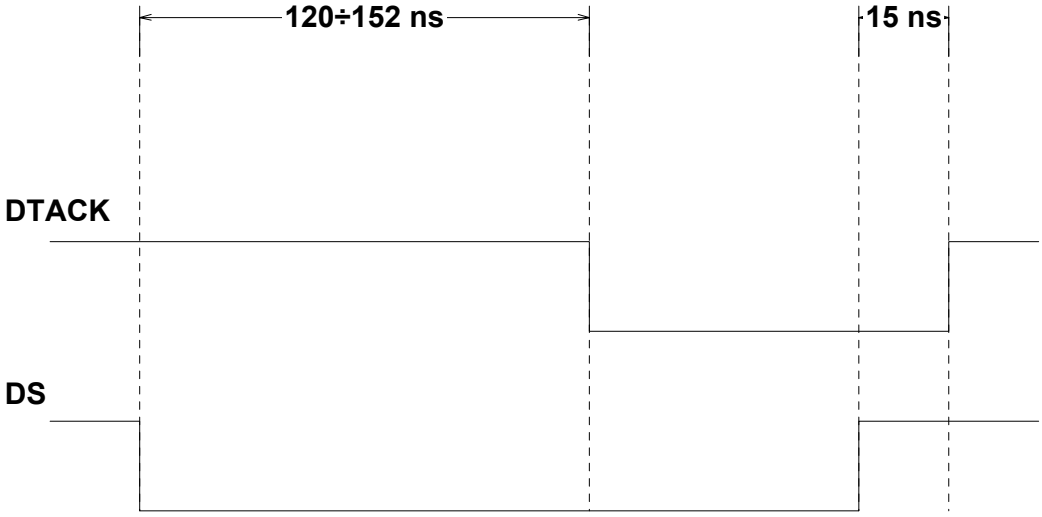


Fig.A.3: VME cycle timing in MBLT/CBLT64 mode

References

[1] C. Cottini, E. Gatti, V. Svelto, “A new method of analog to digital conversion”, NIM vol. 24 p.241, 1963.

[2] C. Cottini, E. Gatti, V. Svelto, “A sliding scale analog to digital converter for pulse height analisys”, in Proc. Int. Symp. Nuclear, Paris, Nov. 1963.

[3] G. Bianchetti et al., “Specification for VMEbus CRATE Type V430”, CERN-EP, January 1990.

[4] -VME64 extensions draft standard, Vita 1.1-199x, draft 1.8, June 13,1997.

[5] -VMEBus for Physics Application, Recommendations & Guidelines, Vita23-199x, draft 1.0, 22 May 1997.

[6] J. Christiansen, “HPTDC Version 2.0”, September 2001