

# **Technical Information Manual**

Revision n. 8  
15 July 2008

**MOD. V965/V965A**

*16/8 CHANNEL  
DUAL RANGE QDC*  
**MANUAL REV.8**

**NPO:**  
**00108/03:V965x.MUTx/08**

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

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



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

### **Disposal of the Product**

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



---

## TABLE OF CONTENTS

|  |           |
|--|-----------|
| <b>1. GENERAL DESCRIPTION.....</b>                         | <b>8</b>  |
| 1.1. OVERVIEW .....  | 8         |
| 1.2. BLOCK DIAGRAM.....                                    | 9         |
| <b>2. PRINCIPLES OF OPERATION .....</b>                    | <b>10</b> |
| 2.1. QAC SECTIONS.....                                     | 10        |
| 2.1.1. <i>The input current-to-current converter</i> ..... | 12        |
| 2.1.2. <i>Pedestal</i> .....                               | 12        |
| 2.1.3. <i>Maximum GATE width</i> .....                     | 12        |
| 2.2. ANALOG TO DIGITAL CONVERSION .....                    | 13        |
| 2.3. ZERO SUPPRESSION .....                                | 13        |
| 2.4. OVERFLOW SUPPRESSION .....                            | 15        |
| 2.5. MULTIPLE EVENT BUFFER (MEB) .....                     | 15        |
| 2.6. EVENT COUNTER.....                                    | 16        |
| 2.7. BUSY LOGIC .....                                      | 17        |
| 2.8. RESET LOGIC.....                                      | 17        |
| 2.9. FAST CLEAR .....                                      | 18        |
| <b>3. TECHNICAL SPECIFICATIONS .....</b>                   | <b>19</b> |
| 3.1. PACKAGING.....  | 19        |
| 3.2. POWER REQUIREMENTS .....                              | 19        |
| 3.3. FRONT PANEL.....                                      | 20        |
| 3.4. EXTERNAL CONNECTORS.....                              | 21        |
| 3.4.1. <i>INPUT connectors</i> .....                       | 21        |
| 3.4.2. <i>CONTROL connectors</i> .....                     | 21        |
| 3.4.3. <i>GATE/COMM connectors</i> .....                   | 21        |
| 3.5. OTHER FRONT PANEL COMPONENTS .....                    | 22        |
| 3.5.1. <i>Displays</i> .....                               | 22        |
| 3.5.2. <i>Switches</i> .....                               | 22        |
| 3.6. INTERNAL HARDWARE COMPONENTS .....                    | 23        |
| 3.6.1. <i>Switches</i> .....                               | 23        |
| 3.6.2. <i>Jumpers</i> .....                                | 23        |
| 3.7. TECHNICAL SPECIFICATIONS TABLE .....                  | 25        |
| <b>4. VME INTERFACE .....</b>                              | <b>26</b> |
| 4.1. ADDRESSING CAPABILITY.....                            | 26        |
| 4.1.1. <i>Addressing via Base Address</i> .....            | 26        |
| 4.1.2. <i>Base addressing examples</i> .....               | 27        |
| 4.1.3. <i>MCST/CBLT addressing</i> .....                   | 27        |
| 4.1.4. <i>MCST/CBLT addressing examples</i> .....          | 29        |

|        |  |    |
|--------|--|----|
| 4.2.   | INTERRUPTER CAPABILITY .....           | 30 |
| 4.2.1. | <i>Interrupt Status/ID</i> .....       | 30 |
| 4.2.2. | <i>Interrupt Level</i> .....           | 30 |
| 4.2.3. | <i>Interrupt Generation</i> .....      | 30 |
| 4.2.4. | <i>Interrupt Request Release</i> ..... | 30 |
| 4.3.   | DATA TRANSFER CAPABILITY .....         | 30 |
| 4.4.   | REGISTER ADDRESS MAP .....             | 30 |
| 4.5.   | OUTPUT BUFFER REGISTER .....           | 34 |
| 4.5.1. | <i>Data storage</i> .....              | 36 |
| 4.5.2. | <i>Data readout</i> .....              | 36 |
| 4.6.   | FIRMWARE REVISION REGISTER.....        | 37 |
| 4.7.   | GEO ADDRESS REGISTER .....             | 38 |
| 4.8.   | MCST/CBLT ADDRESS REGISTER.....        | 38 |
| 4.9.   | BIT SET 1 REGISTER .....               | 39 |
| 4.10.  | BIT CLEAR 1 REGISTER .....             | 39 |
| 4.11.  | INTERRUPT LEVEL REGISTER .....         | 40 |
| 4.12.  | INTERRUPT VECTOR REGISTER.....         | 40 |
| 4.13.  | STATUS REGISTER 1 .....                | 40 |
| 4.14.  | CONTROL REGISTER 1 .....               | 42 |
| 4.15.  | ADDRESS DECODER HIGH REGISTER.....     | 43 |
| 4.16.  | ADDRESS DECODER LOW REGISTER.....      | 43 |
| 4.17.  | SINGLE SHOT RESET REGISTER .....       | 43 |
| 4.18.  | MCST/CBLT CONTROL REGISTER.....        | 44 |
| 4.19.  | EVENT TRIGGER REGISTER .....           | 44 |
| 4.20.  | STATUS REGISTER 2 .....                | 45 |
| 4.21.  | EVENT COUNTER_LOW REGISTER.....        | 45 |
| 4.22.  | EVENT COUNTER_HIGH REGISTER.....       | 46 |
| 4.23.  | INCREMENT EVENT REGISTER .....         | 46 |
| 4.24.  | INCREMENT OFFSET REGISTER.....         | 46 |
| 4.25.  | FAST CLEAR WINDOW REGISTER .....       | 47 |
| 4.26.  | BIT SET 2 REGISTER .....               | 47 |
| 4.27.  | BIT CLEAR 2 REGISTER .....             | 49 |
| 4.28.  | W MEMORY TEST ADDRESS REGISTER .....   | 50 |
| 4.29.  | MEMORY TEST WORD_HIGH REGISTER.....    | 50 |
| 4.30.  | MEMORY TEST WORD_LOW REGISTER.....     | 50 |
| 4.31.  | CRATE SELECT REGISTER.....             | 51 |
| 4.32.  | TEST EVENT WRITE REGISTER .....        | 51 |
| 4.33.  | EVENT COUNTER RESET REGISTER .....     | 52 |

|           |  |           |
|-----------|--|-----------|
| 4.34.     | IPED REGISTER .....                          | 52        |
| 4.34.1.   | <i>Piggy back description</i> .....          | 52        |
| 4.35.     | R MEMORY TEST ADDRESS REGISTER .....         | 52        |
| 4.36.     | SW COMM REGISTER .....                       | 53        |
| 4.37.     | SLIDE CONSTANT REGISTER .....                | 53        |
| 4.38.     | AAD REGISTER.....                            | 53        |
| 4.39.     | BAD REGISTER .....                           | 53        |
| 4.40.     | THRESHOLDS MEMORY .....                      | 54        |
| 4.41.     | ROM MEMORY .....                             | 54        |
| <b>5.</b> | <b>OPERATING MODES .....</b>                 | <b>55</b> |
| 5.1.      | SAFETY INFORMATION .....                     | 55        |
| 5.2.      | PRODUCT DAMAGE PRECAUTIONS .....             | 55        |
| 5.3.      | INSTALLATION.....                            | 55        |
| 5.4.      | POWER ON SEQUENCE .....                      | 55        |
| 5.5.      | POWER ON STATUS .....                        | 56        |
| 5.6.      | OPERATION SEQUENCE .....                     | 56        |
| 5.7.      | TEST MODES .....                             | 57        |
| 5.7.1.    | <i>Random Memory Access Test Mode</i> .....  | 57        |
| 5.7.2.    | <i>Acquisition Test Mode</i> .....           | 58        |
| 5.8.      | BLOCK TRANSFER MODE .....                    | 59        |
| 5.9.      | ADVANCED SETTING AND READOUT MODES .....     | 59        |
| 5.9.1.    | <i>Chained Block Transfer Mode</i> .....     | 60        |
| 5.9.2.    | <i>Multicast Commands</i> .....              | 61        |
| <b>6.</b> | <b>REFERENCES .....</b>                      | <b>62</b> |
|           | <b>APPENDIX A .....</b>                      | <b>63</b> |
|           | <i>VME INTERFACE TIMING</i> .....            | 63        |
| A.1       | VME CYCLE TIMING IN D16/D32 MODE .....       | 64        |
| A.2       | VME CYCLE TIMING IN BLT / CBLT MODE.....     | 64        |
| A.3       | VME CYCLE TIMING IN MBLT / CBLT64 MODE ..... | 65        |

---

## ***LIST OF FIGURES***

|           |   |    |
|-----------|---|----|
| FIG. 1.1: | MODEL V965 BLOCK DIAGRAM .....                    | 9  |
| FIG. 2.1: | SIMPLIFIED BLOCK DIAGRAM OF THE QAC SECTION ..... | 10 |
| FIG. 2.2: | SIGNAL CONVERSION TIMING .....                    | 11 |
| FIG. 2.3: | QAC SECTION .....                                 | 12 |

|   |    |
|---|----|
| FIG. 2.4: BLOCK DIAGRAM OF THE SLIDING SCALE SECTION .....                          | 13 |
| FIG. 2.5: ZERO SUPPRESSION (BIT 8 OF BIT SET 2 REGISTER = 0, DEFAULT SETTING) ..... | 14 |
| FIG. 2.6: ZERO SUPPRESSION (BIT 8 OF BIT SET 2 REGISTER = 1) .....                  | 14 |
| FIG. 2.7: MULTI-EVENT BUFFER: WRITE POINTER AND READ POINTER .....                  | 16 |
| FIG. 2.8: FAST CLEAR WINDOW .....   | 18 |
| FIG. 3.1: MODEL V965/V965A FRONT PANELS .....                                       | 20 |
| FIG. 3.2: COMPONENTS LOCATION (COMPONENTS SIDE) .....                               | 24 |
| FIG. 4.3: BASE ADDRESSING: EXAMPLE 1 .....  | 27 |
| FIG. 4.4: MCST/CBLT ADDRESSING EXAMPLE .....  | 29 |
| FIG. 4.5: OUTPUT BUFFER: THE HEADER .....   | 34 |
| FIG. 4.6: OUTPUT BUFFER: THE DATA WORD FORMAT .....                                 | 34 |
| FIG. 4.7: OUTPUT BUFFER: THE END OF BLOCK .....                                     | 34 |
| FIG. 4.8: OUTPUT BUFFER: NOT VALID DATUM .....                                      | 35 |
| FIG. 4.9: DATA READOUT AND SUPPRESSION .....  | 36 |
| FIG. 4.10: MULTI-EVENT BUFFER: DATA STRUCTURE EXAMPLE .....                         | 37 |
| FIG. 4.11: FIRMWARE REVISION REGISTER .....   | 37 |
| FIG. 4.12: GEOGRAPHICAL ADDRESS REGISTER .....                                      | 38 |
| FIG. 4.13: MCST/CBLT ADDRESS REGISTER .....   | 38 |
| FIG. 4.14: BIT SET 1 REGISTER .....   | 39 |
| FIG. 4.15: INTERRUPT LEVEL REGISTER .....   | 40 |
| FIG. 4.16: INTERRUPT VECTOR REGISTER .....  | 40 |
| FIG. 4.17: STATUS REGISTER 1 .....  | 41 |
| FIG. 4.18: CONTROL REGISTER 1 .....   | 42 |
| FIG. 4.19: ADER HIGH REGISTER .....   | 43 |
| FIG. 4.20: ADER LOW REGISTER .....  | 43 |
| FIG. 4.21: MCST ADDRESS REGISTER .....  | 44 |
| FIG. 4.22: EVENT TRIGGER REGISTER .....   | 44 |
| FIG. 4.23: STATUS REGISTER 2 .....  | 45 |
| FIG. 4.24: EVENT COUNTER LOW REGISTER .....   | 45 |
| FIG. 4.25: EVENT COUNTER HIGH REGISTER .....  | 46 |
| FIG. 4.26: FAST CLEAR WINDOW REGISTER .....   | 47 |
| FIG. 4.27: BIT SET 2 REGISTER .....   | 48 |
| FIG. 4.28: W MEMORY TEST ADDRESS REGISTER .....                                     | 50 |
| FIG. 4.29: TEST WORD_HIGH REGISTER .....  | 50 |
| FIG. 4.30: TEST WORD_LOW REGISTER .....   | 50 |
| FIG. 4.31: CRATE SELECT REGISTER .....  | 51 |

|   |    |
|---|----|
| FIG. 4.32: TEST EVENT WRITE REGISTER .....          | 51 |
| FIG. 4.33: IPED REGISTER .....                      | 52 |
| FIG. 4.34: R MEMORY TEST ADDRESS REGISTER .....     | 52 |
| FIG. 4.35: SLIDE CONSTANT REGISTER .....            | 53 |
| FIG. 4.36: AAD REGISTER .....                       | 53 |
| FIG. 4.37: BAD REGISTER.....                        | 53 |
| FIG. 4.38: THRESHOLD REGISTER.....                  | 54 |
| FIG.A.1: VME CYCLE TIMING IN D16 MODE .....         | 64 |
| FIG.A.2: VME CYCLE TIMING IN BLT/CBLT MODE .....    | 64 |
| FIG.A.3: VME CYCLE TIMING IN MBLT/CBLT64 MODE ..... | 65 |

---

## ***LIST OF TABLES***

|  |    |
|--|----|
| TABLE 3.1: MOD. V965 AND V965A POWER REQUIREMENTS .....    | 19 |
| TABLE 3.2 : MODEL V965 MAIN TECHNICAL SPECIFICATIONS ..... | 25 |
| TABLE 4.1: MODULE RECOGNISED ADDRESS MODIFIER.....         | 26 |
| TABLE 4.2: ADDRESS MAP FOR THE MODEL V965.....             | 32 |
| TABLE 4.3: ADDRESS MAP IN CBLT OPERATION .....             | 33 |
| TABLE 4.4: ADDRESS MAP IN MCST OPERATIONS.....             | 33 |
| TABLE 4.5: ROM ADDRESS MAP FOR THE MODEL V965.....         | 54 |

---

# 1. General description

---

## 1.1. Overview

The **Model V965** is a 1-unit wide VME 6U module housing 16 Charge-to-Digital Conversion channels with current integrating negative inputs ( $50 \Omega$  impedance). For each channel, the input charge is converted to a voltage level by a QAC (Charge to Amplitude Conversion) section. Each QAC output is then converted by two ADCs in parallel; one ADC is preceded by a x1 gain stage, the other by a x8<sup>1</sup> gain stage: a dual input range is then featured:  $0 \div 900$  pC (200 fC LSB) and  $0 \div 100$  pC (25 fC LSB); this allows to avoid saturation with big charge pulses while increasing resolution with small ones.

The outputs of the QAC sections are multiplexed and subsequently converted by two fast 12-bit ADCs. The ADCs use a sliding scale technique to improve the differential non-linearity. Moreover the module allows Zero and Overflow suppression.

The Mod. V965 converted pulses are stored into a VME-accessible 32 events buffer.

The **VME** interface is VME64X standard compliant and features the **A24/A32** and **MultiCast** addressing modes. The data readout occurs either in **D32**, **BLT32**, **MBLT64** mode, or in daisy chain with 32/64 bit **Chained Block Transfers**. The module features a fully programmable **RORA interrupter**.

The board hosts the P1 and P2 VME connectors and fits into both V430 and standard VME 6U crates. It also supports the "live insertion", allowing the User to insert (or remove) the board into (or from) the crate without switching it off.

The Mod. V965A is the 8 channel version of the board, featuring all the other characteristics of the V965, except as otherwise indicated.

---

<sup>1</sup> Not calibrated



## 1.2. Block diagram

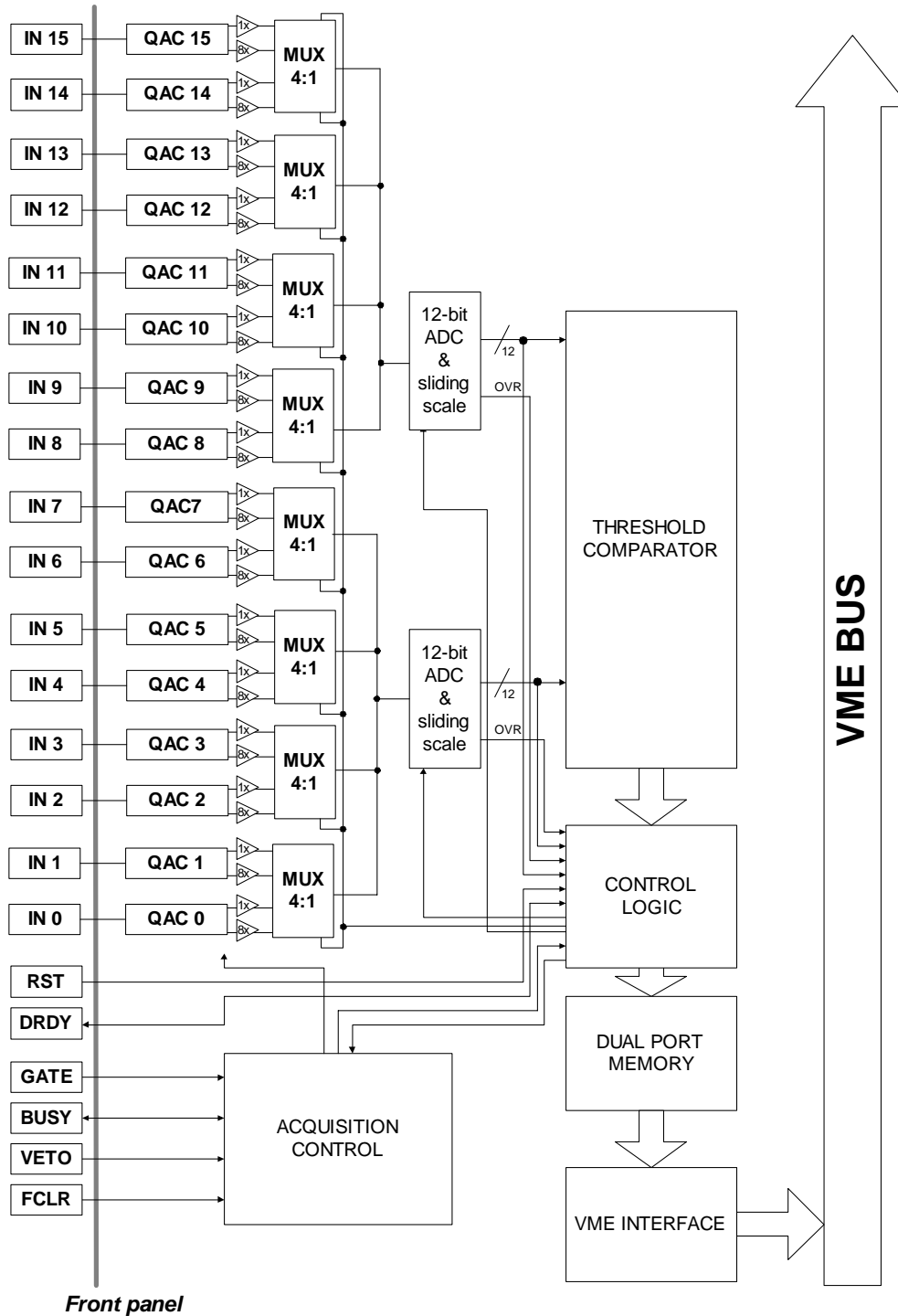


Fig. 1.1: Model V965 Block Diagram

## 2. Principles of operation

The board has 16 input channels (8 for Mod. V965A) and one GATE input common to all channels.

The integrated currents, received from the channel inputs when the GATE input signal is active, are converted into voltage levels by the QAC sections and then multiplexed and converted by two fast 12-bit ADC modules.

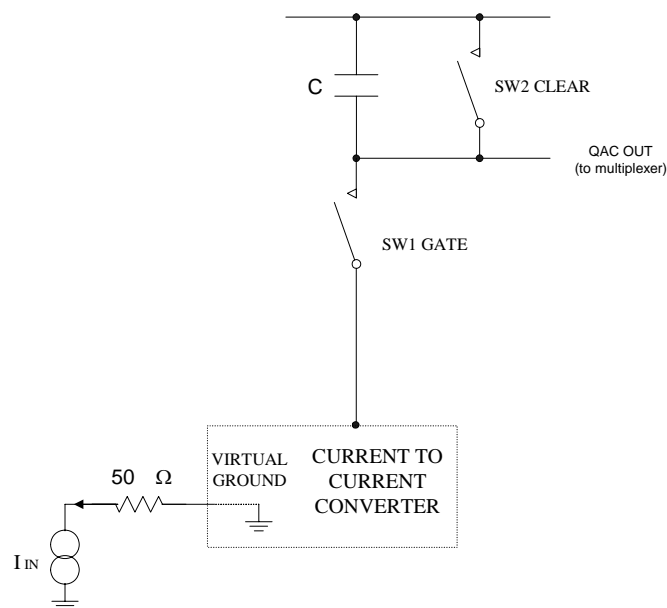
Each input signal is converted into two words, one featuring 25 fC LSB (100 pC full scale) and the other 200 fC LSB (900 pC full scale).

Only the values that are above a programmable threshold (see § 2.3), do not cause overflow (see § 2.4) and are not killed (see § 2.3) will be stored in a dual port data memory accessible via VME.

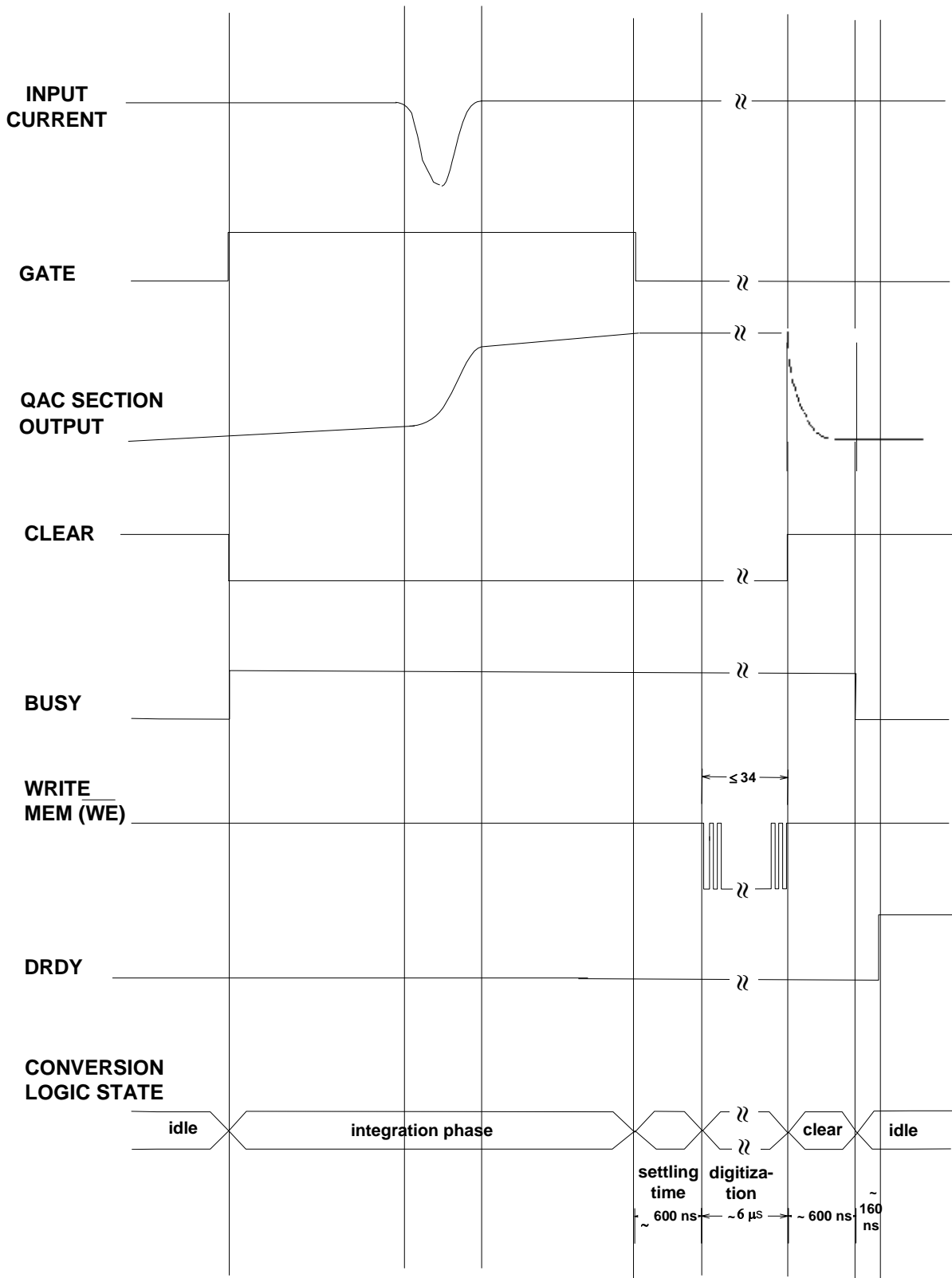
In the following functional sections and operation principles of the module are described in some detail. The block diagram of the module can be found in Fig. 1.1.

### 2.1. QAC sections

The module hosts 16 QAC sections: a simplified block diagram of a QAC section is reported in Fig. 2.1. The GATE signal close the switch SW1 thus allowing the input current to flow through an integrator. The integrator output is amplified and fed to the multiplexer. As the SW1 is open again, the signal is digitised by the 12-bit ADCs. After digitisation the SW2 switch is closed clearing the charge integrated into the capacitor C. The signal conversion timing is shown in Fig. 2.2.



**Fig. 2.1: Simplified block diagram of the QAC section**



**Fig. 2.2: Signal conversion timing**

### 2.1.1. The input current-to-current converter

The QAC sections are hosted on a piggy back board plugged into a motherboard; The input current to current converter (negative inputs) is shown in Fig. 2.3; it uses a differential amplifier and a bipolar transistor; a minimum bias current is required to flow through the transistor T1 in order to:

- Allow T1 to operate in the linear region even for small input signals
- Tolerate small positive input currents still ensuring linear response

This bias current is provided by the  $I_0$  ( $\approx 500 \mu\text{A}$ ) and  $I_1$  (VME programmable, see § 4.34) current generators (see figure below).

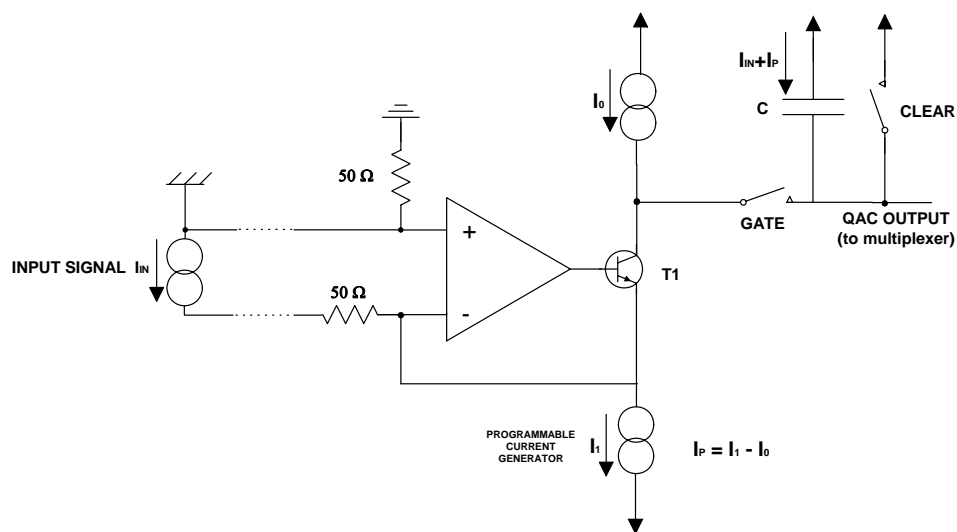


Fig. 2.3: QAC section

### 2.1.2. Pedestal

The QDC pedestal is the value readout when no input signal is present (open inputs). It is mainly due to the  $I_P$  current (see § 2.1.1) integrated by the capacitor C when a GATE signal is applied. It is worth noting that the QDC pedestal linearly depends on GATE width. When the maximum dynamic range is needed, it is possible to program  $I_1$  so that  $I_P$  is close to zero (see also § 4.34 for further details).

### 2.1.3. Maximum GATE width

The pedestal current value  $I_P$  sets an upper limit on the maximum GATE width. The QDC pedestal value is  $T_{\text{GATE}} \cdot I_P$ , where  $T_{\text{GATE}}$  is the GATE width; usually, in order not to spoil the QDC dynamic range, this pedestal value should be 5-10% of the FSR. For example, if  $I_P \sim 1 \mu\text{A}$  and at least 90% of the dynamic range is needed, the GATE width should not exceed  $\sim 40 \mu\text{s}$  (see also § 4.34 for further details).

## 2.2. Analog to digital conversion

The output of each QAC section is split (1x and 8x, in order to obtain a 15 bit input dynamics) and then multiplexed, by couples of channels, and subsequently converted by two fast 12-bit ADCs, each of which operates the conversion on a group of 8 channels (Block A and Block B ADCs): actually each input channel is converted two times, with 200 ps and 25 ps LSB respectively. The ADC section supports the sliding scale technique to reduce the differential non-linearity (see references [1], [2]). This technique (see Fig. 2.4) consists in adding a known value to the analog level to be converted, thus spanning different ADC conversion regions with the same analog value. The known level is then digitally subtracted after the conversion and the final value is sent to the threshold comparator.

If the sliding scale is enabled, it reduces slightly the dynamic range of the ADC: the 12-bit digital output is valid from 0 to 3840, while the values from 3841 to 4095 are not correct.

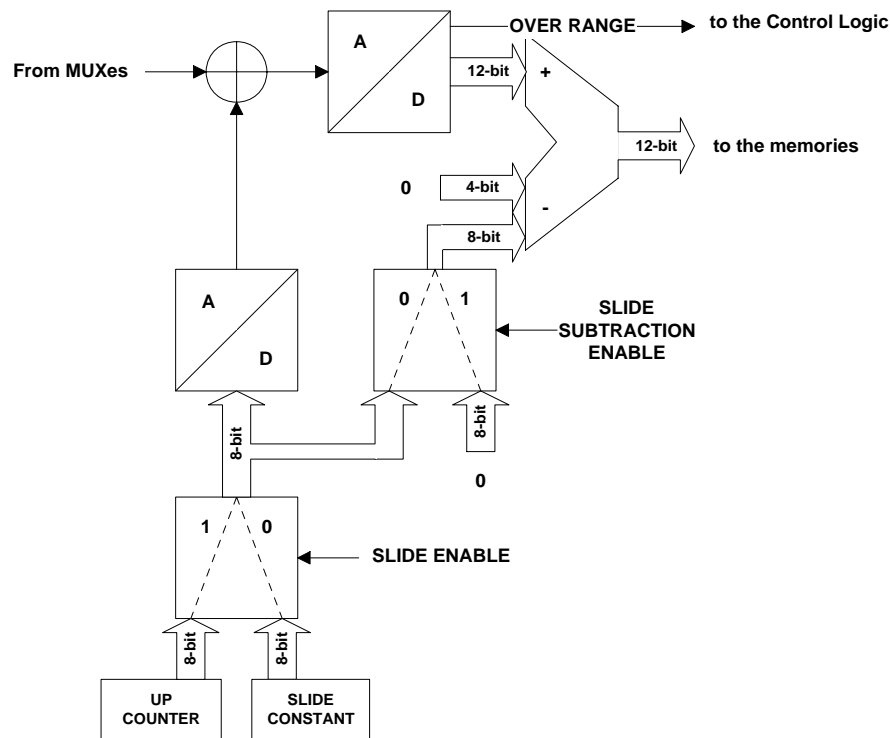


Fig. 2.4: Block diagram of the sliding scale section

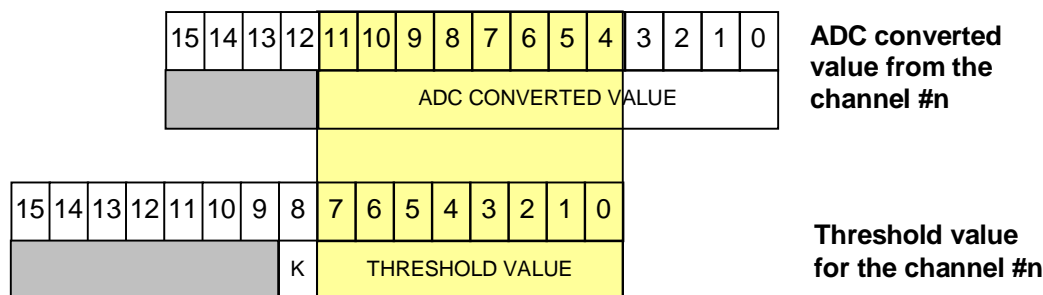
## 2.3. Zero suppression

The output of the ADC is fed to a digital threshold comparator to perform the zero suppression. If the converted value from a channel is greater than (or equal to) the relevant low threshold value set via VME in the Thresholds memory (Base Address + 0x1080 ÷ 0x10BF, see § 4.40), the result is fed to the dual port memory and will be available for the readout.

If the converted value is lower than the threshold, the value is stored in the memory only if the LOW TRESHOLD ENABLE bit of the Bit Set 2 Register is set to 1 (see § 4.26). The fact that the converted value was under the threshold is also flagged in the datum stored

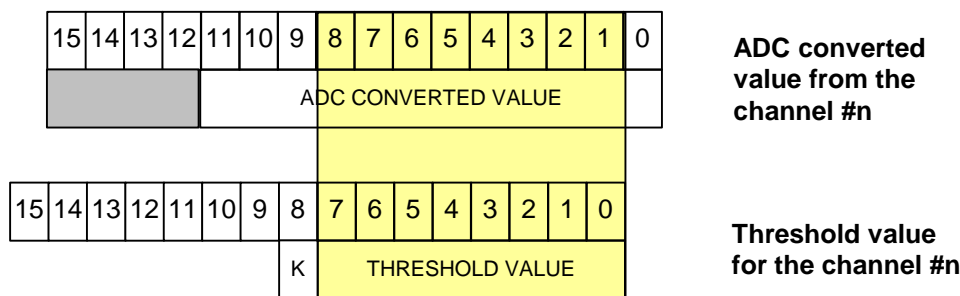
in the memory, where the bit 13 (UNDERTHRESHOLD) of the 16-bit data word is set to 1 (see § 4.5).

The Thresholds memory allows to set a low threshold value for each channel. Default setting corresponds to threshold value = 0. By setting the bit 8 in the Bit Set 2 Register it is possible to program the Threshold values in 16 ADC counts steps over the entire full scale range or in 2 ADC counts steps over 1/8 of full scale range. In more detail, if Bit 8 = 0 (default value) the comparison is performed between the 8 MSB of each 12 bit converted value and the 8 bit threshold value which is stored in the relevant register as illustrated in Fig. 2.5. The threshold values can be programmed over the entire full scale range.



**Fig. 2.5: Zero suppression (Bit 8 of Bit Set 2 Register = 0, default setting)**

if Bit 8 = 1 (in the Bit Set 2 Register) the comparison is performed between the bit 1...8 of each 12 bit converted value and the 8 bit threshold value which is stored in the relevant register as illustrated in the figure below (converted value is under threshold if the value written in the 1...8 bits is smaller than the threshold value and 9...11 bits are 0). The threshold values can be programmed over 1/8 of full scale range.



**Fig. 2.6: Zero suppression (Bit 8 of Bit Set 2 Register =1)**

The comparison is resumed in the following table:

| Bit 8 of Bit Set 2 Register | Comparison                                 |
|-----------------------------|--|
| 1                           | ADC CONVERTED VALUE < THRESHOLD VALUE x 2  |
| 0                           | ADC CONVERTED VALUE < THRESHOLD VALUE x 16 |

If the result of the comparison is true and the Bit 4 (LOW THRESHOLD ENABLE) of the Bit Set 2 Register is set to 0, data are skipped. If the Bit 4 of the Bit Set 2 Register is set

to 1, the true result of the comparison is signalled by Bit 13 (UNDERTHRESHOLD) = 1 in the loaded data 16 bit word.

The content of the Threshold Register includes also a KILL bit, which allows to abort the memorisation of the datum even if it is higher than the threshold set in the register. This bit can thus be used to disable some channels. Refer to § 4.40 for further details.

The threshold values are lost only after switching the board off (a reset operation does not affect the threshold values).

---

## 2.4. Overflow suppression

The overflow suppression allows to abort the memorisation of data which originated an ADC overflow. The control logic provides to check if the output of the ADC is in overflow and, in the case, the value is not stored in the memory.

The overflow suppression can be disabled by means of the OVER RANGE ENABLE bit of the Bit Set 2 Register (see § 4.26): if this bit is set to 1, all the data, independently from the fact that they caused ADC overflow or not, are stored in the memory. In this case, the 16-bit word stored in the memory will have the bit 12 (OVERFLOW) set to 1 (see § 4.5).

---

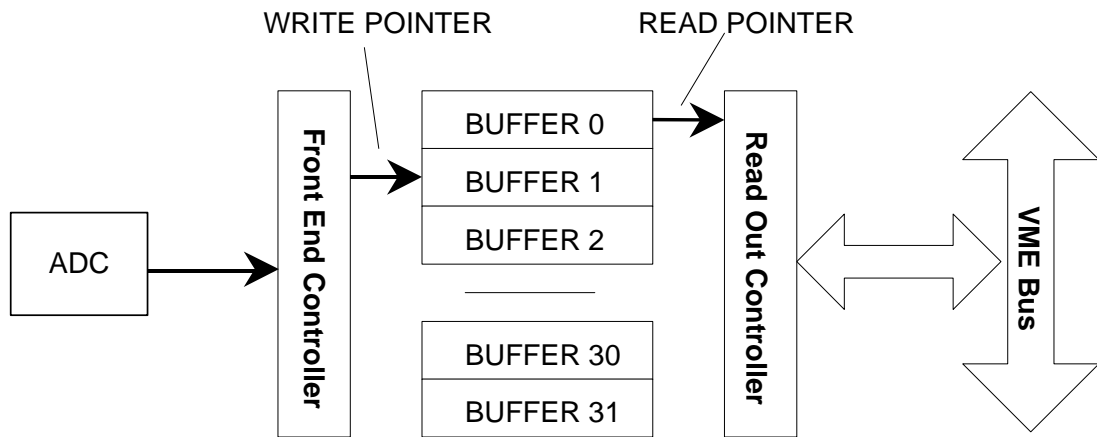
## 2.5. Multiple Event Buffer (MEB)

After the conversion, if there is at least one converted value above the programmed threshold, not causing overflow and not killed, the control logic stores it in the Multi-Event Buffer (MEB).

The Multi-Event Buffer is a Dual Port Memory (34 Words/event) which can store up to 32 events. It is mapped at the VME address: Base Address + 0x0000-0x07FC (see also § 4.5).

In order to trace the event flow, two pointers (Read and Write pointer) are employed. The Read Pointer points to the active read buffer. The Write pointer is incremented automatically via hardware at the end of the channels conversion, while the Read pointer can be either incremented automatically (AUTO INCR. bit of the Bit Set 2 Register set to 1; see § 4.26) or via write access to one of two dummy registers, Increment Event and Increment Offset Registers (see § 4.23 and 4.24). These allow to move the readout pointer to the next event in the output buffer or to the next word, respectively.

The MEB is mapped on a 2 Kbyte wide address space to facilitate BLT across 256 byte boundaries with some brands of CPUs but it is important to point out that any read access to any location within the MEB memory segment always returns the data word marked by the Read Pointer.



**Fig. 2.7: Multi-Event Buffer: Write pointer and Read pointer**

The MEB can be either in a "Full", a "Not empty" or an "Empty" status.

When the 5MSB of the Read pointer and the 5MSB of the Write pointer are different (i.e. point to different events), the MEB is in a "Not empty" status.

When the Read pointer and the Write pointer are equal, the MEB can be either in a "Full" or an "Empty" status. The MEB is full or empty according to the last increment pointer operation performed: if the last increment is the one of the Write pointer, the MEB is Full; if the last increment is the one of the Read pointer, the MEB is Empty.

The status of the MEB is monitored via two Registers, the Status Register 1 and the Status Register 2 (see § 4.13 and § 4.20, respectively).

After the conversion, the accepted data (i.e. the converted values above the programmed threshold, not causing overflow and not killed) are stored in the active event buffer (i.e. the one pointed by the write pointer) in subsequent 32-bit words. These are organised in events. Each event consists of a Header (see Fig. 4.5), a block of data words (Fig. 4.6) and an End-Of-Block (EOB) word (Fig. 4.7).

Each event contains thus from a minimum of 3 32-bit words (Header, one data word and EOB) to a maximum of 34 32-bit words (Header, 32 data words and EOB).

In case there are no accepted data, the User can choose to store anyway in the MEB the Header and the EOB relative to the event (see EMPTY ENABLE bit of the Bit Set 2 Register, see § 4.26): in this case the event is constituted by 2 32-bit words only.

---

## 2.6. Event Counter

The module houses a 24-bit counter that counts the number of GATE signals that the module has received.

The Event Counter can work in two different modes, which can be selected via the Bit 14 (ALL TRG) of the Bit Set 2 Register (see § 4.26):

- Mode A (ALL TRG = 1): it counts all events (default);
- Mode B (ALL TRG = 0): it counts only the accepted events.



In the first case (Mode A), the Event Counter is increased each time a pulse is sent through the GATE input.

In the second case (Mode B), the Event Counter is increased each time a pulse, sent through the GATE input, is accepted (i.e. VETO, FCLR and BUSY are not active).

The value of the Event Counter is stored in the EOB of the Multi-Event Buffer (see § 4.5). The Event Counter is also stored in two registers, the Event Counter\_Low and Event Counter\_High Registers, which respectively contain the 16LSBs and the 8MSBs of the Event Counter (see § 4.21 and § 4.22).

---

## 2.7. Busy Logic

The board is BUSY either during the conversion sequence or during the reset of the analog section or when the MEB is not ready to accept data (MEB Full) or when the board is in Random Memory Access Test mode (see § 5.7.1).

On the occurrence of one of these conditions the front panel BUSY signal is active, the red BUSY LED is on and the bit 2 (BUSY) and bit 3 (GLOBAL BUSY) of the Status Register 1 are set to 1 (see § 4.13). The BUSY LED lights up also while the board is configuring (power ON).

Actually, each module sets to 1 its BUSY output after the leading edge of a pulse on the GATE input and releases it to 0 at the end of the conversion sequence. When the module is busy, it does not accept another GATE pulse.

Several Mod. V965 boards can be connected over the parallel CONTROL Bus; the ECL wired OR and wired NAND of BUSY signals can be read respectively on the BUSY+ and BUSY- lines of the CONTROL Bus and the status of the BUSY+ bidirectional line is flagged by the bit 3 of the Status Register 1 (see § 4.13).

The jumper J12 placed on the PCB (see Fig. 3.2) allows to select board behaviour in response to a BUSY status: if this jumper is set to EXTBSY, the acquisition is stopped as soon as any of the boards connected over the CONTROL Bus is BUSY; if the jumper is set to INTBSY, acquisition is stopped as the board itself is BUSY.

---

## 2.8. Reset Logic

Three different types of RESET operations can be distinguished, according to the effects they have on the module and particularly on the registers. These are:

- *Type A:* Data RESET
- *Type B:* Software RESET
- *Type C:* Hardware RESET

The **Data RESET** clears the data in the output buffer, resets the read and write pointers, the event counter and the QAC sections. It does not affect the registers.

This type of RESET can be forwarded in two ways:

1. setting the Bit 2 (CLEAR DATA) of the Bit Set 2 Register to 1 (see § 4.26); the Reset is released via the Bit Clear 2 Register (see § 4.27);

2. sending a RESET pulse from the front panel with the Bit 4 (PROG RESET) of the Control Register 1 set to 0 (see § 4.14).

The **Software RESET** performs the same actions as the data RESET and, moreover, it resets the registers marked in the column SR (Software Reset) in Table 4.2.

This type of RESET can be forwarded in three ways:

1. setting the Bit 7 (SOFTWARE RESET) of the Bit Set 1 Register to 1 (see § 4.9): this sets the module to a permanent RESET status which is released only via write access, with the relevant bit set to 1, to the Bit Clear Register;
2. sending a RESET pulse from the front panel with the Bit 4 (PROG RESET) of the Control Register 1 set to 1 (see § 4.14);
3. performing a write access to the Single Shot Reset Register (see § 4.17): the RESET lasts as long as the write access itself.

The **Hardware RESET** performs the same actions as the Software RESET and, moreover, it resets further registers. All the registers reset by a Hardware RESET are marked in the column HR (Hardware Reset) in Table 4.2.

This type of RESET is performed:

1. at Power ON of the module;
2. via a VME RESET (SYS\_RES).

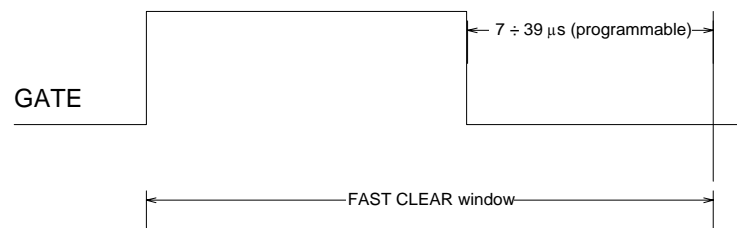
At power ON or after a reset the module must thus be initialised.

---

## 2.9. FAST CLEAR

The FAST CLEAR of the module can be performed via the relevant front panel signal (see § 3.4.2). A FAST CLEAR signal, generated at any time within the FAST CLEAR window, i.e. between the leading edge of the GATE signal and the end of the programmable time value set in the Fast Clear Window Register (see § 4.25), aborts the conversion. Its minimum width must be 30 ns.

N.B.: since a FAST CLEAR operation implies a CLEAR CONVERSION cycle, a new GATE signal is accepted only if it occurs at least 600 ns after the leading-edge of the FAST CLEAR signal.



**Fig. 2.8: Fast Clear window**

---

## 3. Technical specifications

---

### 3.1. Packaging

The Model V965 is housed in a 6U-high, 1U-wide VME unit. The board hosts the VME P1 and P2 connectors.

---

### 3.2. Power requirements

The power requirements of the V965 and V965A modules are as follows:

**Table 3.1: Mod. V965 and V965A power requirements**

| Power supply | V965   | V965A  |
|--------------|--------|--------|
| +12 V        | 850 mA | 500 mA |
| -12 V        | 550 mA | 300 mA |
| +5 V         | 2.1 A  | 2.0 A  |

### 3.3. Front Panel

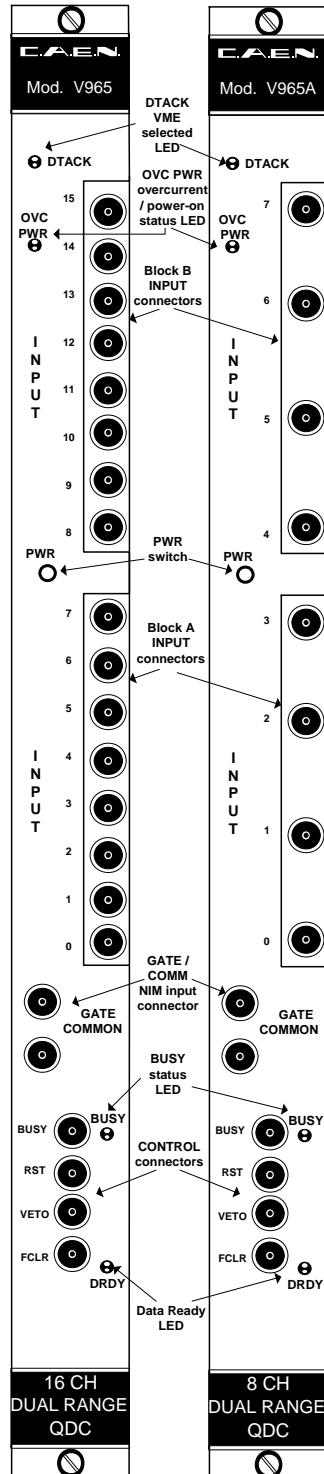


Fig. 3.1: Model V965/V965A front panels

---

## 3.4. External connectors

The location of the connectors is shown in Fig. 2.1. Their function and electro-mechanical specifications are listed in the following subsections.

---

### 3.4.1. INPUT connectors

*Mechanical specifications:*

16 LEMO 00 connectors (8 for Mod. V965A).

*Electrical specifications:*

negative input signals, DC coupled, 50  $\Omega$  impedance.

**BLOCK A INPUT:** input signals from ch. 0 through ch. 7 (0÷3 for V965A)

**BLOCK B INPUT:** input signals from ch. 8 through ch. 15 (4÷7 for V965A)



**WARNING: Max. tolerated positive voltage input is 15 mV; larger values will damage the input stages**

---

### 3.4.2. CONTROL connectors

*Mechanical specifications:*

four LEMO 00 type connectors.

**FCLR:** *Electrical specifications:* NIM std. input signal, 50  $\Omega$  impedance; min. FWHM: 30 ns.

*Function:* FAST CLEAR signal, accepted if sent within the so-called FAST CLEAR window (see Fig. 2.10); it clears the QAC sections of the unit and aborts completely the conversion in progress.

**RST:** *Electrical specifications:* NIM std. input signal, 50  $\Omega$  impedance; min. FWHM: 30 ns.

*Function:* clears the QAC sections, resets the Multi-Event Buffer status, stops pending ADCs conversions and, depending on the User's settings, may clear the control registers.

**VETO:** *Electrical specifications:* NIM std. input signal, 50  $\Omega$  impedance.

*Function:* inhibits the conversion of the QAC signals.

**BUSY:** *Electrical specifications:* NIM output signal.

*Function:* indicates that the board is either converting or resetting or in MEMORY TEST mode or the MEB is full, BUSY status is also flagged by the bit 2 of the Status Register 1.

---

### 3.4.3. GATE /COMM connectors

*Mechanical specifications:*

two bridged 00-type LEMO connectors.

*Electrical specifications:*

NIM std. input signals; high impedance; if this input is used a 50  $\Omega$  termination is required; in daisy-chain configuration, the termination must be inserted on the last board of the chain.

**GATE/COMM:** *Function:* input signal, common to all channels, acting as the temporal window within which the input current is integrated. In the Mod. V965 this signal is internally OR-wired with the GATE of the CONTROL connector

---

## 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:* DATA ACKNOWLEDGE command; it lights up each time a VME access is performed.

**BUSY:** *Colour:* red.  
*Function:* it lights up each time the module is performing a conversion or resetting the analog section or in memory TEST mode or when the Multi-Event Buffer is full; 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.

**OVC/PWR:** *Colour:* green/orange.  
*Function:* it lights up green when the board is inserted into the crate and the crate is powered up; when it is orange, 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.

---

### 3.5.2. Switches

**PWR:** *Type:* miniature flush plunger push-button switch.  
*Function:* after the insertion of the board into the crate, it allows to turn the board ON/OFF by pushing it with a pin. Refer to § 5.4 for the power ON procedure

---

## 3.6. Internal hardware components

The V965 module is constituted by a motherboard with a piggy-back board plugged into it (see also Fig. 1.1 where the functional blocks hosted on the piggy-back board are pointed out). In the following some hardware setting components, located on the boards, are listed. Refer to Fig. 3.2 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. Please refer to Fig. 3.2 for their settings.

---

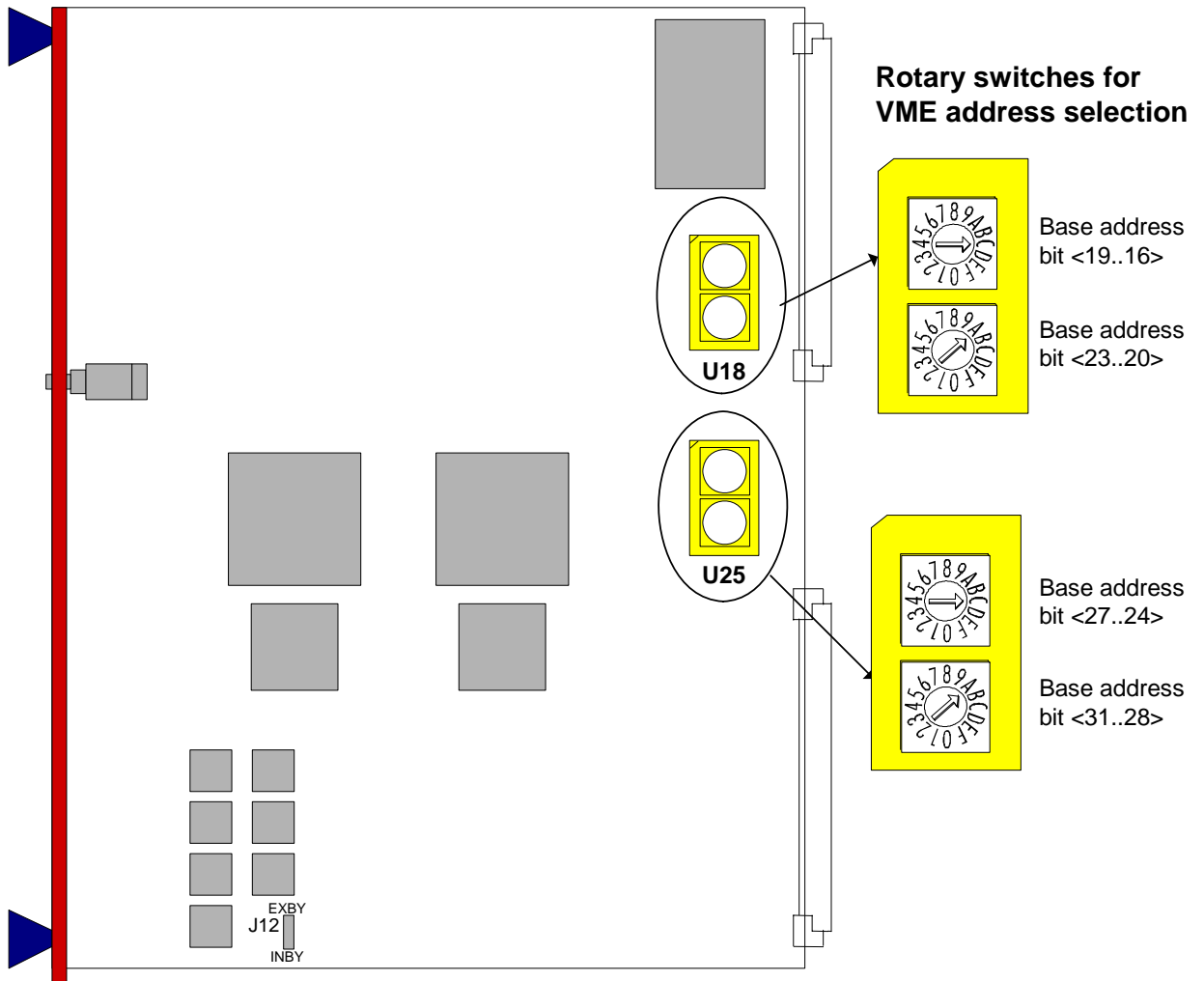
### 3.6.2. Jumpers

**J12:**      *Function:* for Mod. V965 only, it allows to select board behaviour in response to a BUSY status:

**Position A** (high): data acquisition is stopped as soon as any of the boards connected over the CONTROL Bus is BUSY;

**Position B** (low): data acquisition is stopped as the board is BUSY, independently from the status of the other boards on the CONTROL Bus.

Refer to Fig. 3.2 for the exact location of the jumper on the PCB and its setting.



**Fig. 3.2: Components location (components side)**



### 3.7. Technical specifications table

Table 3.2 : Model V965 main technical specifications

|  |   |
|--|---|
| <b>Inputs</b>                                | 16 channels (8 for V965A), 50 $\Omega$ impedance, negative polarity, DC coupling  |
| <b>Input range<sup>2</sup></b>               | Dual range: 0 $\div$ 900 pC / 0 $\div$ 100 pC   |
| <b>Resolution</b>                            | 12 bit <sup>3</sup>   |
| <b>Gain</b>                                  | High range: 200 fC/count; Low range: 25 fC/count  |
| <b>Max. tolerated positive input voltage</b> | +15 mV, larger values will damage the input stages  |
| <b>Reflections</b>                           | <5% with 2 ns fall time input signals   |
| <b>Input offset</b>                          | $\pm$ 2 mV  |
| <b>RMS Noise (typical)</b>                   | 0.7 counts (high range), 1.5 counts (low range)   |
| <b>Interchannel gain uniformity</b>          | $\pm$ 4%  |
| <b>Interchannel Isolation</b>                | > 60 dB   |
| <b>Fast clear time</b>                       | 600 ns  |
| <b>Gate timing</b>                           | The GATE signal must precede the analog input by >15 ns   |
| <b>Conversion time</b>                       | 5.7 $\mu$ s / 16 Channel (2.8 $\mu$ s / 8 Channel for V965A)  |
| <b>Dead time</b>                             | 6.9 $\mu$ s / 16 Channel (3.5 $\mu$ s / 8 Channel for V965A)  |
| <b>Zero suppression</b>                      | Thresholds values programmable in:<br>16 ADC counts step over the entire FSR<br>2 ADC counts steps over 1/8 of FSR  |
| <b>Integral non linearity</b>                | $\pm$ 0.1% of Full Scale Range <sup>4</sup> (T.B.C.)  |
| <b>GATE/COM input</b>                        | Temporal window for current integration   |
| <b>Control inputs</b>                        | <b>GATE:</b> temporal window for peak detection (ECL/NIM)<br><b>RST:</b> resets PEAK sections, MEB status and control registers<br><b>VETO:</b> inhibits the conversion of the peaks<br><b>FCLR:</b> FAST CLEAR of PEAK sections and conversion |
| <b>Control outputs</b>                       | <b>BUSY:</b> indicates the presence of data   |
| <b>MultiEvent Buffer</b>                     | 32 events   |
| <b>Packaging</b>                             | 6U-high, 1U-wide VME unit   |
| <b>Addressing</b>                            | A24 / A32 / Geographical / MultiCast  |
| <b>Register Access</b>                       | D16 / D32 - Read / Write  |
| <b>Data Readout modes</b>                    | D32 / BLT32 / MBLT64 / CBLT32 / CBLT64  |
| <b>Data throughput</b>                       | Max: 40 Mbyte/s   |
| <b>Interrupt</b>                             | 7 level RORA Interrupter  |
| <b>Live Insertion</b>                        | Yes   |
| <b>JAUX</b>                                  | Not used  |
| <b>Power Supplies</b>                        | +5V, +12V, -12V   |
| <b>Compliance</b>                            | VME / VME32 / VME64 / VME64X / VME64XP  |

<sup>2</sup> If Sliding Scale is used FSR is reduced from 4095 to 3840 counts

<sup>3</sup> This value leads to a 15 bit dynamics

<sup>4</sup> from 5% to 95% of FSR=3840 counts

---

## 4. VME interface

---

### 4.1. Addressing capability

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

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

---

#### 4.1.1. 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\_HIGH and ADER\_LOW registers.

The 4 rotary switches for Base Address selection are housed on two piggy-back boards plugged into the main printed circuit board (see Fig. 3.2).

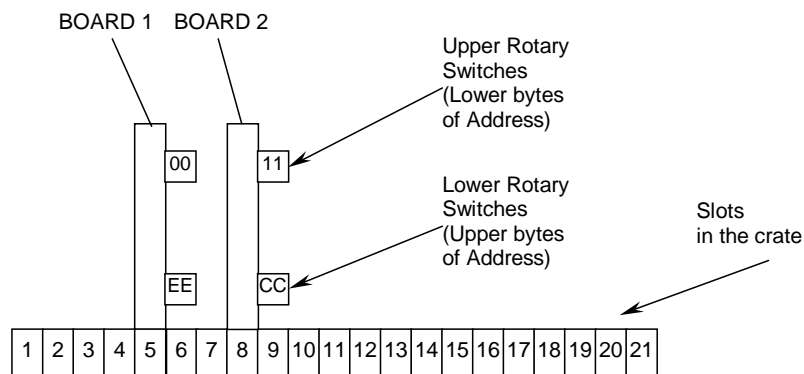
To use this addressing mode the bit 4 of the Bit Set 1 Register (see § 4.9) must be set to 0. This is also the default setting.

The module Base Address can be also fixed by using the Ader\_High and Ader\_Low Registers. These two registers set respectively the A[31:24] and the A[23:16] VME address bits (see § 4.15 and 4.16).

To use this addressing mode bit 4 of the Bit Set 1 Register (see § 4.9) must be set to 1.

### 4.1.2. Base addressing examples

The following is an example of Base Addressing for two V965 boards inserted in a VME crate.



**Fig. 4.3: Base 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

**Board 2:**

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

### 4.1.3. MCST/CBLT addressing

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

- 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) must be written in the MCST/CBLT Address Register (see § 4.8) 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 and MCST operations, the IACKIN/IACKOUT daisy chain is used to pass a token from a 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 § 4.18). 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 § 4.5). It is up to the User to write the GEO address in the GEO register 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.**

#### 4.1.4. MCST/CBLT addressing examples

The following is an example of MCST and CBLT addressing for four V965 boards plugged into a VME crate. To access the boards the steps to be performed are as follows:

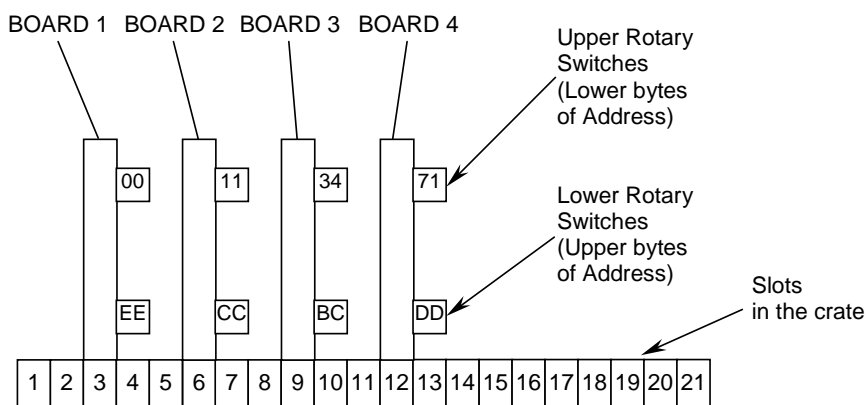
1. Set the MCST address (see § 4.8) for all boards via VME Base Address or geographical addressing (if available);
2. Set the bits F\_B and L\_B of the MCST Control Register (see § 4.18) according to the operational status (active or inactive) of each board and to its position in the chain (first, intermediate 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 (A24 or 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 (0xEE001004, 0xAA, A32, D16) /* set MCST Address=0xAA for board 1 */
vme_write (0xCC111004, 0xAA, A32, D16) /* set MCST Address=0xAA for board 2 */
vme_write (0xBC341004, 0xAA, A32, D16) /* set MCST Address=0xAA for board 3 */
vme_write (0xDD711004, 0xAA, A32, D16) /* set MCST Address=0xAA for board 4 */
vme_write (0xEE00101A, 0x02, A32, D16) /* set board 1 = First */
vme_write (0xCC11101A, 0x03, A32, D16) /* set board 2 = Active */
vme_write (0xBC34101A, 0x00, A32, D16) /* set board 3 = Inactive */
vme_write (0xDD71101A, 0x01, A32, D16) /* set board 4 = Last */
vme_write (0xAA001006, 0x80, 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.

---

## 4.2. Interrupter capability

The Mod. V965 houses a RORA-type VME INTERRUPTER. The INTERRUPTER responds to 8 bit, 16 bit and 32 bit Interrupt Acknowledge cycles by providing an 8-bit STATUS/ID on the VME data lines D00..D07.

---

### 4.2.1. Interrupt Status/ID

The interrupt STATUS/ID is 8-bit wide, and it is contained in the 8LSB of the Interrupt Vector Register (see § 4.12). The register is available at the VME address: Base Address + 0x100C.

---

### 4.2.2. Interrupt Level

The interrupt level corresponds to the value stored in the 3LSB of the Interrupt Level Register (see § 4.11). 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.

---

### 4.2.3. Interrupt Generation

An Interrupt is generated when the number of events stored in the memory equals the value written in the Event Trigger Register at the VME address: Base Address + 0x1020 (see § 4.19). If the value in Event Trigger Register is set to 0 the interrupt is disabled (default setting).

---

### 4.2.4. 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 Event Trigger Register.

---

## 4.3. Data transfer capability

The internal registers are accessible in D16 mode, unless otherwise specified. Access in D32, BLT32, MBLT64, CBLT32 and CBLT64 is available for the data buffer.

---

## 4.4. Register address map

The Address map for the Model V965 is listed in Table 4.2. 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.

The Table gives also information about the effects of RESET on the registers. In particular, column 2 through 4 refer to the following RESET operations:

- **D R → Data RESET;**

- **S R → Software RESET;**
- **H R → Hardware RESET.**

If a register has a mark in these columns, it means that the relevant RESET operation resets that register. For further details on the RESET Logic please refer to § 2.8.

Table 4.3 and Table 4.4 list register addresses (offset) in CBLT and MCST operations, respectively.

The ROM address map is reported in Table 4.5.

**Table 4.2: Address Map for the Model V965**

| Register content       | DR    | SR    | HR    | Address         | Type       | Access mode |
|------------------------|-------|-------|-------|-----------------|------------|-------------|
| Output Buffer          | ✓     | ✓     | ✓     | 0x0000 ÷ 0x07FC | Read only  | D32/D64     |
| Firmware Revision      |       |       |       | 0x1000          | Read only  | D16         |
| Geo Address            |       |       |       | 0x1002          | Read/Write | D16         |
| MCST/CBLT Address      |       |       | ✓     | 0x1004          | Read/Write | D16         |
| Bit Set 1              |       | ✓ (*) | ✓ (*) | 0x1006          | Read/Write | D16         |
| Bit Clear 1            |       | ✓ (*) | ✓ (*) | 0x1008          | Read/Write | D16         |
| Interrupt Level        |       | ✓     | ✓     | 0x100A          | Read/Write | D16         |
| Interrupt Vector       |       | ✓     | ✓     | 0x100C          | Read/Write | D16         |
| Status Register 1      |       | ✓     | ✓     | 0x100E          | Read only  | D16         |
| Control Register 1     |       | ✓ (*) | ✓ (*) | 0x1010          | Read/Write | D16         |
| ADER High              |       |       | ✓     | 0x1012          | Read/Write | D16         |
| ADER Low               |       |       | ✓     | 0x1014          | Read/Write | D16         |
| Single Shot Reset      |       |       |       | 0x1016          | Write only | D16         |
| MCST/CBLT Ctrl         |       |       | ✓     | 0x101A          | Read/Write | D16         |
| Event Trigger Register |       | ✓     | ✓     | 0x1020          | Read/Write | D16         |
| Status Register 2      |       | ✓     | ✓     | 0x1022          | Read only  | D16         |
| Event Counter_L        | ✓ (*) | ✓ (*) | ✓ (*) | 0x1024          | Read only  | D16         |
| Event Counter_H        | ✓ (*) | ✓ (*) | ✓ (*) | 0x1026          | Read only  | D16         |
| Increment Event        |       |       |       | 0x1028          | Write only | D16         |
| Increment Offset       |       |       |       | 0x102A          | Write only | D16         |
| Load Test Register     |       |       |       | 0x102C          | Read/Write | D16         |
| FCLR Window            |       | ✓     | ✓     | 0x102E          | Read/Write | D16         |
| Bit Set 2              |       | ✓     | ✓     | 0x1032          | Read/Write | D16         |
| Bit Clear 2            |       | ✓     | ✓     | 0x1034          | Write only | D16         |
| W Memory Test Address  |       | ✓     | ✓     | 0x1036          | Write only | D16         |
| Memory Test Word_High  |       | ✓     | ✓     | 0x1038          | Write only | D16         |
| Memory Test Word_Low   |       |       |       | 0x103A          | Write only | D16         |
| Crate Select           |       | ✓     | ✓     | 0x103C          | Read/Write | D16         |
| Test Event Write       |       |       |       | 0x103E          | Write only | D16         |
| Event Counter Reset    |       |       |       | 0x1040          | Write only | D16         |
| Iped                   |       | ✓     | ✓     | 0x1060          | Read/Write | D16         |
| R Test Address         |       | ✓     | ✓     | 0x1064          | Write only | D16         |
| SW Comm                |       |       |       | 0x1068          | Write only | D16         |
| AAD                    |       |       |       | 0x1070          | Read only  | D16         |
| BAD                    |       |       |       | 0x1072          | Read only  | D16         |
| Thresholds             |       |       |       | 0x1080 ÷ 0x10BF | Read/Write | D16         |

(\*) not all bits are reset with the same type of RESET: see the description of the relevant register for details.  
The ROM address map is from 0x8000 to 0xFFFF: refer to § 4.41.



**Table 4.3: Address Map in CBLT operation**

| Register content | Address       | Type      | Access mode |
|------------------|---------------|-----------|-------------|
| Output Buffer    | 0x0000÷0x07FF | Read only | D32/D64     |

**Table 4.4: Address Map in MCST operations**

| Register content       | Address         | Type       | Access mode |
|------------------------|-----------------|------------|-------------|
| Bit Set 1              | 0x1006          | Write only | D16         |
| Bit Clear 1            | 0x1008          | Write only | D16         |
| Interrupt Level        | 0x100A          | Write only | D16         |
| Interrupt Vector       | 0x100C          | Write only | D16         |
| Control Register 1     | 0x1010          | Write only | D16         |
| ADER High              | 0x1012          | Write only | D16         |
| ADER Low               | 0x1014          | Write only | D16         |
| Single Shot Reset      | 0x1016          | Write only | D16         |
| Event Trigger Register | 0x1020          | Write only | D16         |
| Increment Event        | 0x1028          | Write only | D16         |
| Increment Offset       | 0x102A          | Write only | D16         |
| Load Test Register     | 0x102C          | Write only | D16         |
| Fast Clear Window      | 0x102E          | Write only | D16         |
| Bit Set 2              | 0x1032          | Write only | D16         |
| Bit Clear 2            | 0x1034          | Write only | D16         |
| W Memory Test Address  | 0x1036          | Write only | D16         |
| Memory Test Word_High  | 0x1038          | Write only | D16         |
| Memory Test Word_Low   | 0x103A          | Write only | D16         |
| Crate Select           | 0x103C          | Write only | D16         |
| Event Counter Reset    | 0x1040          | Write only | D16         |
| Iped                   | 0x1060          | Write only | D16         |
| R Memory Test Address  | 0x1064          | Write only | D16         |
| SW comm                | 0x1068          | Write only | D16         |
| Slide Constant         | 0x106A          | Write only | D16         |
| Thresholds             | 0x1080 ÷ 0x10BF | Write only | D16         |

## 4.5. Output Buffer Register

(Base Address + 0x0000 ÷ 0x07FC, read only)

This register allows the User to access the Multiple Event Buffer to readout the converted values.

The output buffer contains the output data organised in 32-bit words.

The data in the buffer are organised in events.

Each event consists of:

- the **header**, that contains the geographical address, the crate number and the number of converted channels;
- one or more **data words**, each of which contains the geographical address, the number of the channel, the Under-Threshold (UN) bit, the Overflow (OV) bit and the 12-bit converted value;
- the **End Of Block** (EOB), which contains the geographical address and the event counter.

|          |    |    |    |    |    |    |            |    |    |    |    |    |    |          |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|----------|----|----|----|----|----|----|------------|----|----|----|----|----|----|----------|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 31       | 30 | 29 | 28 | 27 | 26 | 25 | 24         | 23 | 22 | 21 | 20 | 19 | 18 | 17       | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| GEO[4:0] |    |    |    | 0  | 1  | 0  | CRATE[7:0] |    |    |    |    | 0  | 0  | CNT[5:0] |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

Fig. 4.5: Output buffer: the Header

V965A:

|          |    |    |    |    |    |    |    |    |               |    |    |    |    |    |    |           |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|----------|----|----|----|----|----|----|----|----|---------------|----|----|----|----|----|----|-----------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 31       | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22            | 21 | 20 | 19 | 18 | 17 | 16 | 15        | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| GEO[4:0] |    |    |    | 0  | 0  | 0  |    |    | CHANNEL [2:0] |    | RG |    |    | UN | OV | ADC[11:0] |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

V965:

|          |    |    |    |    |    |    |    |    |               |    |    |    |    |    |    |           |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|----------|----|----|----|----|----|----|----|----|---------------|----|----|----|----|----|----|-----------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 31       | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22            | 21 | 20 | 19 | 18 | 17 | 16 | 15        | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| GEO[4:0] |    |    |    | 0  | 0  | 0  |    |    | CHANNEL [3:0] |    | RG |    |    | UN | OV | ADC[11:0] |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

Fig. 4.6: Output buffer: the Data Word format

|          |    |    |    |    |    |    |                     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|----------|----|----|----|----|----|----|---------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 31       | 30 | 29 | 28 | 27 | 26 | 25 | 24                  | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| GEO[4:0] |    |    |    | 1  | 0  | 0  | EVENT COUNTER[23:0] |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

Fig. 4.7: Output buffer: the End Of Block

### Header content:

The bits[31...27] contains the GEO address.

The bits[26..24] identify the type of word (010 →header);

The bits[23..16] identify the crate number according to the content of the Crate Select Register (see § 4.31).

The bits[13...8] contain the number of memorised channels.

**Datum content:**

The bits[31...27] contains the GEO address.

The bits[26..24] identify the type of word (000 → datum);

The bits[20..17] identify the channel which the data are coming from; the sequence is explained in § 4.5.1.

The bit [16] identifies the input RANGE of the channel (0 = High; 1 = Low)

The bit[13] is the UNDERTHRESHOLD bit:

= 0 → the datum is over the threshold fixed in the relevant register (see § 4.36);

= 1 → the datum is under the threshold fixed in the relevant register; it is actually possible to make the datum be written in the buffer even if it is under the threshold by using the bits 3 and 4 of the Bit Set 2 Register (see § 4.26);

The bit[12] is the OVERFLOW bit:

= 0 → ADC not in overflow condition;

= 1 → ADC in overflow;

The bits[11...0] contain the converted datum.

**EOB content:**

The bits[31...27] contains the GEO address.

The bits[26..24] identify the type of word (100 → EOB);

The bits[23..0] contain the 24-bit event counter value (see § 4.21).

The bits[31...27] always contains the GEO address (except for the not valid datum, see Fig. 4.8).

The bits[26..24] identify the type of word, according to the following:

- 010 → header;
- 000 → valid datum;
- 100 → end of block;
- 110 → not valid datum.
- *others* → reserved.

If a read access is performed to the buffer when it is empty, the readout will provide a NOT VALID DATUM arranged as shown in Fig. 4.8.

|    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|    |    |    |    |    | 1  | 1  | 0  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

**Fig. 4.8: Output buffer: not valid datum**

---

### 4.5.1. Data storage

The sequence followed to store the data in the buffer is as follows:

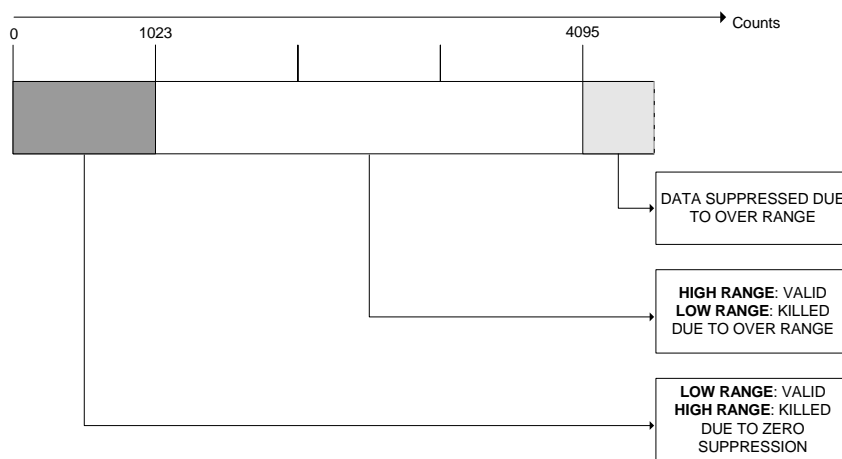
HEADER  
CHANNEL 0 High Range  
CHANNEL 8 High Range  
CHANNEL 0 Low Range  
CHANNEL 8 Low Range  
CHANNEL 1 High Range  
.....  
CHANNEL 7 Low Range  
CHANNEL 15 Low Range  
END OF BLOCK

Please note that some of the above channel data may be missing in the sequence: this is due either to overflow or under threshold conditions (which caused these data not to be stored), or to User's settings to kill some channels.

---

### 4.5.2. Data readout

A smart way to read out only the strictly necessary number of data is to set the Zero Suppression Threshold at  $\sim FSR/8^5$  ( $\sim 500$  counts) on High Range Channels, which will be thus acquiring only pulses larger than 100 pC. Low Range Channels will be automatically suppressed by the Over Range function in the 100 pC ÷ 900 pC interval. Pulses larger than 900 pC will be rejected in any case (on both ranges).



**Fig. 4.9: Data readout and suppression**

---

<sup>5</sup> It is suggested to set this threshold a little smaller than FSR/8, in order to avoid losing valid data

|                                       |     |    |    |    |    |    |                   |    |    |    |    |    |                     |    |                   |    |    |    |            |    |    |    |   |   |   |   |   |   |   |   |   |   |
|---------------------------------------|-----|----|----|----|----|----|-------------------|----|----|----|----|----|---------------------|----|-------------------|----|----|----|------------|----|----|----|---|---|---|---|---|---|---|---|---|---|
|                                       | 31  | 30 | 29 | 28 | 27 | 26 | 25                | 24 | 23 | 22 | 21 | 20 | 19                  | 18 | 17                | 16 | 15 | 14 | 13         | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Write<br>Pointer<br>N<br><br>GATE 5   | GEO |    |    | 0  | 1  | 0  | CRATE NUMBER      |    |    |    |    |    | 0                   | 0  | MEM. CHANNELS (2) |    |    |    |            |    |    |    |   |   |   |   |   |   |   |   |   |   |
|                                       | GEO |    |    | 0  | 0  | 0  |                   |    |    |    |    |    | CHANNEL<br>(2 h.r.) |    |                   |    | UN | OV | ADC COUNTS |    |    |    |   |   |   |   |   |   |   |   |   |   |
|                                       | GEO |    |    | 0  | 0  | 0  |                   |    |    |    |    |    | CHANNEL<br>(2 l.r.) |    |                   |    | UN | OV | ADC COUNTS |    |    |    |   |   |   |   |   |   |   |   |   |   |
|                                       | GEO |    |    | 1  | 0  | 0  | EVENT COUNTER (m) |    |    |    |    |    |                     |    |                   |    |    |    |            |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Write<br>Pointer<br>N+1<br><br>GATE 7 | GEO |    |    | 0  | 1  | 0  | CRATE NUMBER      |    |    |    |    |    | 0                   | 0  | MEM. CHANNELS (2) |    |    |    |            |    |    |    |   |   |   |   |   |   |   |   |   |   |
|                                       | GEO |    |    | 0  | 0  | 0  |                   |    |    |    |    |    | CHANNEL<br>(0 h.r.) |    |                   |    | UN | OV | ADC COUNTS |    |    |    |   |   |   |   |   |   |   |   |   |   |
|                                       | GEO |    |    | 0  | 0  | 0  |                   |    |    |    |    |    | CHANNEL<br>(0 l.r.) |    |                   |    | UN | OV | ADC COUNTS |    |    |    |   |   |   |   |   |   |   |   |   |   |
|                                       | GEO |    |    | 1  | 0  | 0  |                   |    |    |    |    |    |                     |    |                   |    |    |    |            |    |    |    |   |   |   |   |   |   |   |   |   |   |

**Fig. 4.10: Multi-Event Buffer: data structure example**

**N.B.:** the GEO address must be written by the User via a write access to the relevant register (see § 4.6), since the module does not use the PAUX connector. If this operation is not performed, it will be not possible to identify which module the data are coming from when the CBLT access is used.

## 4.6. Firmware Revision Register

(Base Address + 0x1000, read only)

This register contains a 16-bit value identifying the firmware revision. The 16-bit value corresponds to 4 hexadecimal figures which give the firmware revision number. For example, in the figure is shown the register content for the firmware release:

**Rev. 06.02**

|     |    |    |    |     |    |   |   |     |   |   |   |     |   |   |   |
|-----|----|----|----|-----|----|---|---|-----|---|---|---|-----|---|---|---|
| 15  | 14 | 13 | 12 | 11  | 10 | 9 | 8 | 7   | 6 | 5 | 4 | 3   | 2 | 1 | 0 |
| 0   | 0  | 0  | 0  | 0   | 1  | 1 | 0 | 0   | 0 | 0 | 0 | 0   | 0 | 1 | 0 |
| 0x0 |    |    |    | 0x6 |    |   |   | 0x0 |   |   |   | 0x2 |   |   |   |

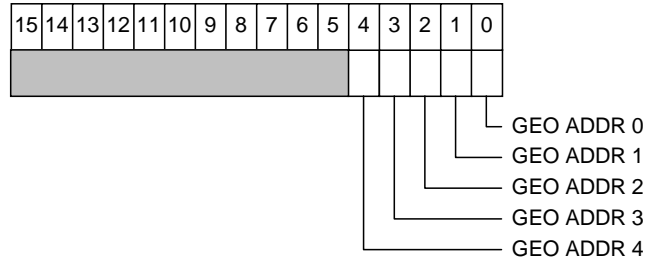
Binary representation  
 Hexadecimal representation

**Fig. 4.11: Firmware Revision Register**

## 4.7. GEO Address Register

(Base Address + 0x1002, read/write)

The register content is the following:



**Fig. 4.12: 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. The bits of the GEO Address register are set to 1 by default. It is up to the User to write the correct GEO address of the module in this register before CBLT operation so that the GEO address will be contained in the HEADER and the END OF BLOCK words for data identification.

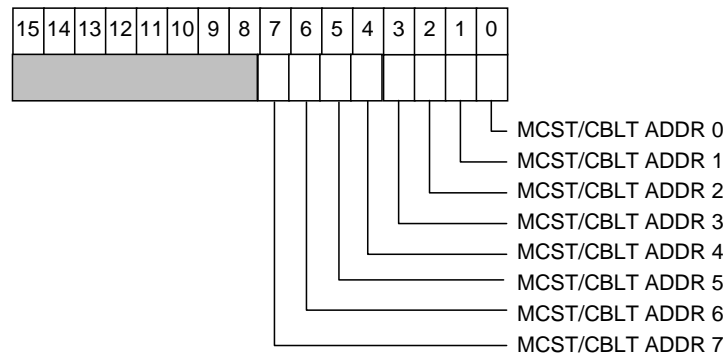
**N.B.:** Since the module has not the PAUX connector, addressing via geographical address is available ONLY for data identification during CBLT operation.

## 4.8. MCST/CBLT Address Register

(Base Address + 0x1004, 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 § 4.1.3 for details about MCST/CBLT addressing mode.

The register content is the following:



**Fig. 4.13: MCST/CBLT address register**

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

## 4.9. Bit Set 1 Register

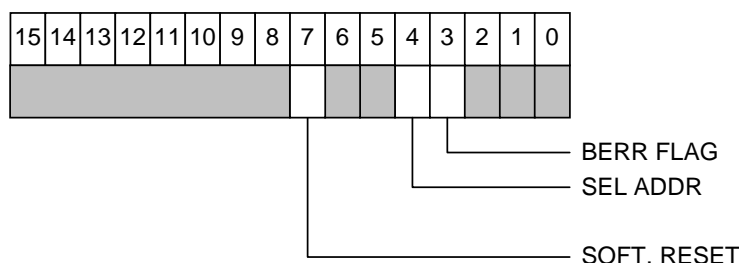
(Base Address + 0x1006, read/write)

This register allows to set the RESET logic of the module and to enable the change of the base address via VME.

A write access with the bits to 1 sets the relevant bits to 1 in the register (i.e. writing 0x10 to this register sets the SEL ADDR bit to 1). A write access with the bits set to 0 does NOT clear the register content; in other words, when 1 is written into one particular bit, such bit is set to 1, if 0 is written, the bit remains unchanged. In order to clear the register content, the Bit Clear 1 Register must be used (see § 4.10).

A read access returns the status of this register.

The register content is the following:



**Fig. 4.14: Bit Set 1 Register**

**BERR FLAG:** Bus Error Flag Bit (meaningful in BLT/CBLT modes only). The User may set this flag for test purposes only. Its content is cleared both via an hardware and via a software reset.

- = 0 board has not generated a Bus Error (default);
- = 1 board has generated a Bus Error.

**SELECT ADDRESS:** Select Address bit.

- = 0 base address is selected via Rotary Switch (default);
- = 1 base address is selected via internal ADER registers.

**SOFTW. RESET:** Sets the module to a permanent RESET status. The RESET is released only via write access with the relevant bit set to 1 in the Bit Clear Register, see § 4.10.

This register is reset via a hardware reset (see § 2.8). Only the bit 3 (BERR FLAG) is reset both via hardware reset and software reset.

## 4.10. Bit Clear 1 Register

(Base Address + 0x1008, read/write)

This register allows to clear the bits in the above described Bit Set 1 Register. A write access with a bit set to 1 resets that bit, e.g. writing 0x8 to this register resets the BERR

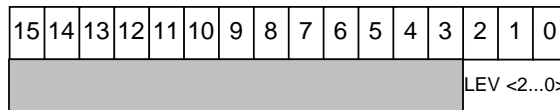
FLAG bit. A write access with the bits set to 0 does NOT clear the register content; in other words, when 1 is written into one particular bit, such bit is set to 0 (cleared), if 0 is written, the bit remains unchanged. The structure of the register is identical to the Bit Set 1 Register.

---

## 4.11. 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.



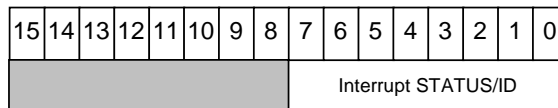
**Fig. 4.15: Interrupt Level Register**

---

## 4.12. Interrupt Vector Register

(Base Address + 0x100C, read/write)

This register contains the STATUS/ID that the V965 INTERRUPTER places on the VME data bus during the Interrupt Acknowledge cycle (Bits 8 to 15 are meaningless). Default setting is 0x00.



**Fig. 4.16: Interrupt Vector Register**

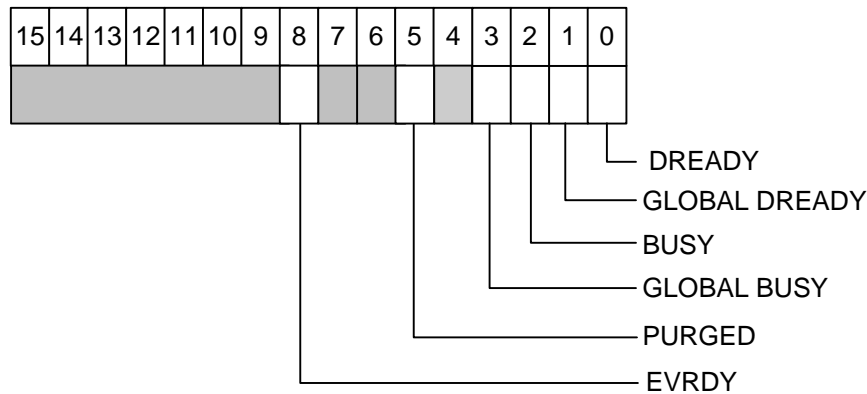
---

## 4.13. Status Register 1

(Base + 0x100E, read only)

This register contains information on the status of the module. The BUSY and DATA READY signals, are available both for the individually addressed module and as a global readout of a system of many units connected together via the CONTROL Bus.





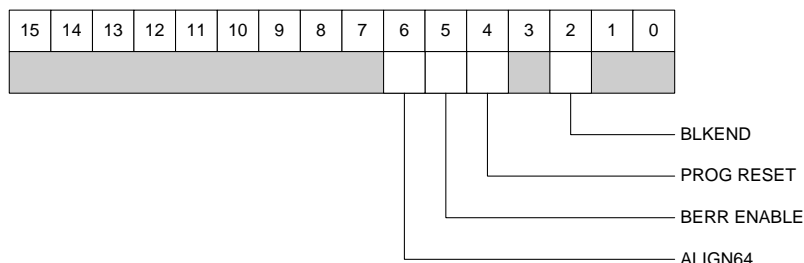
**Fig. 4.17: Status Register 1**

- DREADY:** Indicates that there are data (at least 1 event) in the Output Buffer.  
 = 0 No Data Ready;  
 = 1 Data Ready.
- GLOBAL DREADY:** Indicates that at least one module in the chain has data in the Output Buffer (OR of the READY+ signal of each module in the chain).  
 = 0 No module has Data Ready;  
 = 1 At least one module has Data Ready.
- BUSY:** Busy status indicates that either a conversion is in progress or the board is resetting or the Output Buffer is full or the board is in MEMORY TEST mode.  
 = 0 Module not Busy;  
 = 1 Module Busy.
- GLOBAL BUSY:** Indicates that at least one module connected over the CONTROL Bus is BUSY (OR of the BUSY+ signal of each module in the chain).  
 = 0 No module is Busy;  
 = 1 At least one module is Busy.
- PURGED:** during a CBLT operation it indicates that the board is purged, i.e. the board has finished to send data.  
 = 0 the board is not purged.  
 = 1 the board is purged.
- EVRDY:** is a flag for the Event Trigger Register.  
 = 0 (default) indicates that the number in the Event Trigger Register (see § 4.19) is smaller than the number of events stored in the memory.  
 = 1 indicates that the number in the Event Trigger Register (see § 4.19) is greater than or equal to the number of events stored in the memory and an interrupt request has been generated with interrupt level different from 0 (see § 4.2.3).

## 4.14. Control Register 1

(Base Address + 01010, read/write)

This register allows performing some module's general settings.



**Fig. 4.18: Control Register 1**

- BLKEND:** End of Block bit. Used in Block Transfer modes only.  
= 0 The module sends all requested data to the CPU; when the Output Buffer is empty it will send no valid data. If BERR\_VME is enabled (see bit 5 below, BERR ENABLE), a Bus Error is generated with the readout of the last word in the Output Buffer (default).  
= 1 The module sends all data to the CPU until the first EOB word (end of first event) is reached; afterwards it will send no valid data. If BERR\_VME is enabled, a Bus Error is generated at the readout of the EOB word.
- PROG RESET:** Programmable Reset Mode setting bit.  
= 0 the front panel RESET acts only on data (data reset, default);  
= 1 the front panel RESET acts on the module (software reset).  
**N.B. This bit is cleared only via hardware reset.**
- BERR ENABLE:** 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.
- ALIGN 64:** Allows to add a 32 bit dummy-word (marked as not valid datum, see § 4.5) to an event which is made up of an odd number of words during BLT32 and CBLT32 data readout. In fact some 64 bit CPU's cut off the last 32 bit word of a transferred block if the number of words composing such block is odd, so it is necessary to add a dummy word (which will be then eventually removed via software) in order to avoid data loss. It is used in BLT32 and CBLT32.  
= 0 no dummy word added (default);  
= 1 dummy word added when the number of words is odd.
- (Bits 7 to 15 are meaningless).

This register is reset both via software and via hardware reset (see § 2.8), except for the bit 4 (PROG RESET) which is reset only via hardware reset.

---

## 4.15. Address Decoder High Register

(Base Address + 0x1012, 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 (see also § 4.9). The register content is the following:

|    |    |    |    |    |    |   |   |           |   |   |   |   |   |   |   |
|----|----|----|----|----|----|---|---|-----------|---|---|---|---|---|---|---|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7         | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|    |    |    |    |    |    |   |   | A [31:24] |   |   |   |   |   |   |   |

Fig. 4.19: ADER HIGH Register

---

## 4.16. Address Decoder Low Register

(Base Address + 0x1014 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 (see also § 4.9). The register content is the following:

|    |    |    |    |    |    |   |   |           |   |   |   |   |   |   |   |
|----|----|----|----|----|----|---|---|-----------|---|---|---|---|---|---|---|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7         | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|    |    |    |    |    |    |   |   | A [23:16] |   |   |   |   |   |   |   |

Fig. 4.20: ADER LOW Register

---

## 4.17. Single Shot Reset Register

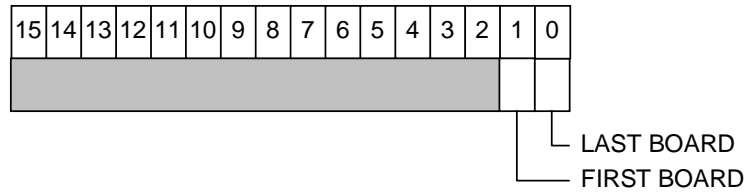
(Base Address + 0x1016, write only)

A write access to this dummy register performs a module reset. This register must be used very carefully and for debugging purposes only. In order to reset the board, it is recommended to use the Bit Set 1 Register (see § 4.9).

## 4.18. MCST/CBLT Control Register

(Base Address + 0x101A, write only)

This register allows performing some general MCST/CBLT settings of the module.



**Fig. 4.21: MCST Address Register**

LAST\_BOARD                      Last Board flag bit (valid in CBLT and MCST modes only)

FIRST\_BOARD                     First Board flag bit (valid in CBLT and MCST modes only)

The status of the boards according to the bit value is the following:

| BOARD STATUS                                    | FIRST BOARD bit | LAST BOARD bit |
|---|-----------------|----------------|
| Board disabled in CBLT or MCST chain            | 0               | 0              |
| First board in CBLT or MCST chain               | 1               | 0              |
| Last board in CBLT or MCST chain                | 0               | 1              |
| Active intermediate board in CBLT or MCST chain | 1               | 1              |

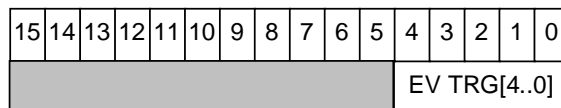
(Bits 2 to 15 are meaningless).

## 4.19. Event Trigger Register

(Base Address + 0x1020, read/write)

This register contains a 5-bit value set by the User: when the number of events stored in the memory equals this value an interrupt request is generated.

Default setting is 0: in this case the interrupt generation is disabled. See also § 4.2.

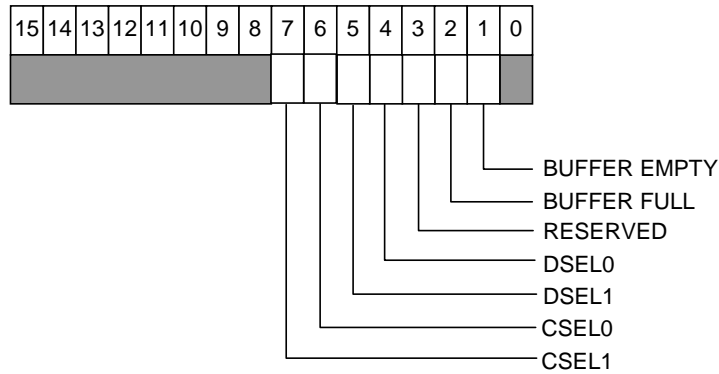


**Fig. 4.22: Event Trigger Register**

## 4.20. Status Register 2

(Base Address + 0x1022, read only)

This register contains further information on the status of the module output buffer and on the type of piggy back plugged into the main board.



**Fig. 4.23: Status Register 2**

**BUFFER EMPTY:** Indicates if the output buffer is empty.  
 =0 buffer not empty;  
 =1 buffer empty.

**BUFFER FULL:** Indicates if the output buffer is full.  
 =0 buffer not full;  
 =1 buffer full.

**CSEL1, CSEL0, DSEL1, DSEL0:** Indicate the type of piggy-back plugged into the board. (0010 for V965, 1110 for V965A)

## 4.21. Event Counter\_Low Register

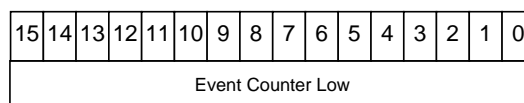
(Base Address + 0x1024, read only)

It contains the 16 LSBs of the event counter. The event counter can work in two different ways (see also § 2.6):

1. it counts all events;
2. it counts only the accepted events.

The two modes can be selected via the Bit 14 (ALL TRG) of the Bit Set 2 Register (see § 4.26).

**EVENT CNT LOW:** 16 LSB of the 24-bit Event Counter.



**Fig. 4.24: Event Counter Low Register**

This register is reset via the Event Counter Reset Register (see § 4.33) or via a software or hardware reset (see § 2.8). However, if the event counter is set so as to work as relative counter (i.e. it counts only the accepted events), this register is reset also with a data reset (see § 2.8).

---

## 4.22. Event Counter\_High Register

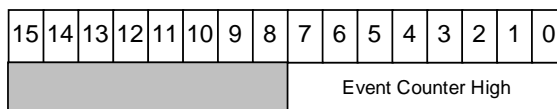
(Base + 0x1026, read only)

It contains the 8 MSB of the 24-bit event counter. The event counter can work in two different ways (see also § 2.6):

1. it counts all events;
2. it counts only the accepted events.

The two modes can be selected via the Bit 14 (ALL TRG) of the Bit Set 2 Register (see § 4.26).

EVENT CNT HIGH:            8 MSB of the 24-bit Event Counter.



**Fig. 4.25: Event Counter High Register**

This register is reset via the Event Counter Reset Register (see § 4.33) or via a software or hardware reset (see § 2.8). However, if the event counter is set so as to work as relative counter (i.e. it counts only the accepted events), this register is reset also with a data reset (see § 2.8).

---

## 4.23. Increment Event Register

(Base Address + 0x1028, write only)

A write access to this dummy register sets the readout pointer on the next event in the output buffer (at the first address).

In particular, if the bit 11 (AUTO INCR) of the Bit Set 2 Register is set to 0 (see § 4.26), the readout pointer is no more automatically incremented, but it can be incremented via a write access to this register or to the Increment Offset Register (see below).

---

## 4.24. Increment Offset Register

(Base Address + 0x102A, write only)

A write access to this dummy register increments the readout pointer of one position (next word, same event if EOB is not encountered; next event if EOB is encountered).

In particular, if the bit 11 (AUTO INCR) of the Bit Set 2 Register is set to 0 (see § 4.26), the readout pointer is no more automatically incremented, but it can be incremented via a write access to this register or to the Increment Event Register (see above).

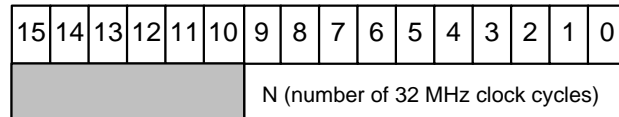
## 4.25. Fast Clear Window Register

(Base Address + 0x102E, read/write)

For the definition of the Fast Clear window refer to Fig. 2.8. By writing a 10 bit number N to this register, it is possible to set the Fast Clear window width  $T_{FC}$  in the range  $7 \div 38.5 \mu\text{s}$  ( $1/32 \mu\text{s}$  steps) according to the following relation:

$$T_{FC} (\mu\text{s}) = N \times T_{\text{CLOCK}} + 7 \mu\text{s}$$

where  $T_{\text{CLOCK}} = 1/32 \mu\text{s}$ .



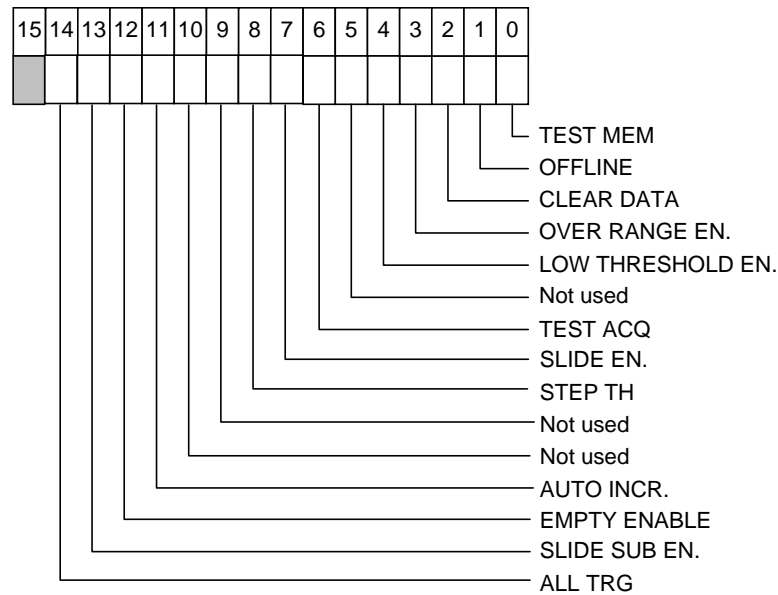
**Fig. 4.26: Fast Clear Window Register**

Please note that the maximum allowed value for N is 3F0 which leads to  $T_{FC} = 38.5 \mu\text{s}$

## 4.26. Bit Set 2 Register

(Base Address + 0x1032, read/write)

This register allows to set the operation mode of the module. A write access with a bit to 1 sets the relevant bit to 1 in the register. A write access with the bit set to 0 does not clear the register content, the Bit Clear 2 Register must be used (see § 4.27). A read access returns the status of the register. The register content is the following:



**Fig. 4.27: Bit set 2 register**

**N.B. DON'T MODIFY RESERVED BITS SETTINGS**

- TEST MEM:** Test bit: allows to select the Random Memory Access Test Mode (see § 5.7.1).  
=0 normal mode (default);  
=1 Random Memory Access Test Mode selected: it is possible to write directly into the memory.
- OFFLINE:** Offline bit: allows to select the ADC controller's status.  
=0 ADC controller online (default);  
=1 ADC controller offline: no conversion is performed.
- CLEAR DATA:** Allows to generate a reset signal which clears the data, the write and read pointers, the event counter and the QAC sections.  
=0 no data reset is generated (default);  
=1 a data reset signal is generated.
- OVER RANGE EN.:** Allows to disable overflow suppression (see also § 2.4).  
=0 over range check enabled: only the data not causing the ADC overflow are written into the output buffer (overflow suppression; default);  
=1 over range check disabled: all the data are written into the output buffer (no overflow suppression).
- LOW THR. EN.:** Allows to disable zero suppression (see also § 2.3).  
=0 low threshold check enabled: only data above the threshold are written into the output buffer (zero suppression; default);  
=1 low threshold check disabled: all the data are written into the output buffer (no zero suppression).
- TEST ACQ:** Allows to select the Acquisition Test Mode (see § 5.7.2).



- =0 normal operation mode, i.e. the data to be stored in the buffer are the real data (default);
- =1 Acquisition Test Mode selected, i.e. the data to be stored in the buffer are taken from an internal FIFO (Test Event Write Register, see § 4.32).

**SLIDE ENABLE:** Allows to enable/disable the sliding scale.

- =0 the sliding scale is disabled and the DAC of the sliding scale is set with a constant value (Slide Constant, see § 4.37);
- =1 the sliding scale is enabled (default).

**STEP TH:** Allows to set the zero suppression threshold resolution (see § 2.3 for details)

- =0  $ADC\ CONVERTED\ VALUE < THRESHOLD\ VALUE \times 16$
- =1  $ADC\ CONVERTED\ VALUE < THRESHOLD\ VALUE \times 2$

**AUTO INCR. EN.:** Allows to enable/disable the automatic increment of the readout pointer.

- =0 the read pointer is not incremented automatically but only by a write access to the Increment Event or Increment Offset Registers (see 4.23 and 4.24);
- =1 the read pointer is incremented automatically (default).

**EMPTY ENABLE:** Allows to choose if writing the header and EOB when there are no accepted channels.

- =0 when there are no accepted channels, nothing is written in the output buffer (default).
- =1 when there are no accepted channels, the Header and the EOB are anyway written in the output buffer.

**SLIDE SUB EN.:** Allows to change operation mode for the sliding scale.

- =0 the sliding scale works normally (default);
- =1 the subtraction section of the sliding scale is disabled (test purposes only).

**ALL TRG:** Allows to choose how to increment the event counter.

- =0 event counter incremented only on accepted triggers.
- =1 event counter incremented on all triggers (default).

---

## 4.27. Bit Clear 2 Register

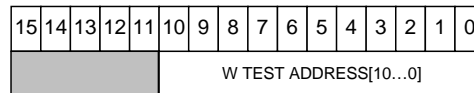
(Base Address + 0x1034 write only)

This register allows clearing the bits of the Bit Set 2 Register (§ 4.26). A write access with a bit set to 1 resets that bit, e.g. writing 0x4 to this register resets the CLEAR DATA bit. A write access with the bits set to 0 does NOT clear the register content. The structure of the register is identical to the Bit Set 2 Register.

## 4.28. W Memory Test Address Register

(Base Address + 0x1036 write only)

This register contains the address of the memory on which data can be written for the memory test.



**Fig. 4.28: W Memory Test Address Register**

**N.B.:** The output buffer is a FIFO, so the read address (R Memory Test Address Register) must be different from the write address (W memory Test Address Register).

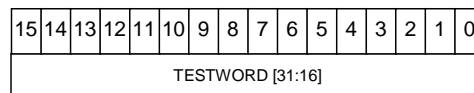
## 4.29. Memory Test Word\_High Register

(Base Address + 0x1038 write only)

The Memory Test Word is a 32-bit word used for the memory test. The higher 16 bits are set via this register, while the lower 16 bits are set via the Test Word\_Low Register. These registers are used in TEST mode as follows:

1. set the module in test mode (see bit 0 of the Bit Set 2 Register, § 4.26);
2. write the memory address (see § 4.28),
3. write the 16 MSBs in the TESTWORD\_HIGH register;
4. write the 16 LSBs in the TESTWORD\_LOW register;

With the latter operation, the 32-bit pattern is transferred to the memory. If operations 3. and 4. are inverted, the content of the 16 MSBs may be meaningless.

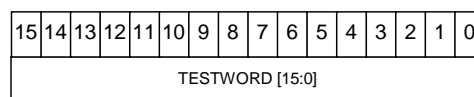


**Fig. 4.29: Test Word\_High Register**

## 4.30. Memory Test Word\_Low Register

(Base Address + 0x103A write only)

This register allows to set the lower 16 bits of the Test Word (see above).



**Fig. 4.30: Test Word\_Low Register**

## 4.31. Crate Select Register

(Base Address + 0x103C read/write)

This register contains the number of the crate which the board is plugged into. This register must be filled at board initialisation and will be part of the data word (see § 4.5).

|    |    |    |    |    |    |   |   |                     |   |   |   |   |   |   |   |
|----|----|----|----|----|----|---|---|---------------------|---|---|---|---|---|---|---|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7                   | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|    |    |    |    |    |    |   |   | CRATE NUMBER[7...0] |   |   |   |   |   |   |   |

Fig. 4.31: Crate Select Register

## 4.32. Test Event Write Register

(Base Address + 0x103E write only)

This register is used in Acquisition Test Mode and its content constitutes the test event to be written in the output buffer.

A write access to this register allows the User to write a set of 32 data into a 32-word FIFO. As the Bit 6 (TEST ACQ) of the Bit Set 2 Register (see § 4.26) is set to 1 and the Acquisition Test Mode is consequently selected, these data are directly written in the output buffer constituting an event which can be used to test the module and/or the acquisition software.

Each 16-bit test word (see the figure below) contains a 12-bit value, acting as the ADC converted value, and an OV bit which indicates the possible overflow.

The 32 test data, corresponding to the data from the 16 channels (dual range), must be written in this FIFO in the same order as they are read from the output buffer, that is:

- test datum for the channel 0 high range
- test datum for the channel 8 high range
- test datum for the channel 0 low range
- .....
- test datum for the channel 15 high range
- test datum for the channel 7 low range
- test datum for the channel 15 low range

For further details on the use of this register in Acquisition Test Mode please refer to § 5.7.2.

**N.B.:** please note that the User must write at least and not more than 32 test words. Actually, since the words are written in a FIFO, if the User writes less than 32 words, some words will be not defined; on the other hand, if the User writes more than 32 words, some words will be overwritten.

|    |    |    |    |                                  |    |   |   |   |   |   |   |   |   |   |   |
|----|----|----|----|----------------------------------|----|---|---|---|---|---|---|---|---|---|---|
| 15 | 14 | 13 | 12 | 11                               | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|    |    |    | OV | TEST ADC CONVERTED VALUE[11...0] |    |   |   |   |   |   |   |   |   |   |   |

Fig. 4.32: Test Event Write Register

### 4.33. Event Counter Reset Register

(Base Address + 0x1040 write only)

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

### 4.34. Iped Register

(Base Address + 0x1060 read/write)

This register allows to program on 8 bits the  $I_1$  current (see § 2.1.1 and § 2.1.2) to set the QDC pedestal (common for all channels).

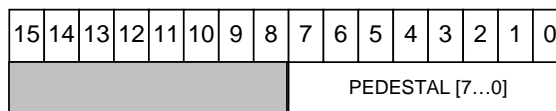


Fig. 4.33: Iped Register

#### 4.34.1. Piggy back description

The following relation holds for the pedestal current (see § 2.1.1 and § 2.1.2):  $I_P = I_1 - I_0$  (where  $I_0 \approx 500 \mu\text{A}$  and  $I_1$  is VME programmable via the Iped register); the  $I_1$  current can be programmed in  $0.5 \mu\text{A}$  steps up to roughly  $620 \mu\text{A}$  (corresponding to PEDESTAL = 255). The default value is PEDESTAL = 180. For correct operation the User should pay attention to set  $I_1 \geq I_0$ ; it is recommended to program PEDESTAL  $\geq 60$ .

For example PEDESTAL = 180 corresponds to  $I_1 \approx 582.5 \mu\text{A}$  (typ.) and thus  $I_P \approx 82.5 \mu\text{A}$ ; if the gate width is 200 ns, this  $I_P$  current gives a QDC pedestal of approximately 80 counts for High Range and 640 counts for Low Range.

### 4.35. R Memory Test Address Register

(Base Address + 0x1064 write only)

This register contains the address of the output buffer from which data can be read for the memory test.

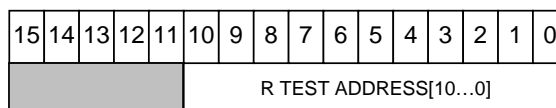


Fig. 4.34: R Memory Test Address Register

**N.B.:** The output buffer is a FIFO, so the read address (R Test Address Register) must be different from the write address (W Test Address Register).

---

## 4.36. SW Comm Register

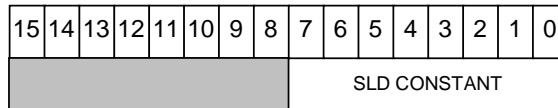
(Base Address + 0x1068 write only)

A write access to this dummy register causes a conversion for test purposes.

---

## 4.37. Slide constant Register

(Base Address + 0x106A read/write)



**Fig. 4.35: Slide Constant Register**

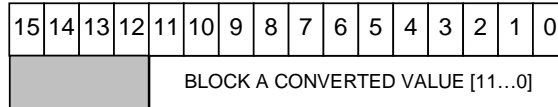
This register contains a 8-bit value corresponding to the constant to which is set the sliding scale DAC when the sliding scale is disabled by means of the SLD\_ENABLE bit of the Bit Set 2 Register (refer to § 4.26).

---

## 4.38. AAD Register

(Base Address + 0x1070 read only)

This register contains the value converted by the ADC of the Block A (refer to the block diagram of Fig. 1.1).



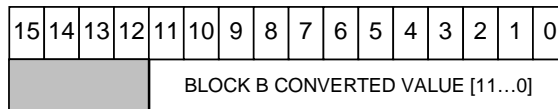
**Fig. 4.36: AAD Register**

---

## 4.39. BAD Register

(Base Address + 0x1072 read only)

This register contains the value converted by the ADC of the Block B (Refer to the block diagram of Fig. 1.1).

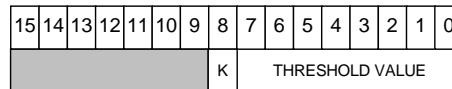


**Fig. 4.37: BAD Register**

## 4.40. Thresholds Memory

(Base Address + 0x1080 ÷ 0x10BE read/write)

This register contains the low threshold and kill option for each channel. The address is different for each channel (V965: ch0 high range → 0x1080, ch0 low range → 0x1082, ..., ch15 high range → 0x10BC, ch15 low range → 0x10BE; V965A: ch0 high range → 0x1080, ch0 low range → 0x1084, ..., ch7 high range → 0x10B8, ch7 low range → 0x10BC). Each threshold is as shown in the figure:



**Fig. 4.38: Threshold Register**

**KILL (K):** allows to abort memorisation of the data from the relevant channel.  
= 0 channel data are memorised;  
= 1 channel data memorisation is aborted.

**THRESHOLD VALUE:** see § 2.3.

Default settings are not defined. Please note that the KILL option can be used to disable some channels.

**N.B.:** the threshold values are reset only with a hardware reset and when the board is switched off.

## 4.41. ROM memory

(Base Address + 0x8000 ÷ 0xFFFFE, read only)

It contains some useful information according to the table below, such as:

- **OUI:** manufacturer identifier (IEEE OUI);
- **Version:** purchased version of the Mod.V965;
- **Board ID:** Board identifier (965);
- **Revision:** hardware revision identifier;
- **Serial MSB:** serial number (MSB);
- **Serial LSB:** serial number (LSB).

**Table 4.5: ROM Address Map for the Model V965**

| Description                      | Address | Content     |
|----------------------------------|---------|-------------|
| <b>OUI MSB</b>                   | 0x8026  | <b>0x00</b> |
| <b>OUI</b>                       | 0x802A  | <b>0x40</b> |
| <b>OUI LSB</b>                   | 0x802E  | <b>0xE6</b> |
| <b>Version</b>                   | 0x8032  | <b>0x00</b> |
| <b>BOARD ID MSB</b>              | 0x8036  | <b>0x00</b> |
| <b>BOARD ID</b>                  | 0x803A  | <b>0x03</b> |
| <b>BOARD ID LSB</b>              | 0x803E  | <b>0xC5</b> |
| <b>Mother Board Revision</b>     | 0x804E  | <b>0x00</b> |
| <b>Piggy Back Board Revision</b> | 0x8052  | <b>0x00</b> |
| <b>Serial MSB</b>                | 0x8F02  | <b>0x00</b> |
| <b>Serial LSB</b>                | 0x8F06  | <b>0x02</b> |

---

## 5. Operating modes

---

### 5.1. Safety information

This section contains the fundamental safety rules for the installation and operation of the module.

Read thoroughly this section before starting any procedure of installation or operation of the product.

---

### 5.2. Product Damage Precautions

**Use Proper Power Source.**

Do not operate this product from a power source that applies more than the voltage specified.

**Provide Proper Ventilation.**

To prevent product overheating, provide proper ventilation.

**Do Not Operate With Suspected Failures.**

If you suspect there is damage to this product, have it inspected by qualified service personnel.

---

### 5.3. Installation

The V965 board can be inserted into both standard and V430 VME 6U crates. Please note that the board supports live insertion/extraction into/from the crate, i.e. it is possible to insert or extract it from the crate without turning the crate off. Moreover, it is possible to switch the board off by the relevant PWR switch (see § 3.5.2) without cutting the interrupt chain off.



## CAUTION

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

---

### 5.4. Power ON sequence

To power ON the board follow this procedure:

1. insert the V965 board into the crate: as the board is inserted, the OVC PWR green LED lights up indicating that the board is powered;

2. after a short time the BUSY and DRDY LEDs will light off: this indicates that the board is ready to acquire data.

**N.B.:** if the OVC PWR LED becomes orange 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 OVC PWR LED is orange 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.

---

## 5.5. 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 Read and Write Pointer are cleared (i.e. Buffer 0 is pointed);
- the Interrupt Level is set to 0x0 (in this case interrupt generation is disabled) and the Interrupt Vector is set to 0x0;
- the values in the threshold memory are not defined (see § 4.40);
- the MCST/CBLT address is set to 0xAA.

Moreover, all other registers marked in the column HR (Hardware RESET) in Table 4.2 are cleared or set to the default value.

At power on or after a hardware reset (see § 2.8) the module must thus be initialised.

---

## 5.6. Operation sequence

After the power ON sequence the module is in the status described above.

Please note that the threshold values are not defined after power ON and consequently before starting the operation of the module it is necessary to set a threshold value for each channel in the Threshold memory (refer to § 4.40).

If the module is not BUSY, a GATE input pulse causes the following:

1. starts the Charge to Amplitude Conversion;
2. increments the event counter according to the User's settings (see § 2.6);
3. sets the BUSY output signal to 1.

If neither RESET nor FAST CLEAR occur (refer to § 2.8 and § 2.9) to abort the Charge to Amplitude Conversion, the control logic starts the following conversion sequence:

1. The outputs of the QAC sections are multiplexed and sampled;



2. The control logic checks if there are accepted data among the converted values, according to the User's settings (zero suppression, overflow suppression and KILL option: see § 2.3 and § 2.4):
  - a) if there are accepted data, these are stored in the active event buffer together with a Header and an EOB;
  - b) if there are no accepted data and the EMPTY ENABLE bit of the Bit Set 2 Register is set to 0 (default setting, see § 4.26), no data will be written in the output buffer.
  - c) if there are no accepted data and the EMPTY ENABLE bit of the Bit Set 2 Register is set to 1 (see § 4.26), the Header and EOB only will be written in the output buffer.
3. If the MEB is not full the QAC sections and the BUSY are cleared and the module is ready for the next acquisition; if the MEB is full the module doesn't accept any GATE and BUSY is not cleared.

---

## 5.7. Test Modes

Two different test modes can be enabled:

- **Random Memory Access Test Mode**
- **Acquisition Test Mode**

The first test mode operation is enabled via the Bit 0 of the Bit Set 2 Register and allows to write directly into the buffer.

The second test mode is enabled via the Bit 6 of the Bit Set 2 Register and allows to test the whole acquisition system by writing a set of 32 data in an internal FIFO which are then transferred to the output buffer at each GATE pulse for the readout.

The test modes will be described in detail in the following subsections.

---

### 5.7.1. Random Memory Access Test Mode

This test mode allows the User to write and read a word in the output buffer.

To perform such test follow these steps:

1. Set to 1 the Bit 0 of the Bit Set 2 Register (see § 4.26);
2. Write into the W Memory Test Address Register (see § 4.28) the 11-bit address where to write the test word;
3. Write the high and low part of the 32-bit test word respectively in the Testword\_High and Testword\_Low Registers (see § 4.29 and § 4.30). As the Testword\_Low register is accessed, the whole test word is written into the memory;
4. Write in the R Test Address Register (see § 4.35) the 11-bit reading memory address and read out the buffer; please note that this address must be different from the write address written in the W Memory Test Address Register.

**N.B.:** please note that the R Memory Test Address must be different from the W Memory Test Address at any step of the procedure. If the User tries to write an

address in one of these registers that is equal to the address contained in the other register, write cycles (step 3. above) will not write the correct value.

---

### 5.7.2. Acquisition Test Mode

This test mode allows the User to simulate the real operation of the board without using any channel input signals but just writing the data into a FIFO via an appropriate register (Test Event Write Register, see § 4.32) and reading them after a GATE signal.

To operate the acquisition test follow these steps:

1. Set to 1 the Bit 6 (TEST ACQ) of the Bit Set 2 Register (see § 4.26); this action selects the Acquisition Test Mode and resets the write pointer in the FIFO;
2. Set to 0 the Bit 6 (TEST ACQ) of the Bit Set 2 Register (see § 4.26); this action resets the read pointer in the FIFO and releases the write pointer;
3. Write 32 data words (each word consisting of a 13-bit word, corresponding to the ADC converted value, + the overflow bit, see § 4.32) in the Test Event Write Register (Base Address + 0x103E). These 32 data constitute the event to obtain as output of the 16 channels (dual range). The 32 test data must be written in this FIFO in the same order as they will be read from the output buffer, that is:

- test datum for the channel 0 high range
- test datum for the channel 8 high range
- test datum for the channel 0 low range
- .....
- test datum for the channel 15 high range
- test datum for the channel 7 low range
- test datum for the channel 15 low range

**N.B.: please note that the User must write at least and not more than 32 test words. Actually, since the words are written in a circular FIFO, if the User writes less than 32 words, some words will be not defined; on the other hand, if the User writes more than 32 words, some words will be overwritten.**

4. Set to 1 the Bit 6 (TEST ACQ) of the Bit Set 2 Register (see § 4.26); this action resets again the write pointer in the FIFO and releases the read pointer;
5. Send a set of GATE input signals: at each GATE signal the data previously written in the FIFO will be transferred to the output buffer. The data will be read via VME in the same order as they were written into the FIFO:

- test datum for the channel 0 high range
- test datum for the channel 8 high range
- test datum for the channel 0 low range
- .....
- test datum for the channel 15 high range
- test datum for the channel 7 low range
- test datum for the channel 15 low range

**N.B.: To operate in normal mode again, the Bit 6 of the Bit Set 2 Register must be set again to 0.**

---

## 5.8. Block Transfer Mode

The module supports the Standard BLT32 and MBLT64 modes.

A standard readout in Block Transfer mode, for example, consists of a readout of the Header for the relevant event and a Block Transfer readout of the number of data words relative to the event (the number of data words referring to the event is the CNT number in the Header, see § 4.5).

A more efficient readout in Block Transfer mode can be performed by using the BLOCK END and BERR-ENABLE bits of the Control Register 1 (see § 4.14).

Some examples of this type of readout in Block Transfer mode are as follows:

**Example A:** BLOCK END = 0, BERR\_ENABLE = 0;  
A Block Transfer readout of 32x34 words (32 events max., each event 34 words max.) allows the readout of all data stored in the buffer: as the buffer is empty, the module will send only not valid data.

**Example B:** BLOCK END = 0, BERR\_ENABLE = 1;  
A Block Transfer readout of 32x34 words (32 events max., each event 34 words max.) allows the readout of all events stored in the buffer: as the buffer is empty, a BERR is generated.

**Example C:** BLOCK END = 1, BERR\_ENABLE = 0;  
A Block Transfer readout of 34 words (each event 34 words max.) allows the readout of one complete event: after the readout of the EOB the module will send only not valid data.

**Example D:** BLOCK END = 1, BERR\_ENABLE = 1;  
A Block Transfer readout of 34 words (each event 34 words max.) allows the readout of one complete event: as the EOB is encountered, a BERR is generated.

The use of the BERR\_ENABLE bit (Examples B and D above) is suggested only if the VME CPU can handle the Bus Error (BERR) in an effective way.

**N.B.: Please note that, according to the VME standard, a Block Transfer readout can be performed with 256 read cycles maximum: as a consequence, a readout with a greater number of read cycles may require more BLT operations.**

This limit is not due to the board itself but only to the VME standard: if it is possible to disable or delay the timeout of the BUS Timer (BTO(x)), a Block Transfer readout with more than 256 read cycles can be performed as well.

---

## 5.9. Advanced Setting and Readout Modes

Chained Block Transfer (CBLT) and Multicast (MCST) operations allow to enhance the set and readout time of the 16 channels. These operations allow accessing several boards at the same time: CBLT operations are used for reading cycles only, while MCST operations are used for write cycles only. For further details on the CBLT/MCST addressing mode please refer to § 4.1.3 and § 4.1.4.

In order to perform CBLT and MCST operations, the higher Base Address bits of all the involved modules (i.e. bits 31 to 24) must be set in common to all boards via the MCST/CBLT Address Register (see § 4.8). This means that all boards must have the same setting on bits 31 to 24.

The resulting MCST (CBLT) Base Address for all boards is:

**MCST (CBLT) Base Address = %NN000000,**

Once the addresses have been set, the first and last board in a chain must have, respectively, only the FIRST\_BOARD and only the LAST\_BOARD bit set to 1 in the MCST Control Register (see § 4.8). Conversely, all intermediate boards must have both the FIRST\_BOARD and the LAST\_BOARD bits set either to 1 or to 0.

---

### 5.9.1. Chained Block Transfer Mode

Once set the address of the boards as described in the above section, the boards can be accessed in Chained Block Transfer mode (CBLT, see [5]). This mode allows for sequential readout of a certain number of contiguous boards in a VME crate. A CBLT access is allowed with the BLT32 and MBLT64 address modifiers only (CBLT32 and CBLT64 accesses respectively).

**N.B.: The CBLT operation can be performed only for the readout of the Multi-Event Buffer: its address in CBLT mode corresponds to the set of offsets listed in Table 4.3 to be added to the address, common to all boards, set by the User via the MCST/CBLT Address Register which contains the most significant bits of the address (see § 4.8).**

The User must perform a number of CBLT accesses that allows for the readout of all data in all boards of the chain in all possible occupancy conditions. E.g.: if the User has a chain of 10 boards, the total number of words for a given event lies between 0 (i.e. no data) and  $34 \times 10 = 340$  32-bit words (i.e. each board has an event, each event consists of a Header + 32 data + End of Block). In order to be sure that a BERR is generated, the User must thus perform 11 CBLT accesses of 34-word each.

In CBLT32 mode the first board of the chain starts sending data (if there are any, i.e. if it is not purged, see § 4.13); as it has sent all data and the EOB is met, the board becomes purged, i.e. the relevant bit (PURGED) of the Status Register 1 is set to 1. This implies that the board will not be involved in the CBLT access any more since it has already sent all the required data. At this point the IACKOUT line is asserted and the next board, if not purged, starts sending data. As the last board receives the token and is purged, it asserts a BERR which acts as a data readout completion flag.

In CBLT64 mode the accesses work as in the CBLT32 one, except for the fact that the address is acknowledged during the first cycle and consequently a DTACK is asserted at least once.

In CBLT mode the Read Pointer must be incremented automatically: if the AUTOINC\_ENABLE bit is set to 1 in the Bit Set 2 Register (see § 4.26), the Read Pointer is automatically incremented with the readout of the End Of Block word of each board; if the AUTOINC\_ENABLE bit is set to 0, the Read Pointer is not automatically incremented and only the Header of the first word is read.

**N.B.: Please note that, according to the VME standard, a Chained Block Transfer readout can be performed with 256 read cycles maximum: as a consequence, a readout with a greater number of read cycles may require more CBLT operations.**

This limit is not due to the board itself but only to the VME standard: it is actually possible to performed a CBLT readout with more than 256 read cycles if the timeout of the BUS Timer (BTO(x)) is disabled or delayed.

If the latter action is not allowed and the CBLT readout stops before having read all data, the new CBLT cycle will start from where the token was left in the previous cycle: this goes on until the last board is reached and all data read, so that a BERR is generated.

---

### **5.9.2. Multicast Commands**

Once set the address of the boards as described in § 5.9, the boards can be accessed in Multicast Commands (MCST) mode. The MCST mode allows to write in the registers of several boards at the same time by accessing a dummy Address only once. The latter is composed by the MCST Base Address plus the offset of the relevant register, according to the list shown in Table 4.4. Refer to § 4.1.3 for details on MCST addressing mode. MCST access can be meaningless (even if possible) for the setting parameters depending on the individual channel characteristics.

**N.B.: the MCST/CBLT Address Register must NEVER be accessed in MCST mode since this can affect the CBLT and MCST operations themselves.**

---

## 6. 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 analysis", 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.

Both documents are available from URL: <http://www.vita.com>

---

## 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.

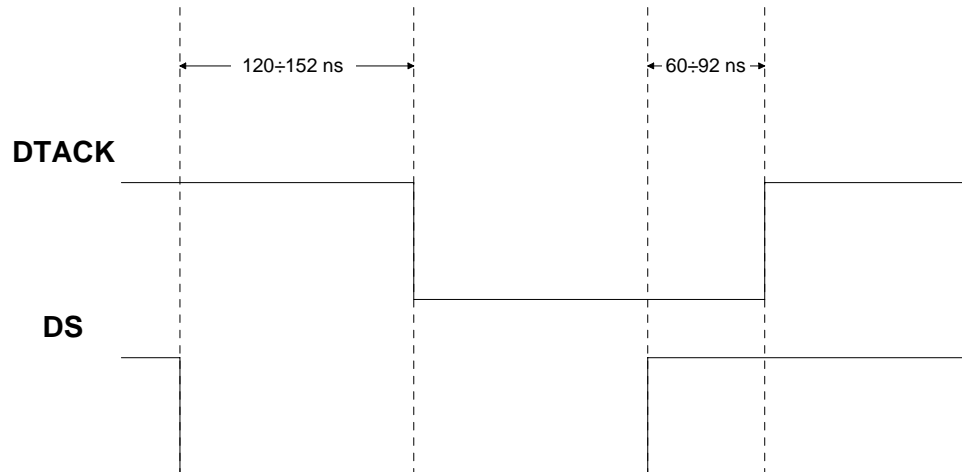


Fig.A.1: VME cycle timing in D16 mode

---

## 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.

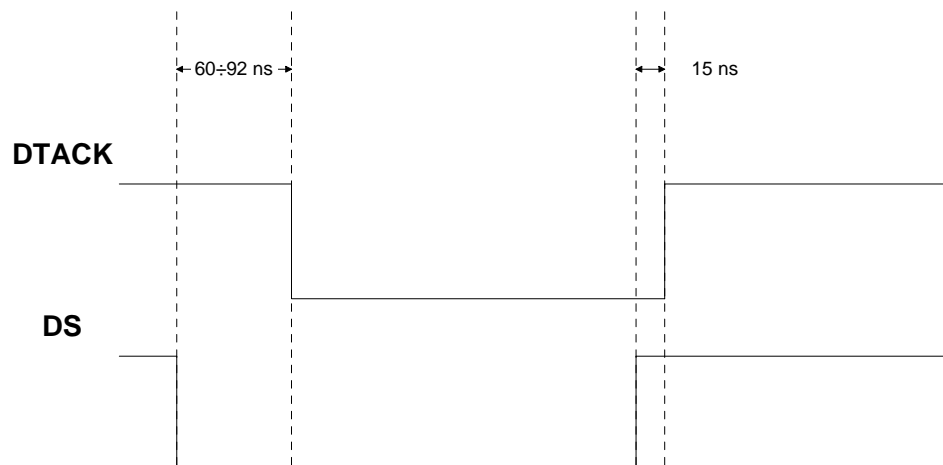


Fig.A.2: VME cycle timing in BLT/CBLT mode



### 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.

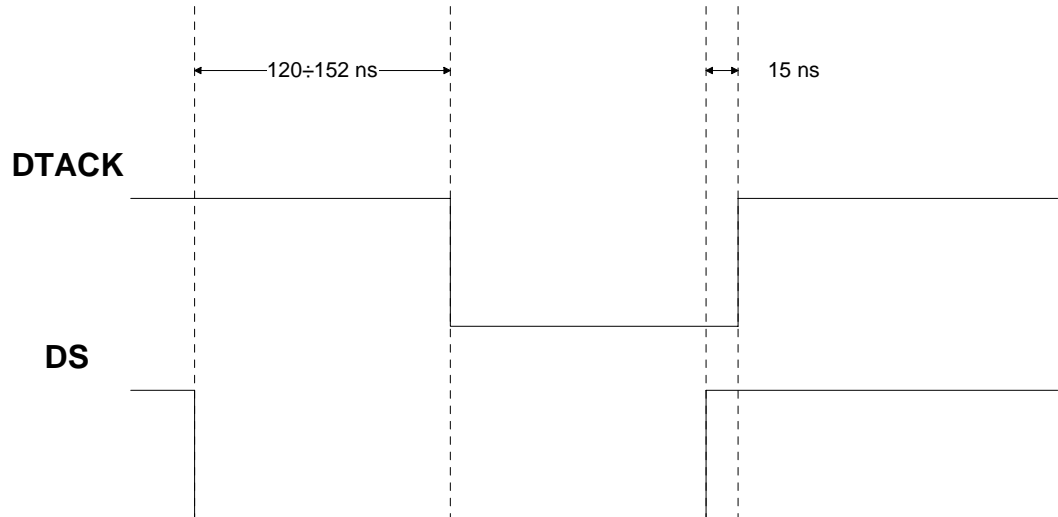


Fig.A.3: VME cycle timing in MBLT/CBLT64 mode