

High Pt Pretrigger Electronics

Message Generator 2

Content

Introduction

Logic Description

Veto Number Distribution

VME Access

VME Instructions

Appendix

1) Board Layout

2) Jumpers

3) Backplane Connector P2

4) Backplane Connector P3

5) TFU Connector P5

6) VETO Number Connector P6

7) Message Bit Partitioning

Introduction

The High-Pt Pretrigger Electronics is built of three different boards:

- The Link Board **LB**, which is located near the detector and provides via optical fibres the fast data transfer to the main trigger logic.
- The Pretrigger Board **PB**, which searches for coincidence pattern as trigger candidates and combines the involved pad information of three detector layers to data sets, which are transmitted to the third board,
- The Message Generator **MG**, which transforms the received data to messages, which are accepted by the Track Finding Unit **TFU** of the HERA-B First Level Trigger System.

This Manual describes the Messages Generator 2, which differs from the first release mainly in the following two aspects:

- The width of the look-up table has been increased from 64 to 72 bits.
- The number of output ports has been increased from one to four. So a content depending distribution of messages to four different TFU links is possible resulting in a reduced load of the connections.

In the first chapter, the logic of the board is shortly presented on the block diagram level. Then the Veto Number Distribution is mentioned, before the next chapters report the control, programming and test facilities provided by VME. Finally an overview of the VME instructions, implemented on the Message Generator, is given.

Logic Description

The block diagram in fig.1 shows the main components of the Message Generator 2:

- The Interface to the Pretrigger Boards, implemented by the **Handshake Logic** and the **Data Input Register**.
- The Message Generation Logic, provided by the **Road Encoder**, the **Look-up Table** and the **Message FIFO**
- The Message Distribution to **PortA..PortD**, controlled by Port Registers **REGA..REGD**.

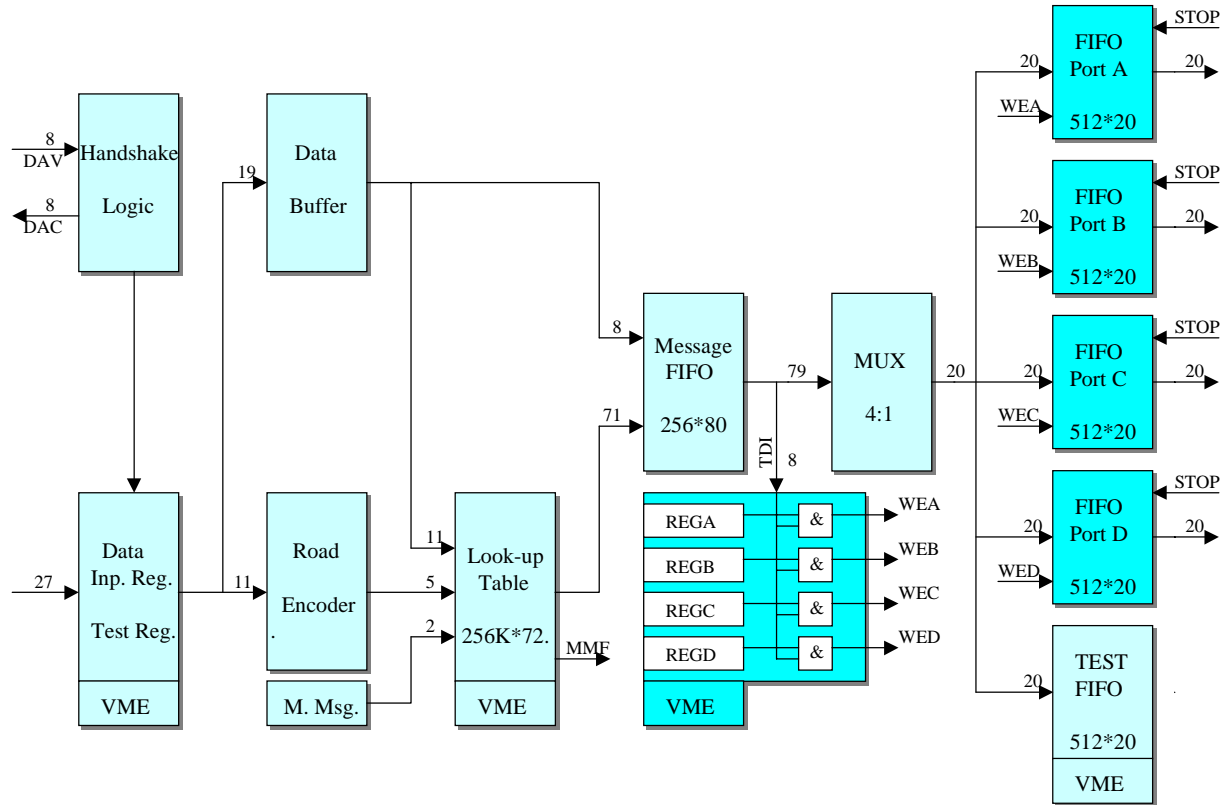


Fig.1: Block Diagram of Message Generator 2

One Message Generator can be linked to 8 Pretrigger Boards. Therefore it has to observe 8 **DAV** flags, each indicating whether there are data available on the different boards.

The **Handshake Logic** is responsible for the data transfer protocol between **PB** and **MG2**. At first it periodically strobes the pattern of **DAV** flags into the input register of an 8 bit priority encoder. If there is at least one flag set, the encoder starts to select the **PB** with the highest priority (highest number) by issuing the corresponding **DAC** signal, which opens the connected data port and enables the 27 bit data pattern to arrive at the **Data Input Register**, where it is stored. Then the next **DAV** flags are selected one after another, until all flags in the encoder input register are served, and then the next **DAV** pattern is stored. That method ensures, that the selection probability is equal for all connected Pretrigger Boards.

The data transfer rate between Pretrigger Boards and Message Generator is 25 MHz, provided, that data are always available.

The 27 bit data set coming from the **PB** contains besides the 8 bit Bunch Number and one Cycle Bit the pattern information of possible coincidences, consisting of one pixel of the first detector layer, which is represented by a 7 bit code (First Pixel Code), together with 5 and 6 pixels of the second and third layer respectively. (For more details see the Pretrigger Board Manual.) This data set is stored in the **Data Input Register** together with a 3 bit code for the data source, generated by the **Handshake Logic** (Data Source Code = PB Board Address -1).

11 bits representing five pads **PIB0..PIB4** of the second detector layer and 6 pads **PIC0..PIC5** of the third detector layer are processed by the **Road Encoder**, which for 18 different coincidences generates a 5 bit code according to the following table:

Coincidence	Code
PIB0*PIC0	0
PIB0*PIC1	1
PIB0*PIC2	2
PIB1*PIC0	3
PIB1*PIC1	4
PIB1*PIC2	5
PIB1*PIC3	6
PIB2*PIC1	7
PIB2*PIC2	8
PIB2*PIC3	9
PIB2*PIC4	10
PIB3*PIC2	11
PIB3*PIC3	12
PIB3*PIC4	13
PIB3*PIC5	14
PIB4*PIC3	15
PIB4*PIC4	16
PIB4*PIC5	17

Table1: Coincidence Code

The resulting Coincidence Code together with 11 bits coming from the **Data Buffer** and two bits provided by a Repetition Counter for generation of multiple messages are forming the 18 bit address of the **Look-up Table**, as shown in the following table:

LUT Address Bits	Provided by
Ad0..Ad1	Repetition Counter
Ad2..Ad6	Coincidence Code
Ad7..Ad13	First Pixel Code
Ad14	Cycle Bit
Ad15..Ad17	Data Source Code

Table2: Look-up Table Address Bits

15 nsec after addressing of the **LUT** the RAM data bits are valid. The allocation of these bits is listed in the following table:

LUT Data Bits	Message Bits	Parameter
LD0		Multiple Message Flag
LD1..LD8	MB0..MB7	TDI
LD9	MB8	nξ
LD10..LD19	MB9..MB18	ξ
LD20..LD27	MB19..MB26	dξ
LD28..LD35	MB27..MB34	dξξ
LD36..LD44	MB35..MB43	η
LD45..LD46	MB44..MB45	ω
LD47	MB46	all
	MB47..MB54	BX Bunch Number
LD48..LD49	MB55..MB56	ID
LD50..LD56	MB57..MB63	P
LD57	MB64	Flag
LD58..LD71	MB65..MB78	

Table3: Message Bit Allocation

The least significant **LUT** data bit is the Multiple Message Flag **MMF**. It increments the Repetition Counter and generates a second message for the same data set. That can be repeated up to three times.

With the exception of **MMF**, all **LUT** data bits (71) together with the 8 bit Bunch Number are stored as 79 bit message in the **Message FIFO**. The messages are generated and written to the FIFO with a frequency of 25 MHz.

Then a multiplexer **MUX** splits the message into four consecutive 20 bit words (see table in the appendix), which at a frequency of 100 MHz are written to the **Test FIFO** and depending on the message part **TDI** (Transfer Direction) and the content of the Port Registers **REGA..REGD** are transmitted through ports **PORTA..PORTD** to the connected TFU's. A certain output port is activated, if the logical AND between **TDI** (MB0..MB7) and the corresponding Port Register (PR0..PR7) is not zero:

$$(MB0*PR0 + MB1*PR1 + MB2*PR2 + MB3*PR3 + MB4*PR4 + MB5*PR5 + MB6*PR6 + MB7*PR7) \neq 0$$

Each output port can be stopped individually by the connected TFU and has it's own FIFO of 512 words length for data buffering.

Veto Number Distribution

The 6 least significant bits of a Veto Bunch Number, which is issued by the Electromagnetic Calorimeter of the HERA-B Detector in order to inhibit certain event trigger, are received by the Message Generator at front connector P6 (Veto Link) and is distributed via P3 to the linked Pretrigger Boards.

VME Access

Via VME access the Message Generator

- has to be set up by programming the Look-up Table and the Port Registers,
- has to be controlled by writing to the Command Register and monitoring the Status Register,
- can be tested by filling the DAV Test Register and Data Test Register and reading the Test FIFO.

The most important Command Register bits are **RUN** and **TSTM** (Test Mode). With **RUN=0** the Finite State Machines are at their ground state and data processing on the board is disabled. **RUN=1** enables the State Machines to react on flags and to process data. **TSTM=0** selects the Pretrigger Boards as data source, while for **TSTM=1** the DAV Test Register emulates Data Available Flags and the Data Test Register provides the data set.

The Status Register contains two status bits of the Test FIFO and 8 Handshake Error Flags, one for each possible data source. Such a flag is set, if the **DAV** bit of a selected **PB** has not been released, when the corresponding Data Accepted Flag goes false. That is a violation of the Handshake Protocol, indicating that there might be a problem with the board. A logical OR of all Handshake Error Flags can cause a VME interrupt, if it is enabled in the Command Register.

A VME access to the Look-up Table is possible only with **RUN=0**. Then an 18 bit Address Counter controls the **LUT** address bits. Therefore at first the Address Counter has to be set up to the desired start value. Then the data can be written or read in parts of 16 bit words. The Address Counter is automatically incremented after an access to the 16 most significant data bits.

A stand-alone board test can be performed by setting **RUN=0** and writing an 8 bit value to the DAV Test Register and a 16 bit value to the Data Test Register. Setting **TSTM=1** and then **RUN=1** starts the State Machines. The **DAV** test pattern is loaded to an encoder and the DAV Test Register is cleared. Then for each true **DAV** flag a data set is processed. The resulting messages can be read back from the Test FIFO and compared with the expected values.

VME Instructions

The VME Interface of the board supports **Short Supervisory Access** and **Short Non Privileged Access** (Address Modifier \$29 and \$2D). The address lines are completely decoded. The six most significant bits **A15..A10** are occupied by the Board Address BAD, which is determined by the slot location. The remaining 9 bits **A9..A1** are forming the address space of the board.

The following table provides a list of all instructions implemented. The second column gives the instruction address (hexadecimal notation), which has to be added to the Base Address of the board.

The Base Address can be calculated from the Board Address BAD by means of the following formula:

$$\text{Base (Byte) Address} = \text{BAD} * \$400$$

The complete instruction (byte) address then is given by:

$$\text{Instruct. Address} = \text{Base Address} + \text{Ad},$$

where **Ad** is given by the following table:

Instruction	Ad	A9..A7	A6	A5	A4	A3	A2	A1	Acc
General Clear	0	0	0	0	0	0	0	0	write
Read Status Register	0	0	0	0	0	0	0	0	read
Write Command Register	2	0	0	0	0	0	0	1	write
Read Command Register	2	0	0	0	0	0	0	1	read
Clear Interrupt Flag	4	0	0	0	0	0	1	0	write
Write DAV Test Register	6	0	0	0	0	0	1	1	write
Read DAV Test Register	6	0	0	0	0	0	1	1	read
Write Data Test Register low	10	0	0	0	1	0	0	0	write
Read Data Test Register low	10	0	0	0	1	0	0	0	read
Write Data Test Register high	12	0	0	0	1	0	0	1	write
Read Data Test Register high	12	0	0	0	1	0	0	1	read
Write LUT Address Counter low	14	0	0	0	1	0	1	0	write
Read LUT Address Counter low	14	0	0	0	1	0	1	0	read
Write LUT Address Counter high	16	0	0	0	1	0	1	1	write
Read LUT Address Counter high	16	0	0	0	1	0	1	1	read
Reset LUT Address Counter	18	0	0	0	1	1	0	0	write
Write LUT Bits 0..15	20	0	0	1	0	0	0	0	write
Read LUT Bits 0..15	20	0	0	1	0	0	0	0	read
Write LUT Bits 16..31	22	0	0	1	0	0	0	1	write
Read LUT Bits 16..31	22	0	0	1	0	0	0	1	read
Write LUT Bits 32..47	24	0	0	1	0	0	1	0	write
Read LUT Bits 32..47	24	0	0	1	0	0	1	0	read
Write LUT Bits 48..63	26	0	0	1	0	0	1	1	write
Read LUT Bits 48..63	26	0	0	1	0	0	1	1	read
Write LUT Bits 64..79 & incr. Address Counter	28	0	0	1	0	1	0	0	write
Read LUT Bits 64..79 & incr. Address Counter	28	0	0	1	0	1	0	0	read
Clear Test FIFO	30	0	0	1	1	0	0	0	write
Read Test FIFO low	30	0	0	1	1	0	0	0	read
Read Test FIFO high	32	0	0	1	1	0	0	1	read
Write Port Register A	40	0	1	0	0	0	0	0	write
Read Port Register A	40	0	1	0	0	0	0	0	read
Write Port Register B	42	0	1	0	0	0	0	1	write
Read Port Register B	42	0	1	0	0	0	0	1	read
Write Port Register C	44	0	1	0	0	0	1	0	write
Read Port Register C	44	0	1	0	0	0	1	0	read
Write Port Register D	46	0	1	0	0	0	1	1	write
Read Port Register D	46	0	1	0	0	0	1	1	read

The instructions are shortly described in the following:

General Clear

This instruction resets the State Machines on the board to a defined Ground State and clears the Command Register.

Read Status Register

HSE7	HSE6	HSE5	HSE4	HSE3	HSE2	HSE1	HSE0	INT	0	0	0	0	0	TFNF	TFNE
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

- TFNE:** 0: Test FIFO is empty
1: Test FIFO is not empty
- TFNF:** 0: Test FIFO is full
1: Test FIFO is not full
- INT:** Interrupt Request Flag
It is set, when the logical equation
(HSE0+ HSE1+ HSE2+ HSE3+ HSE4+ HSE5+ HSE6+ HSE7)&EIN1 + TFNE&IEN2
becomes true.
IEN1, IEN2 are the Interrupt Mask Bits, set in the Command Register.
The Flag is reset by the VME Command “Clear Interrupt Flag”.
- HSE0..HSE7:** Handshake Error Flags.
The Flag **HSEn** indicates, that the Data Available Flag **DAVn** of the data source is not reset,
when the Message Generator releases the Data Accepted Flag **DACn**.
The flags are reset by the VME Command “Clear Interrupt Flag”.

Write/Read Command Register

IL3	IL2	IL1	0	0	0	IEN2	IEN1	IV3	IV2	IV1	IV0	STBY	ENDB	TSTM	RUN
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

- RUN:** 0: Data Processing on the board is disabled, the State Machines are in Halt mode.
One can write to or read from the Look-up Table LUT.
The Output FIFO's are cleared.
1: Data Processing on the board is enabled, the State Machines are operational.
LUT address inputs are controlled by the Pretrigger bits.
- TSTM:** 0: Normal Data Acquisition Mode.
1: Test Mode.
A test cycle is initiated by at first setting **RUN=0**, then setting at least one bit in the DAV Test Register, writing a pattern to the Data Test Register and finally setting **TSTM=1** and then **RUN=1**.
- ENDB:** 0: Double Message Mode is disabled.
1: Double Message Mode is enabled.
- STBY:** 0: 100 MHz oscillator is working.
1: 100 MHz oscillator is in Stand-by Mode.
- IV0..IV3:** Lower Nibble of Interrupt Vector. The upper Nibble is fixed to \$F.
- IEN1:** 0: Handshake Error as interrupt source is disabled.
1: Handshake Error as interrupt source is enabled.
- IEN2:** 0: TFNE Flag as interrupt source is disabled.
1: TFNE Flag as interrupt source is enabled.
- IL1..IL3:** Interrupt Level.

Clear Interrupt Flag

This instruction resets all Handshake Error Flags and the Interrupt Flag.

Write/Read DAV Test Register

x	x	x	x	x	x	x	x	DAV7	DAV6	DAV5	DAV4	DAV3	DAV2	DAV1	DAV0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

- DAV0..DAV7:** 0: Data Available Flag **DAVn** is not set for Test.
1: Data Available Flag **DAVn** is not set for Test.

One should set **RUN=0** (see Command Register) before writing to the Test Register. The test is started by at first setting **TSTM=1** and then **RUN=1**.

Write/Read Data Test Register low

RSF4	RSF3	RSF2	RSF1	RSF0	PIB4	PIB3	PIB2	PIB1	PBC0	PIC5	PIC4	PIC3	PIC2	PIC1	PIC0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Write/Read Data Test Register high

x	x	x	x	x	BxN8	BxN7	BxN6	BxN5	BxN4	BxN3	BxN2	BxN1	CBIT	RSF1	RSF0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

PIC0..PIC5: Pads of 3. Detector Layer.

PIB0..PIB4: Pads of 2. Detector Layer.

RSF0..RSF6: Road Starting Flags.

CBIT: Cycle Bit.

BxN1..BxN8: Bunch Number Bits.

One should set **RUN=0** (see Command Register) before writing to the Test Register. The test is started by at first setting **TSTM=1** and then **RUN=1**.

Write/Read LUT Address Counter low

AD15	AD14	AD13	AD12	AD11	AD10	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Write/Read LUT Address Counter high

x	x	x	x	x	x	x	x	x	x	x	x	x	x	AD17	AD16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

AD0..AD17: LUT Address Counter Bits.

The LUT Address Counter is used to program the Look-up Table and to read it back. Before programming or reading the LUT, it's start address has to be set either by clearing the Address Counter (see below) or by writing a certain value to the Address Counter. Afterwards by writing or reading the most significant LUT bits the Address Counter is incremented automatically.

Reset LUT Address Counter

This dateless instruction resets the LUT Address Counter.

Write/Read LUT Bits 0..15

Data Bits 0..15 are written to (read from) the LUT at the address, given by the LUT Address Counter.

Write/Read LUT Bits 16..31

Data Bits 16..31 are written to (read from) the LUT at the address, given by the LUT Address Counter.

Write/Read LUT Bits 32..47

Data Bits 32..47 are written to (read from) the LUT at the address, given by the LUT Address Counter.

Write/Read LUT Bits 48..63

Data Bits 48..63 are written to (read from) the LUT at the address, given by the LUT Address Counter.

Write/Read LUT Bits 64..71

Data Bits 64..71 are written to (read from) the LUT at the address, given by the LUT Address Counter. Afterwards the LUT Address Counter is incremented. So that command has to be the last one in a writing (reading) cycle.

Read Test FIFO low

TF15	TF14	TF13	TF12	TF11	TF10	TF9	TF8	TF7	TF6	TF5	TF4	TF3	TF2	TF1	TF0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Read Test FIFO high

x	x	x	x	x	x	x	x	x	x	x	VAL	TF19	TF18	TF17	TF16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

TF0..TF19: Test FIFO Bits.

VAL: Valid Bit. VAL=1 flags the first word of a message.

A complete read cycle should start with “Read Test FIFO low” and end with “Read Test FIFO high”, because the last command moves the next word to the FIFO output.

Clear Test FIFO

That dateless instruction clears the Test FIFO. If that command is issued, while data are sent to the TFU, the second, third and fourth word of one message is missing.

Write/Read Port Register A

x	x	x	x	x	x	x	x	PRA7	PRA6	PRA5	PRA4	PRA3	PRA2	PRA1	PRA0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

PRA0..PRA7: Mask Bits for Port A

The Register is cleared by the VME instruction “General Clear”.

A message is sent via Port A, if the logical equation

$PRA0*MB0+PRA1*MB1+PRA2*MB2+PRA3*MB3+PRA4*MB4+PRA5*MB5+PRA6*MB6+PRA7*MB7$ is true.

Write/Read Port Register B

x	x	x	x	x	x	x	x	PRB7	PRB6	PRB5	PRB4	PRB3	PRB2	PRB1	PRB0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

PRB0..PRB7: Mask Bits for Port B

The Register is cleared by the VME instruction “General Clear”.

A message is sent via Port A, if the logical equation

$PRB0*MB0+PRB1*MB1+PRB2*MB2+PRB3*MB3+PRB4*MB4+PRB5*MB5+PRB6*MB6+PRB7*MB7$ is true

Write/Read Port Register C

x	x	x	x	x	x	x	x	PRC7	PRC6	PRC5	PRC4	PRC3	PRC2	PRC1	PRC0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

PRC0..PRC7: Mask Bits for Port C

The Register is cleared by the VME instruction “General Clear”.

A message is sent via Port A, if the logical equation

$PRC0*MB0+PRC1*MB1+PRC2*MB2+PRC3*MB3+PRC4*MB4+PRC5*MB5+PRC6*MB6+PRC7*MB7$ is true

Write/Read Port Register D

x	x	x	x	x	x	x	x	PRD7	PRD6	PRD5	PRD4	PRD3	PRD2	PRD1	PRD0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

PRD0..PRD7: Mask Bits for Port D

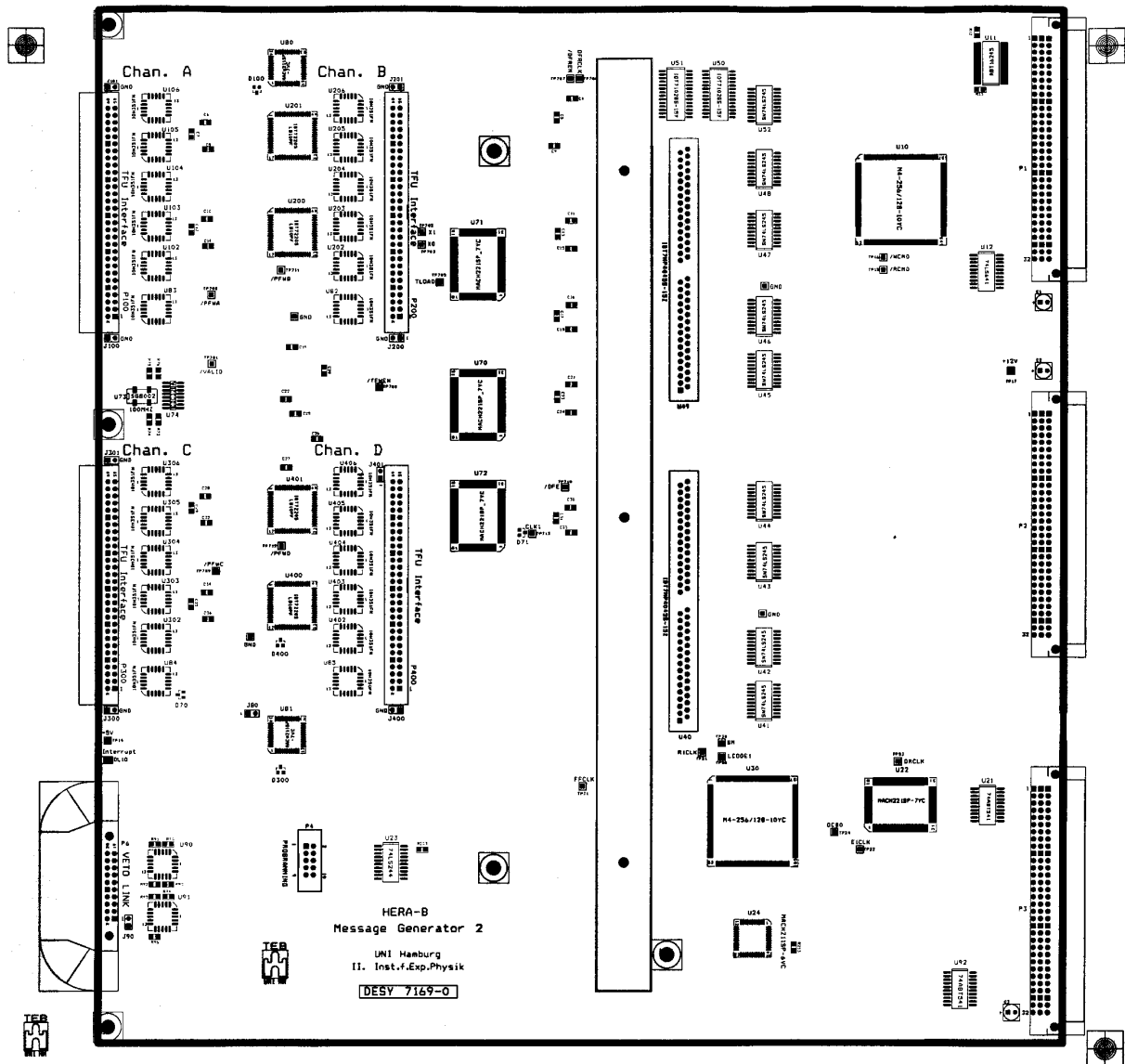
The Register is cleared by the VME instruction “General Clear”.

A message is sent via Port A, if the logical equation

$PRD0*MB0+PRD1*MB1+PRD2*MB2+PRD3*MB3+PRD4*MB4+PRD5*MB5+PRD6*MB6+PRD7*MB7$
is true

Appendix

1) Board Layout



DESY 7169-0 StMask 1

3) Backplane Connector P2

Standard P2/J2 VME Connector, Column b not used

Pin	a	b	b
1	BAD0		BAD1
2	BAD2		BAD3
3	BAD4		BAD5
4	GND		GND
5			
6			
7			
8			
9			
10			
11	GND		GND
12	GND		GND
13	GND		GND
14	GND		GND
15			
16			
17			
18			
19			
20			
21			
22			
23			
24			
25			
26	GND		GND
27	GND		GND
28			
29			
30			
31			
32			

BAD5..BAD0: Board Address Bits, corresponding to **A15..A10**

4) Backplane Connector P3

96 Pin Connector, connected to Backplane of type VME-J1

Pin	a	b	b
1	/DAV1		/DAC1
2	/DAV2		/DAC2
3	/DAV3		/DAC3
4	/DAV4		/DAC4
5	/DAV5		/DAC5
6	/DAV6		/DAC6
7	/DAV7		/DAC7
8	/DAV8		/DAC8
9	GND		GND
10	BCLK		
11	GND		
12	/DAV9		
13	/DAC9		BXN0
14	BXN1		BXN2
15	GND		BXN3
16		BXN4	BXN5
17	GND	BXN6	BXN7
18	BXN8	RSF0	RSF1
19	GND	RSF2	RSF3
20	RSF4	GND	RSF5
21		spare	RSF6
22		spare	PB0
23	PB1	GND	PB2
24	PB3		PB4
25	PC0		PC1
26	PC2		PC3
27	PC4		PC5
28	V0		V1
29	V2		V3
30	V4		VVAL
31	-12V		+12V
32	+5V	+5V	+5V

/DAV9../DAV1: Data Available Flags
/DAC9../DAC1: Data Accepted Flags
BXN8..BXN1,BXN0: Bunch Number, Cycle Bit
RSF6..RSF0: Road Starting Flags
PIB4..PIB0: Pixel of Layer 2
PIC5..PIC0: Pixel of Layer 3
V4..V0,VVAL: Veto Number Bits, Veto Number Valid Flag
BCLK: Bunch Clock

5) TFU Connector P5

Pin		Pin	
1	GND	31	/TD7
2	GND	32	TD7
3	CLK	33	TD8
4	/CLK	34	/TD8
5	GND	35	TD9
6	GND	36	/TD9
7	/VAL	37	/TD10
8	VAL	38	TD10
9	+12V	39	/TD11
10	+12V	40	TD11
11	/STOP	41	TD12
12	STOP	42	/TD12
13	+12V	43	TD13
14	+12V	44	/TD13
15	+12V	45	/TD14
16	+12V	46	TD14
17	TD0	47	/TD15
18	/TD0	48	TD15
19	TD1	49	TD16
20	/TD1	50	/TD16
21	/TD2	51	TD17
22	TD2	52	/TD17
23	/TD3	53	/TD18
24	TD3	54	TD18
25	TD4	55	/TD19
26	/TD4	56	TD19
27	TD5	57	GND
28	/TD5	58	GND
29	/TD6	59	GND
30	TD6	60	GND

TD0..TD19: Transfer Data Bits
CLK: 100 MHz Clock
VAL: Data Valid Bit
STOP: Stop Flag, provided by TFU

6) VETO Number Connector P6

Pin		Pin	
1	GND	11	/V4
2	GND	12	V4
3	/BCLK	13	/V3
4	BCLK	14	V3
5	GND	15	/V2
6	GND	16	V2
7	/VVAL	17	/V1
8	VVAL	18	V1
9	GND	19	/V0
10	GND	20	V0

V0..V5:

BCLK:

VVAL:

Bunch Number Bits

Bunch Clock

Veto Valid Bit

7) Message Bit Partitioning

Bit #	Word 0		Word 1		Word 2		Word 3	
0	LD1		LD2		LD3		LD4	
1	LD5		LD6		LD7		LD8	
2	LD9		LD10		LD11		LD12	
3	LD13		LD14		LD15		LD16	
4	LD17		LD18		LD19		LD20	
5	LD21		LD22		LD23		LD24	
6	LD25		LD26		LD27		LD28	
7	LD29		LD30		LD31		LD32	
8	LD33		LD34		LD35		LD36	
9	LD37		LD38		LD39		LD40	
10	LD41		LD42		LD43		LD44	
11	LD45		LD46		LD47		BXN0	
12	BXN1		BXN2		BXN3		BXN4	
13	BXN5		BXN6		BXN7		LD48	
14	LD49		LD50		LD51		LD52	
15	LD53		LD54		LD55		LD56	
16	LD57		LD58		LD59		LD60	
17	LD61		LD62		LD63		LD64	
18	LD65		LD66		LD67		LD68	
19	LD69		LD70		LD71			

LD1..LD71: LUT Data Bits
BXN0..BXN7: Bunch Number Bits