

VME Event Generator (VME-EVG-230)

Technical Reference

Firmware Version E402

Contents

Introduction	3
Event Stream Details	3
Event Codes.....	3
Distributed Bus and Data Transmission	4
Event Sources	4
Trigger Events	4
Upstream Events.....	5
Event Sequencer	6
Distributed Bus.....	9
Timestamping Inputs	10
Multiplexed Counters	10
Configurable Size Data Buffer	11
AC Line Synchronisation	13
Front Panel Inputs and Universal I/O Inputs	13
Event Clock	13
RF Clock and Event Clock.....	14
Fractional Synthesiser	14
Non-Volatile Storage for Frequency Configuration.....	15
Connections	15
Front Panel Connections.....	15
VME P2 User I/O Pin Configuration	16
Programming Details.....	17
CR/CSR Support	17
Event Generator Function 0 Registers.....	18
Register Map	19
Network Interface.....	28
Assigning an IP Address to the Module.....	28
Using Telnet to Configure Module.....	28
Boot Configuration (command b)	29
Memory dump (command d).....	29
Memory modify (commands d and m).....	30
Upgrading IP2022 Microprocessor Software (command u).....	30
Linux	30
Windows.....	30
Upgrading FPGA Configuration File	31

Linux	31
Windows.....	31
Linux	31
Windows.....	31
UDP Remote Programming Protocol	32
Read Access (Type 0x01).....	32
Write Access (Type 0x02).....	33

Introduction

The Event Generator is responsible of creating and sending out timing events to an array of Event Receivers. High configurability makes it feasible to build a whole timing system with a single Event Generator without external counters etc.

Events are sent out by the event generator as event frames (words) which consist of an eight bit event code and an eight bit distributed bus data byte. The event transfer rate is derived from an external RF clock or optionally an on-board clock generator. The optical event stream transmitted by the Event Generator is phase locked to the clock reference.

There are several sources of events: trigger events, sequence events, software events and events received from an upstream Event Generator. Events from different sources have different priority which is resolved in a priority encoder.

In addition to events the Event Generator enables the distribution of eight simultaneous signals sampled with the event clock rate, the distributed bus. Distributed bus signals may be provided externally or generated on-board by programmable multiplexed counters.

Event Stream Details

The structure of the event stream is described to help understand the functioning of the event system. The event stream should be considered as a continuous flow of event frames which consist of two bytes, the event code and distributed bus data byte.

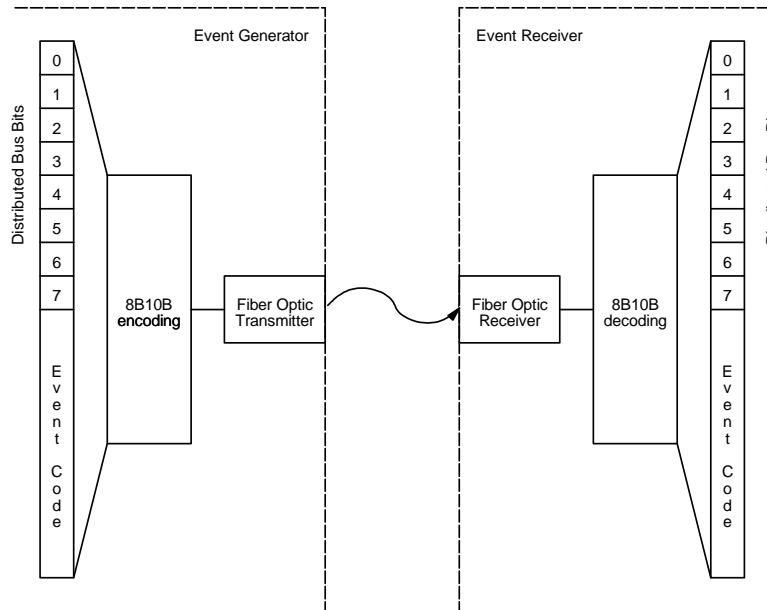


Figure 1: Event Frame

Event Codes

There are 256 event codes from which a few have special functions. The special function event codes are listed below. Only one event code may be transferred at a time. If there is no event code to be transferred, the null event code (0x00) is transmitted. Every now and then a special 8B10B

character K28.5 is transmitted instead of the null event code. The K28.5 comma character is transmitted to allow the event receivers to synchronise on the correct word boundary in the serial bit stream.

Event Code	Code Name	EVG Function	EVR Function
0x00	Null Event Code	-	-
0x01 – 0x6F	-	User Defined	User Defined
0x70	Seconds ‘0’	-	Shift in ‘0’ to LSB of Seconds Shift Register
0x71	Seconds ‘1’	-	Shift in ‘1’ to LSB of Seconds Shift Register
0x72 – 0x79	-	User Defined	User Defined
0x7A	Heartbeat	-	Reset Heartbeat Monitor
0x7B	Synchronise Prescalers	-	Synchronise Prescaler Outputs
0x7C	Timestamp Counter Increment	-	Increment Timestamp Counter
0x7D	Timestamp Counter Reset	-	Reset Timestamp Counter
0x7F	End of Sequence	Stop Sequence	-

Distributed Bus and Data Transmission

The distributed bus allows transmission of eight simultaneous signals with the event clock rate time resolution (8 ns at 125 MHz event clock rate). The source for distributed bus signals may come from an external source or the signals may be generated with programmable multiplexed counters (MXC) inside the event generator. The distributed bus signals may be programmed to be available as hardware outputs on the event receiver.

In latest firmware versions the distributed bus bandwidth may be shared by transmission of a configurable size data buffer to up to 2 kbytes. When data transmission is enabled the distributed bus bandwidth is halved. The remaining bandwidth is reserved for transmitting data with a speed up to 62.5 Mbytes/s (event clock rate divided by two).

Event Sources

Trigger Events

There are eight trigger event sources that send out an event code on a stimulus. Each trigger event has its own programmable event code register and various enable bits. The event code transmitted is determined by contents of the corresponding event code register. The stimulus may be a detected rising edge on an external signal or a rising edge of a multiplexed counter output.

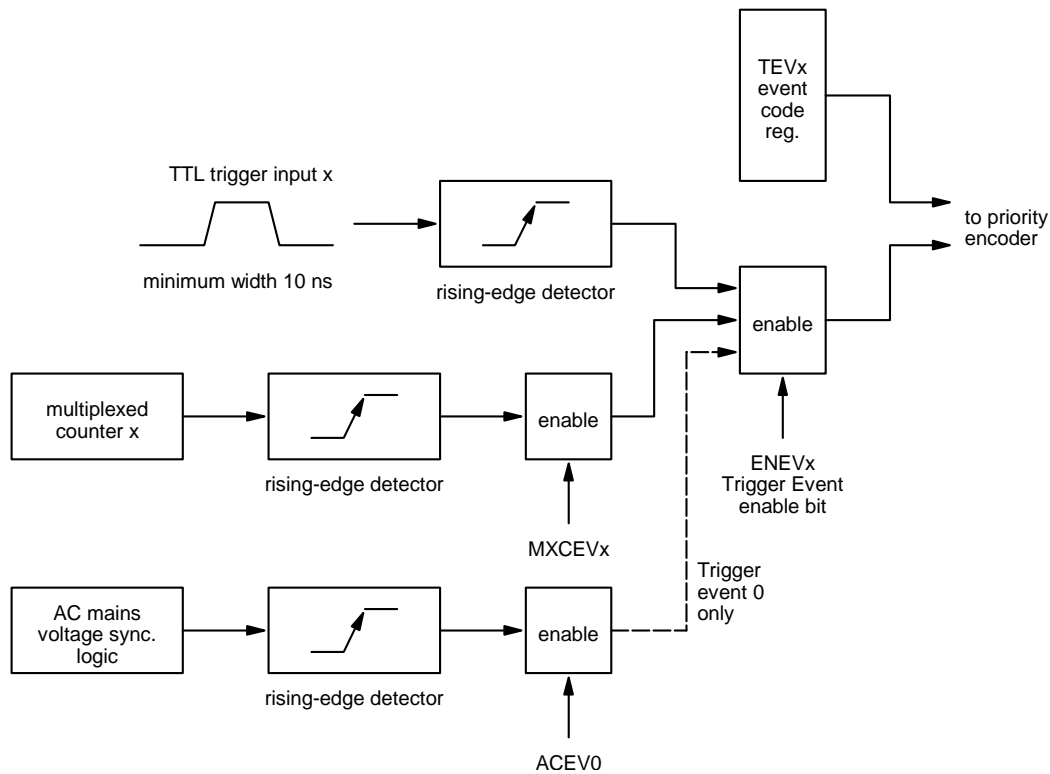


Figure 2: Trigger Events

Trigger Event 0 has also the option of being triggered by a rising edge of the AC mains voltage synchronization logic output signal.

The external input accepts TTL level signals. The input logic is edge sensitive and the signals are synchronized internally to the event clock.

Upstream Events

Event Generators may be cascaded. The event generator receiver includes a first-in-first-out (FIFO) memory to synchronize incoming events which may be synchronized to a clock unrelated to the event clock. Usually there are no events in the FIFO. An event code from an upstream EVG is transmitted as soon as there is no other event code to be transmitted.

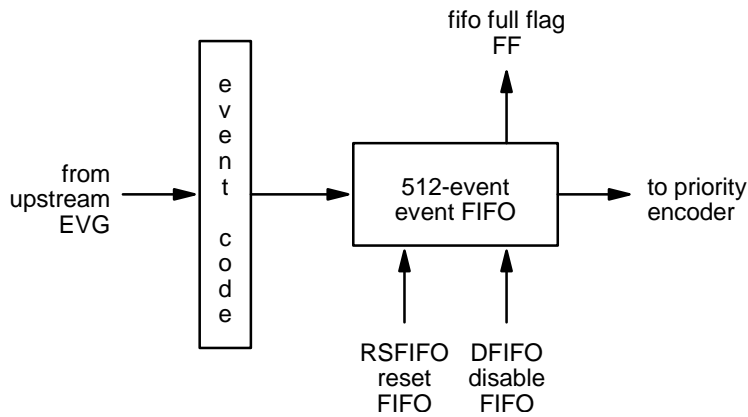


Figure 3: Upstream Event FIFO

Event Sequencer

Event sequencers provide a method of transmitting or playing back sequences of events stored in random access memory with defined timing. In the event generator there are two event sequencers. The 8-bit event codes are stored in a RAM table each attached with a 32-bit timestamp relative to the start of sequence. Both sequencers can hold up to 2048 event code – timestamp pairs.

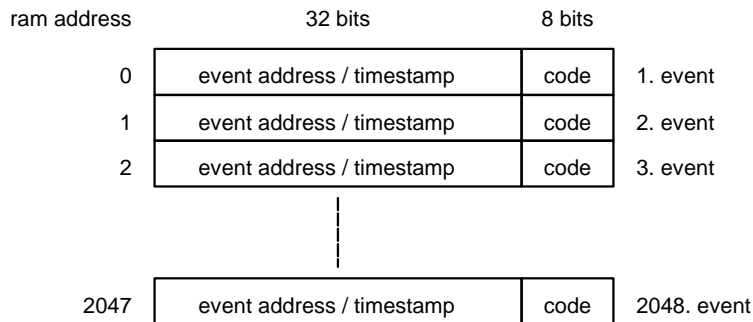


Figure 4: Sequencer RAM Structure

The contents of a sequencer RAM may be altered at any time, however, it is recommended only to modify RAM contents when the RAM is disabled. The RAMs are addressed indirectly: there are separate registers for address and data.

The sequencer clock may be selected to be applied externally or down converted directly from the event clock with a 16-bit prescaler. The clock is selected by a 16-bit prescaler. When the prescaler value is zero an externally supplied clock is used. The sequencers are capable of operating at full event clock speed to up to 125 MHz.

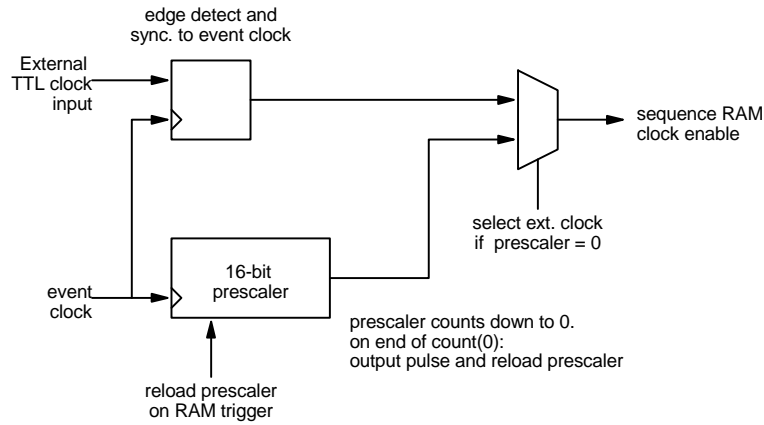


Figure 5: Sequencer Clock

Both sequencers have their own prescalers, but by setting the CMODE configuration bit sequence RAM 2 may be forced to use the same clock, trigger and reset signals as sequence RAM 1.

The Sequencers may be triggered from several sources including hardware triggering from external TTL input, software triggering by VME access, triggering on a multiplexed counter output or AC mains voltage synchronization logic output.

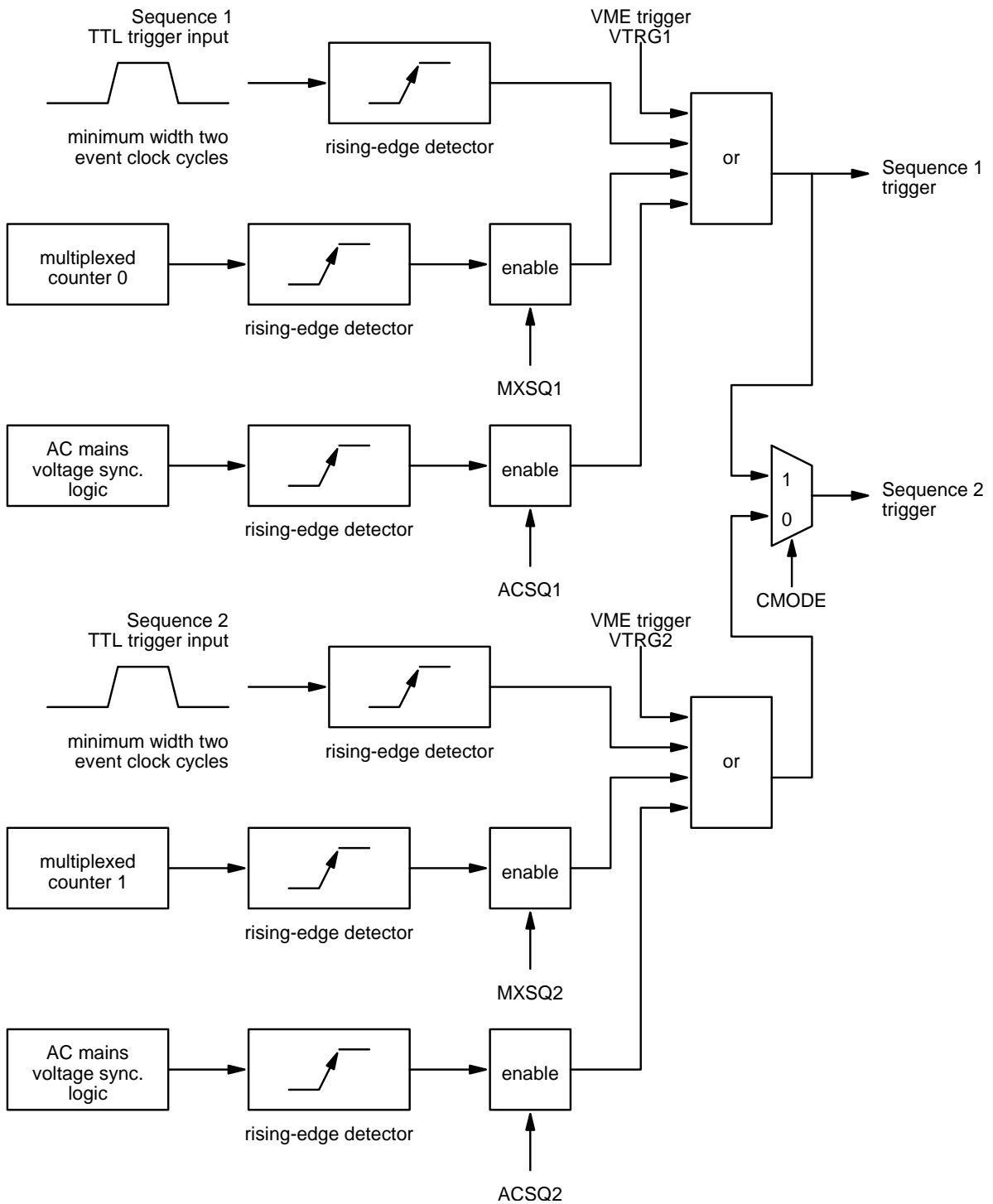


Figure 6: Sequencer Triggering

The sequencers are enabled by setting bit ENSQ_x in the Event Enable Register. The RAMs may be disabled any time by clearing the corresponding bits. Disabling sequence RAMs does not reset the RAM address and timestamp registers. By setting the bit SEQ_x in the Control Register the sequencer is both disabled and the RAM address and timestamp register is reset.

When the sequencer is triggered the internal event address counters starts counting. The counter value is compared to the event address of the next event in the RAM table. When the counter value matches the timestamp in the RAM table, the attached event code is transmitted. The time offset between two consecutive events in the RAM is allowed to be 1 to 2^{32} sequence clock cycles i.e. the internal event address counter rolls over when to 0 when 0xffffffff is reached.

There are two special event codes which are not transmitted, the null event code 0x00 and end sequence code 0x7f. The null event code may be used if the time between two consecutive events should exceed 2^{32} event clock cycles. The end sequence code resets the sequencer RAM table address and timestamp register and depending on configuration bits, disables the sequencer (single sequence, SSEQx=1) or restarts the sequence either immediately (recycle sequence, RCYLx=1) or waits for a new trigger (RCYLx=0).

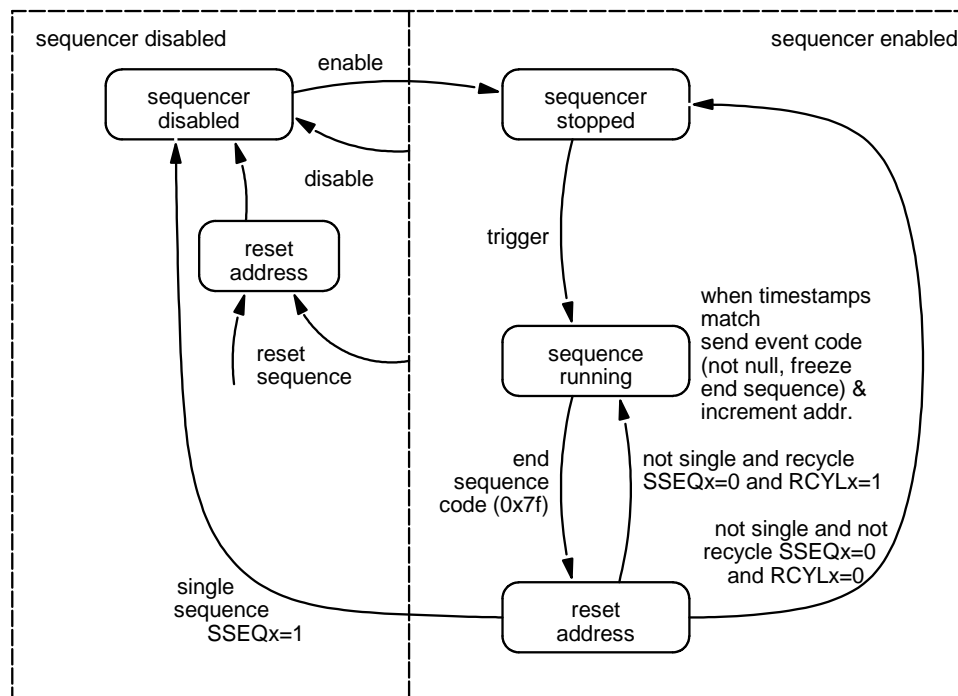


Figure 7: Sequencer Control

Distributed Bus

The bits of the distributed bus are sampled at the event rate from external signals; alternatively the distributed bus signals may be generated by multiplexed counter outputs. If there is an upstream EVG, the state of all distributed bus bits may be forwarded by the EVG.

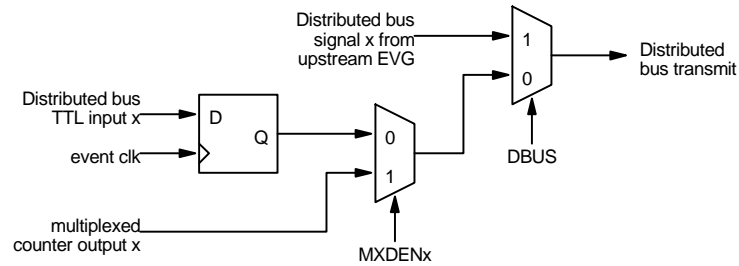


Figure 8: Distributed Bus

Timestamping Inputs

Starting from firmware version E306 a few distributed bus input signals have dual function: transition board input DBUS5-7 can be used to generate special event codes controlling the timestamping in Event Receivers.

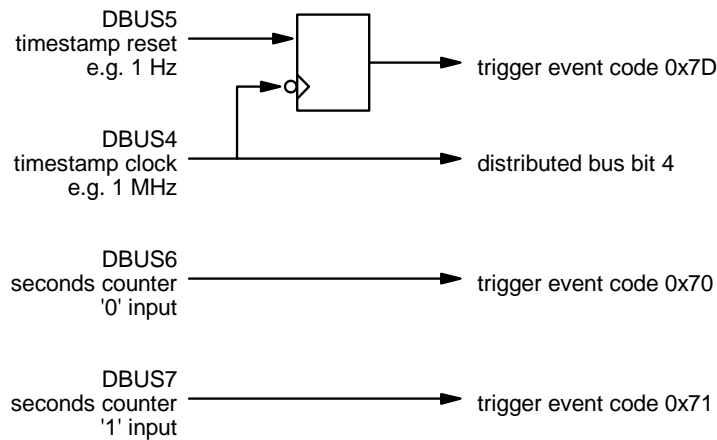


Figure 9: Timestamping Inputs

The two clocks, timestamp clock and timestamp reset clock, are assumed to be rising edge aligned. In the EVG the timestamp reset clock is sampled with the falling edge of the timestamp clock. This is to prevent a race condition between the reset and clock signals. In the EVR the reset is synchronised with the timestamp clock.

The two seconds counter events are used to shift in a 32-bit seconds value between consecutive timestamp reset events. In the EVR the value of the seconds shift register is transferred to the seconds counter at the same time the higher running part of the timestamp counter is reset.

The distributed bus event inputs can be enabled independently through the distributed bus event enable register. The events generated through these distributed bus input ports are given lowest priority.

Multiplexed Counters

Eight 32-bit multiplexed counters generate clock signals with programmable frequencies from event clock/ $2^{32}-1$ to event clock/2. Even divisors create 50% duty cycle signals. The counter outputs may be programmed to trigger events, drive distributed bus signals and trigger sequence RAMs. The output of multiplexed counter 7 is hard-wired to the mains voltage synchronization logic.

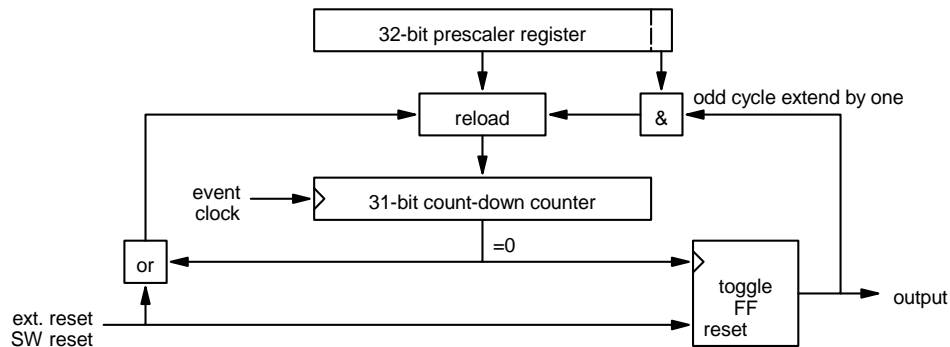


Figure 10: Multiplexed Counter

Each multiplexed counter consists of a 32-bit prescaler register and a 31-bit count-down counter which runs at the event clock rate. When count reaches zero, the output of a toggle flip-flop changes and the counter is reloaded from the prescaler register. If the least significant bit of the prescaler register is one, all odd cycles are extended by one clock cycle to support odd dividers.

Prescaler value	Duty Cycle	Frequency at 125 MHz Event Clock
0, 1 not allowed	undefined	undefined
2	50/50	62.5 MHz
3	33/66	41.7 MHz
4	50/50	31.25 MHz
5	40/60	25 MHz
...
$2^{32} - 1$	approx. 50/50	0.029 Hz

The multiplexed counters may be reset by software or hardware input. The reset state is defined by the multiplexed counter polarity register.

Configurable Size Data Buffer

Starting from firmware version E305 transmission of a configurable size data buffer over the event system link is possible. The buffer size can be programmed in four byte increments (long words) from 4 bytes to 2048 bytes.

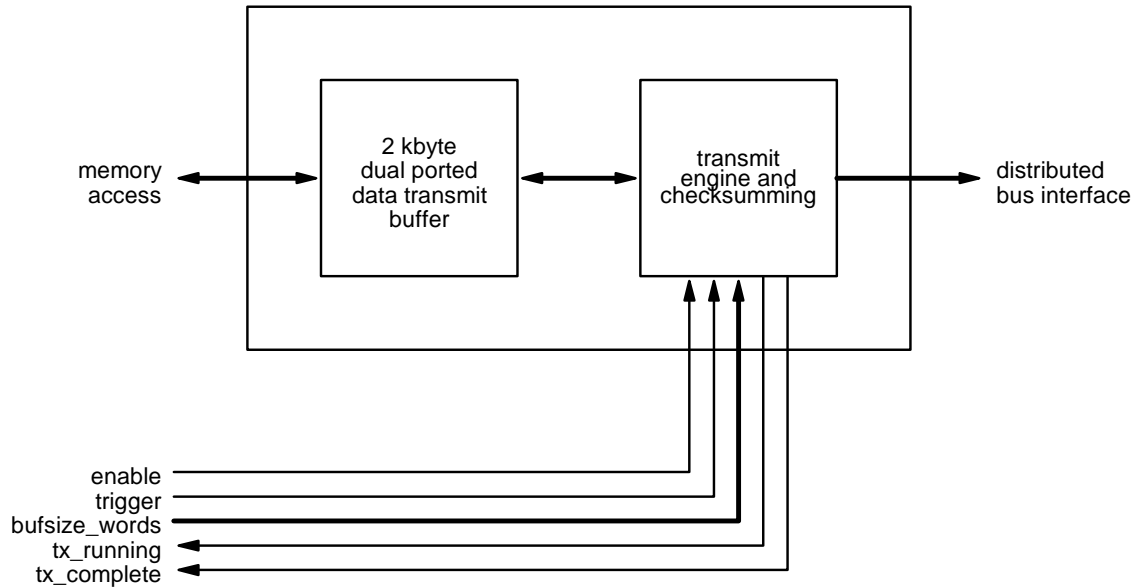


Figure 11: Configurable size transmit data buffer

When the EVG is configured for data transmission (*mode* = 1 in data buffer control register) the bandwidth of the distributed bus is shared with data transmission: half of the bandwidth remains for the distributed bus and the other half is reserved for data transmission.

The data to be transmitted is stored in a 2 kbyte dual-ported memory starting from the lowest address 0. This memory is directly accessible from VME. The transfer size is determined by *bufsize* register bits in four byte increments. The transmission is trigger by software. Two flags *tx_running* and *tx_complete* represent the status of transmission.

Transmission utilises two K-characters to mark the start and end of the data transfer payload, the protocol looks following:

Table 1: Data Transmission Protocol

8B10B-character	Description
K28.0	Start of data transfer
Dxx.x	1 st data byte (address 0)
Dxx.x	2 nd data byte (address 1)
Dxx.x	3 rd data byte (address 2)
Dxx.x	4 th data byte (address 3)
...	...
Dxx.x	n th data byte (address n-1)
K28.1	End of data
Dxx.x	Checksum (LSB)
Dxx.x	Checksum(MSB)

AC Line Synchronisation

The Event Generator provides synchronization to the mains voltage frequency or another external clock. The mains voltage frequency can be divided by an eight bit programmable divider. The output of the divider may be delayed by 0 to 25.5 ms by a phase shifter in 0.1 ms steps to be able to adjust the triggering position relative to mains voltage phase. After this the signal synchronized to the event clock or the output of multiplexed counter 7.

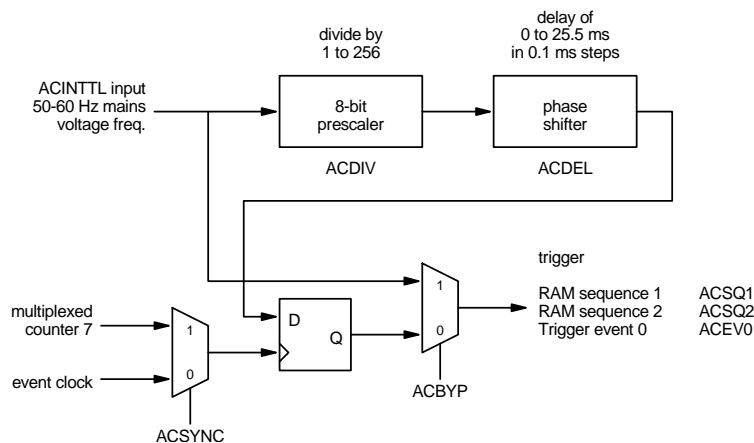


Figure 12: AC Input

The phase shifter operates with a clock of 1 MHz which introduces jitter. If the prescaler and phase shifter are not required this circuit may be bypassed. This also reduces jitter because the external trigger input is sampled directly with the event clock.

Front Panel Inputs and Universal I/O Inputs

The VME-EVG has two configurable front panel inputs and up to four Universal I/O inputs that can be used to trigger events or provide data on the distributed bus. Each of the inputs can be programmed to trigger any of the eight trigger events or drive any distributed bus bit. Two input can drive the same trigger however all enabled inputs for the same destination are internally combined using a logical or operation.

Event Clock

All operations on the event generator are synchronised to the event clock which is derived from an externally provided RF clock. For laboratory testing purposes an on-board fractional synthesiser may be used to deliver the event clock. The serial link bit rate is 20 times the event clock rate. The acceptable range for the event clock and bit rate is shown in the following table.

	Event Clock	Bit Rate
Minimum	50 MHz	1.0 Gb/s
Maximum	125 MHz	2.5 Gb/s

During operation the reference frequency should not be changed more than ± 100 ppm.

RF Clock and Event Clock

The event clock may be derived from an external RF clock signal. The front panel RF input is 50 ohm terminated and AC coupled to a LVPECL logic input, so either an ECL level clock signal or sine-wave signal with a level of maximum +10 dBm can be used.

Divider	RF Input Frequency	Event Clock	Bit Rate
÷ 1	50 MHz – 125 MHz	50 MHz – 125 MHz	1.0 Gb/s – 2.5 Gb/s
÷ 2	100 MHz – 250 MHz	50 MHz – 125 MHz	1.0 Gb/s – 2.5 Gb/s
÷ 3	150 MHz – 375 MHz	50 MHz – 125 MHz	1.0 Gb/s – 2.5 Gb/s
÷ 4	200 MHz – 500 MHz	50 MHz – 125 MHz	1.0 Gb/s – 2.5 Gb/s
÷ 5	250 MHz – 625 MHz	50 MHz – 125 MHz	1.0 Gb/s – 2.5 Gb/s
÷ 6	300 MHz – 900 MHz	50 MHz – 125 MHz	1.0 Gb/s – 2.5 Gb/s
÷ 7	350 MHz – 1025 MHz	50 MHz – 125 MHz	1.0 Gb/s – 2.5 Gb/s
÷ 8	400 MHz – 1 GHz	50 MHz – 125 MHz	1.0 Gb/s – 2.5 Gb/s
÷ 9	450 MHz – 1.125 GHz	50 MHz – 125 MHz	1.0 Gb/s – 2.5 Gb/s
÷ 10	500 MHz – 1.25 GHz	50 MHz – 125 MHz	1.0 Gb/s – 2.5 Gb/s
÷ 11	550 MHz – 1.375 GHz	50 MHz – 125 MHz	1.0 Gb/s – 2.5 Gb/s
÷ 12	600 MHz – 1.5 GHz	50 MHz – 125 MHz	1.0 Gb/s – 2.5 Gb/s
÷ 14	700 MHz – 1.6 GHz *)	50 MHz – 114 MHz	1.0 Gb/s – 2.285 Gb/s
÷ 15	750 MHz – 1.6 GHz *)	50 MHz – 106 MHz	1.0 Gb/s – 2.133 Gb/s
÷ 16	800 MHz – 1.6 GHz *)	50 MHz – 100 MHz	1.0 Gb/s – 2.000 Gb/s
÷ 17	850 MHz – 1.6 GHz *)	50 MHz – 94 MHz	1.0 Gb/s – 1.882 Gb/s
÷ 18	900 MHz – 1.6 GHz *)	50 MHz – 88 MHz	1.0 Gb/s – 1.777 Gb/s
÷ 19	950 MHz – 1.6 GHz *)	50 MHz – 84 MHz	1.0 Gb/s – 1.684 Gb/s
÷ 20	1.0 GHz – 1.6 GHz *)	50 MHz – 80 MHz	1.0 Gb/s – 1.600 Gb/s
÷ 21	1.05 GHz – 1.6 GHz *)	50 MHz – 76 MHz	1.0 Gb/s – 1.523 Gb/s
÷ 22	1.1 GHz – 1.6 GHz *)	50 MHz – 72 MHz	1.0 Gb/s – 1.454 Gb/s
÷ 23	1.15 GHz – 1.6 GHz *)	50 MHz – 69 MHz	1.0 Gb/s – 1.391 Gb/s
÷ 24	1.2 GHz – 1.6 GHz *)	50 MHz – 66 MHz	1.0 Gb/s – 1.333 Gb/s
÷ 25	1.25 GHz – 1.6 GHz *)	50 MHz – 64 MHz	1.0 Gb/s – 1.280 Gb/s
÷ 26	1.3 GHz – 1.6 GHz *)	50 MHz – 61 MHz	1.0 Gb/s – 1.230 Gb/s
÷ 27	1.35 GHz – 1.6 GHz *)	50 MHz – 59 MHz	1.0 Gb/s – 1.185 Gb/s
÷ 28	1.4 GHz – 1.6 GHz *)	50 MHz – 57 MHz	1.0 Gb/s – 1.142 Gb/s
÷ 29	1.45 GHz – 1.6 GHz *)	50 MHz – 55 MHz	1.0 Gb/s – 1.103 Gb/s
÷ 30	1.5 GHz – 1.6 GHz *)	50 MHz – 53 MHz	1.0 Gb/s – 1.066 Gb/s
÷ 31	1.55 GHz – 1.6 GHz *)	50 MHz – 51 MHz	1.0 Gb/s – 1.032 Gb/s
÷ 32	1.6 GHz *)	50 MHz	1.0 Gb/s

*) Range limited by AD9515 maximum input frequency of 1.6 GHz

Fractional Synthesiser

For laboratory testing purposes the event clock may be generated on-board the event generator using a fractional synthesiser. A Micrel (<http://www.micrel.com>) SY87739L Protocol Transparent Fractional-N Synthesiser with a reference clock of 24 MHz is used. The following table lists programming bit patterns for a few frequencies.

Event Rate	Configuration Bit	Reference Output	Precision
------------	-------------------	------------------	-----------

	Pattern		(theoretical)
499.8 MHz/4 = 124.95 MHz	0x00FE816D	124.95 MHz	0
499.654 MHz/4 = 124.9135 MHz	0x0C928166	124.907 MHz	-52 ppm
476 MHz/4 = 119 MHz	0x018741AD	119 MHz	0
106.25 MHz (fibre channel)	0x049E81AD	106.25 MHz	0
499.8 MHz/5 = 99.96 MHz	0x025B41ED	99.956 MHz	-40 ppm
50 MHz	0x009743AD	50.0 MHz	0
499.8 MHz/10 = 49.98 MHz	0x025B43AD	49.978 MHz	-40 ppm
499.654 MHz/4 = 124.9135 MHz	0x0C928166	124.907 MHz	-52 ppm
50 MHz	0x009743AD	50.0 MHz	0

Non-Volatile Storage for Frequency Configuration

The reference clock setting etc. initialisation values are stored in non-volatile memory inside the IP2022 microcontroller.

Connections

Front Panel Connections

The front panel of the Event Generator is shown in Figure 13.

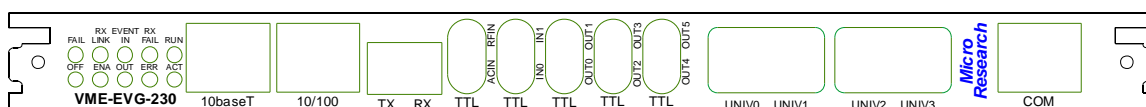


Figure 13: Event Generator Front Panel

The front panel of the Event Generator includes the following connections and status leds:

Connector / Led	Style	Level	Description
FAIL	Red Led		Module Failure
OFF	Blue Led		Module Powered Down
RX LINK	Green Led		Receiver Link Signal OK
ENA	Green Led		Event Generator Enabled
EVENT IN	Yellow Led		Incoming Event (RX)
EVENT OUT	Yellow Led		Outgoing Event (TX)
RX FAIL	Red Led		Receiver Violation
ERR	Red Led		SY87739L reference not locked
RUN	Green Led		Ubicom IP2022 Running
ACT	Yellow Led		Ubicom IP2022 Telnet connection active
10baseT	RJ45	10baseT	Ubicom 10baseT Ethernet

			Connection with link (green) and active (amber) leds
10/100	RJ45		(reserved)
TX	LC	optical	Transmit Optical Output (TX)
RX	LC	optical	Receiver Optical Input (RX)
ACIN	LEMO-EPY	TTL	Trigger input
RFIN	LEMO-EPY	RF +10 dBm	RF Reference Input
IN0	LEMO-EPY	TTL	Configurable front panel input
IN1	LEMO-EPY	TTL	Configurable front panel input
OUT0	LEMO-EPY	TTL	Multiplexed Counter 0 Output
OUT1	LEMO-EPY	TTL	Multiplexed Counter 1 Output
OUT2	LEMO-EPY	TTL	Multiplexed Counter 7 Output
OUT3	LEMO-EPY	TTL	Injection Trigger
OUT4	LEMO-EPY	TTL	Reserved Output
OUT5	LEMO-EPY	TTL	Reserved Output
UNIV0	Universal I/O		Configurable input
UNIV1	Universal I/O		Configurable input
UNIV2	Universal I/O		Configurable input
UNIV3	Universal I/O		Configurable input
COM	RJ45	RS232	Reserved

VME P2 User I/O Pin Configuration

The following table lists the connections to the VME P2 User I/O Pins.

Pin	Signal
A1	Transition board ID0
A2	Transition board ID1
A3-A10	Ground
A11	Transition board ID2
A12	Transition board ID3
A13-A15	Ground
A16	Transition board handle switch
A17-A26	Ground
A27-A31	+5V
A32	Power control for transition board
C1	external event trigger input 0
C2	external event trigger input 1
C3	external event trigger input 2
C4	external event trigger input 3
C5	external event trigger input 4
C6	external event trigger input 5
C7	external event trigger input 6
C8	external event trigger input 7
C9	sequence RAM 1 clock
C10	sequence RAM 1 trigger
C11	sequence RAM 1 reset

C12	sequence RAM 2 clock
C13	sequence RAM 2 trigger
C14	sequence RAM 2 reset
C15	(reserved input)
C16	(reserved input)
C17	multiplexed counter 0 reset
C18	multiplexed counter 1 reset
C19	multiplexed counter 2 reset
C20	multiplexed counter 3 reset
C21	multiplexed counter 4 reset
C22	multiplexed counter 5 reset
C23	multiplexed counter 6 reset
C24	multiplexed counter 7 reset (main external reset)
C25	distributed bus input 0
C26	distributed bus input 1
C27	distributed bus input 2
C28	distributed bus input 3
C29	distributed bus input 4 / timestamp clock
C30	distributed bus input 5 / timestamp reset input
C31	distributed bus input 6 / timestamp seconds '0' input
C32	distributed bus input 7 / timestamp seconds '1' input

Programming Details

CR/CSR Support

The Event Generator module provides CR/CSR Support as specified in the VME64x specification. The CR/CSR Base Address Register is determined after reset by the inverted state of VME64x P1 connector signal pins GA4*-GA0*. In case the parity signal GAP* does not match the GAx* pins the CR/CSR Base Address Register is loaded with the value 0xf8 which corresponds to slot number 31.

Note: the board can be used in standard VME crates where geographical pins do not exist, in this case the user may either insert jumpers to set the geographical address or use the default setting when the board's CR/CSR base address will be set to 0xf8.

After power up or reset the board responds only to CR/CSR accesses with its geographical address. Prior to accessing Event Generator functions the board has to be configured by accessing the boards CSR space.

The Configuration ROM (CR) contains information about manufacturer, board ID etc. to identify boards plugged in different VME slots. The following table lists the required field to locate an Event Generator module.

CR address	Register	EVG
------------	----------	-----

0x27, 0x2B, 0x2F	Manufacturer's ID (IEEE OUI)	0x000EB2
0x33, 0x37, 0x3B, 0x3F	Board ID	0x454700E6

For convenience functions are provided to locate VME64x capable boards in the VME crate.

```
STATUS vmeCRFindBoard(int slot, UINT32 ieee_oui, UINT32 board_id,
                    int *p_slot);
```

To locate the first Event Generator in the crate starting from slot 1, the function has to be called following:

```
#include "vme64x_cr.h"
int slot = 1;
int slot_ev;
vmeCRFindBoard(slot, 0x000EB2, 0x454700E6, &slot_ev);
or
vmeCRFindBoard(slot, MRF_IEEE_OUI, MRF_4CHTIM_BID, &slot_ev);
```

If this function returns OK, an Event Generator board was found in slot `slot_ev`.

Event Generator Function 0 Registers

The Event Generator specific register are accessed via Function 0 as specified in the VME64x specification. To enable Function 0, the address decoder compare register for Function 0 in CSR space has to be programmed. For convenience a function to perform this is provided:

```
STATUS vmeCSRWriteADER(int slot, int func, UINT32 ader);
```

To configure Function 0 of an Event Generator board in slot 3 to respond to A16 accesses at the address range 0x1800-0x1FFF the function has to be called with following values:

```
vmeCSRWriteADER(3, 0, 0x18A4);
```

ADER contents are composed of the address mask and address modifier, the above is the same as:

```
vmeCSRWriteADER(3, 0, (slot << 11) | (VME_AM_SUP_SHORT_IO << 2));
```

To get the memory mapped pointer to the configured Function 0 registers on the Event Generator board the following VxWorks function has to be called:

```
MrfEvgStruct *pEvg;
sysBusToLocalAdrs(VME_AM_SUP_SHORT_IO, (char *) (slot << 11),
                (void *) pEvg);
```

Note: using the data transmission capability requires reserving more than 4 kbytes for function 0 i.e. use of addressing mode A24 is suggested, following:

```
vmeCSRWriteADER(3, 0, (slot << 19) | (VME_AM_STD_USR_DATA << 2));
MrfEvgStruct *pEvg;
sysBusToLocalAdrs(VME_AM_STD_USR_DATA, (char *) (slot << 19),
                (void *) pEvg);
```

Register Map

Address Offset	Register	Type	Description
0x000	Control	UINT16	Control/Status Register
0x002	EventEnable	UINT16	Event Enable Register
0x004	SWEvent	UINT16	Software Event
0x006	(reserved)	UINT16	Reserved
0x008	(reserved)	UINT16	Reserved
0x00A	(reserved)	UINT16	Reserved
0x00C	(reserved)	UINT16	Reserved
0x00E	EventMap0	UINT16	Event 0 Mapping Register
0x010	EventMap1	UINT16	Event 1 Mapping Register
0x012	EventMap2	UINT16	Event 2 Mapping Register
0x014	EventMap3	UINT16	Event 3 Mapping Register
0x016	EventMap4	UINT16	Event 4 Mapping Register
0x018	EventMap5	UINT16	Event 5 Mapping Register
0x01A	EventMap6	UINT16	Event 6 Mapping Register
0x01C	EventMap7	UINT16	Event 7 Mapping Register
0x01E	MXCEnable	UINT16	Multiplexed Counter Enable Register
0x020	(reserved)	UINT16	Reserved
0x022	(reserved)	UINT16	Reserved
0x024	Seq1ClockSel	UINT16	Sequencer 1 Clock Select Register
0x026	Seq2ClockSel	UINT16	Sequencer 2 Clock Select Register
0x028	ACEnable	UINT16	AC Sync. Enable Register
0x02A	MXCControl	UINT16	Multiplexed Counter Control Register
0x02C	MXCPrescaler	UINT16	Multiplexed Counter Prescaler Register
0x02E	FirmwareVersion	UINT16	Firmware Version Number
0x030	(reserved)	UINT32	Reserved
0x034	(reserved)	UINT32	Reserved
0x038	(reserved)	UINT32	Reserved
0x03C	(reserved)	UINT32	Reserved
0x040	RFControl	UINT16	RF Clock Select Register
0x042	MXCPolarity	UINT16	Multiplexed Counter Reset Polarity
0x044	Seq1Addr	UINT16	Sequencer 1 RAM Address Register
0x046	Seq1Code	UINT16	Sequencer 1 RAM Event Code
0x048	Seq1Time	UINT32	Sequencer 1 RAM Timestamp
0x04C	Seq1Pos	UINT32	Sequencer 1 Sequence Current Time
0x050	Seq2Addr	UINT16	Sequencer 2 RAM Address Register
0x052	Seq2Code	UINT16	Sequencer 2 RAM Event Code
0x054	Seq2Time	UINT32	Sequencer 2 RAM Timestamp
0x058	Seq2Pos	UINT32	Sequencer 2 Sequence Current Time

0x05C	EvanControl	UINT16	Event Analyser Control Register
0x05E	EvanEvent	UINT16	Event Analyser Distributed Bus and Event Code Register
0x060	EvanTimeHigh	UINT32	Event Analyser Time Counter (bits 63 – 32)
0x064	EvanTimeLow	UINT32	Event Analyser Time Counter (bits 31 – 0)
0x068	uSecDivider	UINT16	Divider to get from Event Clock to 1 MHz
0x06A	DataBufControl	UINT16	Data Buffer Control Register
0x06C	DataBufSize	UINT16	Data Buffer Size Register
0x06E	DBusEvents	UINT16	Distributed Bus Event Enable Register
0x070	(reserved)	UINT32	TTL IN0 Mapping Register
0x072	(reserved)	UINT32	TTL IN1 Mapping Register
0x074	(reserved)	UINT32	UNIV IN0 Mapping Register
0x076	(reserved)	UINT32	UNIV IN1 Mapping Register
0x078	(reserved)	UINT32	UNIV IN2 Mapping Register
0x07A	(reserved)	UINT32	UNIV IN3 Mapping Register
0x07C	(reserved)	UINT32	Reserved
0x080	FracDiv	UINT32	SY87739L Fractional Divider Configuration Word
0x084	(reserved)	UINT32	Reserved
0x088	RxDelay	UINT32	Receive Delay (Controlled by PPC)
0x08C	TxDelay	UINT32	Transmit Delay (Controlled by PPC)
0x090	ADICtrl	UINT32	ADN2812 Control Word
0x094	FbTxFrac	UINT32	Transmit Feedback Fraction
0x098	(reserved)	UINT32	Reserved
0x09C	RxDelayInit	UINT32	Receive Init Delay
0x0A0	TxDelayInit	UINT32	Transmit Init Delay
0x0A4	(reserved)	UINT32	Reserved
0x0A8	(reserved)	UINT32	Reserved
0x0AC	(reserved)	UINT32	Reserved
0x0B0	(reserved)	UINT32	Reserved
0x0B4	(reserved)	UINT32	Reserved
0x0B8	(reserved)	UINT32	Reserved
0x0BC	(reserved)	UINT32	Reserved
0x0C0	(reserved)	UINT32	Reserved
0x0C4	(reserved)	UINT32	Reserved
0x0C8	(reserved)	UINT32	Reserved
0x0CC	(reserved)	UINT32	Reserved
0x0D0	(reserved)	UINT32	Reserved
0x0D4	(reserved)	UINT32	Reserved
0x0D8	(reserved)	UINT32	Reserved
0x0DC	(reserved)	UINT32	Reserved

0x0E0	(reserved)	UINT32	Reserved
0x0E4	(reserved)	UINT32	Reserved
0x0E8	(reserved)	UINT32	Reserved
0x0EC	(reserved)	UINT32	Reserved
0x0F0	(reserved)	UINT32	Reserved
0x0F4	(reserved)	UINT32	Reserved
0x0F8	(reserved)	UINT32	Reserved
0x0FC	(reserved)	UINT32	Reserved
0x100-0x7FF	(reserved)		Reserved
0x800-0xFFF	DataBuf		Data Buffer Transmit Memory

Control and Status Register

address	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
0x000	MSDIS	FF	RSFIFO	DFIFO	ERRLD			VTRG1

Bit	Function
MSDIS	Master Disable, write 1 to disable EVG
FF	FIFO full flag, write 1 to clear.
RSFIFO	Reset FIFO, write 1 to reset FIFO. The FIFO has to be disabled prior to resetting.
DFIFO	Disable FIFO/receiver, write 1 to disable upstream receiver, this disabled reception and retransmission of event codes from an upstream EVG.
ERRLD	Controls the front panel ERROR-led when the upstream receiver is disabled.
VTRG1	Software trigger sequence RAM 1, Write 1 to trigger.

address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
0x001	VTRG2	RCYL1	RCYL2			SEQ1	SEQ2	RXVIO

Bit	Function
VTRG2	Software trigger sequence RAM 2, Write 1 to trigger.
RCYL1	Sequence RAM 1 recycle mode select
RCYL2	Sequence RAM 2 recycle mode select
SEQ1	Stop and Reset Sequence RAM 1
SEQ2	Stop and Reset Sequence RAM 2
RXVIO	Receiver violation flag, write 1 to clear flag

Event Enable Register

address	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
0x002	DBUS		SSEQ1	SSEQ2	CMODE	ENEV7	ENEV6	ENEV5

Bit	Function
DBUS	When set, distributed bus state of upstream EVG is forwarded
SSEQ1	Sequence RAM 1 Single Sequence Mode
SSEQ2	Sequence RAM 2 Single Sequence Mode
CMODE	Select RAM 1 control signals for RAM 2
ENEVx	Master enable for trigger event x.

address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
0x003	ENEV4	ENEV3	ENEV2	ENEV1	ENEV0	ENSQ1	ENSQ2	ENVME

Bit	Function
ENEVx	Master enable for trigger event x.
ENSQ1	Sequence RAM 1 Enable/Disable Sequence
ENSQ2	Sequence RAM 2 Enable/Disable Sequence
ENVME	Enable VME Events

Software Event Register

address	bit 7	bit 0
0x005	Software event register (write only)	

Distributed Bus Data Register

address	bit 7	bit 0
0x005	Distributed bus data register (read only)	

Trigger Event 0 Mapping Register

address	bit 7	bit 0
0x00f	Trigger Event 0 Mapping Registers	

Trigger Event 1 Mapping Register

address	bit 7	bit 0
0x011	Trigger Event 1 Mapping Registers	

Trigger Event 2 Mapping Register

address	bit 7	bit 0
0x013	Trigger Event 2 Mapping Registers	

Trigger Event 3 Mapping Register

address	bit 7	bit 0
0x015	Trigger Event 3 Mapping Registers	

Trigger Event 4 Mapping Register

address	bit 7	bit 0
0x017	Trigger Event 4 Mapping Registers	

Trigger Event 5 Mapping Register

address	bit 7	bit 0
0x019	Trigger Event 5 Mapping Registers	

Trigger Event 6 Mapping Register

address	bit 7	bit 0

0x01b

Trigger Event 6 Mapping Registers

Trigger Event 7 Mapping Register

address **bit 7** **bit 0**
 0x01d

Trigger Event 7 Mapping Registers

Multiplexed Counter Enable Register

address	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
0x01e	MXDB7	MXDB6	MXDB5	MXDB4	MXDB3	MXDB2	MXDB1	MXDB0

Bit	Function
MXDBx	When set, map multiplexed counter output x to distributed bus bit x.

address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
0x01f	MXEV7	MXEV6	MXEV5	MXEV4	MXEV3	MXEV2	MXEV1	MXEV0

Bit	Function
MXEVx	Enable multiplexed counter output x to generate trigger events.

Sequence RAM 1 Clock Select Register

address **bit 15** **bit 0**
 0x024

Sequence RAM 1 Clock Select Register

Sequence RAM 2 Clock Select Register

address **bit 15** **bit 0**
 0x026

Sequence RAM 2 Clock Select Register

AC Input Control Register

address	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
0x028	ACSQ2	ACSQ1	ACEV0	ACSYNC	ACBYP			DLYSEL

Bit	Function
ACSQ1	Enable AC mains voltage synchronization logic output to trigger sequence RAM 1. (Note! Sequence RAM 2 triggered if CMODE is 1)
ACSQ2	Enable AC mains voltage synchronization logic output to trigger sequence RAM 2. (Note! Only when CMODE is 0)
ACEV0	Enable AC mains voltage synchronization logic output to generate trigger event 0.
ACSYNC	Synchronization select (0 = event clock, 1 = multiplexed counter 7 output)
ACBYP	AC divider and phase shifter bypass (0 = divider/phase shifter enabled, 1 = divider/phase shifter bypassed)
DLYSEL	AC data register select bit, 0 for divider, 1 for delay.

address **bit 7** **bit 0**
 0x029

AC Input Divider / Delay Register

Multiplexed Counter Control Register

address	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
0x02A	MXRS7	MXRS6	MXRS5	MXRS4	MXRS3	MXRS2	MXRS1	MXRS0

Bit	Function
MXRSx	Write 1 to reset multiplexed counter x.

address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
0x02B	MXSQ2	MXSQ1			MXHSEL	MXSEL2	MXSEL1	MXSEL0

Bit	Function
MXSQ1	Enable multiplexed counter 0 output to trigger sequence RAM 1. (Note! Sequence RAM 2 triggered if CMODE is 1)
MXSQ2	Enable multiplexed counter 1 output to trigger sequence RAM 2. (Note! Only when CMODE is 0)
MXHSEL	Multiplexed counter select prescaler high word
MXSEL2-0	Multiplexed counter select register

Multiplexed Counter Prescaler Register

address	bit 15	bit 0
0x02C	Multiplexed Counter Prescaler Register	

RF Clock Select Register

address	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
0x040					TRRES	TXPWD	TXRES	TCSEL

address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
0x041	ECSEL	BRSEL	RFSEL5	RFSEL4	RFSEL3	RFSEL2	RFSEL1	RFSEL0

Bits	Function
TRRES	Multi-gigabit transceiver reset: 0 – normal operation 1 – transceiver reset
TXPWD	Multi-gigabit transceiver power down 0 – normal operation 1 – transceiver powered down
TXRES	Transmitter reset: 0 – normal operation 1 – transmitter in reset
TCSEL	Transmitter reference clock select: 0 – on-chip fractional synthesizer 1 – external RF / recovered event clock from upstream EVG
ECSEL	Event clock select: 0 – on-chip fractional synthesizer 1 – external RF / recovered event clock from upstream EVG
BRSEL	BREFCLK2 source select:

0 – recovered event clock from upstream EVG
 1 – external RF

RFSEL5-0 RF divider select:

000000 – RF/1
 000001 – RF/2
 000010 – RF/3
 000011 – RF/4
 000100 – RF/5
 000101 – RF/6
 000110 – RF/7
 000111 – RF/8
 001000 – RF/9
 001001 – RF/10
 001010 – RF/11
 001011 – RF/12
 001100 – OFF
 001101 – RF/14
 001110 – RF/15
 001111 – RF/16
 010000 – RF/17
 010001 – RF/18
 010010 – RF/19
 010011 – RF/20

MXC Polarity Select Register

address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
0x043	MXCP7	MXCP6	MXCP5	MXCP4	MXCP3	MXCP2	MXCP1	MXCP0

Bits **Function**
 MXCPx Multiplexed Counter Reset State
 0 – falling edged aligned
 1 – rising edges aligned

Sequencer 1 RAM Address Register

address	bit 15	bit 10	bit 9	bit 0
0x044				Sequencer 1 RAM address register

Sequencer 1 RAM Event Code Register

address	bit 15	bit 8	bit 7	bit 0
0x046				Sequencer 1 RAM Event Code Register

Sequencer 1 RAM Event Timestamp Register

address	bit 31	bit 0
0x048	Sequencer 1 RAM Event Timestamp Register	

Sequencer 2 RAM Address Register

address	bit 15	bit 10	bit 9	bit 0
0x050				Sequencer 2 RAM address register

Sequencer 2 RAM Event Code Register

address	bit 15	bit 8	bit 7	bit 0
0x052				Sequencer 2 RAM Event Code Register

Sequencer 2 RAM Event Timestamp Register

address	bit 31	bit 0
0x054	Sequencer 2 RAM Event Timestamp Register	

Event Analyser Control Register

address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
0x05D				EVANE	EVARS	EVAOF	EVAEN	EVACR

Bits	Function
EVANE	Event Analyser FIFO not empty flag: 0 – FIFO empty 1 – FIFO not empty, events in FIFO
EVARS	Event Analyser Reset 0 – not in reset 1 – reset
EVAOF	Event Analyser FIFO overflow flag: 0 – no overflow 1 – FIFO overflow
EVAEN	Event Analyser enable 0 – Event Analyser disabled 1 – Event Analyser enabled
EVACR	Event Analyser 64 bit counter reset 0 – Counter running 1 – Counter reset to zero.

Event Analyser Data Register

address	bit 15	bit 8	bit 7	bit 0
0x05E	(reserved)		Event Code	

Event Analyser Counter Registers

address	bit 31	bit 0
0x060	Event Analyser Counter Register (bits 63 – 32)	

address	bit 31	bit 0
0x064	Event Analyser Counter Register (bits 31 – 0)	

Data Buffer Control Register

address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
0x06B				TXCPT	TXRUN	TRIG	ENA	MODE

Bits	Function
TXCPT	Data Buffer Transmission Complete
TXRUN	Data Buffer Transmission Running – set when data transmission has been triggered and has not been completed yet
TRIG	Data Buffer Trigger Transmission Write ‘1’ to start transmission of data in buffer
ENA	Data Buffer Transmission enable ‘0’ – data transmission engine disabled ‘1’ – data transmission engine enabled
MODE	Distributed bus sharing mode ‘0’ – distributed bus not shared with data transmission ‘1’ – distributed bus shared with data transmission

Distributed Bus Event Enable Register

address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
0x06F	DBEV7	DBEV6	DBEV5					

Bits	Function
DBEV5	Distributed bus input 5 “Timestamp reset” 0x7D event enable
DBEV6	Distributed bus input 6 “Seconds ‘0’” 0x70 event enable
DBEV7	Distributed bus input 7 “Seconds ‘1’” 0x71 event enable

TTL Input Mapping Register

address	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
0x070	IN0DB7	IN0DB6	IN0DB5	IN0DB4	IN0DB3	IN0DB2	IN0DB1	IN0DB0

address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
0x071	IN0EV7	IN0EV6	IN0EV5	IN0EV4	IN0EV3	IN0EV2	IN0EV1	IN0EV0

address	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
0x072	IN1DB7	IN1DB6	IN1DB5	IN1DB4	IN1DB3	IN1DB2	IN1DB1	IN1DB0

address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
0x073	IN1EV7	IN1EV6	IN1EV5	IN1EV4	IN1EV3	IN1EV2	IN1EV1	IN1EV0

Bits	Function
INxDBy	Forward TTLINx to distributed bus bit y
INxEVy	Trigger Event y from TTLINx

Universal I/O Input Mapping Register

address	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
0x074	UI0DB7	UI0DB6	UI0DB5	UI0DB4	UI0DB3	UI0DB2	UI0DB1	UI0DB0

address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
0x075	UI0EV7	UI0EV6	UI0EV5	UI0EV4	UI0EV3	UI0EV2	UI0EV1	UI0EV0

address	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
0x076	UI1DB7	UI1DB6	UI1DB5	UI1DB4	UI1DB3	UI1DB2	UI1DB1	UI1DB0

address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
0x077	UI1EV7	UI1EV6	UI1EV5	UI1EV4	UI1EV3	UI1EV2	UI1EV1	UI1EV0

address	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
0x078	UI2DB7	UI2DB6	UI2DB5	UI2DB4	UI2DB3	UI2DB2	UI2DB1	UI2DB0

address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
0x079	UI2EV7	UI2EV6	UI2EV5	UI2EV4	UI2EV3	UI2EV2	UI2EV1	UI2EV0

address	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
0x07A	UI3DB7	UI3DB6	UI3DB5	UI3DB4	UI3DB3	UI3DB2	UI3DB1	UI3DB0

address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
0x07B	UI3EV7	UI3EV6	UI3EV5	UI3EV4	UI3EV3	UI3EV2	UI3EV1	UI3EV0

Bits	Function
UIxDB _y	Forward Universal I/O Input x to distributed bus bit y
UIxEV _y	Trigger Event y from Universal I/O Input UNIV _x

Network Interface

A 10baseT network interface is provided to upgrade the FPGA firmware and set up boot options. It is also possible to control the module over the network interface.

Assigning an IP Address to the Module

By default the modules uses DHCP (dynamic host configuration protocol) to acquire an IP address. In case a lease cannot be acquired the IP address set randomly in the 169.254.x.x subnet. The board can be programmed to use a static address instead if DHCP is not available.

The module can be located looking at the lease log of the DHCP server or using a Windows tool called Locator.exe.

Using Telnet to Configure Module

To connect to the configuration utility of the module issue the following command:

```
telnet 192.168.1.32 23
```

The latter parameter is the telnet port number and is required in Linux to prevent negotiation of telnet parameters which the telnet server of the module is not capable of.

The telnet server responds to the following commands:

Command	Description
b	Show/change boot parameters, IP address etc.
d	Dump 16 bytes of memory
h / ?	Show Help
i	Read & show dynamic configuration values from FPGA
m <address> [<data>]	Read/Write FPGA CR/CSR, Function 0
r	Reset Board
s	Save boot configuration & dynamic configuration values into non-volatile memory
u	Update IP2022 software
q	Quit Telnet

Boot Configuration (command b)

Command b displays the current boot configuration parameters of the module. The parameter may be changed by giving a new parameter value. The following parameters are displayed:

Parameter	Description
Use DHCP	0 = use static IP address, 1 = use DHCP to acquire address, net mask etc.
IP address	IP address of module
Subnet mask	Subnet mask of module
Default GW	Default gateway
FPGA mode	FPGA configuration mode 0 – FPGA is not configured after power up 1 – FPGA configured from internal Flash memory 2 – FPGA is configured from FTP server
FTP server	FTP server IP address where configuration bit file resides
Username	FTP server username
Password	FTP server password
FTP Filename	FTP server configuration file name
Flash Filename	Configuration file name on internal flash
µs divider	Integer divider to get from event clock to 1MHz, e.g. 125 for 124.9135 MHz
Fractional divider configuration word	Micrel SY87739UMI fractional divider configuration word to set reference for event clock

Note that after changing parameters the parameters have to be saved to internal flash by issuing the Save boot configuration (s) command. The changes are applied only after resetting the module using the reset command or hardware reset/power sequencing.

Memory dump (command d)

This command dumps 16 bytes of memory starting at the given address, if the address is omitted the previous address value is increased by 16 bytes.

The most significant byte of the address determines the function of the access:

Address	Function
0x00000000	CR/CSR space access
0x80000000	EVG registers access

To dump the start of the EVG register map issue the 'd' command from the telnet prompt:

```
VME-EVG-230 -> d 80000000 ↵
Addr 80000000: d000 0001 0000 0000 0000 0000 0000 0000
VME-EVG-230 -> d ↵
Addr 80000010: 0000 0000 0000 0000 0000 0000 0000 0000
VME-EVG-230 ->
```

Memory modify (commands d and m)

The access size is always a short word i.e. two bytes.

To check the status register from the telnet prompt:

```
VME-EVG-230 -> m 80000000 ↵
Addr 80000000 data d000
VME-EVG-230 ->
```

To enable the EVG issue:

```
VME-EVG-230 -> m 80000000 0000 ↵
Addr 80000000 data 4001
VME-EVG-230 ->
```

Upgrading IP2022 Microprocessor Software (command u)

To upgrade the Uvicom IP2022 microprocessor software download the upgrade image containing the upgrade to the module using TFTP:

Linux

In Linux use e.g. interactive tftp:

```
$ tftp 192.168.1.32
tftp> bin
tftp> put upgrade.bin /fw
tftp> quit
```

Windows

In Windows command prompt issue the following command:

```
C:\> tftp -i 192.168.1.32 PUT upgrade.bin /fw
```

When the upgrade image has been downloaded and verified, enter at the telnet prompt following:

```
VME-EVG-230 -> u ↵
Really update firmware (yes/no) ? yes ↵
```

Self programming triggered.

The Event Generator starts programming the new software and restarts.

Upgrading FPGA Configuration File

When the FPGA configuration file resides in internal flash memory a new file system image has to be downloaded to the module. This is done using TFTP protocol:

Linux

In Linux use e.g. interactive tftp:

```
$ tftp 192.168.1.32
tftp> bin
tftp> put filesystem.bin /
tftp> quit
```

Windows

In Windows command prompt issue the following command:

```
C:\> tftp -i 192.168.1.32 PUT filesystem.bin /
```

Now the FPGA configuration file has been upgraded and the new configuration is loaded after next reset/power sequencing.

Note! Due to the UDP protocol it is recommended to verify (read back and compare) the filesystem image before restarting the module. This is done following:

Linux

In Linux use e.g. interactive tftp:

```
$ tftp 192.168.1.32
tftp> bin
tftp> get / verify.bin
tftp> quit
$ diff filesystem.bin verify.bin
$
```

If files differ you should get following message:

```
Binary files filesystem.bin and verify.bin differ
```

Windows

In Windows command prompt issue the following command:

```
C:\> tftp -i 192.168.1.32 GET / verify.bin
C:\> fc /b filesystem.bin verify.bin
Comparing files filesystem.bin and verify.bin
FC: no differences encountered
```

UDP Remote Programming Protocol

The VME-EVG can be remotely programmed using the 10baseT Ethernet interface with a protocol over UDP (User Datagram Protocol) which runs on top of IP (Internet Protocol). The default port for remote programming is UDP port 2000. The UDP commands are built upon the following structure:

access_type (1 byte)	status (1 byte)	data (2 bytes)
address (4 bytes)		
ref (4 bytes)		

The first field defines the access type:

access_type	Description
0x01	Read Register from module
0x02	Write and Read back Register from module

The second field tells the status of the access:

Status	Description
0	Command OK
-1	Bus ERROR (Invalid read/write address)
-2	Timeout (FPGA did not respond)
-3	Invalid command

The access size is always a short word i.e. two bytes. The most significant byte of the address determines the function of the access:

Address	Function
0x00000000	CR/CSR space access
0x80000000	EVG registers access

Read Access (Type 0x01)

The host sends a UDP packet to port 2000 of the VME-EVG with the following contents:

access_type (1 byte) 0x01	status (1 byte) 0x00	data (2 bytes) 0x0000
address (4 bytes) 0x80000000 (Control and Status register Function 0 address)		
ref (4 bytes) 0x00000000		

If the read access is successful the VME-EVG replies to the same host and port the message came from with the following packet:

access_type (1 byte) 0x01	status (1 byte) 0x00	data (2 bytes) 0xD000
address (4 bytes) 0x80000000 (Control and Status register Function 0 address)		

ref (4 bytes) 0x00000000

Write Access (Type 0x02)

The host sends a UDP packet to port 2000 of the VME-EVG with the following contents:

access_type (1 byte) 0x02	status (1 byte) 0x00	data (2 bytes) 0x0001
Address (4 bytes) 0x80000002 (Event enable register Function 0 address)		
ref (4 bytes) 0x00000000		

If the write access is successful the VME-EVG replies to the same host and port the message came from with the following packet:

access_type (1 byte) 0x02	status (1 byte) 0x00	data (2 bytes) 0x0001
address (4 bytes) 0x80000002 (Event enable register Function 0 address)		
ref (4 bytes) 0x00000000		

Notice that in the reply message the data returned really is the data read from the address specified in the address field so one can verify that the data really was written ok.