Micro−Research Finland Oy

Event System with Delay Compensation

VME-EVM-300,
VME-EVR-300,
mTCA-EVR-300,
PCIe-EVR-300DC

Technical Reference
Firmware 0207 (PRELIMINARY)

October 18, 2016
http://www.mrf.fi/

Contents

1. The MRF Timing System

The MRF Timing System provides a complete timing distribution system including timing signal generation with only a few components.

The system is capable of generating and synchronous frequencies, trigger signals and sequences of events, etc. synchronous to an externally provided master clock reference and mains voltage phase signal. Support for timestamps makes the system a global timebase and allows attaching timestamps to collected data and performed actions.

1.1. Timing System Topology

As a basic setup the timing system consists of an Event Generator (EVG), the distribution layer (Fan-Out) and Event Receivers (EVR). With the active delay compensation feature the Event Generator and distribution layer have been integrated into a signle product, the Event Master (EVM).

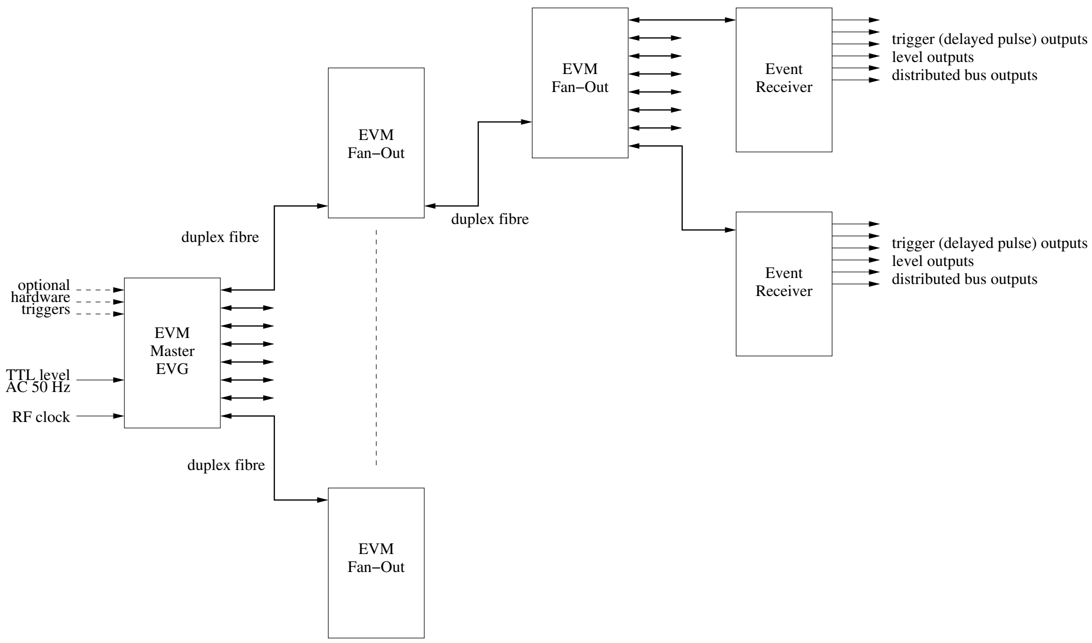

Figure 1: Timing System Topology

1.2. Active Delay Compensation

Delay compensation is achived in measuring the propagation delay of events from the delay compensation master EVM through the distribution network up to the Event Receivers. At the last stage the EVR is aware of the delay through the network and adjusts an internal FIFO depth to match a programmed traget delay value.

1.3. 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.

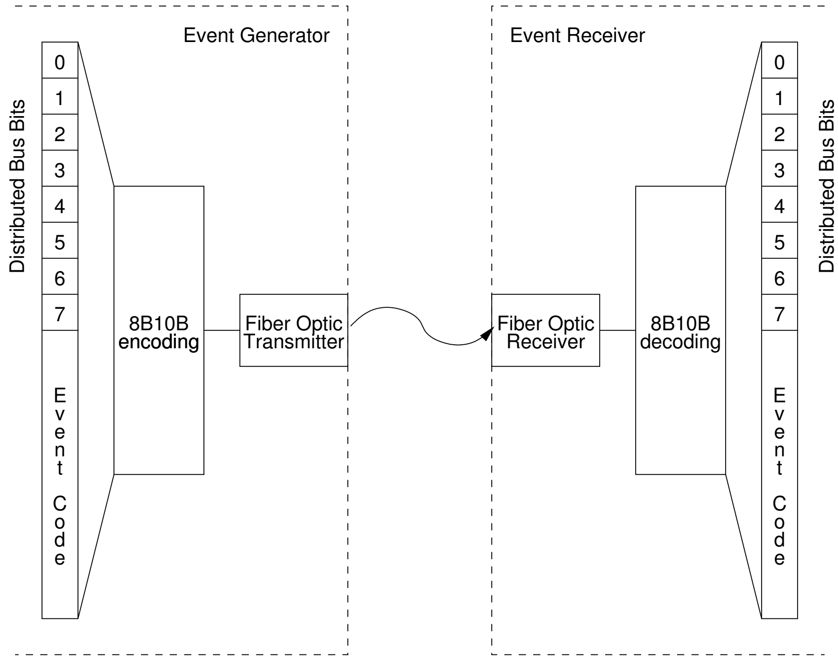

Figure 2: Event Frame

1.3.1. 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 on the serial bit stream.

Table 1: Event Codes

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 – 0x78	-	User Defined	User Defined
0x79	Stop Event Log	-	Stop writing events to Event log
0x7A	Heartbeat (Beacon event)	Beacon event for Delay Compensation	Reset Heartbeat Monitor Delay Compensation signal
0x7B	Synchronise Prescalers	-	Synchronise Prescaler Outputs
0x7C	Timestamp Counter Increment	-	Increment Timestamp Counter

Event Code	Code Name	EVG Function	EVR Function
0x7D	Timestamp Counter Reset	-	Reset Timestamp Counter
0x7F	End of Sequence	Stop Sequence	-
0x80-FF	-	User Defined	User Defined

1.3.2. Protocol Details

The MRF Timing system protocol is based on 8B10B encoded characters. Two characters are transmitted on every event clock cycle. The first encoded byte is an event code and the second encoded byte is shared by the distributed bus and synchronous data buffer.

Event codes are encoded as characters D01.0 through D31.7. Character D00.0 is reserved and transmitted when there is no event code to transmit. To synchronize the receivers every fourth empty event slot is transmitted as a synchronisation character K28.5.

The example below shows the following details:

- Synchronisation characters at cycle 0, 4, 8, 12 and 20. Please not that event code transmission overrides synchronisation character transmission at cycle 16
- Event code transmissions at following cycles:
  - 2 - Beacon event (beacon event code is reserved and shall not be used for user events)
  - 6 – User event code 0x10
  - 16 – User event code 0x20
- A clock with frequency event clock / 4 on distributed bus bit zero:
  - ‘0’ at cycle 0, 4, 8, 12, 16, 20
  - ‘1’ at cycle 2, 6, 10, 14, 18, 22
- Data packet transmission on the synchronous data bus:
  - Four bytes 0xC0FFEE99 are transmitted to address 0x0A0 of the data buffer

Table 2: Event Protocol Example

Cycle	Event slot	Distributed Bus / Data slot
0	K28.5 Sync character, no event	D00.0 Distributed bus byte
1	D00.0 Null event	D00.0 Data buffer, null data
2	D26.3 Beacon event	D01.0 Distributed bus byte
3	D00.0 Null event	D00.0 Data buffer, null data
4	K28.5 Sync character, no event	D00.0 Distributed bus byte
5	D00.0 Null event	K28.0 Data buffer, start of transfer
6	D16.0 Event code 0x10	D01.0 Distributed bus byte
7	D00.0 Null event	D10.0 Data buffer, segment address 0xA0
8	K28.5 Sync character, no event	D00.0 Distributed bus byte

Cycle	Event slot		Distributed Bus / Data slot	
9	D00.0	Null event	D00.6	Data buffer, data byte 0xC0
10	D00.0	Null event	D01.0	Distributed bus byte
11	D00.0	Null event	D31.7	Data buffer, data byte 0xFF
12	K28.5	Sync character, no event	D00.0	Distributed bus byte
13	D00.0	Null event	D14.7	Data buffer, data byte 0xEE
14	D00.0	Null event	D01.0	Distributed bus byte
15	D00.0	Null event	D25.4	Data buffer, data byte 0x99
16	D00.1	Event code 0x20	D00.0	Distributed bus byte
17	D00.0	Null event	K28.1	Data buffer, end transfer
18	D00.0	Null event	D01.0	Distributed bus byte
19	D00.0	Null event	D28.7	Data buffer, checksum MSB 0xFFFF – 0xA0 – 0xC0 – 0xFF – 0xEE – 0x99 = 0xFC19, MSB 0xFC
20	K28.5	Sync character, no event	D00.0	Distributed bus byte
21	D00.0	Null event	D25.0	Data buffer, checksum LSB 0x19
22	D00.0	Null event	D01.0	Distributed bus byte
23	D00.0	Null event	D00.0	Data buffer, null data

1.4. Migrating to 0205 Firmware

It is important to notice that firmware version 0205 for both the EVM and EVR has some changes that are incompatible with earlier firmware versions. All components in a system have to be running either firmware 0200 to 0204 firmware or 0205 or later firmware.

The most important and incompatible change is moving the delay compensation data segment in the segmented data buffer from the beginning of the buffer (segment 0) to the last segment address.

In addition to this the EVR firmware has been added the capability to remain compatible with earlier event generators. The original non-segmented data buffer has been returned to its original memory location and the segmented data buffer has been given new addresses. In addition to this the segmented data buffer now has a receive data length register for each segment.

2. Event Master

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.

2.1. Distributed Bus

The distributed bus allows transmission of eight simultaneous signals with half of the event clock rate time resolution (20 ns at 100 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.

2.2. 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.

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

2.3. 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.

2.4. 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. 8-bit event

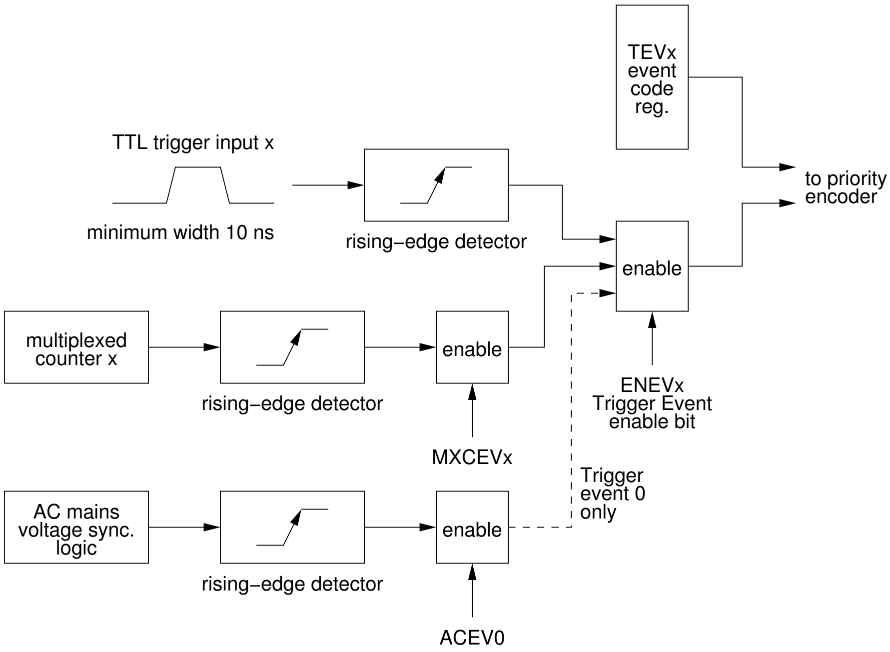

Figure 3: Trigger Event

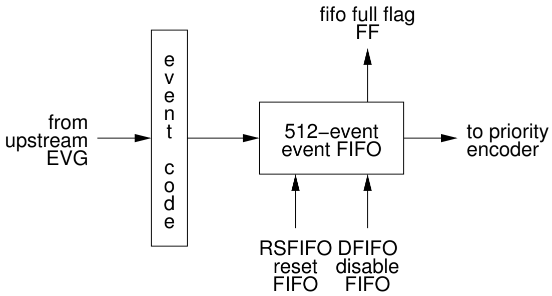

Figure 4: Upstream Events

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.

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 sequencer runs at the event clock rate. Starting with firmware version 0200 a mask field has been added. Bits in the mask field allow masking events from being send out based on external signal input states or software mask bits.

The Sequencers may be triggered from several sources including software triggering, triggering on a multiplexed counter output or AC mains voltage synchronization logic output.

The sequencers are enabled by writing a ‘1’ bit to SQxEN in the Sequence RAM control Register. The RAMs may be disabled any time by writing a ‘1’ to SQxDIS bit. Disabling sequence RAMs does not reset the RAM address and timestamp registers. By writing a ‘1’ to the bit SQxRES 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 or is greater than the timestamp in the RAM table, the attached event code is transmitted. The time offset

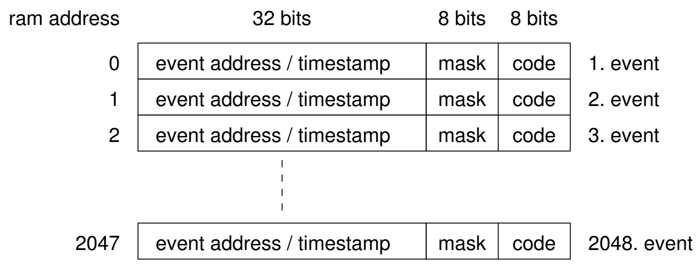

Figure 5: Sequencer RAM Structure

between two consecutive events in the RAM is allowed to be 1 to 232 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 232 event clock cycles by inserting a null event with a timestamp value of 0xffffffff. The end sequence code resets the sequencer RAM table address and timestamp register and depending on configuration bits, disables the sequencer (single sequence, SQxSNG=1) or restarts the sequence either immediately (recycle sequence, SQxREC=1) or waits for a new trigger (SQxREC=0).

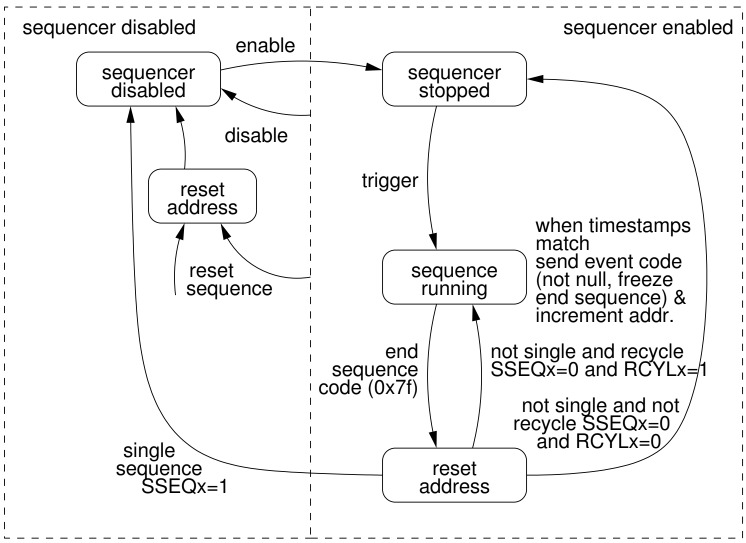

Figure 6: Sequencer RAM Control

2.4.1. Sequencer Interrupt Support

The sequencers provide several interrupts: a sequence start and sequence stop interrupt and a two interrupts based on the position of the playback pointer in the sequencer RAM: a sequence halfway through interrupt and a sequence roll-over interrupt. The sequence start interrupt is issued when a sequencer is in enabled state, gets triggered and was not running before the trigger.

A sequence stop interrupt is issued when the sequence is running and reaches the ‘end of sequence’ code.

2.5. 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.

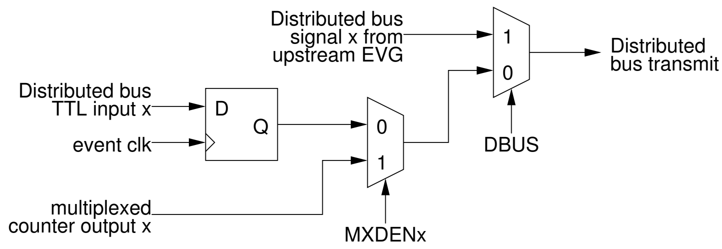

Figure 7: Distributed Bus

2.6. 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.

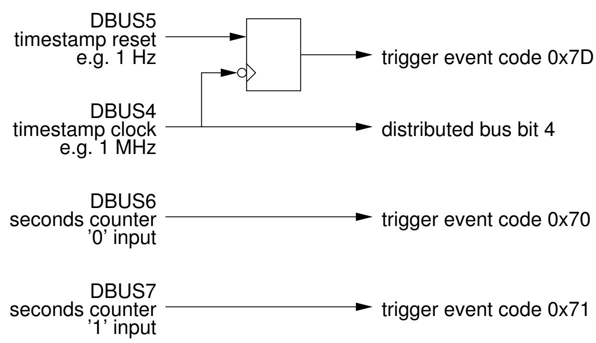

Figure 8: Timestamping Inputs

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.

2.7. Timestamp Generator

Logic has been added to automatically increment and send out the 32-bit seconds value. Using this feature requires the two externally supplied clocks as shown above, but the events 0x70 and 0x71 get generated automatically.

After the rising edge of the slower clock on DBUS4, the internal seconds counter is incremented and the 32 bit binary value is sent out LSB first as 32 events 0x70 and 0x71. The seconds counter can be updated by software by using the TSValue and TSControl registers.

2.8. 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.

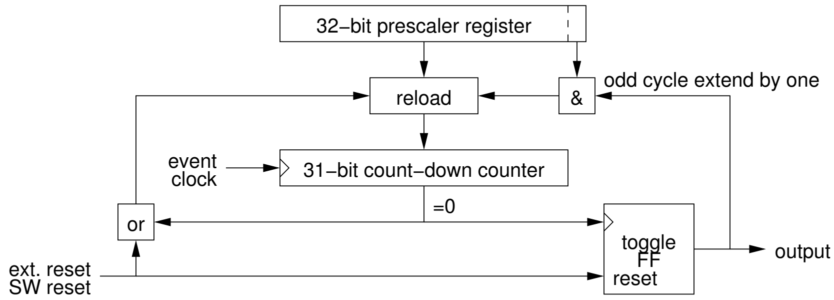

Figure 9: Multiplexed Counters

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.

Table 3: Multiplexed Counter Prescaler values

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.

2.9. Segmented Data Buffer

With the introduction of active transmission path delay compensation in firmware version 0200 use of the “data buffer mode” has become mandatory. Delay compensation logic does use the beginning of the data buffer memory for propagating delay compensation information. With this change the data buffer memory now is divided into 256 16 byte segments and it is possible to transmit the contents of a single segment or a block of consecutive segments without affecting contents of other segments.

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 4: Data Transfer Example

8B10B-character	**Description**
K28.0	Start of data transfer
Dxx.x	Block address of 16 byte segment
Dxx.x	1st data byte (address 0)
Dxx.x	2nd data byte (address 1)
Dxx.x	3rd data byte (address 2)
Dxx.x	4th data byte (address 3)
...	...
Dxx.x	nth data byte (address n-1)
K28.1	End of data
Dxx.x	Checksum (MSB)
Dxx.x	Checksum(LSB)

2.10. Programmable Outputs

All the outputs are programmable: multiplexed counters and distributed bus bits can be mapped to any output. The mapping is shown in table below.

Table 5: Mapping IDs

Mapping ID	**Signal**
0 to 31	(Reserved)
32	Distributed bus bit 0 (DBUS0)
...	...
39	Distributed bus bit 7 (DBUS7)
40	Multiplexed Counter 0
...	...
47	Multiplexed Counter 7
48	AC trigger logic output
49 to 61	(Reserved)
62	Force output high (logic 1)
63	Force output low (logic 0)

Mapping ID	Signal

2.11. 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.

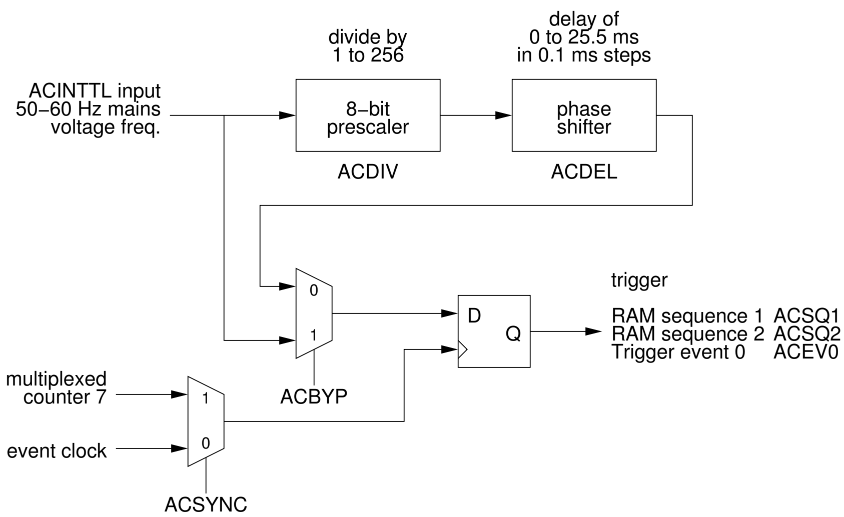

Figure 10: AC Input Logic

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.

2.12. Event Clock RF Source

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.

Table 6: Event Clock Requirement

	Event Clock	Bit Rate
Minimum	50 MHz	1.0 Gb/s
Maximum	142.8 MHz	2.9 Gb/s

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

2.13. 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.

Table 7: RF Input Requirements

Divider	**RF Input Frequency**	**Event Clock**	**Bit Rate**
÷ 1	50 MHz – 142.8 MHz	50 MHz – 142.8 MHz	1.0 Gb/s – 2.9 Gb/s
÷ 2	100 MHz – 285.6 MHz	50 MHz – 142.8 MHz	1.0 Gb/s – 2.9 Gb/s
÷ 3	150 MHz – 428.4 MHz	50 MHz – 142.8 MHz	1.0 Gb/s – 2.9 Gb/s
÷ 4	200 MHz – 571.2 MHz	50 MHz – 142.8 MHz	1.0 Gb/s – 2.9 Gb/s
÷ 5	250 MHz – 714 MHz	50 MHz – 142.8 MHz	1.0 Gb/s – 2.9 Gb/s
÷ 6	300 MHz – 856.8 MHz	50 MHz – 142.8 MHz	1.0 Gb/s – 2.9 Gb/s
÷ 7	350 MHz – 999.6 MHz	50 MHz – 142.8 MHz	1.0 Gb/s – 2.9 Gb/s
÷ 8	400 MHz – 1.142 GHz	50 MHz – 142.8 MHz	1.0 Gb/s – 2.9 Gb/s
÷ 9	450 MHz – 1.285 MHz	50 MHz – 142.8 MHz	1.0 Gb/s – 2.9 Gb/s
÷ 10	500 MHz – 1.428 GHz	50 MHz – 142.8 MHz	1.0 Gb/s – 2.9 Gb/s
÷ 11	550 MHz – 1.571 GHz	50 MHz – 142.8 MHz	1.0 Gb/s – 2.9 Gb/s
÷ 12	600 MHz – 1.6 GHz	50 MHz – 133 MHz	1.0 Gb/s – 2.667 Gb/s
÷ 14	700 MHz – 1.6 GHz *)	50 MHz – 114 MHz	1.0 Gb/s – 2.286 Gb/s
÷ 15	750 MHz – 1.6 GHz *)	50 MHz – 107 MHz	1.0 Gb/s – 2.133 Gb/s
÷ 16	800 MHz – 1.6 GHz *)	50 MHz – 100 MHz	1.0 Gb/s – 2.0 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

2.13.1. 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.

Table 8: Fractional Synthesiser

Event Rate	Configuration Bit Pattern	Reference Output	Precision (theoretical)
142.8 MHz	0x0891C100	142.857 MHz	0
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

2.14. VME-EVM-300 Front Panel Connections

The front panel of the Event Generator is shown in Figure 11.

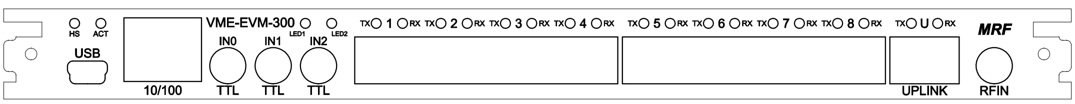

Figure 11: VME-EVM-300 Front Panel

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

Table 9: VME-EVM-300 Front Panel Connections

Connector / Led	Style	Level	Description
HS	Red Led		Module Failure
HS	Blue Led		Module Powered Down
ACT	3-color Led		SAM3X Activity Led
USB	Micro-USB		SAM3X Serial port / JTAG interface
10/100	RJ45		SAM3X Ethernet Interface
IN0	LEMO	TTL	ACIN / TTL0 Trigger input
IN1	LEMO	TTL	Configurable front panel input

Connector / Led	Style	Level	Description
IN2	LEMO	TTL	Configurable front panel input
TX 1	LC	optical	Fan-Out Port 1 Transmit (TX 1)
RX 1	LC	optical	Concentrator Port 1 Receiver (RX 1)
TX 2	LC	optical	Fan-Out Port 2 Transmit (TX 2)
RX 2	LC	optical	Concentrator Port 2 Receiver (RX 2)
TX 3	LC	optical	Fan-Out Port 3 Transmit (TX 3)
RX 3	LC	optical	Concentrator Port 3 Receiver (RX 3)
TX 4	LC	optical	Fan-Out Port 4 Transmit (TX 4)
RX 4	LC	optical	Concentrator Port 4 Receiver (RX 4)
TX 5	LC	optical	Fan-Out Port 5 Transmit (TX 5)
RX 5	LC	optical	Concentrator Port 5 Receiver (RX 5)
TX 6	LC	optical	Fan-Out Port 6 Transmit (TX 6)
RX 6	LC	optical	Concentrator Port 6 Receiver (RX 6)
TX 7	LC	optical	Fan-Out Port 7 Transmit (TX 7)
RX 7	LC	optical	Concentrator Port 7 Receiver (RX 7)
TX 8	LC	optical	Fan-Out Port 8 Transmit (TX 8)
RX 8	LC	optical	Concentrator Port 8 Receiver (RX 8)
TX UP	LC	optical	Upstream Transmit Optical Output (TX)
RX UP	LC	optical	Upstream Receiver Optical Input (RX)
RFIN	LEMO	RF +10 dBm	RF Reference Input

2.14.1. TTL Input Levels

The VME-EVM-300 has three front panel TTL inputs. The inputs are terminated with 50 ohm to ground and are 5V tolerant even when powered down.

Input specifications are following:

parameter	value
connector type	LEMO EPK.00.250.NTN
input impedance	50 ohm
V_{IH}	> 2.3 V
V_{IL}	< 1.0 V

2.15. VME-EVM-300 VME P2 User I/O Pin Configuration

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

Table 10: VME-EVM-300 VME P2 User I/O Pin Configuration

Pin	Signal
A1	Transition board ID0

Pin	Signal
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	Transition board input 0
C2	Transition board input 1
C3	Transition board input 2
C4	Transition board input 3
C5	Transition board input 4
C6	Transition board input 5
C7	Transition board input 6
C8	Transition board input 7
C9	Transition board input 8
C10	Transition board input 9
C11	Transition board input 10
C12 – C27	(reserved input)
C28	Transition board input 11
C29	Transition board input 12
C30	Transition board input 13
C31	Transition board input 14
C32	Transition board input 15

2.16. VME-EVM-300 CR/CSR Support

The VME 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.

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.

Table 11: VME-EVM-300 CR/CSR

CR address	Register	EVG
0x27, 0x2B, 0x2F	Manufacturer's ID (IEEE OUI)	0x000EB2
0x33, 0x37, 0x3B, 0x3F	Board ID	0x4547012C

2.16.1. Function 0 and 1 Registers

The Event Generator specific register are accessed via Function 0 or 1 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.

```
vmeCSRWriteADER(3, 0, (slot << 19) | (VME_AM_STD_USR_DATA << 2));

MrfEvgStruct *pEvg;

sysBusToLocalAdrs(VME_AM_STD_USR_DATA, (char *) (slot << 19),
        (void *) pEvg);
```

2.16.2. Function 2 Registers

The Fan-Out/Concentrator specific register are accessed via Function 2 as specified in the VME64x specification. To enable Function 2, the address decoder compare register for Function 2 in CSR space has to be programmed.

2.17. EVG Function Register Map

Table 12: Event Generator Register Map

Address	**Register**	**Type**	**Description**
0x000	Status	UINT32	Status Register
0x004	Control	UINT32	Control Register
0x008	IrqFlag	UINT32	Interrupt Flag Register
0x00C	IrqEnable	UINT32	Interrupt Enable Register
0x010	ACControl	UINT32	AC divider control
0x014	ACMap	UINT32	AC trigger event mapping
0x018	SWEvent	UINT32	Software event register
0x01C	SegBufControl	UINT32	Segmented Data Buffer Control Register
0x020	DataBufControl	UINT32	Data Buffer Control Register
0x024	DBusMap	UINT32	Distributed Bus Mapping Register
0x028	DBusEvents	UINT32	Distributed Bus Timestamping Events Register
0x02C	FWVersion	UINT32	Firmware Version Register
0x034	TSControl	UINT32	Timestamp event generator control register
0x038	TSValue	UINT32	Timestamp event generator value register
0x040	FPInput	UINT32	Front Panel Input state register

Address	Register	Type	Description
0x044	UnivInput	UINT32	Universal Input state register
0x048	TBInput	UINT32	Transition Board Input state register
0x04C	UsecDivider	UINT32	Divider to get from Event Clock to 1 MHz
0x050	ClockControl	UINT32	Event Clock Control Register
0x060	EvanControl	UINT32	Event Analyser Control Register
0x064	EvanCode	UINT32	Event Analyser Distributed Bus and Event Code Register
0x068	EvanTimeHigh	UINT32	Event Analyser Time Counter (bits 63 – 32)
0x06C	EvanTimeLow	UINT32	Event Analyser Time Counter (bits 31 – 0)
0x070	SeqRamCtrl0	UINT32	Sequence RAM 0 Control Register
0x074	SeqRamCtrl1	UINT32	Sequence RAM 1 Control Register
0x080	FracDiv	UINT32	Micrel SY87739L Fractional Divider Configuration Word
0x0A0	SPIData	UINT32	SPI Data Register
0x0A4	SPIControl	UINT32	SPI Control Register
0x100	EvTrig0	UINT32	Event Trigger 0 Register
0x104	EvTrig1	UINT32	Event Trigger 1 Register
0x108	EvTrig2	UINT32	Event Trigger 2 Register
0x10C	EvTrig3	UINT32	Event Trigger 3 Register
0x110	EvTrig4	UINT32	Event Trigger 4 Register
0x114	EvTrig5	UINT32	Event Trigger 5 Register
0x118	EvTrig6	UINT32	Event Trigger 6 Register
0x11C	EvTrig7	UINT32	Event Trigger 7 Register
0x180	MXCCtrl0	UINT32	Multiplexed Counter 0 Control Register
0x184	MXCPresc0	UINT32	Multiplexed Counter 0 Prescaler Register
0x188	MXCCtrl1	UINT32	Multiplexed Counter 1 Control Register
0x18C	MXCPresc1	UINT32	Multiplexed Counter 1 Prescaler Register
0x190	MXCCtrl2	UINT32	Multiplexed Counter 2 Control Register
0x194	MXCPresc2	UINT32	Multiplexed Counter 2 Prescaler Register
0x198	MXCCtrl3	UINT32	Multiplexed Counter 3 Control Register
0x19C	MXCPresc3	UINT32	Multiplexed Counter 3 Prescaler Register
0x1A0	MXCCtrl4	UINT32	Multiplexed Counter 4 Control Register
0x1A4	MXCPresc4	UINT32	Multiplexed Counter 4 Prescaler Register
0x1A8	MXCCtrl5	UINT32	Multiplexed Counter 5 Control Register
0x1AC	MXCPresc5	UINT32	Multiplexed Counter 5 Prescaler Register
0x1B0	MXCCtrl6	UINT32	Multiplexed Counter 6 Control Register
0x1B4	MXCPresc6	UINT32	Multiplexed Counter 6 Prescaler Register
0x1B8	MXCCtrl7	UINT32	Multiplexed Counter 7 Control Register

Address	Register	Type	Description
0x1BC	MXCPresc7	UINT32	Multiplexed Counter 7 Prescaler Register
0x400	FPOutMap0	UINT16	Front Panel Output 0 Mapping Register
0x402	FPOutMap1	UINT16	Front Panel Output 1 Mapping Register
0x404	FPOutMap2	UINT16	Front Panel Output 2 Mapping Register
0x406	FPOutMap3	UINT16	Front Panel Output 3 Mapping Register
0x440	UnivOutMap0	UINT16	Universal Output 0 Mapping Register
0x442	UnivOutMap1	UINT16	Universal Output 1 Mapping Register
0x444	UnivOutMap2	UINT16	Universal Output 2 Mapping Register
0x446	UnivOutMap3	UINT16	Universal Output 3 Mapping Register
0x448	UnivOutMap4	UINT16	Universal Output 4 Mapping Register
0x44A	UnivOutMap5	UINT16	Universal Output 5 Mapping Register
0x44C	UnivOutMap6	UINT16	Universal Output 6 Mapping Register
0x44E	UnivOutMap7	UINT16	Universal Output 7 Mapping Register
0x450	UnivOutMap8	UINT16	Universal Output 8 Mapping Register
0x452	UnivOutMap9	UINT16	Universal Output 9 Mapping Register
0x480	TBOutMap0	UINT16	Transition Board Output 0 Mapping Register
0x482	TBOutMap1	UINT16	Transition Board Output 1 Mapping Register
0x484	TBOutMap2	UINT16	Transition Board Output 2 Mapping Register
0x486	TBOutMap3	UINT16	Transition Board Output 3 Mapping Register
0x488	TBOutMap4	UINT16	Transition Board Output 4 Mapping Register
0x48A	TBOutMap5	UINT16	Transition Board Output 5 Mapping Register
0x48C	TBOutMap6	UINT16	Transition Board Output 6 Mapping Register
0x48E	TBOutMap7	UINT16	Transition Board Output 7 Mapping Register
0x490	TBOutMap8	UINT16	Transition Board Output 8 Mapping Register
0x492	TBOutMap9	UINT16	Transition Board Output 9 Mapping Register
0x494	TBOutMap10	UINT16	Transition Board Output 10 Mapping Register
0x496	TBOutMap11	UINT16	Transition Board Output 11 Mapping Register
0x498	TBOutMap12	UINT16	Transition Board Output 12 Mapping Register
0x49A	TBOutMap13	UINT16	Transition Board Output 13 Mapping Register
0x49C	TBOutMap14	UINT16	Transition Board Output 14 Mapping Register
0x49E	TBOutMap15	UINT16	Transition Board Output 15 Mapping Register
0x500	FPInMap0	UINT32	Front Panel Input 0 Mapping Register
0x504	FPInMap1	UINT32	Front Panel Input 1 Mapping Register
0x508	FPInMap2	UINT32	Front Panel Input 2 Mapping Register
0x540	UnivInMap0	UINT32	Front Panel Universal Input 0 Map Register
0x544	UnivInMap1	UINT32	Front Panel Universal Input 1 Map Register
0x548	UnivInMap2	UINT32	Front Panel Universal Input 2 Map Register
0x54C	UnivInMap3	UINT32	Front Panel Universal Input 3 Map Register

Address	Register	Type	Description
0x550	UnivInMap4	UINT32	Front Panel Universal Input 4 Map Register
0x554	UnivInMap5	UINT32	Front Panel Universal Input 5 Map Register
0x558	UnivInMap6	UINT32	Front Panel Universal Input 6 Map Register
0x55C	UnivInMap7	UINT32	Front Panel Universal Input 7 Map Register
0x560	UnivInMap8	UINT32	Front Panel Universal Input 8 Map Register
0x564	UnivInMap9	UINT32	Front Panel Universal Input 9 Map Register
0x600	TBInMap0	UINT32	Transition Board Input 0 Mapping Register
0x604	TBInMap1	UINT32	Transition Board Input 1 Mapping Register
0x608	TBInMap2	UINT32	Transition Board Input 2 Mapping Register
0x60C	TBInMap3	UINT32	Transition Board Input 3 Mapping Register
0x610	TBInMap4	UINT32	Transition Board Input 4 Mapping Register
0x614	TBInMap5	UINT32	Transition Board Input 5 Mapping Register
0x618	TBInMap6	UINT32	Transition Board Input 6 Mapping Register
0x61C	TBInMap7	UINT32	Transition Board Input 7 Mapping Register
0x620	TBInMap8	UINT32	Transition Board Input 8 Mapping Register
0x624	TBInMap9	UINT32	Transition Board Input 9 Mapping Register
0x628	TBInMap10	UINT32	Transition Board Input 10 Mapping Register
0x62C	TBInMap11	UINT32	Transition Board Input 11 Mapping Register
0x630	TBInMap12	UINT32	Transition Board Input 12 Mapping Register
0x634	TBInMap13	UINT32	Transition Board Input 13 Mapping Register
0x638	TBInMap14	UINT32	Transition Board Input 14 Mapping Register
0x63C	TBInMap15	UINT32	Transition Board Input 15 Mapping Register
0x800 – 0xFFF	DataBuf		Data Buffer Transmit Memory
0x1000 – 0x10FF	configROM		
0x1100 – 0x11FF	scratchRAM		
0x1200 – 0x12FF	SFPEEPROM		Upstream SFP Transceiver EEPROM contents (SFP address 0xA0)
0x1300 – 0x13FF	SFPDIAG		Upstream SFP Transceiver diagnostics (SFP address 0xA2)
0x2000 – 0x27FF	SegBuf		Segmented Data Buffer Transmit Memory
0x8000 – 0xBFFF	SeqRam0		Sequence RAM 0
0xC000 – 0xFFFF	SeqRam1		Sequence RAM 1

2.17.1. Status Register

address	bit 31	bit 30	bit 29	bit 28	bit 27	bit 26	bit 25	bit 24
0x000	RDB7	RDB6	RDB5	RDB4	RDB3	RDB2	RDB1	RDB0

address	bit 23	bit 22	bit 21	bit 20	bit 19	bit 18	bit 17	bit 16
0x001	TDB7	TDB6	TDB5	TDB4	TDB3	TDB2	TDB1	TDB0

Bit	Function
RDB7	Status of received distributed bus bit 7 (from upstream EVG)
RDB6	Status of received distributed bus bit 6 (from upstream EVG)
RDB5	Status of received distributed bus bit 5 (from upstream EVG)
RDB4	Status of received distributed bus bit 4 (from upstream EVG)
RDB3	Status of received distributed bus bit 3 (from upstream EVG)
RDB2	Status of received distributed bus bit 2 (from upstream EVG)
RDB1	Status of received distributed bus bit 1 (from upstream EVG)
RDB0	Status of received distributed bus bit 0 (from upstream EVG)
TDB7	Status of transmitted distributed bus bit 7
TDB6	Status of transmitted distributed bus bit 6
TDB5	Status of transmitted distributed bus bit 5
TDB4	Status of transmitted distributed bus bit 4
TDB3	Status of transmitted distributed bus bit 3
TDB2	Status of transmitted distributed bus bit 2
TDB1	Status of transmitted distributed bus bit 1
TDB0	Status of transmitted distributed bus bit 0

address	bit 31	bit 30	bit 29	bit 28	bit 27	bit 26	bit 25	bit 24
0x004	EVGEN	RXDIS	RXPWD	FIFORS		SRST	LEMDE	MXCRES

address	bit 23	bit 22	bit 21	bit 20	bit 19	bit 18	bit 17	bit 16
0x005	BCGEN	DCMST						SRALT

Bit	Function
EVGEN	Event Generator Master enable
RXDIS	Disable event reception
RXPWD	Receiver Power down
FIFORS	Reset RX Event Fifo
SRST	Soft reset IP
LEMDE	Little endian mode (cPCI-EVG-300) 0 – PCI core in big endian mode (power up default) 1 – PCI core in little endian mode
MXCRES	Write 1 to reset multiplexed counters
BCGEN	Delay Compensation Beacon generator enable 0 – Beacon generator disabled 1 – Beacon generator enabled, sends out 0x7A events regularly
DCMST	Delay Compensation Master enable 0 – Delay Compensation Master disabled 1 – Delay Compensation Master enabled – sends out delay compensation data packets regularly

Bit	Function
SRALT	(reserved)

2.17.2. Interrupt Flag Register

address	bit 31	bit 30	bit 29	bit 28	bit 27	bit 26	bit 25	bit 24
0x008								
address	**bit 23**	**bit 22**	**bit 21**	**bit 20**	**bit 19**	**bit 18**	**bit 17**	**bit 16**
0x009			IFSOV1	IFSOV0			IFSHF1	IFSHF0
address	**bit 15**	**bit 14**	**bit 13**	**bit 12**	**bit 11**	**bit 10**	**bit 9**	**bit 8**
0x00A			IFSSTO1	IFSSTO0			IFSSTA1	IFSSTA0
address	**bit 7**	**bit 6**	**bit 5**	**bit 4**	**bit 3**	**bit 2**	**bit 1**	**bit 0**
0x00B		IFEXT	IFDBUF				IFFF	IFVIO

Bit	Function
IFSOV1	Sequence RAM 1 sequence roll over interrupt flag
IFSOV0	Sequence RAM 0 sequence roll over interrupt flag
IFSHF1	Sequence RAM 1 sequence halfway through interrupt flag
IFSHF0	Sequence RAM 0 sequence halfway through interrupt flag
IFSSTO1	Sequence RAM 1 sequence stop interrupt flag
IFSSTO0	Sequence RAM 0 sequence stop interrupt flag
IFSSTA1	Sequence RAM 1 sequence start interrupt flag
IFSSTA0	Sequence RAM 0 sequence start interrupt flag
IFEXT	External Interrupt flag
IFDBUF	Data buffer flag
IFFF	RX Event FIFO full flag
IFVIO	Receiver violation flag

2.17.3. Interrupt Enable Register

address	bit 31	bit 30	bit 29	bit 28	bit 27	bit 26	bit 25	bit 24
0x00C	IRQEN	PCIIE						
address	**bit 23**	**bit 22**	**bit 21**	**bit 20**	**bit 19**	**bit 18**	**bit 17**	**bit 16**
0x00D			IESOV1	IESOV0			IESHF1	IESHF0
address	**bit 15**	**bit 14**	**bit 13**	**bit 12**	**bit 11**	**bit 10**	**bit 9**	**bit 8**
0x00E			IESSTO1	IESSTO0			IESSTA1	IESSTA0

address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
0x00F		IEEXT	IEDBUF				IEFF	IEVIO

Bit	Function
IRQEN	Master interrupt enable
PCIIE	PCI core interrupt enable (cPCI-EVG-300) This bit is used by the low level driver to disable further interrupts before the first interrupt has been handled in user space
IESOV1	Sequence RAM 1 sequence roll over interrupt enable
IESOV0	Sequence RAM 0 sequence roll over interrupt enable
IESHF1	Sequence RAM 1 sequence halfway through interrupt enable
IESHF0	Sequence RAM 0 sequence halfway through interrupt enable
IESSTO1	Sequence RAM 1 sequence stop interrupt enable
IESSTO0	Sequence RAM 0 sequence stop interrupt enable
IESSTA1	Sequence RAM 1 sequence start interrupt enable
IESSTA0	Sequence RAM 0 sequence start interrupt enable
IEEXT	External interrupt enable
IEDBUF	Data buffer interrupt enable
IEFF	Event FIFO full interrupt enable
IEVIO	Receiver violation interrupt enable

2.17.4. AC Trigger Control Register

address	bit 23	bit 22	bit 21	bit 20	bit 19	bit 18	bit 17	bit 16
0x011							ACBYP	ACSYNC

address	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
0x012	AC Trigger Divider							

address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
0x013	AC Trigger Phase Shift							

Bit	Function
ACBYP	AC divider and phase shifter bypass (0 = divider/phase shifter enabled, 1 = divider/phase shifter bypassed)
ACSYNC	Synchronization select (0 = event clock, 1 = multiplexed counter 7 output)

2.17.5. AC Trigger Mapping Register

address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
0x017	ACM7	ACM6	ACM5	ACM4	ACM3	ACM2	ACM1	ACM0

Bit	Function
ACM7	If set AC circuit triggers Event Trigger 7
ACM6	If set AC circuit triggers Event Trigger 6
ACM5	If set AC circuit triggers Event Trigger 5
ACM4	If set AC circuit triggers Event Trigger 4
ACM3	If set AC circuit triggers Event Trigger 3
ACM2	If set AC circuit triggers Event Trigger 2
ACM1	If set AC circuit triggers Event Trigger 1
ACM0	If set AC circuit triggers Event Trigger 0

2.17.6. Software Event Register

address	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
0x01A							SWPEND	SWENA

address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
0x01B	Event Code to be sent out							

Bit	Function
SWPEND	Event code waiting to be sent out (read-only). A new event code may be written to the event code register when this bit reads '0'.
SWENA	Enable software event When enabled '1' a new event will be sent out when event code is written to the event code register.

2.17.7. Segmented Data Buffer Control Register

address	bit 31	bit 30	bit 29	bit 28	bit 27	bit 26	bit 25	bit 24
0x01C	SADDR(7:0)							

address	bit 23	bit 22	bit 21	bit 20	bit 19	bit 18	bit 17	bit 16
0x01D				TXCPT	TXRUN	TRIG	ENA	1

address	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
0x01E						DTSZ(10:8)		

address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
0x01F	DTSZ(7:2)						0	0

Bit	Function
SADDR	Transfer Start Segment Address (16 byte segments)
TXCPT	Data Buffer Transmission Complete

Bit	Function
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
DTSZ(10:8)	Data Transfer size 4 bytes to 2k in four byte increments

2.17.8. Data Buffer Control Register

address	bit 31	bit 30	bit 29	bit 28	bit 27	bit 26	bit 25	bit 24
0x020								

address	bit 23	bit 22	bit 21	bit 20	bit 19	bit 18	bit 17	bit 16
0x021				TXCPT	TXRUN	TRIG	ENA	1

address	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
0x022						DTSZ(10:8)		

address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
0x023	DTSZ(7:2)						0	0

Bit	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
DTSZ	Data Transfer size 4 bytes to 2k in four byte increments

2.17.9. Distributed Bus Mapping Register

address	bit 31	bit 30	bit 29	bit 28	bit 27	bit 26	bit 25	bit 24
0x024	DBMAP7(3:0)				DBMAP6(3:0)			

address	bit 23	bit 22	bit 21	bit 20	bit 19	bit 18	bit 17	bit 16
0x025	DBMAP5(3:0)				DBMAP4(3:0)			

address	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
0x026	DBMAP3(3:0)				DBMAP2(3:0)			

address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
0x027	DBMAP1(3:0)				DBMAP0(3:0)			

Bit	Function
DBMAP7(3:0)	Distributed Bus Bit 7 Mapping: 0 – Off, output logic ‘0’ 1 – take bus bit from external input 2 – Multiplexed counter output mapped to distributed bus bit 3 – Distributed bus bit forwarded from upstream EVG
DBMAP6(3:0)	Distributed Bus Bit 7 Mapping (see above for mappings)
DBMAP5(3:0)	Distributed Bus Bit 7 Mapping (see above for mappings)
DBMAP4(3:0)	Distributed Bus Bit 7 Mapping (see above for mappings)
DBMAP3(3:0)	Distributed Bus Bit 7 Mapping (see above for mappings)
DBMAP2(3:0)	Distributed Bus Bit 7 Mapping (see above for mappings)
DBMAP1(3:0)	Distributed Bus Bit 7 Mapping (see above for mappings)
DBMAP0(3:0)	Distributed Bus Bit 7 Mapping (see above for mappings)

2.17.10. Distributed Bus Event Enable Register

address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
0x02B	DBEV7	DBEV6	DBEV5					

Bit	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

2.17.11. FPGA Firmware Version Register

address	bit 31	bit 30	bit 29	bit 28	bit 27	bit 26	bit 25	bit 24
0x02C	EVG = 0x2				Form Factor			

address	bit 23	bit 22	bit 21	bit 20	bit 19	bit 18	bit 17	bit 16
0x02D	Reserved							

address	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
0x02E	Firmware ID							

address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
0x02F	Revision ID							

Bit	Function
Form Factor	0 – CompactPCI 3U 1 – PMC 2 – VME64x 3 – CompactRIO 4 – CompactPCI 6U 6 – PXIe 7 – PCIe 8 – mTCA.4
Firmware ID	0 – Modular Register Map firmware (no delay compensation) 1 – Reserved 2 – Delay Compensation firmware
Revision ID	See end of manual

2.17.12. Timestamp Generator Control Register

address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
0x037							TSGENA	TSGLOAD

Bit	Function
TSGENA	Timestamp Generator Enable (‘0’ = disable, ‘1’ = enable)
TSGLOAD	Timestamp Generator Load new value into Timestamp Counter Write ‘1’ to load new value

2.17.13. Microsecond Divider Register

address	bit 15 … bit 0
0x04E	Rounded integer value of 1 μs * event clock

This register shall be written with an integer value of the event clock rate in MHz. For 100 MHz event clock this register should read 100, for 50 MHz event clock this register should read 50. This value is used to set the parameters for the clock cleaner PLL and e.g. for the phase shifter in the AC input logic.

2.18. Clock Control Register

address	bit 31	bit 30	bit 29	bit 28	bit 27	bit 26	bit 25	bit 24
0x050	PLLLOCK	BWSEL(2:0)				RFSEL(2:0)		

address	bit 23	bit 22	bit 21	bit 20	bit 19	bit 18	bit 17	bit 16
0x051			RFDIV(5:0)					

address	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
0x052							CGLOCK	

Bit	**Function**
PLLLOCK	Clock cleaner locked
BWSEL2:0	PLL Bandwidth Select (see Silicon Labs Si5317 datasheet) 000 – Si5317, BW setting HM (lowest loop bandwidth) 001 – Si5317, BW setting HL 010 – Si5317, BW setting MH 011 – Si5317, BW setting MM 100 – Si5317, BW setting ML (highest loop bandwidth)
RFDIV5-0	External 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 010100 – RF/21 010101 – RF/22 010110 – RF/23 010111 – RF/24 011000 – RF/25 011001 – RF/26 011010 – RF/27 011011 – RF/28 011100 – RF/29 011101 – RF/30 011110 – RF/31 011111 – RF/32

Bit	Function
RFSEL2-0	RF reference select: 000 – Use internal reference (fractional synthesizer) 001 – Use external RF reference (front panel input through divider) 010 – PXIe 100 MHz clock 100 – Use recovered RX clock, Fan-Out mode 101 – Use external RF reference for downstream ports, internal reference for upstream port, Fan-Out mode, event rate down conversion 110 – PXIe 10 MHz clock through clock multiplier 111 – Recovered clock /2 decimate mode, event rate is halved
CGLOCK	Micrel SY87739L reference clock locked (read-only)

2.18.1. Event Analyser Control Register

address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
0x063				EVANE	EVARS	EVAOF	EVAEN	EVACR

Bit	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.

2.18.2. Sequence RAM Control Registers

address	bit 31	bit 30	bit 29	bit 28	bit 27	bit 26	bit 25	bit 24
0x070							SQ0RUN	SQ0ENA

address	bit 23	bit 22	bit 21	bit 20	bit 19	bit 18	bit 17	bit 16
0x071	SQ0XTR	SQ0XEN	SQ0SWT	SQ0SNG	SQ0REC	SQ0RES	SQ0DIS	SQ0EN

address	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
0x072	SQSWMASK(3:0)				SQSWENA(3:0)			

address	bit 7						bit 0
0x073	SQ0TSEL						

address	bit 31	bit 30	bit 29	bit 28	bit 27	bit 26	bit 25	bit 24
0x074							SQ1RUN	SQ1ENA

address	bit 23	bit 22	bit 21	bit 20	bit 19	bit 18	bit 17	bit 16
0x075	SQ1XTR	SQ1XEN	SQ1SWT	SQ1SNG	SQ1REC	SQ1RES	SQ1DIS	SQ1EN

address	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
0x076	SQSWMASK(3:0)				SQSWENA(3:0)			

address	bit 7						bit 0
0x077	SQ1TSEL						

Bit	**Function**
SQxRUN	Sequence RAM running flag (read-only)
SQxENA	Sequence RAM enabled flag (read_only)
SQxSWT	Sequence RAM software trigger, write ‘1’ to trigger
SQxSNG	Sequence RAM single mode
SQxREC	Sequence RAM recycle mode
SQxRES	Sequence RAM reset, write ‘1’ to reset
SQxDIS	Sequence RAM disable, write ‘1’ to disable
SQxEN	Sequence RAM enable, write ‘1’ to enable/arm
SQxXEN	Sequence RAM allow external enable, ‘1’ - allow
SQxXTR	Sequence RAM allow external trigger enable, ‘1’ - allow
SQSWMASK	Sequence RAM SW mask register, the mask bits are common for all RAMS
SQSWENA	Sequence RAM SW enable register, the mask bits are common for all RAMS
SQxTSEL	Sequence RAM trigger select: 0 – trigger from MXC0 1 – trigger from MXC1 2 – trigger from MXC2 3 – trigger from MXC3 4 – trigger from MXC4 5 – trigger from MXC5 6 – trigger from MXC6 7 – trigger from MXC7 16 – trigger from AC synchronization logic 17 – trigger from sequence RAM 0 software trigger 18 – trigger from sequence RAM 1 software trigger 19 – trigger always immediately when enabled 24 – trigger from sequence RAM 0 external trigger 25 – trigger from sequence RAM 1 external trigger 31 – trigger disabled (default after power up)

Bit	Function

2.18.3. SY87739L Fractional Divider Configuration Word

Configuration Word	Frequency with 24 MHz reference oscillator
0x0891C100	142.857 MHz
0x00DE816D	125 MHz
0x00FE816D	124.95 MHz
0x0C928166	124.9087 MHz
0x018741AD	119 MHz
0x072F01AD	114.24 MHz
0x049E81AD	106.25 MHz
0x008201AD	100 MHz
0x025B41ED	99.956 MHz
0x0187422D	89.25 MHz
0x0082822D	81 MHz
0x0106822D	80 MHz
0x019E822D	78.900 MHz
0x018742AD	71.4 MHz
0x0C9282A6	62.454 MHz
0x009743AD	50 MHz
0x0C25B43AD	49.978 MHz
0x0176C36D	49.965 MHz

2.19. SPI Configuration Flash Registers

address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
0x0A3	SPIDATA(7:0)							

address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
0x0A7	E	RRDY	TRDY	TMT	TOE	ROE	OE	SSO

Bit	Function
SPIDATA(7:0)	Read SPI data byte / Write SPI data byte
E	Overrun Error flag
RRDY	Receiver ready, if ‘1’ data byte waiting in SPI_DATA
TRDY	Transmitter ready, if ‘1’ SPI_DATA is ready to accept new transmit data byte
TMT	Transmitter empty, if ‘1’ data byte has been transmitted
TOE	Transmitter overrun error
ROE	Receiver overrun error

Bit	Function
OE	Output enable for SPI pins, ‘1’ enable SPI pins
SSO	Slave select output enable for SPI slave device, ‘1’ device selected

2.19.1. Event Trigger Registers

address	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
0x102								EVEN0

address	bit 7 … bit 0
0x103	EVCD0(7:0)

address	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
0x106								EVEN1

address	bit 7 … bit 0
0x107	EVCD1(7:0)

address	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
0x10A								EVEN2

address	bit 7 … bit 0
0x10B	EVCD2(7:0)

address	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
0x10E								EVEN3

address	bit 7 … bit 0
0x10F	EVCD3(7:0)

address	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
0x112								EVEN4

address	bit 7 … bit 0
0x113	EVCD4(7:0)

address	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
0x116								EVEN5

address	bit 7 … bit 0
0x117	EVCD5(7:0)

address	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
0x11A								EVEN6

address	bit 7 … bit 0
0x11B	EVCD6(7:0)

address	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
0x11E								EVEN7

address	bit 7 … bit 0
0x11F	EVCD7(7:0)

Bit	Function
EVENx	Enable Event Trigger x
EVCDx	Event Trigger Code for Event trigger x

2.19.2. Multiplexed Counter Registers

address	bit 31	bit 30	bit 29	bit 28	bit 27	bit 26	bit 25	bit 24
0x180	MXC0	MXCP0						

address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
0x183	MX0EV7	MX0EV6	MX0EV5	MX0EV4	MX0EV3	MX0EV2	MX0EV1	MX0EV0

address	bit 31 … bit 0
0x184	Multiplexed Counter 0 Prescaler

address	bit 31	bit 30	bit 29	bit 28	bit 27	bit 26	bit 25	bit 24
0x188	MXC1	MXCP1						

address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
0x18B	MX1EV7	MX1EV6	MX1EV5	MX1EV4	MX1EV3	MX1EV2	MX1EV1	MX1EV0

address	bit 31 … bit 0
0x18C	Multiplexed Counter 1 Prescaler

address	bit 31	bit 30	bit 29	bit 28	bit 27	bit 26	bit 25	bit 24
0x190	MXC2	MXCP2						

address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
0x193	MX2EV7	MX2EV6	MX2EV5	MX2EV4	MX2EV3	MX2EV2	MX2EV1	MX2EV0

address	bit 31 … bit 0
0x194	Multiplexed Counter 2 Prescaler

address	bit 31	bit 30	bit 29	bit 28	bit 27	bit 26	bit 25	bit 24
0x198	MXC3	MXCP3						

address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
0x19B	MX3EV7	MX3EV6	MX3EV5	MX3EV4	MX3EV3	MX3EV2	MX3EV1	MX3EV0

address	bit 31 … bit 0
0x19C	Multiplexed Counter 3 Prescaler

address	bit 31	bit 30	bit 29	bit 28	bit 27	bit 26	bit 25	bit 24
0x1A0	MXC4	MXCP4						

address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
0x1A3	MX4EV7	MX4EV6	MX4EV5	MX4EV4	MX4EV3	MX4EV2	MX4EV1	MX4EV0

address	bit 31 … bit 0
0x1A4	Multiplexed Counter 4 Prescaler

address	bit 31	bit 30	bit 29	bit 28	bit 27	bit 26	bit 25	bit 24
0x1A8	MXC5	MXCP5						

address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
0x1AB	MX5EV7	MX5EV6	MX5EV5	MX5EV4	MX5EV3	MX5EV2	MX5EV1	MX5EV0

address	bit 31 … bit 0
0x1AC	Multiplexed Counter 5 Prescaler

address	bit 31	bit 30	bit 29	bit 28	bit 27	bit 26	bit 25	bit 24
0x1B0	MXC6	MXCP6						

address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
0x1B3	MX6EV7	MX6EV6	MX6EV5	MX6EV4	MX6EV3	MX6EV2	MX6EV1	MX6EV0

address	bit 31 … bit 0
0x1B4	Multiplexed Counter 6 Prescaler

address	bit 31	bit 30	bit 29	bit 28	bit 27	bit 26	bit 25	bit 24
0x1B8	MXC7	MXCP7						

address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
0x1BB	MX7EV7	MX7EV6	MX7EV5	MX7EV4	MX7EV3	MX7EV2	MX7EV1	MX7EV0

address	bit 31 … bit 0
0x1BC	Multiplexed Counter 7 Prescaler

Bit	Function
MXCx	Multiplexed counter output status (read-only)
MXPx	Multiplexed counter output polarity
MXxEV7	Map rising edge of multiplexed counter x to send out event trigger 7
MXxEV6	Map rising edge of multiplexed counter x to send out event trigger 6
MXxEV5	Map rising edge of multiplexed counter x to send out event trigger 5
MXxEV4	Map rising edge of multiplexed counter x to send out event trigger 4
MXxEV3	Map rising edge of multiplexed counter x to send out event trigger 3
MXxEV2	Map rising edge of multiplexed counter x to send out event trigger 2
MXxEV1	Map rising edge of multiplexed counter x to send out event trigger 1

Bit	Function
MXxEV0	Map rising edge of multiplexed counter x to send out event trigger 0

2.19.3. Transition Board Output Mapping Registers

address	bit 15 … bit 0
0x480	Transition Board Output 0 Mapping ID (see Table 5 on page 14 for mapping IDs)

address	bit 15 … bit 0
0x482	Transition Board Output 1 Mapping ID

. . .

address	bit 15 … bit 0
0x41E	Transition Board Output 15 Mapping ID

2.19.4. Front Panel Input Mapping Registers

address	bit 31	bit 30	bit 29	bit 28	bit 27	bit 26	bit 25	bit 24
0x500	FP0SQMK(3:0)				FP0SQEEN3:0			
								FP0IRQ

address	bit 23	bit 22	bit 21	bit 20	bit 19	bit 18	bit 17	bit 16
0x501	FP0DB7	FP0DB6	FP0DB5	FP0DB4	FP0DB3	FP0DB2	FP0DB1	FP0DB0

address	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
0x502			FP0SEN1	FP0SEN0			FP0SEQ1	FP0SEQ0

address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
0x503	FP0EV7	FP0EV6	FP0EV5	FP0EV4	FP0EV3	FP0EV2	FP0EV1	FP0EV0

address	bit 31	bit 30	bit 29	bit 28	bit 27	bit 26	bit 25	bit 24
0x504	FP1SQMK(3:0)				FP1SQEEN3:0			
								FP1IRQ

address	bit 23	bit 22	bit 21	bit 20	bit 19	bit 18	bit 17	bit 16
0x505	FP1DB7	FP1DB6	FP1DB5	FP1DB4	FP1DB3	FP1DB2	FP1DB1	FP1DB0

address	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
0x506			FP1SEN1	FP1SEN0			FP1SEQ1	FP1SEQ0

address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
0x507	FP1EV7	FP1EV6	FP1EV5	FP1EV4	FP1EV3	FP1EV2	FP1EV1	FP1EV0

address	bit 31	bit 30	bit 29	bit 28	bit 27	bit 26	bit 25	bit 24
0x508	FP2SQMK(3:0)				FP2SQEEN3:0			
								FP2IRQ

address	bit 23	bit 22	bit 21	bit 20	bit 19	bit 18	bit 17	bit 16
0x509	FP2DB7	FP2DB6	FP2DB5	FP2DB4	FP2DB3	FP2DB2	FP2DB1	FP2DB0

address	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
0x50A			FP2SEN1	FP2SEN0			FP2SEQ1	FP2SEQ0

address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
0x50B	FP2EV7	FP2EV6	FP2EV5	FP2EV4	FP2EV3	FP2EV2	FP2EV1	FP2EV0

Bit	Function
FPxSQMKy	Map Front panel Input x to Sequence Event Mask bit y
FPxSQEENy	Map Front panel Input x to Sequence Event Enable bit y
FPxIRQ	Map Front panel Input x to External Interrupt
FPxDB7	Map Front panel Input x to Distributed Bus bit 7
FPxDB6	Map Front panel Input x to Distributed Bus bit 6
FPxDB5	Map Front panel Input x to Distributed Bus bit 5
FPxDB4	Map Front panel Input x to Distributed Bus bit 4
FPxDB3	Map Front panel Input x to Distributed Bus bit 3
FPxDB2	Map Front panel Input x to Distributed Bus bit 2
FPxDB1	Map Front panel Input x to Distributed Bus bit 1
FPxDB0	Map Front panel Input x to Distributed Bus bit 0
FPxSEN1	Map Front panel Input x to Sequence External Enable 1
FPxSEN0	Map Front panel Input x to Sequence External Enable 0
FPxSEQ1	Map Front panel Input x to Sequence Trigger 1
FPxSEQ0	Map Front panel Input x to Sequence Trigger 0
FPxEV7	Map Front panel Input x to Event Trigger 7
FPxEV6	Map Front panel Input x to Event Trigger 6
FPxEV5	Map Front panel Input x to Event Trigger 5
FPxEV4	Map Front panel Input x to Event Trigger 4
FPxEV3	Map Front panel Input x to Event Trigger 3
FPxEV2	Map Front panel Input x to Event Trigger 2
FPxEV1	Map Front panel Input x to Event Trigger 1
FPxEV0	Map Front panel Input x to Event Trigger 0

2.19.5. Transition Board Input Mapping Registers

address	bit 31	bit 30	bit 29	bit 28	bit 27	bit 26	bit 25	bit 24
0x540	TI0SQMK(3:0)				TI0SQEEN3:0			
								TI0IRQ

address	bit 23	bit 22	bit 21	bit 20	bit 19	bit 18	bit 17	bit 16
0x541	TI0DB7	TI0DB6	TI0DB5	TI0DB4	TI0DB3	TI0DB2	TI0DB1	TI0DB0

address	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
0x542			TI0SEN1	TI0SEN0			TI0SEQ1	TI0SEQ0

address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
0x543	TI0EV7	TI0EV6	TI0EV5	TI0EV4	TI0EV3	TI0EV2	TI0EV1	TI0EV0

address	bit 31	bit 30	bit 29	bit 28	bit 27	bit 26	bit 25	bit 24
0x544	TI1SQMK(3:0)				TI1SQEEN3:0			
								TI1IRQ

address	bit 23	bit 22	bit 21	bit 20	bit 19	bit 18	bit 17	bit 16
0x545	TI1DB7	TI1DB6	TI1DB5	TI1DB4	TI1DB3	TI1DB2	TI1DB1	TI1DB0

address	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
0x546			TI1SEN1	TI1SEN0			TI1SEQ1	TI1SEQ0

address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
0x547	TI1EV7	TI1EV6	TI1EV5	TI1EV4	TI1EV3	TI1EV2	TI1EV1	TI1EV0

. . .

address	bit 31	bit 30	bit 29	bit 28	bit 27	bit 26	bit 25	bit 24
0x55C	TI15SQMK(3:0)				TI15SQEEN3:0			
								TI15IRQ

address	bit 23	bit 22	bit 21	bit 20	bit 19	bit 18	bit 17	bit 16
0x55D	TI15DB7	TI15DB6	TI15DB5	TI15DB4	TI15DB3	TI15DB2	TI15DB1	TI15DB0

address	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
0x55E			TI15SEN1	TI15SEN0			TI15SEQ1	TI15SEQ0

address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
0x55F	TI15EV7	TI15EV6	TI15EV5	TI15EV4	TI15EV3	TI15EV2	TI15EV1	TI15EV0

Bit	**Function**
TIxSQMKy	Map Transition Board Input x to Sequence Event Mask bit y
TIxSQEENy	Map Transition Board Input x to Sequence Event Enable bit y
TIxIRQ	Map Transition Board Input x to External Interrupt
TIxDB7	Map Transition Board Input x to Distributed Bus bit 7
TIxDB6	Map Transition Board Input x to Distributed Bus bit 6
TIxDB5	Map Transition Board Input x to Distributed Bus bit 5
TIxDB4	Map Transition Board Input x to Distributed Bus bit 4
TIxDB3	Map Transition Board Input x to Distributed Bus bit 3
TIxDB2	Map Transition Board Input x to Distributed Bus bit 2
TIxDB1	Map Transition Board Input x to Distributed Bus bit 1
TIxDB0	Map Transition Board Input x to Distributed Bus bit 0

Bit	Function
TIxSEN1	Map Transition Board Input x to Sequence External Enable 1
TIxSEN0	Map Transition Board Input x to Sequence External Enable 0
TIxSEQ1	Map Transition Board Input x to Sequence Trigger 1
TIxSEQ0	Map Transition Board Input x to Sequence Trigger 0
TIxEV7	Map Transition Board Input x to Event Trigger 7
TIxEV6	Map Transition Board Input x to Event Trigger 6
TIxEV5	Map Transition Board Input x to Event Trigger 5
TIxEV4	Map Transition Board Input x to Event Trigger 4
TIxEV3	Map Transition Board Input x to Event Trigger 3
TIxEV2	Map Transition Board Input x to Event Trigger 2
TIxEV1	Map Transition Board Input x to Event Trigger 1
TIxEV0	Map Transition Board Input x to Event Trigger 0

Note: all enabled input signals are OR'ed together. So if e.g. distributed bus bit 0 has two sources from universal input 0 and 1, if either of the inputs is active high also the distributed bus is active high.

2.20. FCT Function Register Map

Address	Register	Type	Description
0x000	Status	UINT32	Status Register
0x004	Control	UINT32	Control Register
0x010	UpDCValue	UINT32	Upstream Data Compensation Delay Value
0x014	FIFODCValue	UINT32	Receive FIFO Data Compensation Delay Value
0x018	IntDCValue	UINT32	FCT Internal Datapath Data Compensation Delay Value
0x040	Port1DCValue	UINT32	Downstream link port 1 loop delay value
0x044	Port2DCValue	UINT32	Downstream link port 2 loop delay value
0x048	Port3DCValue	UINT32	Downstream link port 3 loop delay value
0x04C	Port4DCValue	UINT32	Downstream link port 4 loop delay value
0x050	Port5DCValue	UINT32	Downstream link port 5 loop delay value
0x054	Port6DCValue	UINT32	Downstream link port 6 loop delay value
0x058	Port7DCValue	UINT32	Downstream link port 7 loop delay value
0x05C	Port8DCValue	UINT32	Downstream link port 8 loop delay value
0x1000 – 0x10FF	SFP1EEPROM		Port 1 SFP Transceiver EEPROM contents (SFP address 0xA0)
0x1100 – 0x11FF	SFP1DIAG		Port 1 SFP Transceiver diagnostics (SFP address 0xA2)
0x1200 – 0x12FF	SFP2EEPROM		Port 2 SFP Transceiver EEPROM contents (SFP address 0xA0)

Address	Register	Type	Description
0x1300 – 0x13FF	SFP2DIAG		Port 2 SFP Transceiver diagnostics (SFP address 0xA2)
0x1400 – 0x14FF	SFP3EEPROM		Port 3 SFP Transceiver EEPROM contents (SFP address 0xA0)
0x1500 – 0x15FF	SFP3DIAG		Port 3 SFP Transceiver diagnostics (SFP address 0xA2)
0x1600 – 0x16FF	SFP4EEPROM		Port 4 SFP Transceiver EEPROM contents (SFP address 0xA0)
0x1700 – 0x17FF	SFP4DIAG		Port 4 SFP Transceiver diagnostics (SFP address 0xA2)
0x1800 – 0x18FF	SFP5EEPROM		Port 5 SFP Transceiver EEPROM contents (SFP address 0xA0)
0x1900 – 0x19FF	SFP5DIAG		Port 5 SFP Transceiver diagnostics (SFP address 0xA2)
0x1A00 – 0x1AFF	SFP6EEPROM		Port 6 SFP Transceiver EEPROM contents (SFP address 0xA0)
0x1B00 – 0x1BFF	SFP6DIAG		Port 6 SFP Transceiver diagnostics (SFP address 0xA2)
0x1C00 – 0x1CFF	SFP7EEPROM		Port 7 SFP Transceiver EEPROM contents (SFP address 0xA0)
0x1D00 – 0x1DFF	SFP7DIAG		Port 7 SFP Transceiver diagnostics (SFP address 0xA2)
0x1E00 – 0x1EFF	SFP8EEPROM		Port 8 SFP Transceiver EEPROM contents (SFP address 0xA0)
0x1F00 – 0x1FFF	SFP8DIAG		Port 8 SFP Transceiver diagnostics (SFP address 0xA2)

2.20.1. Status Register

address	bit 23	bit 22	bit 21	bit 20	bit 19	bit 18	bit 17	bit 16
0x001	LINK8	LINK7	LINK6	LINK5	LINK4	LINK3	LINK2	LINK1

address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
0x003	VIO8	VIO7	VIO6	VIO5	VIO4	VIO3	VIO2	VIO1

Bit	Function
LINK8	Downstream Link 8 Status, 1 – link up, 0 – link down
LINK7	Downstream Link 7 Status, 1 – link up, 0 – link down
LINK6	Downstream Link 6 Status, 1 – link up, 0 – link down
LINK5	Downstream Link 5 Status, 1 – link up, 0 – link down
LINK4	Downstream Link 4 Status, 1 – link up, 0 – link down
LINK3	Downstream Link 3 Status, 1 – link up, 0 – link down
LINK2	Downstream Link 2 Status, 1 – link up, 0 – link down

Bit	Function
LINK1	Downstream Link 1 Status, 1 – link up, 0 – link down
VIO8	Downstream Link 8 Violation
VIO7	Downstream Link 7 Violation
VIO6	Downstream Link 6 Violation
VIO5	Downstream Link 5 Violation
VIO4	Downstream Link 4 Violation
VIO3	Downstream Link 3 Violation
VIO2	Downstream Link 2 Violation
VIO1	Downstream Link 1 Violation

2.20.2. Control Register

address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
0x007	CLRV8	CLRV7	CLRV6	CLRV5	CLRV4	CLRV3	CLRV2	CLRV1

Bit	Function
CLRV8	Clear Violation Port 8
CLRV7	Clear Violation Port 7
CLRV6	Clear Violation Port 6
CLRV5	Clear Violation Port 5
CLRV4	Clear Violation Port 4
CLRV3	Clear Violation Port 3
CLRV2	Clear Violation Port 2
CLRV1	Clear Violation Port 1

2.21. Firmware Version Change Log

FW Version	Date	Changes	Affected HW
0200	11.06.2015	- Prototype release	VME-EVM-300
0201	24.09.2015	- Added segmented data buffer - Fixed Port 1 TX polarity	VME-EVM-300
010202	01.10.2015	- Changed receive FIFO delay target to 00060000 - Added LED test mode (production testing) - Removed test signals from TBOUT	VME-EVM-300
010203	23.11.2015	- Added changes for running with a slower clock on fan-out.	VME-EVM-300
020203	18.12.2015	- Changes to data buffer forwarding - Changes for rate conversion forwarding, using internal div/2.	VME-EVM-300

FW Version	Date	Changes	Affected HW
0204	12.01.2016	- /2 rate conversion working on events, dbuf and dbits. - Improvements to delay measurement system.	VME-EVM-300
0205	13.04.2016	- Moved delay compensation segment from segment 0 to last segment in memory. - Fixed front panel TTL input order. - Fixed race condition in segmented memory buffer transmission that caused dropped software buffers.	VME-EVM-300

3. Event Receiver

Event Receivers decode timing events and signals from an optical event stream transmitted by an Event Generator. Events and signals are received at predefined rate the event clock that is usually divided down from an accelerators main RF reference. The event receivers lock to the phase event clock of the Event Generator and are thus phase locked to the RF reference. Event Receivers convert event codes transmitted by an Event Generator to hardware outputs. They can also generate software interrupts and store the event codes with globally distributed timestamps into FIFO memory to be read by a CPU.

3.1. Functional Description

After recovering the event clock the Event Receiver demultiplexes the event stream to 8-bit distributed bus data and 8-bit event codes. The distributed bus may be configured to share its bandwidth with time deterministic data transmission.

3.1.1. Event Decoding

The Event Receiver provides two mapping RAMs of 256 × 128 bits. Only one of the RAMs can be active at a time, however both RAMs may be modified at any time. The event code is applied to the address lines of the active mapping RAM. The 128-bit data programmed into a specific memory location pointed to by the event code determines what actions will be taken.

Event code	Offset	Internal functions	Pulse Triggers	'Set' Pulse	'Reset' Pulse
0x00	0x0000	4 bytes/32 bits	4 bytes/32 bits	4 bytes/32 bits	4 bytes/32 bits
0x01	0x0010	4 bytes/32 bits	4 bytes/32 bits	4 bytes/32 bits	4 bytes/32 bits
0x02	0x0020	4 bytes/32 bits	4 bytes/32 bits	4 bytes/32 bits	4 bytes/32 bits
...	...	...	...	...	...
0xFF	0x0FF0	4 bytes/32 bits	4 bytes/32 bits	4 bytes/32 bits	4 bytes/32 bits

There are 32 bits reserved for internal functions which are by default mapped to the event codes shown in table . The remaining 96 bits control internal pulse generators. For each pulse generator there is one bit to trigger the pulse generator, one bit to set the pulse generator output and one bit to clear the pulse generator output.

Map bit	Default event code	Function
127	n/a	Save event in FIFO
126	n/a	Latch timestamp
125	n/a	Led event
124	n/a	Forward event from RX to TX
123	0x79	Stop event log
122	n/a	Log event
102 to 121	n/a	(Reserved)
101	0x7a	Hearbeat / Beacon event
100	0x7b	Reset Prescalers

Map bit	Default event code	Function
99	0x7d	Timestamp reset event (TS counter reset)
98	0x7c	Timestamp clock event (TS counter increment)
97	0x71	Seconds shift register ‘1’
96	0x70	Seconds shift register ‘0’
80 to 95		(Reserved)
79		Trigger pulse generator 15
...		...
64		Trigger pulse generator 0
48 to 63		(Reserved)
47		Set pulse generator 15 output high
...		...
32		Set pulse generator 0 output high
16 to 31		(Reserved)
15		Reset pulse generator 15 output low
...		...
0		Reset pulse generator 0 output low

3.1.2. Heartbeat Monitor

A heartbeat monitor is provided to receive heartbeat events. Event code $7A is by default set up to reset the heartbeat counter. If no heartbeat event is received the counter times out (approx. 1.6 s) and a heartbeat flag is set. The Event Receiver may be programmed to generate a heartbeat interrupt.

With the Delay Compensation feature added in firmware version 0x0200 the former heartbeat event is used as a delay compensation beacon, but the same event can still be used as a heartbeat only that it is automatically sent out by the Event Master.

3.1.3. Event FIFO and Timestamp Events

The Event System provides a global timebase to attach timestamps to collected data and performed actions. The time stamping system consists of a 32-bit timestamp event counter and a 32-bit seconds counter. The timestamp event counter either counts received timestamp counter clock events or runs freely with a clock derived from the event clock. The event counter is also able to run on a clock provided on a distributed bus bit.

The event counter clock source is determined by the prescaler control register. The timestamp event counter is cleared at the next event counter rising clock edge after receiving a timestamp event counter reset event. The seconds counter is updated serially by loading zeros and ones (see mapping register bits) into a shift register MSB first. The seconds register is updated from the shift register at the same time the timestamp event counter is reset.

The timestamp event counter and seconds counter contents may be latched into a timestamp latch. Latching is determined by the active event map RAM and may be enabled for any event code.

An event FIFO memory is implemented to store selected event codes with attached timing information. The 80-bit wide FIFO can hold up to 511 events. The recorded event is stored along with 32-bit seconds counter contents and 32-bit timestamp event counter contents at the time of reception. The event FIFO as well as the timestamp counter and latch are accessible by software.

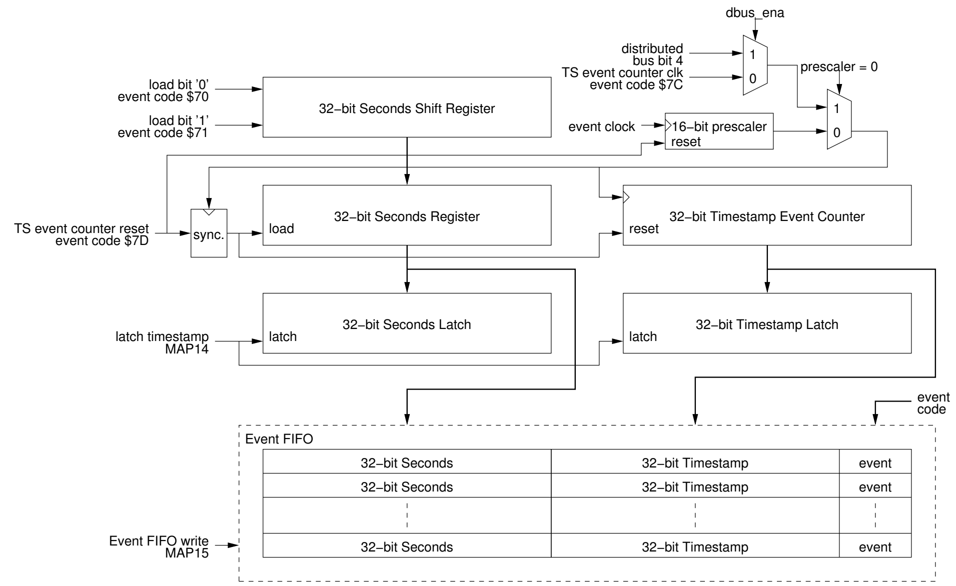

Figure 12: Event FIFO and Timestamping

3.1.4. Event Log

Up to 512 events with timestamping information can be stored in the event log. The log is implemented as a ring buffer and is accessible as a memory region. Logging events can be stopped by an event or software.

3.1.5. Distributed Bus and Data Transmission

The distributed bus is able to carry eight simultaneous signals sampled with half the event clock rate over the fibre optic transmission media. The distributed bus signals may be output on programmable front panel outputs.

The distributed bus bandwidth is shared by transmission of a configurable size data buffer to up to 2 kbytes.

3.1.6. Pulse Generators

The structure of the pulse generation logic is shown in Figure 13. Three signals from the mapping RAM control the output of the pulse: trigger, 'set' pulse and 'reset' pulse. A trigger causes the delay counter to start counting, when the end-of-count is reached the output pulse changes to the 'set' state and the width counter starts counting. At the end of the width count the output pulse is cleared. The mapping RAM signal 'set' and 'reset' cause the output to change state immediately without any delay.

Starting from firmware version 0200 pulse generators can also be triggered from rising edges of distributed bus signals or EVR internal prescalers.

32 bit registers are reserved for both counters and the prescaler, however, the prescaler is not necessarily implemented for all channels and may be hard coded to 1 in case the prescaler is omitted. Software may write 0xFFFFFFFF to these registers and read out the actual width or hard-coded value of the register. For example if the width counter is limited to 16 bits a read will return 0x0000FFFF after a write of 0xFFFFFFFF.

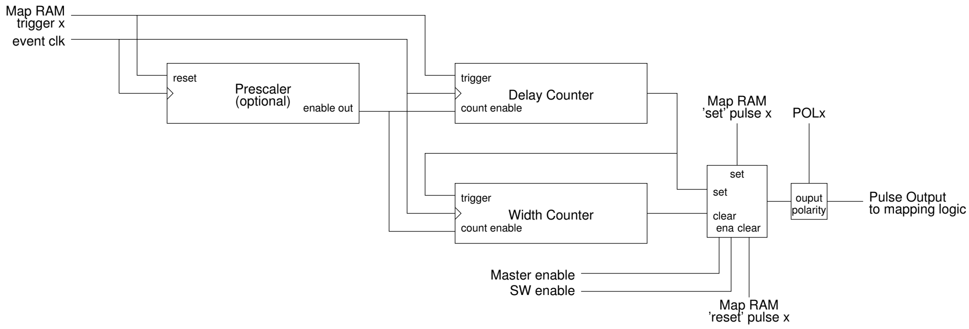

Figure 13: Pulse Generator

3.1.7. Pulse Generator Gates

Depending on firmware revision/form factor a number of pulse generators are configured as event triggered gates only and can be used to mask or enable pulse generator triggers. VME-EVR-300 has four pulse generators configured as gates, pulse generators 28 to 31 which correspond gates 0 to 3.

3.1.8. Prescalers

The Event Receiver provides a number of programmable prescalers. The frequencies are programmable and are derived from the event clock. A special event code reset prescalers $7B causes the prescalers to be synchronously reset, so the frequency outputs will be in same phase across all event receivers.

3.1.9. Programmable Front Panel, Universal I/O and Backplane Connections

All outputs are programmable: each pulse generator output, prescaler and distributed bus bit can be mapped to any output. Starting with firmware version 0200 each output can have two sources which are logically OR'ed together. The mapping for a single source is shown in table below.

Each output has a two byte mapping register and each byte corresponds a single source. An unused mapping source should be set to 63 (0x3f). In case of a bidirectional signal to tri-state set both bytes to 61 (0x3d).

Table 13: Output mapping values

Mapping ID	Signal
0 to n-1	Pulse generator output (number n of pulse generators depends on HW and firmware version)
n to 31	(Reserved)
32	Distributed bus bit 0 (DBUS0)
...	...
39	Distributed bus bit 7 (DBUS7)
40	Prescaler 0
41	Prescaler 1
42	Prescaler 2

Mapping ID	Signal
43 to 58	(Reserved)
59	Event clock output (only on PXIe-EVR-300)
60	Event clock output with 180° phase shift (only on PXIe-EVR-300)
61	Tri-state output (for PCIe-EVR-300DC with input module populated in IFB-300's Universal I/O slot)
62	Force output high (logic 1)
63	Force output low (logic 0)

3.1.10. Front Panel Universal I/O Slots

Universal I/O slots provide different types of output with exchangeable Universal I/O modules. Each module provides two outputs e.g. two TTL output, two NIM output or two optical outputs. The source for these outputs is selected with mapping registers.

3.1.11. VME-EVR-300 GTX Front Panel Outputs

The VME-EVR-300 has four GTX front panel outputs, two in Universal I/O slot UNIV6/UNIV7 and CML outputs CML0 and CML1. The GTX Outputs provide low jitter signals with special outputs. The outputs can work in different configurations: pulse mode, pattern mode and frequency mode. The difference compared to the CML output of the VME-EVR-230RF is that instead of 20 bits per event clock cycle the GTX outputs have 40 bits per event clock cycle doubling the resolution to 200 ps/bit at an event clock of 125 MHz.

3.1.12. GTX Pulse Mode

The source for these outputs is selected in a similar way than the standard outputs using mapping registers, however, the output logic monitors the state of this signal and distinguishes between state low (00), rising edge (01), high state (11) and falling edge (10). Based on the state a 40 bit pattern is sent out with a bit rate of 40 times the event clock rate.

- When the source for a GTX output is low and was low one event clock cycle earlier (state low), the GTX output repeats the 40 bit pattern stored in pattern_00 register.
- When the source for a GTX output is high and was low one event clock cycle earlier (state rising), the GTX output sends out the 40 bit pattern stored in pattern_01 register.
- When the source for a GTX output is high and was high one event clock cycle earlier (state high), the GTX output repeats the 40 bit pattern stored in pattern_11 register.
- When the source for a GTX output is low and was high one event clock cycle earlier (state falling), the GTX output sends out the 40 bit pattern stored in pattern_10 register.

For an event clock of 125 MHz the duration of one single GTX output bit is 200 ps. These outputs allow for producing fine grained adjustable output pulses and clock frequencies.

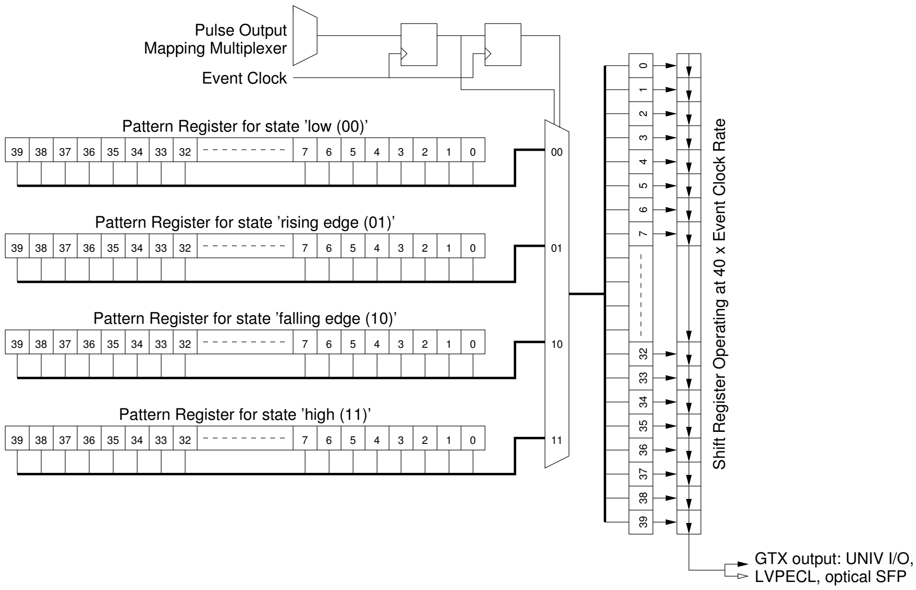

Figure 14: GTX Output

3.1.13. GTX Frequency Mode

In frequency mode one can generate clocks where the clock period can be defined in steps of 1/40th part of the event clock cycle i.e. 200 ps step with an event clock of 125 MHz. There are some limitations, however:

- Clock high time and clock low time must be $\geq$ 40/40th event clock period steps
- Clock high time and clock low time must be < 65536/40th event clock period steps

The clock output can be synchronized by one of the pulse generators, distributed bus signal etc. When a rising edge of the mapped output signal is detected the frequency generator takes its output value from the trigger level bit and the counter value from the trigger position register. Thus one can adjust the phase of the synchronized clock in 1/40th steps of the event clock period.

To change the generated clock phase in respect to the trigger we can select the trigger polarity by bit CMLTL in the CML Control register and the trigger position also in the CML Control register.

3.1.14. GTX Pattern Mode

In pattern mode one can generate arbitrary bit patterns taking into account following:

- The pattern length is a multiple of 40 bits, where each bit is 1/40th of the event clock period
- Maximum length of the arbitrary pattern is 40 × 2048 bits
- A pattern can be triggered from any pulse generator, distributed bus bit etc. When triggered the pattern generator starts sending 40 bit words from the pattern memory sequentially starting from position 0. This goes on until the pattern length set by the samples register has been reached.

- If the pattern generator is in recycle mode the pattern continues immediately from position 0 of the pattern memory.
- If the pattern generator is in single pattern mode, the pattern stops and the 40 bit word from the last position of the pattern memory (2047) is sent out until the pattern generator is triggered again.

3.1.15. Configurable Size Data Buffer

Pre-DC (Delay Compensation) event systems provided a way to to transmit configurable size data packets that may be transmitted over the event system link. The buffer transmission size is configured in the Event Generator to up to 2 kbytes. The Event Receiver is able to receive buffers of any size from 4 bytes to 2 kbytes in four byte (long word) increments. Support for using the event receivers with older pre-DC hardware has been added with firmware version 0205. The newer segmented data buffer has been moved to a different address location in the register map.

The configurable size data buffer works only in an enviroment with a pre-DC event generator and fan-outs. In a delay compensation environment the configurable size data buffer would receive too many buffers for the CPU to handle.

3.1.16. Segmented Data Buffer

Starting with firmware version 0200 the data buffer is divided into 16 byte segments that allow updating only part of the buffer memory with the remaining segments left untouched.

When starting a data transmission the Event Generator first sends the starting segment number that defines the starting address in the buffer. The data buffer address offset is the segment number * 16 bytes. The Event Receiver writes the received bytes into the data buffer and when transmission is complete rises a receive complete flag for the starting segment of the packet transmission. The transmission can overlap several segments, however, the flag is risen only for the starting segment. If there is a checksum mismatch the checksum error flag for the starting segment is set. In case the receive complete flag already was set before the new data was received an segment overflow flag is set. Flags are cleared by writing a ‘1’ to the receive flag. Each segment has a receive data counter and after completion of the transfer the receive data counter of the starting segment is updated with the actual number of bytes received in the transmission.

The procedure to receive a segmented data buffer is following:

- check that receive complete flag for received segment is set
- check that starting segment overflow flag is cleared
- read transmission size from segment receive data counter
- copy segment data from segmented data buffer memory into system RAM
- verify that starting segment overflow flag is still cleared
- clear segment receive complete flag

Starting with firmware 0205 the delay compensation logic uses the last 16 byte segment of the segmented data buffer for delay compensation data.

3.1.17. Interrupt Generation

The Event Receiver has multiple interrupt sources which all have their own enable and flag bits. The following events may be programmed to generate an interrupt:

- Receiver link state change
- Receiver violation: bit error or the loss of signal.
- Lost heartbeat: heartbeat monitor timeout.
- Write operation of an event to the event FIFO.
- Event FIFO is full.
- Data Buffer reception complete.

In addition to the events listed above an interrupt can be generated from one of the pulse generator outputs, distributed bus bits or prescalers. The pulse interrupt can be mapped in a similar way as the front panel outputs.

3.1.18. External Event Input

An external hardware input is provided to be able to take an external pulse to generate an internal event. This event will be handled as any other received event.

3.2. Programmable Reference Clock

The event receiver requires a reference clock to be able to synchronise on the incoming event stream sent by the event generator. For flexibility a programmable reference clock is provided to allow the use of the equipment in various applications with varying frequency requirements.

3.2.1. Fractional Synthesiser

The clock reference for the event receiver is generated on-board the event receiver 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 Pattern	Reference Output	Precision (theoretical)
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

The event receiver reference clock is required to be in ±100 ppm range of the event generator event clock.

3.3. Hardware Configuration Summary

<table>
<tr><th colspan="2"></th><th>VME-EVR-300</th><th>mTCA-EVR-300</th><th>PCIe-EVR-300DC</th></tr>
<tr><td colspan="2">Pulse Generators</td><td>24</td><td>16</td><td>16</td></tr>
<tr><td colspan="2">FP TTL inputs</td><td>2</td><td>2</td><td>0</td></tr>
<tr><td colspan="2">FP TTL outputs</td><td>0</td><td>4</td><td>0</td></tr>
<tr><td colspan="2">FP GTX outputs</td><td>4[1]</td><td>0</td><td>0</td></tr>
<tr><td colspan="2">FP UNIV I/O / slots</td><td>4</td><td>0</td><td>16 / 8[2]</td></tr>
<tr><td colspan="2">FP UNIV GPIO pins / slots</td><td>16/4</td><td>0 / 0</td><td>0 / 0</td></tr>
<tr><td colspan="2">TB Outputs</td><td>16</td><td>32</td><td>0</td></tr>
<tr><td colspan="2">TB Inputs</td><td>16</td><td>32</td><td>0</td></tr>
<tr><td colspan="2">Prescalers</td><td>8 x 32 bit</td><td>8 x 32 bit4</td><td>83 x 3216 bit</td></tr>
<tr><td rowspan="16">Pulse Generator Prescaler, Delay, Pulse Width Range (bits)</td><td>0</td><td>16, 32, 32</td><td>16, 32, 32</td><td>16, 32, 32</td></tr>
<tr><td>1</td><td>16, 32, 32</td><td>16, 32, 32</td><td>16, 32, 32</td></tr>
<tr><td>2</td><td>16, 32, 32</td><td>16, 32, 32</td><td>16, 32, 32</td></tr>
<tr><td>3</td><td>16, 32, 32</td><td>16, 32, 32</td><td>16, 32, 32</td></tr>
<tr><td>4</td><td>0, 32, 16</td><td>0, 32, 16</td><td>0, 32, 16</td></tr>
<tr><td>5</td><td>0, 32, 16</td><td>0, 32, 16</td><td>0, 32, 16</td></tr>
<tr><td>6</td><td>0, 32, 16</td><td>0, 32, 16</td><td>0, 32, 16</td></tr>
<tr><td>7</td><td>0, 32, 16</td><td>0, 32, 16</td><td>0, 32, 16</td></tr>
<tr><td>8</td><td>0, 32, 16</td><td>0, 32, 16</td><td>0, 32, 16</td></tr>
<tr><td>9</td><td>0, 32, 16</td><td>0, 32, 16</td><td>0, 32, 16</td></tr>
<tr><td>10</td><td>0, 32, 16</td><td>0, 32, 16</td><td>0, 32, 16</td></tr>
<tr><td>11</td><td>0, 32, 16</td><td>0, 32, 16</td><td>0, 32, 16</td></tr>
<tr><td>12</td><td>0, 32, 16</td><td>0, 32, 16</td><td>0, 32, 16</td></tr>
<tr><td>13</td><td>0, 32, 16</td><td>0, 32, 16</td><td>0, 32, 16</td></tr>
<tr><td>14</td><td>0, 32, 16</td><td>0, 32, 16</td><td>0, 32, 16</td></tr>
<tr><td>15</td><td>0, 32, 16</td><td>0, 32, 16</td><td>0, 32, 16</td></tr>
</table>

[1] One Universal I/O slot (2 outputs), 2 x CML output
[2] Universal I/O is available on the external I/O box

3.4. VME-EVR-300 Front Panel Connections

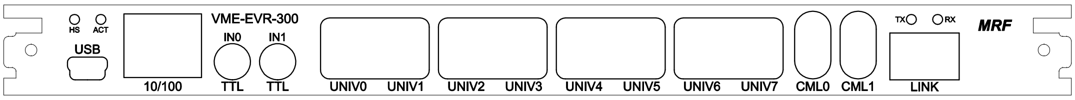

Figure 15: VME-EVR-300 Front Panel

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

Connector / Led	Style	Level	Description
HS	Red Led		Module Failure
HS	Blue Led		Module Powered Down
ACT	3-color Led		SAM3X Activity Led
USB	Micro-USB		SAM3X Serial port / JTAG interface
10/100	RJ45		SAM3X Ethernet Interface
IN0	LEMO	TTL (3.3V / 5V)	FPTTL0 Trigger input
IN1	LEMO	TTL (3.3V / 5V)	FPTTL1 Trigger input
UNIV0/1	Universal slot		Universal Output 0/1
UNIV2/3	Universal slot		Universal Output 2/3
UNIV4/5	Universal slot		Universal Output 4/5
UNIV6/7	Universal slot		Universal Output 6/7 The output signals come through CML/GTX logic block 0/1
CML0	LEMO EPY	CML	Mapped as Universal Output 8 The output signals come through CML/GTX logic block 2
CML1	LEMO EPY	CML	Mapped as Universal Output 9 The output signals come through CML/GTX logic block 3
Link TX (SFP)	LC	Optical 850 nm	Event link Transmit
Link RX (SFP)	LC	Optical 850 nm	Event link Receiver

3.4.1. VME TTL Input Levels

The VME-EVR-300 has two front panel TTL inputs. The inputs have a configurable input termination than can be set by a jumper. The input can be terminated with 50 ohm to ground or 220 ohm to +3.3V. The front panel inputs are 5V tolerant even when powered down.

Input specifications are following:

parameter	**value**
connector type	LEMO EPK.00.250.NTN
input impedance	50 ohm (jumper position Pull-down to GND)
input impedance	220 ohm (jumper position Pull-up to +3.3V)
V_{IH}	> 2.3 V
V_{IL}	< 1.0 V

3.5. PCIe-EVR-300DC and IFB-300 Connections

Due to its small bracket the PCIe-EVR-300DC has only a SFP transceiver and a micro-SCSI type connector to interface to the IFB-300. The cable between the PCIe-EVR-300DC and IFB-300 should be connected/disconnected only when powered down.

Connector / Led	Style	Level	Description
Link TX (SFP)	LC	Optical 850 nm	Event link Transmit Green: TX enable Red: Fract.syn. not locked Blue: Event out
Link RX (SFP) Next to micro-SCSI	LC	Optical 850 nm	Event link Receiver Green: link up Red: link violation detected Blue: event led

The interface board IFB-300 has eight Universal I/O slots which can be populated with various types of Universal I/O modules. If an input module is populated in any slot a jumper has to be mounted in that slot's two pin header with marking "Insert jumper for input module". Please note that if an input module is mounted the corresponding Universal Output Mapping has to be tri-stated. Refer to Table 1: Signal mapping IDs for details.

Universal Slot 0/1 signals are hard-wired to the TTLIN 0/1 signals.

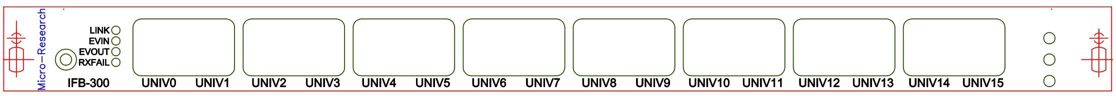

Figure 16: IFB-300 Front Panel

Connector / Led	Style	Level	Description
UNIV0/1	Universal slot		TTL Input / Universal I/O 0/1
UNIV2/3	Universal slot		Universal I/O 2/3
UNIV4/5	Universal slot		Universal I/O 4/5
UNIV6/7	Universal slot		Universal I/O 6/7
UNIV8/9	Universal slot		Universal I/O 8/9
UNIV10/11	Universal slot		Universal I/O 10/11
UNIV12/13	Universal slot		Universal I/O 12/13

Connector / Led	Style	Level	Description
UNIV14/15	Universal slot		Universal I/O 14/15
LINK	Green led		RX link up
EVIN	Yellow led		RX event in
EVOUT	Yellow led		RX event led (mapped)
RXFAIL	Red led		RX violation detected

3.6. mTCA-EVR-300 Connections

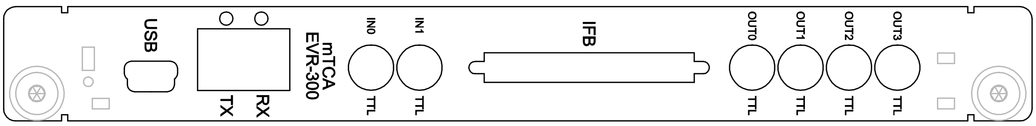

Figure 17: mTCA-EVR-300 Front Panel

Connector / Led	Style	Level	Description
USB	Micro-USB		MMC diagnostics serial port / JTAG interface
Link TX (SFP)	LC	Optical 850 nm	Event link Transmit Green: TX enable Red: Fract.syn. not locked Blue: Event out
Link RX (SFP)	LC	Optical 850 nm	Event link Receiver Green: link up Red: link violation detected Blue: event led
IFB	VHDCI	LVDS	IFB-300 Interface Box connection
IN0	LEMO	TTL	FPTTL0 Trigger input
IN1	LEMO	TTL (3.3V / 5V)	FPTTL1 Trigger input
OUT0	LEMO	3.3V LVTTL	TTL Front panel output 0
OUT1	LEMO	3.3V LVTTL	TTL Front panel output 1
OUT2	LEMO	3.3V LVTTL	TTL Front panel output 2
OUT3	LEMO	3.3V LVTTL	TTL Front panel output 3
RX17	mTCA.4	MLVDS	Backplane output 0
TX17	mTCA.4	MLVDS	Backplane output 1
RX18	mTCA.4	MLVDS	Backplane output 2
TX18	mTCA.4	MLVDS	Backplane output 3
RX19	mTCA.4	MLVDS	Backplane output 4
TX19	mTCA.4	MLVDS	Backplane output 5
RX20	mTCA.4	MLVDS	Backplane output 6
TX20	mTCA.4	MLVDS	Backplane output 7

Connector / Led	Style	Level	Description

3.7. PCIe-EVR-300DC Firmware Upgrade

The PCIe-EVR-300DC firmware image can be upgraded with the following command after loading the driver in Linux:

```
dd if=new_image.bit of=/dev/era1
```

A power cycle is required to load the new configuration image on the PCIe-EVR-300DC.

3.8. Register Map

Address	Register	Type	Description
0x000	Status	UINT32	Status Register
0x004	Control	UINT32	Control Register
0x008	IrqFlag	UINT32	Interrupt Flag Register
0x00C	IrqEnable	UINT32	Interrupt Enable Register
0x010	PulseIrqMap	UINT32	Mapping register for pulse interrupt
0x018	SWEvent	UINT32	Software event register
0x01C	PCIIrqEnable	UINT32	PCI Interrupt Enable Register
0x020	DataBufCtrl	UINT32	Data Buffer Control and Status Register
0x024	TxDataBufCtrl	UINT32	TX Data Buffer Control and Status Register
0x028	TxSegBufCtrl	UINT32	TX Segmented Data Buffer Control and Status Register
0x02C	FWVersion	UINT32	Firmware Version Register
0x040	EvCntPresc	UINT32	Event Counter Prescaler
0x04C	UsecDivider	UINT32	Divider to get from Event Clock to 1 MHz
0x050	ClockControl	UINT32	Event Clock Control Register
0x05C	SecSR	UINT32	Seconds Shift Register
0x060	SecCounter	UINT32	Timestamp Seconds Counter
0x064	EventCounter	UINT32	Timestamp Event Counter
0x068	SecLatch	UINT32	Timestamp Seconds Counter Latch
0x06C	EvCntLatch	UINT32	Timestamp Event Counter Latch
0x070	EvFIFOSec	UINT32	Event FIFO Seconds Register
0x074	EvFIFOEvCnt	UINT32	Event FIFO Event Counter Register
0x078	EvFIFOCode	UINT16	Event FIFO Event Code Register
0x07C	LogStatus	UINT32	Event Log Status Register
0x080	FracDiv	UINT32	Micrel SY87739L Fractional Divider Configuration Word

Address	Register	Type	Description
0x090	GPIODir	UINT32	Front Panel UnivIO GPIO signal direction
0x094	GPIOIn	UINT32	Front Panel UnivIO GPIO input register
0x098	GPIOOut	UINT32	Front Panel UnivIO GPIO output register
0x0A0	SPIData	UINT32	SPI Data Register
0x0A4	SPIControl	UINT32	SPI Control Register
0x0B0	DCTarget	UINT32	Delay Compensation Target Value
0x0B4	DCRxValue	UINT32	Delay Compensation Transmission Path Delay Value
0x0B8	DCIntValue	UINT32	Delay Compensation Internal Delay Value
0x0BC	DCStatus	UINT32	Delay Compensation Status Register
0x0E0	SeqRamCtrl	UINT32	Sequence RAM Control Register
0x100	Prescaler0	UINT32	Prescaler 0 Divider
0x104	Prescaler1	UINT32	Prescaler 1 Divider
0x108	Prescaler2	UINT32	Prescaler 2 Divider
0x10C	Prescaler3	UINT32	Prescaler 3 Divider
0x110	Prescaler4	UINT32	Prescaler 4 Divider
0x114	Prescaler5	UINT32	Prescaler 5 Divider
0x118	Prescaler6	UINT32	Prescaler 6 Divider
0x11C	Prescaler7	UINT32	Prescaler 7 Divider
0x140	PrescTrig0	UINT32	Prescaler 0 Pulse Generator Trigger Register
0x144	PrescTrig1	UINT32	Prescaler 1 Pulse Generator Trigger Register
0x148	PrescTrig2	UINT32	Prescaler 2 Pulse Generator Trigger Register
0x14C	PrescTrig3	UINT32	Prescaler 3 Pulse Generator Trigger Register
0x150	PrescTrig4	UINT32	Prescaler 4 Pulse Generator Trigger Register
0x154	PrescTrig5	UINT32	Prescaler 5 Pulse Generator Trigger Register
0x158	PrescTrig6	UINT32	Prescaler 6 Pulse Generator Trigger Register
0x15C	PrescTrig7	UINT32	Prescaler 7 Pulse Generator Trigger Register
0x180	DBusTrig0	UINT32	DBus Bit 0 Pulse Generator Trigger Register
0x184	DBusTrig1	UINT32	DBus Bit 1 Pulse Generator Trigger Register
0x188	DBusTrig2	UINT32	DBus Bit 2 Pulse Generator Trigger Register
0x18C	DBusTrig3	UINT32	DBus Bit 3 Pulse Generator Trigger Register
0x190	DBusTrig4	UINT32	DBus Bit 4 Pulse Generator Trigger Register
0x194	DBusTrig5	UINT32	DBus Bit 5 Pulse Generator Trigger Register
0x198	DBusTrig6	UINT32	DBus Bit 6 Pulse Generator Trigger Register
0x19C	DBusTrig7	UINT32	DBus Bit 7 Pulse Generator Trigger Register
0x200	Pulse0Ctrl	UINT32	Pulse 0 Control Register
0x204	Pulse0Presc	UINT32	Pulse 0 Prescaler Register
0x208	Pulse0Delay	UINT32	Pulse 0 Delay Register

Address	Register	Type	Description
0x20C	Pulse0Width	UINT32	Pulse 0 Width Register
0x210			Pulse 1 Registers
0x220			Pulse 2 Registers
...	...	...	...
0x32F0			Pulse 3115 Registers
0x400	FPOutMap0	UINT16	Front Panel Output 0 Map Register
0x402	FPOutMap1	UINT16	Front Panel Output 1 Map Register
0x404	FPOutMap2	UINT16	Front Panel Output 2 Map Register
0x406	FPOutMap3	UINT16	Front Panel Output 3 Map Register
0x408	FPOutMap4	UINT16	Front Panel Output 4 Map Register
0x40A	FPOutMap5	UINT16	Front Panel Output 5 Map Register
0x40C	FPOutMap6	UINT16	Front Panel Output 6 Map Register
0x40E	FPOutMap7	UINT16	Front Panel Output 7 Map Register
0x440	UnivOutMap0	UINT16	Front Panel Universal Output 0 Map Register
0x442	UnivOutMap1	UINT16	Front Panel Universal Output 1 Map Register
0x444	UnivOutMap2	UINT16	Front Panel Universal Output 2 Map Register
0x446	UnivOutMap3	UINT16	Front Panel Universal Output 3 Map Register
0x448	UnivOutMap4	UINT16	Front Panel Universal Output 4 Map Register
0x44A	UnivOutMap5	UINT16	Front Panel Universal Output 5 Map Register
0x44C	UnivOutMap6	UINT16	Front Panel Universal Output 6 Map Register
0x44E	UnivOutMap7	UINT16	Front Panel Universal Output 7 Map Register
0x450	UnivOutMap8	UINT16	Front Panel Universal Output 8 Map Register
0x452	UnivOutMap9	UINT16	Front Panel Universal Output 9 Map Register
0x480	TBOutMap0	UINT16	Transition Board Output 0 Map Register
0x482	TBOutMap1	UINT16	Transition Board Output 1 Map Register
0x484	TBOutMap2	UINT16	Transition Board Output 2 Map Register
0x486	TBOutMap3	UINT16	Transition Board Output 3 Map Register
0x488	TBOutMap4	UINT16	Transition Board Output 4 Map Register
0x48A	TBOutMap5	UINT16	Transition Board Output 5 Map Register
0x48C	TBOutMap6	UINT16	Transition Board Output 6 Map Register
0x48E	TBOutMap7	UINT16	Transition Board Output 7 Map Register
0x490	TBOutMap8	UINT16	Transition Board Output 8 Map Register
0x492	TBOutMap9	UINT16	Transition Board Output 9 Map Register
0x494	TBOutMap10	UINT16	Transition Board Output 10 Map Register
0x496	TBOutMap11	UINT16	Transition Board Output 11 Map Register
0x498	TBOutMap12	UINT16	Transition Board Output 12 Map Register
0x49A	TBOutMap13	UINT16	Transition Board Output 13 Map Register
0x49C	TBOutMap14	UINT16	Transition Board Output 14 Map Register

Address	Register	Type	Description
0x49E	TBOutMap15	UINT16	Transition Board Output 15 Map Register
0x4A0	TBOutMap16	UINT16	Transition Board Output 16 Map Register
0x4A2	TBOutMap17	UINT16	Transition Board Output 17 Map Register
0x4A4	TBOutMap18	UINT16	Transition Board Output 18 Map Register
0x4A6	TBOutMap19	UINT16	Transition Board Output 19 Map Register
0x4A8	TBOutMap20	UINT16	Transition Board Output 20 Map Register
0x4AA	TBOutMap21	UINT16	Transition Board Output 21 Map Register
0x4AC	TBOutMap22	UINT16	Transition Board Output 22 Map Register
0x4AE	TBOutMap23	UINT16	Transition Board Output 23 Map Register
0x4B0	TBOutMap24	UINT16	Transition Board Output 24 Map Register
0x4B2	TBOutMap25	UINT16	Transition Board Output 25 Map Register
0x4B4	TBOutMap26	UINT16	Transition Board Output 26 Map Register
0x4B6	TBOutMap27	UINT16	Transition Board Output 27 Map Register
0x4B8	TBOutMap28	UINT16	Transition Board Output 28 Map Register
0x4BA	TBOutMap29	UINT16	Transition Board Output 29 Map Register
0x4BC	TBOutMap30	UINT16	Transition Board Output 30 Map Register
0x4BE	TBOutMap31	UINT16	Transition Board Output 31 Map Register
0x4C0	BPOutMap0	UINT16	Backplane Output 0 Map Register
0x4C2	BPOutMap1	UINT16	Backplane Output 1 Map Register
0x4C4	BPOutMap2	UINT16	Backplane Output 2 Map Register
0x4C6	BPOutMap3	UINT16	Backplane Output 3 Map Register
0x4C8	BPOutMap4	UINT16	Backplane Output 4 Map Register
0x4CA	BPOutMap5	UINT16	Backplane Output 5 Map Register
0x4CC	BPOutMap6	UINT16	Backplane Output 6 Map Register
0x4CE	BPOutMap7	UINT16	Backplane Output 7 Map Register
0x500	FPInMap0	UINT32	Front Panel Input 0 Mapping Register
0x504	FPInMap1	UINT32	Front Panel Input 1 Mapping Register
0x610	GTX0Ctrl	UINT32	UNIV6 / GTX0CML Output Control Register
0x614	GTX0HP	UINT16	UNIV6 / GTX0 Output High Period Count
0x616	GTX0LP	UINT16	UNIV6 / GTX0 Output Low Period Count
0x618	GTX0Samp	UINT32	UNIV6 / GTX0 Output Number of 40 bit word patterns
0x630	GTX1Ctrl	UINT32	UNIV7 / GTX1CML 5 Output Control Register
0x634	GTX1HP	UINT16	UNIV7 / GTX1 Output High Period Count
0x636	GTX1LP	UINT16	UNIV7 / GTX1 Output Low Period Count
0x638	GTX1Samp	UINT32	UNIV7 / GTX1 Output Number of 40 bit word patterns
0x650	GTX2Ctrl	UINT32	CML 0 / GTX2 Output Control Register

Address	Register	Type	Description
0x654	GTX2HP	UINT16	CML 0 / GTX2 Output High Period Count
0x656	GTX2LP	UINT16	CML 0 / GTX2 Output Low Period Count
0x658	GTX2Samp	UINT32	CML 0 / GTX2 Output Number of 40 bit word patterns
0x670	GTX3Ctrl	UINT32	CML1 / GTX3 Output Control Register
0x674	GTX3HP	UINT16	CML1 / GTX3 Output High Period Count
0x676	GTX3LP	UINT16	CML1 / GTX3 Output Low Period Count
0x678	GTX3Samp	UINT32	CML1 / GTX3 Output Number of 40 bit word patterns
0x780 – 0x78F	DataBufSIrqEna		Data Buffer Segment Interrupt Enable Register
0x7A0 – 0x7AF	DataBufCSFlag		Data Buffer Segment Checksum Flags
0x7C0 – 0x7CF	DataBufOVFlag		Data Buffer Segment Overflow Flags
0x7E0 – 0x7EF	DataBufRxFlag		Data Buffer Segment Receive Flags
0x800 – 0xFFF	DataBuf		Data Buffer Receive Memory
0x1000 – 0x17FF			Diagnostics counters
0x1800 – 0x1FFF	TxDataBuf		Data Buffer Transmit Memory
0x2000 – 0x3FFF	EventLog		512 x 16 byte position Event Log
0x4000 – 0x5FFF	MapRam1		Event Mapping RAM 1
0x6000 – 0x7FFF	MapRam2		Event Mapping RAM 2
0x8000 – 0x80FF	configROM		
0x8100 – 0x81FF	scratchRAM		
0x8200 – 0x82FF	SFPEEPROM		SFP Transceiver EEPROM contents (SFP address 0xA0)
0x8300 – 0x83FF	SFPDIAG		SFP Transceiver diagnostics (SFP address 0xA2)
0x8800	DataBufRXSize0	UINT32	Segmented Data Buffer Segment 0 Receive Size Register
0x8804	SDataBufRXSize0	UINT32	Segmented Data Buffer Segment 1 Receive Size Register
...	...	...	...
0x89FC	SDataBufRXSize127	UINT32	Segmented Data Buffer Segment 127 Receive Size Register
0x8F80 – 0x8F8F	SDataBufSIrqEna		Segmented Data Buffer Segment Interrupt Enable Register
0x8FA0 – 0x8FAF	SDataBufCSFlag		Segmented Data Buffer Segment Checksum Flags
0x8FC0 – 0x8FCF	SDataBufOVFlag		Segmented Data Buffer Segment Overflow Flags
0x8FE0 – 0x8FEF	SDataBufRxFlag		Segmented Data Buffer Segment Receive Flags
0x9000 – 0x97FF	SDataBufData		Segmented Data Buffer Segment Data Memory
0xC000 – 0xFFFF	SeqRam		Sequence RAM
0x20000 – 0x23FFF	GTX0MEM		Pattern memory: 16k bytes GTX output 0 (VME-EVR-300)

Address	Register	Type	Description
0x24000 – 0x27FFF	GTX1MEM		Pattern memory: 16k bytes GTX output 1 (VME-EVR-300)
0x28000 – 0x2BFFF	GTX2MEM		Pattern memory: 16k bytes GTX output 2 (VME-EVR-300)
0x2C000 – 0x2FFFF	GTX3MEM		Pattern memory: 16k bytes GTX output 3 (VME-EVR-300)

3.8.1. Status Register

address	bit 31	bit 30	bit 29	bit 28	bit 27	bit 26	bit 25	bit 24
0x000	DBUS7	DBUS6	DBUS5	DBUS4	DBUS3	DBUS2	DBUS1	DBUS0

address	bit 23	bit 22	bit 21	bit 20	bit 19	bit 18	bit 17	bit 16
0x001								LEGVIO

address	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
0x002								

address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
0x003	SFPMOD	LINK	FIFOSTP					

Bit	Function
DBUS7	Read status of DBUS bit 7
DBUS6	Read status of DBUS bit 6
DBUS5	Read status of DBUS bit 5
DBUS4	Read status of DBUS bit 4
DBUS3	Read status of DBUS bit 3
DBUS2	Read status of DBUS bit 2
DBUS1	Read status of DBUS bit 1
DBUS0	Read status of DBUS bit 0
LEGVIO	Legacy VIO (series 100, 200 and 230)
SFPMOD	SFP module status: ‘0’ – plugged in ‘1’ – no module installed
LINK	Link status: ‘0’ – link down ‘1’ – link up
FIFOSTP	Event FIFO stopped flag

3.8.2. Control Register

address	bit 31	bit 30	bit 29	bit 28	bit 27	bit 26	bit 25	bit 24
0x004	EVREN	EVFWD	TXLP	RXLP	OUTEN	SRST	LEMDE	GTXIO

address	bit 23	bit 22	bit 21	bit 20	bit 19	bit 18	bit 17	bit 16
0x005		DCENA						

address	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
0x006		TSDBUS	RSTS			LTS	MAPEN	MAPRS

address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
0x007	LOGRS	LOGEN	LOGDIS	LOGSE	RSFIFO			

Bit	Function
EVREN	Event Receiver Master enable
EVFWD	Event forwarding enable: 0 – Events not forwarded 1 – Events received with forward bit in mapping RAM set are sent back on TX
TXLP	Transmitter loopback: 0 – Receive signal from SFP transceiver (normal operation) 1 – Loopback EVR TX into EVR RX
RXLP	Receiver loopback: 0 – Transmit signal from EVR on SFP transceiver TX 1 – Loopback SFP RX on SFP TX
OUTEN	Output enable for FPGA external components / IFB-300 (cPCI-EVRTG-300, PCIe-EVR-300, PXIe-EVR-300I) 0 – disable outputs 1 – enable outputs
SRST	Soft reset IP
LEMDE	Little endian mode (cPCI-EVR-300, PCIe-EVR-300) 0 – PCI core in big endian mode (power up default) 1 – PCI core in little endian mode
GTXIO	GUN-TX output hardware inhibit override 0 – honor hardware inhibit signal (default) 1 – inhibit override, don't care about hardware inhibit input state
DCENA	Delay compensation mode enable 0 – Delay compensation mode disable (receive FIFO depth controlled by DC Target). 1 – Delay compensation mode enable (receive FIFO depth controlled by DC Target - Datapath Delay).
TSDBUS	Use timestamp counter clock on DBUS4
RSTS	Reset Timestamp. Write 1 to reset timestamp event counter and timestamp latch.
LTS	Latch Timestamp: Write 1 to latch timestamp from timestamp event counter to timestamp latch.
MAPEN	Event mapping RAM enable.
MAPRS	Mapping RAM select bit for event decoding: 0 – select mapping RAM 1 1 – select mapping RAM 2.

Bit	Function
LOGRS	Reset Event Log. Write 1 to reset log.
LOGEN	Enable Event Log. Write 1 to (re)enable event log.
LOGDIS	Disable Event Log. Write 1 to disable event log.
LOGSE	Log Stop Event Enable.
RSFIFO	Reset Event FIFO. Write 1 to clear event FIFO.

3.8.3. Interrupt Flag Register

address	bit 31	bit 30	bit 29	bit 28	bit 27	bit 26	bit 25	bit 24
0x008								

address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
0x00b	IFSEGD	IFLINK	IFDBUF	IFHW	IFEV	IFHB	IFFF	IFVIO

Bit	Function
IFSEGD	Segmented data buffer interrupt flag
IFLINK	Link state change interrupt flag
IFDBUF	Data buffer interrupt flag
IFHW	Hardware interrupt flag (mapped signal)
IFEV	Event interrupt flag
IFHB	Heartbeat interrupt flag
IFFF	Event FIFO full flag
IFVIO	Receiver violation flag

3.8.4. Interrupt Enable Register

address	bit 31	bit 30	bit 29	bit 28	bit 27	bit 26	bit 25	bit 24
0x00c	IRQEN							

address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
0x00f	IESEGD	IELINK	IEDBUF	IEHW	IEEV	IEHB	IEFF	IEVIO

Bit	Function
IRQEN	Master interrupt enable: 0 – disable all interrupts 1 – allow interrupts
IESEGD	Segmented data buffer interrupt flag
IELINK	Link state change interrupt flag
IEDBUF	Data buffer interrupt enable

Bit	Function
IEHW	Hardware interrupt enable (mapped signal)
IEEV	Event interrupt enable
IEHB	Heartbeat interrupt enable
IEFF	Event FIFO full interrupt enable
IEVIO	Receiver violation interrupt enable

3.8.5. Hardware Interrupt Mapping Register

address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
0x013	Mapping ID (see Table 1 for mapping IDs)							

3.8.6. Software Event Register

address	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
0x01A							SWPEND	SWENA

address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
0x01B	Event Code to be inserted into receive event stream							

Bit	Function
SWPEND	Event code waiting to be inserted (read-only). A new event code may be written to the event code register when this bit reads '0'.
SWENA	Enable software event When enabled '1' a new event will be inserted into the receive event stream when event code is written to the event code register.

3.8.7. PCI Interrupt Enable Register

address	bit 31	bit 30	bit 29	bit 28	bit 27	bit 26	bit 25	bit 24
0x01c		PCIIE						

Bit	Function
PCIIE	PCI core interrupt enable (PCIe-EVR-300DC, mTCA-EVR-300) This bit is used by the low level driver to disable further interrupts before the first interrupt has been handled in user space

3.8.8. Receive Data Buffer Control and Status Register

address	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
0x022	DBRX/ DBENA	DBRDY/ DBDIS	DBCS	DBEN	RXSIZE(11:8)			

address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
0x023	RXSIZE(7:0)							

Bit	Function
DBRX	Data Buffer Receiving (read-only)
DBENA	Set-up for Single Reception (write ‘1’ to set-up)
DBRDY	Data Buffer Transmit Complete / Interrupt Flag
DBDIS	Stop Reception (write ‘1’ to stop/disable)
DBCS	Data Buffer Checksum Error (read-only) Flag is cleared by writing ‘1’ to DBRX or DBRDY or disabling data buffer
DBEN	Data Buffer Enable Data Buffer Mode ‘0’ – Distributed bus not shared with data transmission, full speed distributed bus ‘1’ – Distributed bus shared with data transmission, half speed distributed bus
RXSIZE	Data Buffer Received Buffer Size (read-only)

3.8.9. Transmit Data Buffer Control Register

address	bit 31	bit 30	bit 29	bit 28	bit 27	bit 26	bit 25	bit 24
0x024								
address	**bit 23**	**bit 22**	**bit 21**	**bit 20**	**bit 19**	**bit 18**	**bit 17**	**bit 16**
0x025				TXCPT	TXRUN	TRIG	ENA	1
address	**bit 15**	**bit 14**	**bit 13**	**bit 12**	**bit 11**	**bit 10**	**bit 9**	**bit 8**
0x026						DTSZ(10:8)		
address	**bit 7**	**bit 6**	**bit 5**	**bit 4**	**bit 3**	**bit 2**	**bit 1**	**bit 0**
0x027	DTSZ(7:2)						0	0

Bit	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
DTSZ(10:8)	Data Transfer size 4 bytes to 2k in four byte increments

3.8.10. Transmit Segemented Data Buffer Control Register

address	bit 31	bit 30	bit 29	bit 28	bit 27	bit 26	bit 25	bit 24
0x028	SADDR(7:0)							

address	bit 23	bit 22	bit 21	bit 20	bit 19	bit 18	bit 17	bit 16
0x029				TXCPT	TXRUN	TRIG	ENA	MODE

address	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
0x02A						DTSZ(10:8)		

address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
0x02B	DTSZ(7:2)						0	0

Bit	Function
SADDR	Transfer Start Segment Address (16 byte segments)
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
DTSZ(10:8)	Data Transfer size 4 bytes to 2k in four byte increments

3.8.11. FPGA Firmware Version Register

address	bit 31	bit 30	bit 29	bit 28	bit 27	bit 26	bit 25	bit 24
0x02C	EVR = 0x1				Form Factor			

address	bit 23	bit 22	bit 21	bit 20	bit 19	bit 18	bit 17	bit 16
0x02D	Reserved							

address	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
0x02E	Firmware ID							

address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
0x02F	Revision ID							

Bit	Function
Form Factor	0 – CompactPCI 3U 1 – PMC 2 – VME64x 3 – CompactRIO 4 – CompactPCI 6U 6 – PXIe 7 – PCIe 8 – mTCA.4
Firmware ID	0 – Modular Register Map firmware (no delay compensation) 1 – Reserved 2 – Delay Compensation firmware
Revision ID	See end of manual

3.8.12. Clock Control Register

address	bit 31	bit 30	bit 29	bit 28	bit 27	bit 26	bit 25	bit 24
0x050	PLLLOCK	BWSEL(2:0)					INTCLK	

address	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
0x052							CGLOCK	

Bit	Function
PLLLOCK	Clock cleaner PLL Locked (read-only) to receiver recovered clock
BWSEL2:0	PLL Bandwidth Select (see Silicon Labs Si5317 datasheet) 000 – Si5317, BW setting HM (lowest loop bandwidth) 001 – Si5317, BW setting HL 010 – Si5317, BW setting MH 011 – Si5317, BW setting MM 100 – Si5317, BW setting ML (highest loop bandwidth)
INTCLK	Internal Reference Clock Operation Mode. When this bit is set the Event Receiver operates in stand-alone mode and can generate its own events and event act as a simple event generator by forwarding the internally generated events through its SFP TX port.
CGLOCK	Micrel fractional synthesizer SY87739L locked (read-only). This serves as the reference clock for the FPGA internal transceiver and indicates that a valid configuration word has been set in the FracDiv control register.

3.8.13. Event FIFO

Note that reading the FIFO event code registers pulls the event code and timestamp/seconds value from the FIFO for access. The correct order to read an event from FIFO is to first read the event code register and after this the timestamp/seconds registers in any order. Every read access to the FIFO event register pulls a new event from the FIFO if it is not empty.

3.8.14. SY87739L Fractional Divider Configuration Word

Configuration Word	Frequency with 24 MHz reference oscillator
0x0891C100	142.857 MHz
0x00DE816D	125 MHz
0x00FE816D	124.95 MHz
0x0C928166	124.9087 MHz
0x018741AD	119 MHz
0x072F01AD	114.24 MHz
0x049E81AD	106.25 MHz
0x008201AD	100 MHz
0x025B41ED	99.956 MHz
0x0187422D	89.25 MHz
0x0082822D	81 MHz
0x0106822D	80 MHz
0x019E822D	78.900 MHz
0x018742AD	71.4 MHz
0x0C9282A6	62.454 MHz
0x009743AD	50 MHz
0x0C25B43AD	49.978 MHz
0x0176C36D	49.965 MHz

3.8.15. SPI Configuration Flash Registers

address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
0x0A3	SPIDATA(7:0)							

address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
0x0A7	E	RRDY	TRDY	TMT	TOE	ROE	OE	SSO

Bit	Function
SPIDATA(7:0)	Read SPI data byte / Write SPI data byte
E	Overrun Error flag
RRDY	Receiver ready, if ‘1’ data byte waiting in SPI_DATA
TRDY	Transmitter ready, if ‘1’ SPI_DATA is ready to accept new transmit data byte
TMT	Transmitter empty, if ‘1’ data byte has been transmitted
TOE	Transmitter overrun error
ROE	Receiver overrun error
OE	Output enable for SPI pins, ‘1’ enable SPI pins
SSO	Slave select output enable for SPI slave device, ‘1’ device selected

3.8.16. Delay Compensation Status Register

address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
0x0BF					DLYL	DLYS		DCLOCK

Bit	Function
DLYL	Delay setting too long (delay shorter than target)
DLYS	Delay setting too short (delay longer than target)
DCLOCK	Delay compensation locked

3.8.17. Sequence RAM Control Register

address	bit 31	bit 30	bit 29	bit 28	bit 27	bit 26	bit 25	bit 24
0x0E0							SQRUN	SQENA

address	bit 23	bit 22	bit 21	bit 20	bit 19	bit 18	bit 17	bit 16
0x0E1			SQSWT	SQSNG	SQREC	SQRES	SQDIS	SQEN

address	bit 7 … bit 0
0x0E3	SQTSEL

Bit	Function
SQRUN	Sequence RAM running flag (read-only)
SQENA	Sequence RAM enabled flag (read_only)
SQSWT	Sequence RAM software trigger, write ‘1’ to trigger
SQSNG	Sequence RAM single mode
SQREC	Sequence RAM recycle mode
SQRES	Sequence RAM reset, write ‘1’ to reset
SQDIS	Sequence RAM disable, write ‘1’ to disable
SQEN	Sequence RAM enable, write ‘1’ to enable/arm
SQTSEL	Sequence RAM trigger select, see mapping bits in table 13 on page 48.

3.8.18. Prescaler Pulse Trigger Registers

Each bit in the Prescaler Pulse Trigger Register corresponds to one pulse generator trigger. If for instance bit 0 is set, pulse generator 0 gets trigger on each 0 to 1 transition of the corresponding prescaler.

3.8.19. Distributed Bus Pulse Trigger Registers

Each bit in the Distributed Bus Pulse Trigger Register corresponds to one pulse generator trigger. If for instance bit 0 is set, pulse generator 0 gets trigger on each 0 to 1 transition of the corresponding distributed bus bit.

3.8.20. Pulse Generator Registers

address	bit 31	bit 30	bit 29	bit 28	bit 27	bit 26	bit 25	bit 24
0x200	PxMASK(7:0)							

address	bit 23	bit 22	bit 21	bit 20	bit 19	bit 18	bit 17	bit 16
0x201	PxENA(7:0)							

address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
0x203	PxOUT	PxSWS	PxSWR	PxPOL	PxMRE	PxMSE	PxMTE	PxENA

address	bit 31	bit 0
0x204	Pulse Generator 0 Prescaler Register	

address	bit 31	bit 0
0x208	Pulse Generator 0 Delay Register	

address	bit 31	bit 0
0x20C	Pulse Generator 0 Width Register	

Bit	Function
PxMASK(7:0)	Pulse HW Mask Register 0 – HW masking disabled 1 – HW masking enabled. When corresponding gate bit is active '1' pulse triggers are blocked
PxENA(7:0)	Pulse HW Enable Register 0 – HW enabling inactive 1 – HW enabling active. When corresponding gate bit is inactive '0' pulse triggers are blocked
PxOUT	Pulse Generator Output (read-only)
PxSWS	Pulse Generator Software Set
PxSWC	Pulse Generator Software Reset
PxPOL	Pulse Generator Output Polarity 0 – normal polarity 1 – inverted polarity
PxMRE	Pulse Generator Event Mapping RAM Reset Event Enable 0 – Reset events disabled 1 – Mapped Reset Events reset pulse generator output
PxMSE	Pulse Generator Event Mapping RAM Set Event Enable 0 – Set events disabled 1 – Mapped Set Events set pulse generator output
PxMTE	Pulse Generator Event Mapping RAM Trigger Event Enable 0 – Event Triggers disabled 1 – Mapped Trigger Events trigger pulse generator

Bit	Function
PxENA	Pulse Generator Enable 0 – generator disabled 1 – generator enabled

3.8.21. Front Panel Input Mapping Registers

address	bit 31	bit 30	bit 29	bit 28	bit 27	bit 26	bit 25	bit 24
0x500	FPIN0		EXTLV0	BCKLE0	EXTLE0	EXTED0	BCKEV0	EXTEV0

address	bit 23	bit 22	bit 21	bit 20	bit 19	bit 18	bit 17	bit 16
0x501	T0DB7	T0DB6	T0DB5	T0DB4	T0DB3	T0DB2	T0DB1	T0DB0

address	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
0x502	Backward Event Code Register for front panel input 0							

address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
0x503	External Event Code Register for front panel input 0							

address	bit 31	bit 30	bit 29	bit 28	bit 27	bit 26	bit 25	bit 24
0x504	FPIN1		EXTLV1	BCKLE1	EXTLE1	EXTED1	BCKEV1	EXTEV1

address	bit 23	bit 22	bit 21	bit 20	bit 19	bit 18	bit 17	bit 16
0x505	T1DB7	T1DB6	T1DB5	T1DB4	T1DB3	T1DB2	T1DB1	T1DB0

address	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
0x506	Backward Event Code Register for front panel input 1							

address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
0x507	External Event Code Register for front panel input 1							

Bit	Function
FPINx	Front panel Input x state. 0 – low 1 – high
EXTLVx	Backward HW Event Level Sensitivity for input x 0 – active high 1 – active low
BCKLEx	Backward HW Event Level Trigger enable for input x 0 – disable level events 1 – enable level events, send out backward event code every 1 us when input is active (see EXTLVx for level sensitivity)
EXTLEx	External HW Event Level Trigger enable for input x 0 – disable level events 1 – enable level events, apply external event code to active mapping RAM every 1 us when input is active (see EXTLVx for level sensitivity)

Bit	Function
EXTEDx	Backward HW Event Edge Sensitivity for input x 0 – trigger on rising edge 1 – trigger on falling edge
BCKEVx	Backward HW Event Edge Trigger Enable for input x 0 – disable backward HW event 1 – enable backward HW event, send out backward event code on detected edge of hardware input (see EXTEDx bit for edge)
EXTEVx	External HW Event Enable for input x 0 – disable external HW event 1 – enable external HW event, apply external event code to active mapping RAM on edge of hardware input
TxDB7-TxDB0	Backward distributed bus bit enable: 0 – disable distributed bus bit 1 – enable distributed bus bit control from hardware input: e.g. when TxDB7 is '1' the hardware input x state is sent out on distributed bus bit 7.

3.8.22. CML/GTX Output Control Register

address	bit 31 … bit 16
0x610	Frequency mode trigger position

address	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
0x612					GTX3MD	GTX2MD	GTXPH1	GTXPH0

address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
0x613	CMLRC	CMLTL	CMLMD(1:0)			CMLRES	CMLPWD	CMLENA

Bit	Function
GTX3MD	GUN-TX-300 Mode (cPCI-EVRTG-300 only) 0 – CML/GTX Mode 1 – SFP output in GUN-TX-300 Mode
GTX2MD	GUN-TX-203 Mode (cPCI-EVRTG-300 only) 0 – CML/GTX Mode 1 – SFP output in GUN-TX-203 Mode
GTXPH1:0	GUN-TX-203 Trigger output phase shift (cPCI-EVRTG-300 only) 00 – no delay 01 – output pulse delayed by ¼ event clock period (˜2 ns) 10 – output pulse delayed by ½ event clock period (˜4 ns) 11 – output pulse delayed by ¾ event clock period (˜6 ns)
CMLRC	CML Pattern recycle
CMLTL	CML Frequency mode trigger level
CMLMD	CML Mode Select: 00 = classic mode 01 = frequency mode 10 = pattern mode 11 = undefined
CMLRES	CML Reset 1 = reset CML output (default on EVR power up) 0 = normal operation
CMLPWD	CML Power Down 1 = CML outputs powered down (default on EVR power up) 0 = normal operation
CMLENA	CML Enable 0 = CML output disabled (default on EVR power up) 1 = CML output enabled

3.8.23. Data Buffer Segment Interrupt Enable Register

address	bit 31	bit 30	bit 29	bit 28	bit 27	bit 26	bit 25	bit 24
0x8F80	DBIE00	DBIE01	DBIE02	DBIE03	DBIE04	DBIE05	DBIE06	DBIE07

address	bit 23	bit 22	bit 21	bit 20	bit 19	bit 18	bit 17	bit 16
0x8F81	DBIE08	DBIE09	DBIE0A	DBIE0B	DBIE0C	DBIE0D	DBIE0E	DBIE0F

. . .

address	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
0x8F8E	DBIE70	DBIE71	DBIE72	DBIE73	DBIE74	DBIE75	DBIE76	DBIE77

address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
0x8F8F	DBIE78	DBIE79	DBIE7A	DBIE7B	DBIE7C	DBIE7D	DBIE7E	DBIE7F

Bit	Function
DBIExx	Data Buffer Segment (16-byte segments) Interrupt Enable: 0 – Interrupt for segment disabled 1 – Interrupt for segment enabled An interrupt will occur when the segment's receive flag is active. To enable Data Buffer interrupts the IEDBUF bit in the Interrupt Enable Register has to be set.

3.8.24. Data Buffer Checksum Flag Register

address	bit 31	bit 30	bit 29	bit 28	bit 27	bit 26	bit 25	bit 24
0x8FA0	DBCS00	DBCS01	DBCS02	DBCS03	DBCS04	DBCS05	DBCS06	DBCS07

address	bit 23	bit 22	bit 21	bit 20	bit 19	bit 18	bit 17	bit 16
0x8FA1	DBCS08	DBCS09	DBCS0A	DBCS0B	DBCS0C	DBCS0D	DBCS0E	DBCS0F

...

address	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
0x8FAE	DBCS70	DBCS71	DBCS72	DBCS73	DBCS74	DBCS75	DBCS76	DBCS77

address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
0x8FAF	DBCS78	DBCS79	DBCS7A	DBCS7B	DBCS7C	DBCS7D	DBCS7E	DBCS7F

Bit	Function
DBCSxx	Data Buffer Segment (16-byte segments) Checksum Flag: 0 – Checksum OK 1 – Checksum error This flag is cleared by writing a '1' into the segment's DBRXxx bit in the DataBufRxFlag register.

3.8.25. Data Buffer Overflow Flag Register

address	bit 31	bit 30	bit 29	bit 28	bit 27	bit 26	bit 25	bit 24
0x8FC0	DBOV00	DBOV01	DBOV02	DBOV03	DBOV04	DBOV05	DBOV06	DBOV07

address	bit 23	bit 22	bit 21	bit 20	bit 19	bit 18	bit 17	bit 16
0x8FC1	DBOV08	DBOV09	DBOV0A	DBOV0B	DBOV0C	DBOV0D	DBOV0E	DBOV0F

...

address	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
0x8FCE	DBOV70	DBOV71	DBOV72	DBOV73	DBOV74	DBOV75	DBOV76	DBOV77

address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
0x8FCF	DBOV78	DBOV79	DBOV7A	DBOV7B	DBOV7C	DBOV7D	DBOV7E	DBOV7F

Bit	Function
DBOVxx	Data Buffer Segment (16-byte segments) Overflow Flag: 0 – No overflow condition 1 – Overflow: a new packet has been received before the DBRX flag for this segment was cleared This flag is cleared by writing a ‘1’ into the segment’s DBRXxx bit in the DataBufRxFlag register.

3.8.26. Data Buffer Receive Flag Register

address	bit 31	bit 30	bit 29	bit 28	bit 27	bit 26	bit 25	bit 24
0x8FE0	DBRX00	DBRX01	DBRX02	DBRX03	DBRX04	DBRX05	DBRX06	DBRX07

address	bit 23	bit 22	bit 21	bit 20	bit 19	bit 18	bit 17	bit 16
0x8FE1	DBRX08	DBRX09	DBRX0A	DBRX0B	DBRX0C	DBRX0D	DBRX0E	DBRX0F

. . .

address	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
0x8FEE	DBRX70	DBRX71	DBRX72	DBRX73	DBRX74	DBRX75	DBRX76	DBRX77

address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
0x8FEF	DBRX78	DBRX79	DBRX7A	DBRX7B	DBRX7C	DBRX7D	DBRX7E	DBRX7F

Bit	Function
DBRXxx	Data Buffer Segment (16-byte segments) Receive Flag: 0 – No packet received 1 – Data packet received in this segment This flag is cleared by writing a ‘1’ into the segment’s DBRXxx bit.

3.8.27. SFP Module EEPROM and Diagnostics

Small Form Factor Pluggable (SFP) transceiver modules provide a means to identify the module by accessing an EEPROM. As an advanced feature some modules also support reading dynamic information including module temperature, receive and transmit power levels etc. from the module. The EVR gives access to all of this information through a memory window of 2×256 bytes. The first 256 bytes consist of the EEPROM values and the rest of the advanced values.

Byte #	Field size	Notes	Value
		BASE ID FIELDS	
0	1	Type of serial transceiver	0x03 = SFP transceiver
1	1	Extended identifier of type serial transceiver	0x04 = serial ID module definition
2	1	Code for connector type	0x07 = LC
3 – 10	8	Code for electronic compatibility or optical compatibility	

Byte #	Field size	Notes	Value
11	1	Code for serial encoding algorithm	
12	1	Nominal bit rate, units of 100 MBits/sec	
13	1	Reserved	
14	1	Link length supported for 9/125 µm fiber, units of km	
15	1	Link length supported for 9/125 µm fiber, units of 100 m	
16	1	Link length supported for 50/125 µm fiber, units of 10 m	
17	1	Link length supported for 62.5/125 µm fiber, units of 10 m	
18	1	Link length supported for copper, units of meters	
19	1	Reserved	
20 – 35	16	SFP transceiver vendor name (ASCII)	
36	1	Reserved	
37 – 39	3	SFP transceiver vendor IEEE company ID	
40 – 55	16	Part number provided by SFP transceiver vendor (ASCII)	
56 – 59	4	Revision level for part number provided by vendor (ASCII)	
60 – 62	3	Reserved	
63	1	Check code for Base ID Fields	
		EXTENDED ID FIELDS	
64 – 65	2	Indicated which optional SFP signals are implemented	
66	1	Upper bit rate margin, units of %	
67	1	Lower bit rate margin, units of %	
68 – 83	16	Serial number provided by vendor (ASCII)	
84 – 91	8	Vendor's manufacturing date code	
92 – 94	3	Reserved	
95	1	Check code for the Extended ID Fields	
		VENDOR SPECIFIC ID FIELDS	
96 – 127	32	Vendor specific data	
128 – 255		Reserved	
		ENHANCED FEATURE SET MEMORY	
256 – 257	2	Temp H Alarm	Signed twos complement integer in increments of 1/256 °C

Byte #	Field size	Notes	Value
258 – 259	2	Temp L Alarm	Signed twos complement integer in increments of 1/256 °C
260 – 261	2	Temp H Warning	Signed twos complement integer in increments of 1/256 °C
262 – 263	2	Temp L Warning	Signed twos complement integer in increments of 1/256 °C
264 – 265	2	VCC H Alarm	Supply voltage decoded as unsigned integer in increments of 100 µV
266 – 267	2	VCC L Alarm	Supply voltage decoded as unsigned integer in increments of 100 µV
268 – 269	2	VCC H Warning	Supply voltage decoded as unsigned integer in increments of 100 µV
270 – 271	2	VCC L Warning	Supply voltage decoded as unsigned integer in increments of 100 µV
272 – 273	2	Tx Bias H Alarm	Laser bias current decoded as unsigned integer in increment of 2 µA
274 – 275	2	Tx Bias L Alarm	Laser bias current decoded as unsigned integer in increment of 2 µA
276 – 277	2	Tx Bias H Warning	Laser bias current decoded as unsigned integer in increment of 2 µA
278 – 279	2	Tx Bias L Warning	Laser bias current decoded as unsigned integer in increment of 2 µA
280 – 281	2	Tx Power H Alarm	Transmitter average optical power decoded as unsigned integer in increments of 0.1 µW
282 – 283	2	Tx Power L Alarm	Transmitter average optical power decoded as unsigned integer in increments of 0.1 µW
284 – 285	2	Tx Power H Warning	Transmitter average optical power decoded as unsigned integer in increments of 0.1 µW
286 – 287	2	Tx Power L Warning	Transmitter average optical power decoded as unsigned integer in increments of 0.1 µW
288 – 289	2	Rx Power H Alarm	Receiver average optical power decoded as unsigned integer in increments of 0.1 µW
290 – 291	2	Rx Power L Alarm	Receiver average optical power decoded as unsigned integer in increments of 0.1 µW
292 – 293	2	Rx Power H Warning	Receiver average optical power decoded as unsigned integer in increments of 0.1 µW

Byte #	Field size	Notes	Value
294 – 295	2	Rx Power L Warning	Receiver average optical power decoded as unsigned integer in increments of 0.1 μW
296 – 311	16	Reserved	
312 – 350		External Calibration Constants	
351	1	Checksum for Bytes 256 – 350	
352 – 353	2	Real Time Temperature	Signed twos complement integer in increments of 1/256 °C
354 – 355	2	Real Time VCC Power SupplyVoltage	Supply voltage decoded as unsigned integer in increments of 100 μV
356 – 357	2	Real Time Tx Bias Current	Laser bias current decoded as unsigned integer in increment of 2 μA
358 – 359	2	Real Time Tx Power	Transmitter average optical power decoded as unsigned integer in increments of 0.1 μW
360 – 361	2	Real Time Rx Power	Receiver average optical power decoded as unsigned integer in increments of 0.1 μW
362 – 365	4	Reserved	
366	1	Status/Control	bit 7: TX_DISABLE State bit 6 – 3: Reserved bit 2: TX_FAULT State bit 1: RX_LOS State bit 0: Data Ready (Bar)
367	1	Reserved	
368	1	Alarm Flags	bit 7: Temp High Alarm bit 6: Temp Low Alarm bit 5: VCC High Alarm bit 4: VCC Low Alarm bit 3: Tx Bias High Alarm bit 2: Tx Bias Low Alarm bit 1: Tx Power High Alarm bit 0: Tx Power Low Alarm
369	1	Alarm Flags cont.	bit 7: Rx Power High Alarm bit 6: Rx Power Low Alarm bit 5 – 0: Reserved
370 – 371	2	Reserved	
372	1	Warning Flags	bit 7: Temp High Warning bit 6: Temp Low Warning bit 5: VCC High Warning bit 4: VCC Low Warning bit 3: Tx Bias High Warning bit 2: Tx Bias Low Warning bit 1: Tx Power High Warning bit 0: Tx Power Low Warning

Byte #	Field size	Notes	Value
373	1	Warning Flags cont.	bit 7: Rx Power High Warning bit 6: Rx Power Low Warning bit 5 – 0: Reserved
374 – 511		Reserved/Vendor Specific	

3.9. Firmware Version Change Log

FW Version	Date	Changes	Affected HW
0200	11.06.2015	- Prototype release	VME-EVR-300
0201	24.09.2015	- Added segmented data buffer block status flags - Changed delay compensation FIFO depth from 2k to 4k event cycles - Added DCM modulation to improve jitter performance	VME-EVR-300
0203	12.01.2016	- Delay compensation amendments, non-GTX outputs are compensated properly	VME-EVR-300
0204	25.01.2016	- First release for PCIe-EVR-300DC - Fixed segmented data buffer flag writes	all
0204	03.02.2016	- Fixed initial values of GTX outputs - GTX output aligment	VME-EVR-300
0205	07.04.2016	- Changed PCIe-EVR-300DC class code to 0x118000. - Moved delay compensation data from first segment to last segment. - Fixed dual output mapping for transition board outputs. - Added backplane signals to mTCA-EVR. - Added delay compensation disabled mode to be able to use DC capable EVRs with pre-DC EVG and fan-outs.	all
0206	12.08.2016	- Relocated segmented data buffer to new address location. - Replaced earlier data buffer in its original position (maintaining compatibility with 230 series protocol). - Changed segmented data buffer protocol to use K28.2 as a start symbol	all
0207	30.08.2016	- Added stand-alone capability: using its internal reference the EVR can now operate as a stand-alone pulse generator without event link. - EVR can operate as a simple EVG by forwarding internal events - Added software event capability - Added one EVG type sequencer	all

FW Version	Date	Changes	Affected HW

4. Examples

4.1. Event Receiver Standalone Operation

Starting from firmware version 0207 capability to use the EVR as a stand-alone unit has been added. Functionality includes:

- Using the internal fractional synthesizer clock as a reference clock
- Generating internal events by software
- Generating internal event by one EVG type sequencer
- Generating internal event by external signals
- Internal events may be sent out on the TX link by setting the FWD bit for each event in the active mapping RAM

The example code below has been written for the mTCA-EVR-300, but with minor changes (remapping the outputs) it can be used for other form factors as well.

```
int evr_sa(volatile struct MrfErRegs *pEr)
{
  int i;

  EvrEnable(pEr, 1);
  if (!EvrGetEnable(pEr))
    {
      printf(ERROR_TEXT "Could not enable EVR!\n");
      return -1;
    }

  EvrSetIntClkMode(pEr, 1);

  /* Build configuration for EVR map RAMS */

  {
    int ram,code;

    /* Setup MAP ram:
       event code 0x01 to 0x04 trigger pulse generators 0 through 3 */
    ram = 0;
    for (i = 0; i < 4; i++)
      {
        code = 1+i;
        EvrSetLedEvent(pEr, ram, code, 1);
        /* Pulse Triggers start at code 1 */
        EvrSetPulseMap(pEr, ram, code, i, -1, -1);
      }

    /* Setup pulse generators and front panel TTL outputs */
    for (i = 0; i < 4; i++)
      {
EvrSetPulseParams(pEr, i, 1, 100, 100);
```

```
EvrSetPulseProperties(pEr, i, 0, 0, 0, 1, 1);
EvrSetFPOutMap(pEr, i, 0x3f00 | i);
      }
  }

  /* Setup Prescaler 0 */
  EvrSetPrescaler(pEr, 0, 0x07ffff);

  /* Write some RAM events */
  EvrSetSeqRamEvent(pEr, 0, 0, 0, 1);
  EvrSetSeqRamEvent(pEr, 0, 1, 0x001ff, 2);
  EvrSetSeqRamEvent(pEr, 0, 2, 0x002ff, 3);
  EvrSetSeqRamEvent(pEr, 0, 3, 0x003ff, 4);
  EvrSetSeqRamEvent(pEr, 0, 4, 0x04000, 0x7f);
  /* Setup sequence RAM to trigger from prescaler 0 */
  EvrSeqRamControl(pEr, 0, 1, 0, 0, 0, C_EVR_SIGNAL_MAP_PRESC+0);

  EvrMapRamEnable(pEr, 0, 1);

  EvrOutputEnable(pEr, 1);

  return 0;
}
```