**Micro-Research Finland Oy**
Välitalontie 83 C, FI-00660 Helsinki, Finland



# Event Receiver (VME-EVR and VME-EVR-RF)

# Technical Reference

# Firmware Version D308

## Contents

# Introduction

Event Receivers (EVR) recover the clock signal from the event stream transmitted by an Event Generator and generate an event clock that is phase locked to the Event Generator event clock and thus 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 module from the VME bus.

## *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. Introduced in EVR firmware version D307, the distributed bus may be configured to share its bandwidth with data transmission.

### Event Decoding

The Event Receiver provides two mapping RAMs of 8 × 16 bits. Only one of the RAMs can be active at a time, the other one may be modified from VME. The event code is applied to the address lines of the active mapping RAM. The 16-bit data programmed into a specific memory location pointed to by the event code determines what actions will be taken. In addition to the mapping RAMs each of bit 0 to 6 of the event code may generate a trigger event which is a pulse with the length of a single event cycle. There are also a few special event codes to reset prescaler outputs, control the timestamp event counter and reset the heartbeat timeout counter.

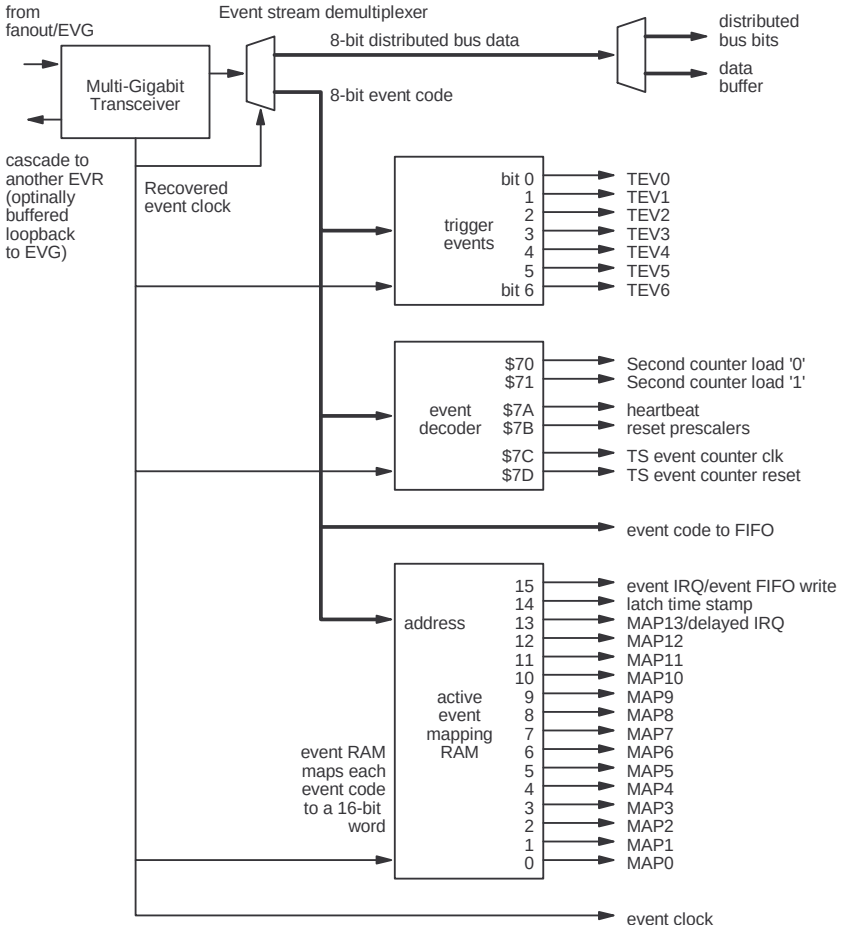


**Figure 1: Event Stream Decoding**

A heartbeat monitor is provided to receive heartbeat events (event code $7A). The heartbeat counter is reset upon receiving the heartbeat event code. If no heartbeat 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.

## 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. Starting from firmware version D308 the event counter is also able to run on a clock provided on the distributed bus bit 4. The event counter clock source is determined by the prescaler value and distributed bus enable register. When the prescaler value is greater than 0 the prescaler output is used. Otherwise the clock source is defined by the distributed bus enable 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 (event code $70) and ones (event code $71) into a shift register MSB first. The seconds register is updated from the shift register at the same time the timestamp event counter is cleared.

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 from VME.

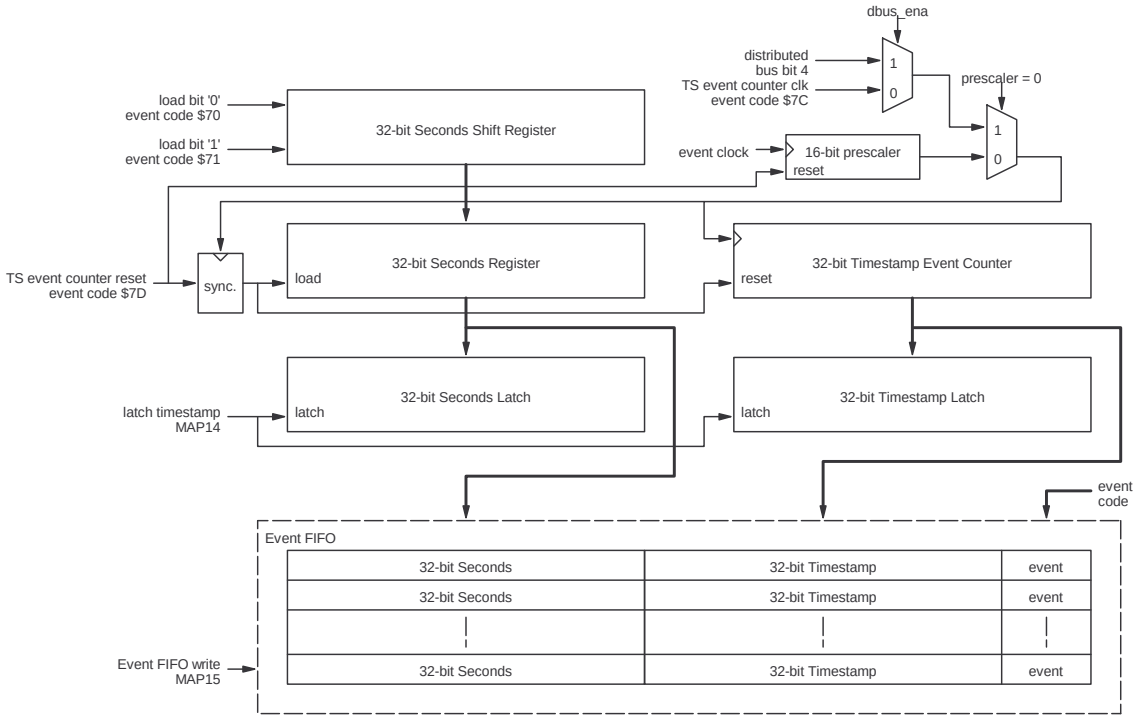


**Figure 2: Event FIFO and Timestamping**

## Distributed Bus and Data Transmission

The distributed bus is able to carry eight simultaneous signals sampled with the event clock rate over the fibre optic transmission media. The distributed bus signals may be output on shared OTP/DBUS signals via the transition board or programmable front panel outputs.

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

## Hardware Outputs

The Event Receiver can generate up to 32 simultaneous outputs on the VME P2 connector. Transition modules provide TTL and optical outputs. The outputs may be selected from multiple sources. There are also a few outputs available in the front panel.

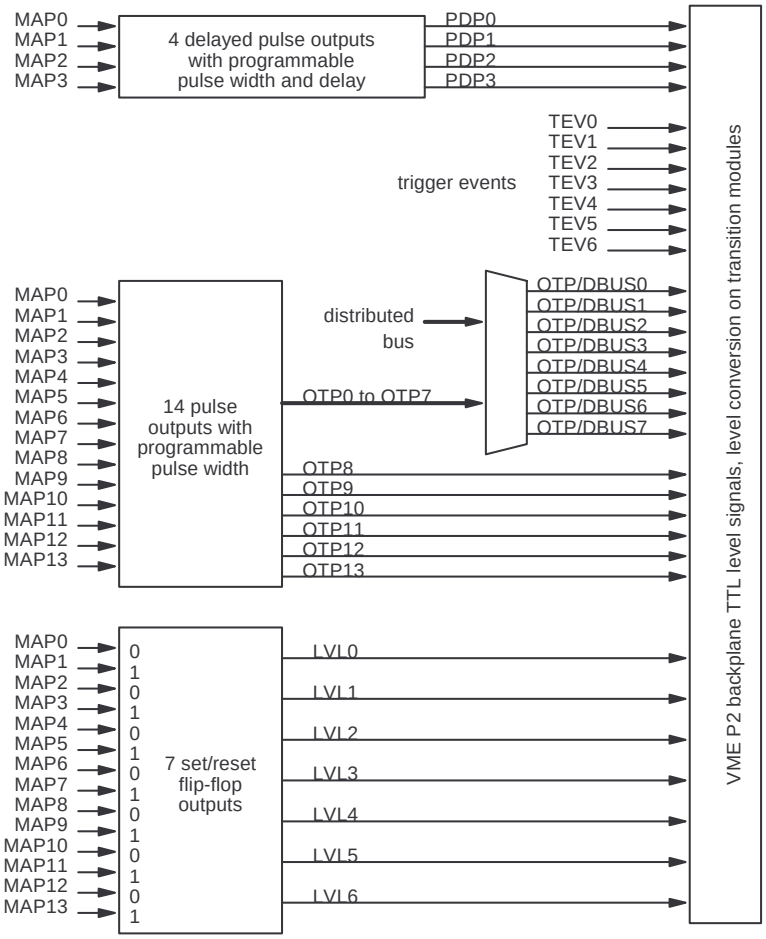


**Figure 3: Event Receiver Hardware Outputs**

There are fourteen pulse outputs (called OTP for historical reasons) of programmable delay, width and polarity. For each channel the pulse delay may be adjusted from 0 to $32^2$-1 event clock cycles (up to 34.3 s with event clock of 125 MHz) and the pulse width may be adjusted from 1 to 65535 event clock cycles (8 ns to 524 µs with event clock of 125 MHz). Eight pulse outputs share the output pin with the distributed bus signals. The mapping (pulse/distributed bus pin) for each of the shared pins may be selected independently.

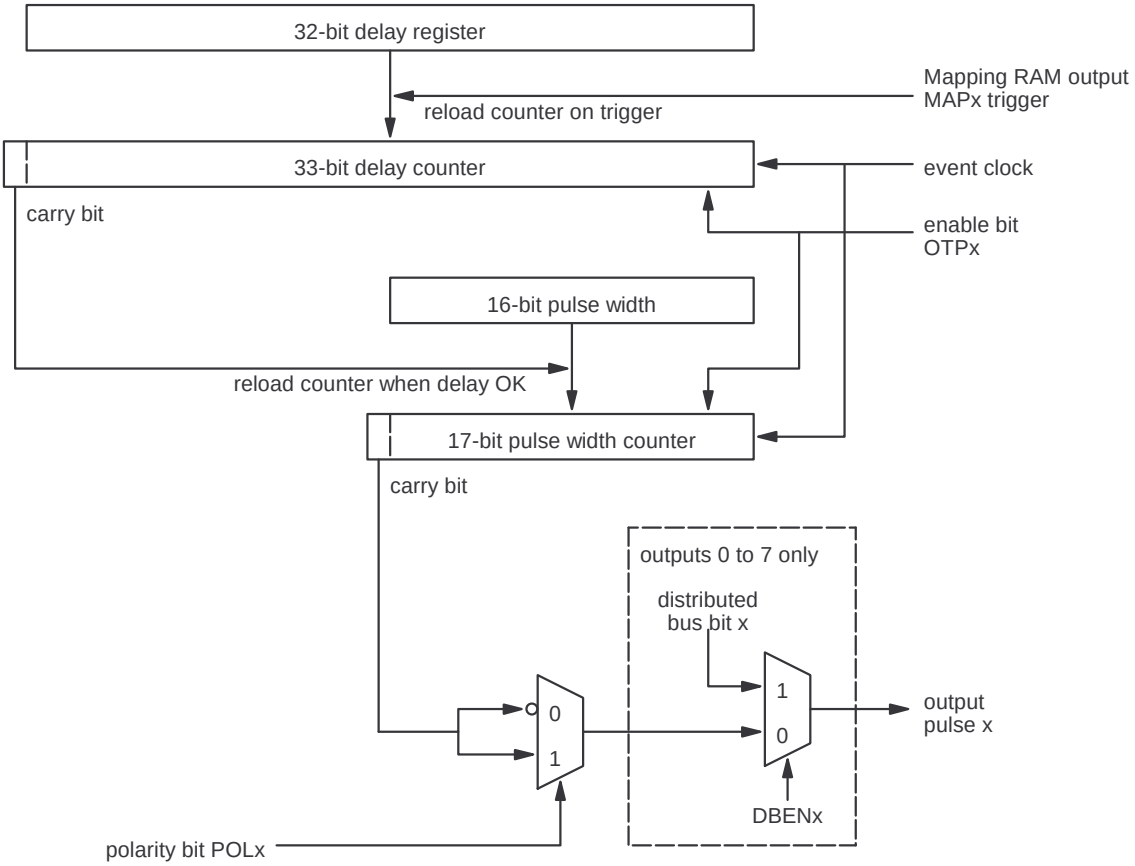


**Figure 4: Programmable Width Pulse Outputs**

Flip flop outputs may be programmed to change their state on desired event codes.

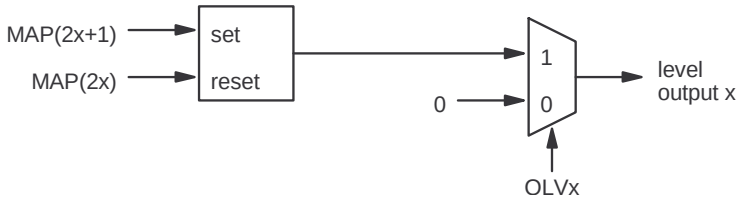


**Figure 5: Level Outputs**

Four extended delayed pulse outputs (called DGP for historical reasons) provide programmable delay, width and polarity like the pulse outputs. The delay and width counters share a 16-bit programmable prescaler which generates counting frequencies from event clock/65536 to the event clock rate. The delay and width counters both are 32-bits wide and thus allow maximum delays and pulse widths up to 625 h at event clock rate of 125 MHz.

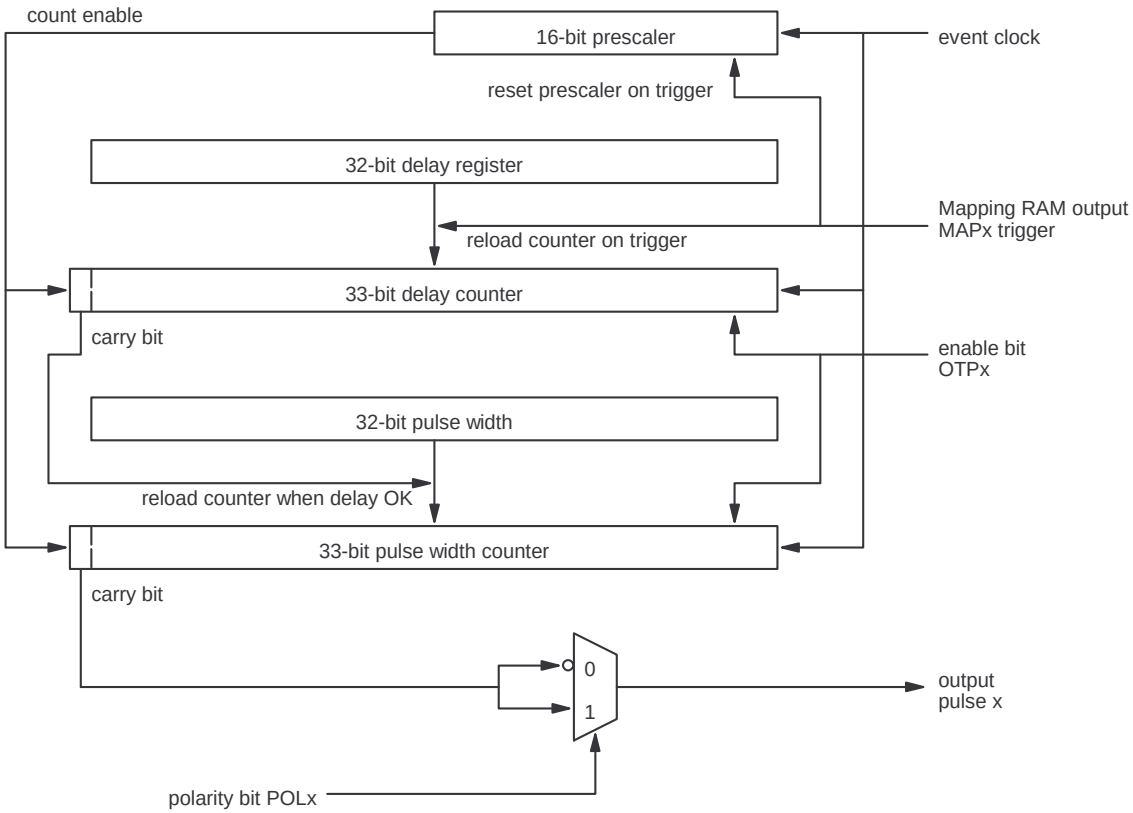


**Figure 6: Programmable Delayed Pulse Outputs**

Bits 0 to 6 of the received event code may be used to generate trigger event outputs. The width of a trigger event is one event clock cycle.

## Prescaler Outputs

The Event Receiver provides three programmable prescaler outputs which may be mapped to front panel outputs. The frequencies 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.

## Configurable Size Data Buffer

Starting from EVR firmware version D307 the reception of data over the event system link is possible. The buffer 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.

Data reception is enabled by changing the distributed bus mode for data transmission (*mode* = 1 in Data Buffer Control Register). This halves the distributed bus update rate. Before a data buffer can be received the data buffer receiver has to be enabled (write *enable* = 1 in control register). This clears the checksum error flag and sets the rx_enable flag. When a data buffer has been received the rx_enable flag is cleared and rx_complete flag is set. If the received and computed checksums do not match the checksum error flag is set.

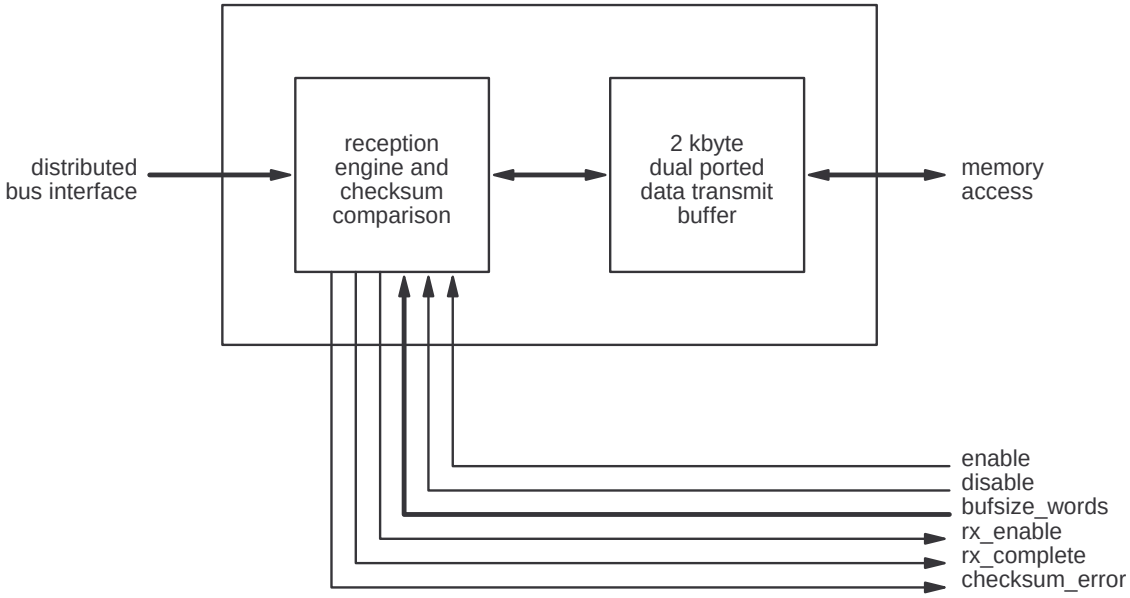


**Figure 7: Data Receive Buffer**

The size of the data buffer transfer can be read from the control register. An interrupt may be generated after reception of a data buffer.

## 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 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 receive complete.

In addition to the events listed above a delayed interrupt is provided. The delayed interrupt is triggered by event map RAM bit 13. A 16-bit prescaler running with the event clock frequency and a 16-bit delay counter determine the interrupt delay.

## 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.

# *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.

## 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/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 |

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

## Jitter Cleaning Circuit

The reference clock is passed on to the FPGA where the RocketIO Multi-Gigabit Transceiver (MGT) locks to the event stream sent by the event generator. Special configuration of the MGT allows the locking to the data stream in always the same phase. Because of the construction of the RocketIO transceiver the recovered bit clock jumps by a maximum of $1/16^{th}$ of a bit period every eight recovered clock cycles. To overcome jitter generated by the RocketIO an external jitter cleaning circuit is introduced.

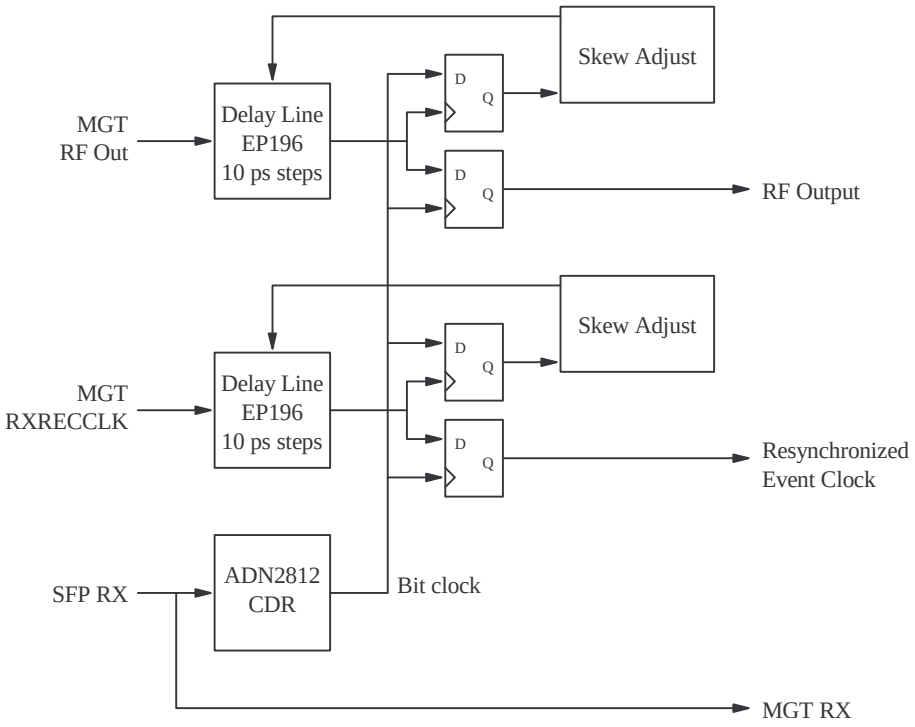


**Figure 8: Jitter Cleaning Circuit for Event Clock and RF Recovery**

## Non-Volatile Storage for Frequency Configuration

The reference clock setting and delay line initialisation values are stored in non-volatile memory inside the IP2022 microcontroller. The part to part differences of the delay lines require tuning of the delays when a new operating frequency is selected.

# Connections

## *Front Panel Connections*

The front panel of the Event Receiver is shown in Figure 9.

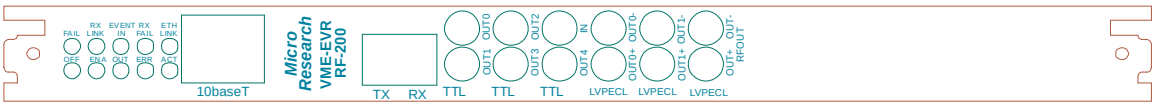


**Figure 9: Event Receiver Front Panel**

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

| Connector / Led | Style | Level | Description |
|---|---|---|---|
| FAIL | Red Led | | Module Failure |
| OFF | Blue Led | | Module not Configured/Powered Down |
| RX LINK | Green Led | | Receiver Link Signal OK |
| ENA | Green Led | | Event Receiver Enabled |

| | | | |
|---|---|---|---|
| EVENT IN | Yellow Led | | Incoming Event (RX) |
| EVENT OUT | Yellow Led | | Active HW output |
| RX FAIL | Red Led | | Receiver Violation |
| ERR | Red Led | | SY87739L reference not locked |
| ETH LINK | Green Led | | 10baseT Activity Led |
| ACT | Yellow Led | | Ubicom IP2022 Active (Flashing) |
| 10baseT | RJ45 | 10baseT | 10baseT Ethernet Connection |
| TX | LC | optical | Transmit Optical Output (TX) |
| RX | LC | optical | Receiver Optical Input (RX) |
| TTL OUT0 | LEMO-EPY | TTL | Programmable TTL Output 0 |
| TTL OUT1 | LEMO-EPY | TTL | Programmable TTL Output 1 |
| TTL OUT2 | LEMO-EPY | TTL | Programmable TTL Output 2 |
| TTL OUT3 | LEMO-EPY | TTL | Programmable TTL Output 3 |
| TTL OUT4 | LEMO-EPY | TTL | Programmable TTL Output 4 |
| TTL IN | LEMO-EPY | TTL | External Event Input |
| LVPECL OUT0± | LEMO-EPY | Diff. LVPECL | Programmable LVPECL Output 0 |
| LVPECL OUT1± | LEMO-EPY | Diff. LVPECL | Programmable LVPECL Output 1 |
| RF OUT± | LEMO-EPY | Diff. LVPECL | Recovered RF Output (VME-EVR-RF only) |

## VME P2 User I/O Pin Configuration

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

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

| | |
|---|---|
| C12 | programmable width pulse / distributed bus output 0 |
| C13 | programmable width pulse / distributed bus output 1 |
| C14 | programmable width pulse / distributed bus output 2 |
| C15 | programmable width pulse / distributed bus output 3 |
| C16 | programmable width pulse / distributed bus output 4 |
| C17 | programmable width pulse / distributed bus output 5 |
| C18 | programmable width pulse / distributed bus output 6 |
| C19 | programmable width pulse / distributed bus output 7 |
| C20 | programmable width pulse output 8 |
| C21 | programmable width pulse output 9 |
| C22 | programmable width pulse output 10 |
| C23 | programmable width pulse output 11 |
| C24 | programmable width pulse output 12 |
| C25 | programmable width pulse output 13 |
| C26 | level output 0 |
| C27 | level output 1 |
| C28 | level output 2 |
| C29 | level output 3 |
| C30 | level output 4 |
| C31 | level output 5 |
| C32 | level output 6 |

# Programming Details

## *CR/CSR Support*

The Event Receiver 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 Receiver 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 Receiver module.

| CR address | Register | VME-EVR | VME-EVR-RF |
|---|---|---|---|
| 0x27, 0x2B, 0x2F | Manufacturer's ID (IEEE OUI) | 0x000EB2 | 0x000EB2 |
| 0x33, 0x37, 0x3B, 0x3F | Board ID | 0x455246C8 | 0x455246C8 |

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

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

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

```
#include “vme64x_cr.h”
int slot = 1;
int slot_evr;
vmeCRFindBoard(slot, MRF_IEEE_OUI, MRF_EVR200RF_BID, &slot_evr);
```

If this function returns `OK`, an Event Receiver board was found in slot `slot_evr`.

## *Event Receiver Function 0 Registers*

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

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

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

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

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

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

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

```
MrfEvrStruct *pEvr;
sysBusToLocalAdrs(VME_AM_SUP_SHORT_IO, (char *) (slot << 11),
                  (void *) pEvr);
```

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

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

## Register Map

| Address Offset | Register | Type | Description |
|---|---|---|---|

| | | | |
|---|---|---|---|
| 0x000 | Control | UINT16 | Control/Status Register |
| 0x002 | MapAddr | UINT16 | Mapping RAM Address Register (8 bit) |
| 0x004 | MapData | UINT16 | Mapping RAM Data Register |
| 0x006 | PulseEnable | UINT16 | Output Pulse Enable Register |
| 0x008 | LevelEnable | UNIT16 | Level Output Enable Register |
| 0x00A | TriggerEnable | UINT16 | Trigger Pulse Enable Register |
| 0x00C | EventCounter | UNIT16 | Timestamp Event Counter (LSW) |
| 0x00E | EventCounter | UINT16 | Timestamp Event Counter (MSW) |
| 0x010 | TSLatch | UINT16 | Timestamp Latch (LSW) |
| 0x012 | TSLatch | UINT16 | Timestamp Latch (LSW) |
| 0x014 | EventFIFO | UINT16 | Series 100 Compatible Event FIFO (LSW)<br>bits 15 - 8 – LSB of Timestamp Counter<br>bits 7 - 0 – Event Code |
| 0x016 | EventFIFO | UINT16 | Series 100 Compatible Event FIFO (MSW)<br>bits 23 - 8 of Timestamp Counter |
| 0x018 | PDPEnable | UINT16 | Delayed Pulse Enable Register |
| 0x01A | PDPSelect | UINT16 | Delayed Pulse Select Register |
| 0x01C | PDPDelay | UINT16 | Series 100 Compatible Multiplexed Delay Register (16 LSB only) |
| 0x01E | PDPWidth | UINT16 | Series 100 Compatible Multiplexed Width Register (16 LSB only) |
| 0x020 | IrqVector | UINT16 | VME Interrupt Vector Register |
| 0x022 | IrqEnable | UINT16 | Interrupt Enable Register |
| 0x024 | DBusEnable | UINT16 | Distributed Bus Enable Register |
| 0x026 | DBusData | UINT16 | Distributed Bus Data Register |
| 0x028 | PDPPrescaler | UINT16 | Multiplexed Prescaler Register |
| 0x02A | EventPrescaler | UINT16 | Event Counter Prescaler Register |
| 0x02C | (Reserved) | UINT16 | (Reserved) |
| 0x02E | FirmwareVersion | UINT16 | Event Receiver Firmware Version Register |
| 0x030 | (Reserved) | UINT32 | (Reserved) |
| 0x034 | (Reserved) | UINT32 | (Reserved) |
| 0x038 | (Reserved) | UINT32 | (Reserved) |
| 0x03C | (Reserved) | UINT32 | (Reserved) |
| 0x040 | FPMap0 | UINT16 | Front Panel TTL Output 0 Map Register |
| 0x042 | FPMap1 | UINT16 | Front Panel TTL Output 1 Map Register |
| 0x044 | FPMap2 | UINT16 | Front Panel TTL Output 2 Map Register |
| 0x046 | FPMap3 | UINT16 | Front Panel TTL Output 3 Map Register |
| 0x048 | FPMap4 | UINT16 | Front Panel TTL Output 4 Map Register |
| 0x04A | FPMap5 | UINT16 | Front Panel LVPECL Output 0 Map Register |
| 0x04C | FPMap6 | UINT16 | Front Panel LVPECL Output 1 Map |

| | | | |
|---|---|---|---|
| | | | Register |
| 0x04E | UsecDivider | UINT16 | Divider to get from Event Clock to 1 MHz |
| 0x050 | ExtEvent | UINT16 | External Event Code Register |
| 0x052 | ClockControl | UINT16 | Event Clock Control Register |
| 0x054 | SecondsSR | UINT32 | Seconds Shift Register |
| 0x058 | TSSec | UINT32 | Timestamp Latch Seconds Register |
| 0x05C | (Reserved) | UINT32 | (Reserved) |
| 0x060 | EvFIFOSec | UINT32 | Event FIFO Seconds Register |
| 0x064 | EvFIFOEvCnt | UINT32 | Event FIFO Event Counter Register |
| 0x068 | OutputPolarity | UINT32 | Output Polarity Register (for pulse outputs) |
| 0x06C | ExtDelay | UINT32 | Multiplexed Delay Register |
| 0x070 | ExtWidth | UINT32 | Multiplexed Width Register |
| 0x074 | Prescaler_0 | UINT16 | Prescaler 0 divider |
| 0x076 | Prescaler_1 | UINT16 | Prescaler 1 divider |
| 0x078 | Prescaler_2 | UINT16 | Prescaler 2 divider |
| 0x07A | DataBufCtrl | UINT16 | Data Buffer Control and Status Register |
| 0x07C | RFPattern | UINT32 | RF Pattern Register |
| 0x080 | FracDiv | UINT32 | SY87739L Fractional Divider Configuration Word |
| 0x084 | RFDelay | UINT32 | RF Recovery Delay |
| 0x088 | RxDelay | UINT32 | Recovered Event Clock Delay |
| 0x08C – 0x7FF | (Reserved) | UINT32 | Reserved |
| 0x800 – 0xFFF | DataBuf | | Data Buffer Receive Memory |

## Control and Status Register

| address | bit 15 | bit 14 | bit 13 | bit 12 | bit 11 | bit 10 | bit 9 | bit 8 |
|---|---|---|---|---|---|---|---|---|
| 0x000 | EVREN | IRQEN | RSTS | HRTBT | IRQFL | LTS | MAPEN | MAPRS |

| Bit | Function |
|---|---|
| EVREN | Event Receiver Master enable. |
| IRQEN | VME irq enable. When 0 all interrupts are disabled. |
| RSTS | Write 1 to reset timestamp event counter and timestamp latch. |
| HRTBT | Lost heartbeat flag. Write 1 to reset. |
| IRQFL | Event FIFO interrupt flag. Write 1 to reset. |
| LTS | 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 - mapping RAM 1, 1 - mapping RAM 2. |

| address | bit 7 | bit 6 | bit 5 | bit 4 | bit 3 | bit 2 | bit 1 | bit 0 |
|---|---|---|---|---|---|---|---|---|
| 0x001 | NFRAM | VMERS | AUTOI | RSADR DIRQ | RSFIFO | FF | FNE RSDIRQ | RXVIO |

| Bit | Function |
|---|---|
| NFRAM | Write 1 to null fill (clear) mapping RAM selected by VMERS. This bit changed to 0 when the selected RAM has been cleared. |
| VMERS | Mapping RAM select bit for VME access. 0 - mapping RAM 1, 1 - mapping RAM 2. |
| AUTOI | Enable mapping RAM auto increment mode. When set the address is automatically incremented by one upon every access from VME. |
| RSADR | Write 1 to reset mapping RAM address register (write only). |
| DIRQ | Delayed interrupt flag. |
| RSFIFO | Write 1 to clear event FIFO. |
| FF | Event FIFO full flag. Write 1 to reset flag. |
| FNE | FIFO not empty flag. Indicated whether there are event in event FIFO. |

## Mapping RAM address register

| address | bit 7 | bit 6 | bit 5 | bit 4 | bit 3 | bit 2 | bit 1 | bit 0 |
|---|---|---|---|---|---|---|---|---|
| 0x003 | Mapping RAM common address register | | | | | | | |

## Mapping RAM data register

| address | bit 15 | bit 0 |
|---|---|---|
| 0x004 | Mapping RAM common data register | |

## Output pulse enable register

| address | bit 15 | bit 14 | bit 13 | bit 12 | bit 11 | bit 10 | bit 9 | bit 8 |
|---|---|---|---|---|---|---|---|---|
| 0x006 | | | OTP13 | OTP12 | OTP11 | OTP10 | OTP9 | OTP8 |

| address | bit 7 | bit 6 | bit 5 | bit 4 | bit 3 | bit 2 | bit 1 | bit 0 |
|---|---|---|---|---|---|---|---|---|
| 0x007 | OTP7 | OTP6 | OTP5 | OTP4 | OTP3 | OTP2 | OTP1 | OTP0 |

| Bit | Function |
|---|---|
| OTPx | Enable programmable width output pulse x. |

## Output level enable register

| address | bit 7 | bit 6 | bit 5 | bit 4 | bit 3 | bit 2 | bit 1 | bit 0 |
|---|---|---|---|---|---|---|---|---|
| 0x009 | | OLV6 | OLV5 | OLV4 | OLV3 | OLV2 | OLV1 | OLV0 |

| Bit | Function |
|---|---|
| OLVx | Enable level output pulse x. |

## Trigger event enable register

| address | bit 7 | bit 6 | bit 5 | bit 4 | bit 3 | bit 2 | bit 1 | bit 0 |
|---|---|---|---|---|---|---|---|---|
| 0x00B | | TEV6 | TEV5 | TEV4 | TEV3 | TEV2 | TEV1 | TEV0 |

| Bit | Function |
|---|---|
| TEVx | Enable trigger event output pulse x. |

## Timestamp event counter register

| address | bit 15 | bit 0 |
|---|---|---|
| 0x00C | Timestamp event counter (LSW, read only) | |

| address | bit 15 | bit 0 |
|---|---|---|
| 0x00E | Timestamp event counter (MSW, read only) | |

## Timestamp event latch register

| address | bit 15 | bit 0 |
|---|---|---|
| 0x010 | Timestamp event latch (LSW, read only) | |

| address | bit 15 | bit 0 |
|---|---|---|
| 0x012 | Timestamp event latch (MSW, read only) | |

## Event FIFO data register

| address | bit 15 | bit 8 |
|---|---|---|
| 0x014 | Event FIFO data register, bits 7 – 0 of timestamp event counter | |

| address | bit 7 | bit 0 |
|---|---|---|
| 0x015 | Event FIFO data register, event code | |

| address | bit 15 | bit 0 |
|---|---|---|
| 0x016 | Event FIFO data register, bits 23 – 8 of timestamp event counter | |

## Programmable delayed pulse output enable register

| address | bit 7 | bit 6 | bit 5 | bit 4 | bit 3 | bit 2 | bit 1 | bit 0 |
|---|---|---|---|---|---|---|---|---|
| 0x019 | POL3 | POL2 | POL1 | POL0 | PDP3 | PDP2 | PDP1 | PDP0 |

| Bit | Function |
|---|---|
| POLx | Delayed pulse output x polarity: 0 - active high, 1 - active low. |
| PDPx | Delayed pulse output x enable. |

## Programmable pulse / delay select register

| address | bit 7 | bit 6 | bit 5 | bit 4 | bit 3 | bit 2 | bit 1 | bit 0 |
|---|---|---|---|---|---|---|---|---|
| 0x01b | | | | DSEL4 | DSEL3 | DSEL2 | DSEL1 | DSEL0 |

| DSEL | Register selected |
|---|---|
| 00000 | Programmable delayed pulse 0 |
| 00001 | Programmable delayed pulse 1 |
| 00010 | Programmable delayed pulse 2 |
| 00011 | Programmable delayed pulse 3 |
| 00100 | Delayed interrupt |
| 10000 | Programmable width pulse 0 |
| 10001 | Programmable width pulse 1 |
| 10010 | Programmable width pulse 2 |
| 10011 | Programmable width pulse 3 |

| | |
|---|---|
| 10100 | Programmable width pulse 4 |
| 10101 | Programmable width pulse 5 |
| 10110 | Programmable width pulse 6 |
| 10111 | Programmable width pulse 7 |
| 11000 | Programmable width pulse 8 |
| 11001 | Programmable width pulse 9 |
| 11010 | Programmable width pulse 10 |
| 11011 | Programmable width pulse 11 |
| 11100 | Programmable width pulse 12 |
| 11101 | Programmable width pulse 13 |

## Programmable Delayed Pulse / Delayed Interrupt Delay Register

| address | bit 15 | | | | | | | bit 0 |
|---|---|---|---|---|---|---|---|---|
| 0x01C | Programmable Delayed Pulse / Delayed Interrupt Delay Register | | | | | | | |

## Programmable Width Pulse / Delayed Pulse Width Register

| address | bit 15 | | | | | | | bit 0 |
|---|---|---|---|---|---|---|---|---|
| 0x01E | Programmable Width Pulse / Delayed Pulse Width Register | | | | | | | |

## VME Interrupt Vector Register

| address | bit 7 | | | | | | | bit 0 |
|---|---|---|---|---|---|---|---|---|
| 0x021 | VME interrupt vector register | | | | | | | |

## Interrupt configuration register

| address | bit 7 | bit 6 | bit 5 | bit 4 | bit 3 | bit 2 | bit 1 | bit 0 |
|---|---|---|---|---|---|---|---|---|
| 0x023 | | | IEDBUF | IEDIRQ | IEEVT | IEHRT | IEFF | IEVIO |

| Bit | Function |
|---|---|
| IEDBUF | Data Buffer interrupt enable. |
| IEDIRQ | Delayed interrupt enable. |
| IEEVT | Event interrupt enable. |
| IEHRT | Lost heartbeat interrupt enable. |
| IEFF | Event FIFO full interrupt enable. |
| IEVIO | Receiver violation interrupt enable. |

## Distributed bus enable register

| address | bit 15 | bit 14 | bit 13 | bit 12 | bit 11 | bit 10 | bit 9 | bit 8 |
|---|---|---|---|---|---|---|---|---|
| 0x024 | | | | DBEVC | | | | |

| address | bit 7 | bit 6 | bit 5 | bit 4 | bit 3 | bit 2 | bit 1 | bit 0 |
|---|---|---|---|---|---|---|---|---|
| 0x025 | DBEN7 | DBEN6 | DBEN5 | DBEN4 | DBEN3 | DBEN2 | DBEN1 | DBEN0 |

| Bit | Function |
|---|---|
| DBEVC | Event counter clock source when prescaler = 0<br>0 – Event counter is counting increment timestamp event codes 0x7c<br>1 – Event counter is running on signal distributed on DBUS bit 4 |
| DBENx | OTPx output select: |

0 - programmable width pulse x,
1 - distributed bus bit x.

## Distributed bus data register

| address | bit 7 | | | | | | | bit 0 |
|---|---|---|---|---|---|---|---|---|
| 0x027 | Distributed bus data register (read only) | | | | | | | |

## Programmable Delayed Pulse / Delayed Interrupt Prescaler Register

| address | bit 15 | | | | | | | bit 0 |
|---|---|---|---|---|---|---|---|---|
| 0x028 | Programmable Delayed Pulse / Delayed Interrupt Prescaler Register | | | | | | | |

## Timestamp Event Counter Clock Prescaler Register

| address | bit 15 | | | | | | | bit 0 |
|---|---|---|---|---|---|---|---|---|
| 0x02A | Timestamp Event Counter Clock Prescaler Register | | | | | | | |

## Front Panel Output Multiplexer Registers

| address | bit 7 | bit 6 | bit 5 | bit 4 | bit 3 | bit 2 | bit 1 | bit 0 |
|---|---|---|---|---|---|---|---|---|
| 0x040 | | | FP0SEL5 | FP0SEL4 | FP0SEL3 | FP0SEL2 | FP0SEL1 | FP0SEL0 |
| 0x042 | | | FP1SEL5 | FP1SEL4 | FP1SEL3 | FP1SEL2 | FP1SEL1 | FP1SEL0 |
| 0x044 | | | FP2SEL5 | FP2SEL4 | FP2SEL3 | FP2SEL2 | FP2SEL1 | FP2SEL0 |
| 0x046 | | | FP3SEL5 | FP3SEL4 | FP3SEL3 | FP3SEL2 | FP3SEL1 | FP3SEL0 |
| 0x048 | | | FP4SEL5 | FP4SEL4 | FP4SEL3 | FP4SEL2 | FP4SEL1 | FP4SEL0 |
| 0x04A | | | FP5SEL5 | FP5SEL4 | FP5SEL3 | FP5SEL2 | FP5SEL1 | FP5SEL0 |
| 0x04C | | | FP6SEL5 | FP6SEL4 | FP6SEL3 | FP6SEL2 | FP6SEL1 | FP6SEL0 |

| FPxSEL *) | Signal selected |
|---|---|
| 000000 | Programmable Delayed Pulse Output 0 |
| 000001 | Programmable Delayed Pulse Output 1 |
| 000010 | Programmable Delayed Pulse Output 2 |
| 000011 | Programmable Delayed Pulse Output 3 |
| 000100 | Trigger Event Output 0 |
| 000101 | Trigger Event Output 1 |
| 000110 | Trigger Event Output 2 |
| 000111 | Trigger Event Output 3 |
| 001000 | Trigger Event Output 4 |
| 001001 | Trigger Event Output 5 |
| 001010 | Trigger Event Output 6 |
| 001011 | Programmable Width Pulse Output 0 |
| 001100 | Programmable Width Pulse Output 1 |
| 001101 | Programmable Width Pulse Output 2 |
| 001110 | Programmable Width Pulse Output 3 |
| 001111 | Programmable Width Pulse Output 4 |
| 010000 | Programmable Width Pulse Output 5 |
| 010001 | Programmable Width Pulse Output 6 |
| 010010 | Programmable Width Pulse Output 7 |
| 010011 | Programmable Width Pulse Output 8 |
| 010100 | Programmable Width Pulse Output 9 |

| | |
|---|---|
| 010101 | Programmable Width Pulse Output 10 |
| 010110 | Programmable Width Pulse Output 11 |
| 010111 | Programmable Width Pulse Output 12 |
| 011000 | Programmable Width Pulse Output 13 |
| 011001 | Level Output 0 |
| 011010 | Level Output 1 |
| 011011 | Level Output 2 |
| 011100 | Level Output 3 |
| 011101 | Level Output 4 |
| 011110 | Level Output 5 |
| 011111 | Level Output 6 |
| 100000 | Distributed Bus Data 0 |
| 100001 | Distributed Bus Data 1 |
| 100010 | Distributed Bus Data 2 |
| 100011 | Distributed Bus Data 3 |
| 100100 | Distributed Bus Data 4 |
| 100101 | Distributed Bus Data 5 |
| 100110 | Distributed Bus Data 6 |
| 100111 | Distributed Bus Data 7 |
| 101000 | Prescaler 0 |
| 101001 | Prescaler 1 |
| 101010 | Prescaler 2 |

*) FP0 to FP4 are front panel TTL outputs 0 to 4, FP5 and FP6 are front panel LVPECL outputs 0 and 1

## External Event Code Register

| address | bit 7 | | | | | | | bit 0 |
|---|---|---|---|---|---|---|---|---|
| 0x051 | External Event Code Register | | | | | | | |

## Clock Control Register

| address | bit 7 | bit 6 | bit 5 | bit 4 | bit 3 | bit 2 | bit 1 | bit 0 |
|---|---|---|---|---|---|---|---|---|
| 0x053 | CGLOCK | PSDONE | RFFB | RXFB/ PSDEC | DCML/ PSINC | DCMS/ RSSMPL | RSDCM | EVSRC |

| Bit | Function |
|---|---|
| EVSRC | Event Clock Source:<br>0 – Use externally resynchronized recovered clock<br>1 – Use recovered clock directly from MGT |
| RSDCM | Reset event DCM (always write 0) |
| DCMS | Event DCM stopped (read-only) |
| RSSMPL | Reset (always write 0) |
| DCML | Event DCM locked (read-only) |
| PSINC | Event DCM phase-shift increment (always write 0) |
| RXFB | External recovered clock resynchronization feedback (read-only) |
| PSDEC | Event DCM phase-shift decrement (always write 0) |
| RFFB | External RF resynchronization feedback (read-only) |
| PSDONE | Event DCM phase-shift DONE (read-only) |
| CGLOCK | Micrel SY87739L locked (read-only) |

## Seconds Shift Register

| address | bit 31 | | | | | | | bit 0 |
|---|---|---|---|---|---|---|---|---|
| 0x054 | Seconds Shift Register (read-only) | | | | | | | |

## Timestamp Latch Seconds Register

| address | bit 31 | | | | | | | bit 0 |
|---|---|---|---|---|---|---|---|---|
| 0x058 | Timestamp Latch Seconds Register (read-only) | | | | | | | |

## Event FIFO Extended Timestamp Registers

| address | bit 31 | | | | | | | bit 0 |
|---|---|---|---|---|---|---|---|---|
| 0x060 | Event FIFO Seconds Register (read-only) | | | | | | | |
| 0x064 | Event FIFO Event Counter Register (read-only) | | | | | | | |

## Polarity Register

| address | bit 31 | bit 30 | bit 29 | bit 28 | bit 27 | bit 26 | bit 25 | bit 24 |
|---|---|---|---|---|---|---|---|---|
| 0x068 | | | | | | | | POL24 |

| address | bit 23 | bit 22 | bit 21 | bit 20 | bit 19 | bit 18 | bit 17 | bit 16 |
|---|---|---|---|---|---|---|---|---|
| 0x069 | POL23 | POL22 | POL21 | POL20 | POL19 | POL18 | POL17 | POL16 |

| address | bit 15 | bit 14 | bit 13 | bit 12 | bit 11 | bit 10 | bit 9 | bit 8 |
|---|---|---|---|---|---|---|---|---|
| 0x06A | POL15 | POL14 | POL13 | POL12 | POL11 | | | |

| address | bit 7 | bit 6 | bit 5 | bit 4 | bit 3 | bit 2 | bit 1 | bit 0 |
|---|---|---|---|---|---|---|---|---|
| 0x068 | | | | | POL3 | POL2 | POL1 | POL0 |

| Bit | Function |
|---|---|
| POL0-3 | Programmable Delayed Pulse Output Polarity<br>0 – normal polarity (pulse active high)<br>1 – inverted polarity |
| POL11 | Output Pulse 0 (OTP0) Polarity<br>0 – normal polarity (pulse active high)<br>1 – inverted polarity |
| … | … |
| POL24 | Output Pulse 13 (OTP13) Polarity<br>0 – normal polarity (pulse active high)<br>1 – inverted polarity |

## Data Buffer Control and Status Register

| address | bit 15 | bit 14 | bit 13 | bit 12 | bit 11 | bit 10 | bit 9 | bit 8 |
|---|---|---|---|---|---|---|---|---|
| 0x07A | 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 |
|---|---|---|---|---|---|---|---|---|
| 0x07B | 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)<br>Flag is cleared by writing '1' to DBRX or DBRDY or disabling data buffer |
| DBEN | Data Buffer Enable Data Buffer Mode<br>'0' – Distributed bus not shared with data transmission, full speed distributed bus<br>'1' – Distributed bus shared with data transmission, half speed distributed bus |
| RXSIZE | Data Buffer Received Buffer Size (read-only) |

### SY87739L Fractional Divider Configuration Word

| address | bit 31 | bit 0 |
|---|---|---|
| 0x080 | SY87739L Fractional Divider Configuration Word | |

| Configuration Word | Frequency with 24 MHz reference oscillator |
|---|---|
| 0x0C928166 | 124.907 MHz |
| 0x009743AD | 50 MHz |
| 0xC25B43AD | 49.978 MHz |

## Network Interface

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

### *Changing the IP Address of the Module*

The IP address of the module may be changed by sending an ICMP echo request packet with the modules MAC (Media Access Control) address and the IP address the module should respond to. This method is called ARP/PING. Both IP addresses have to be on the same subnet with the subnet mask programmed into the module (defaults to 255.255.255.0).

#### Linux

To set the IP address of an Event Receiver module to 192.168.1.32 on a Linux machine (as root):

```
# /sbin/arp –s 192.168.1.32 00:0E:B2:00:00:16
# ping 192.168.1.32
```

Now the board should respond to the echo request with echo replies.

#### Windows

To set the IP address of an Event Receiver module to 192.168.1.32 on a Windows machine (in command prompt):

```
C:\> arp –s 192.168.1.32 00-0E-B2-00-00-16

C:\> ping 192.168.1.32
```

Now the board should respond to the echo request with echo replies.

## *Using Telnet to Configure Module*

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

```
telnet 192.168.1.32 23
```

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

The telnet server responds to the following commands:

| Command | Description |
|---|---|
| b | Show/change boot parameters, IP address etc. |
| h / ? | Show Help |
| i | Read & show dynamic configuration values from FPGA |
| m <address> [<data>] | Read/Write FPGA CR/CSR, Function 0 |
| r | Reset Board |
| s | Save boot configuration & dynamic configuration values into non-volatile memory |
| t | Set event clock rate and tune module delay lines |
| u | Update IP2022 software |
| q | Quit Telnet |

## Boot Configuration (command b)

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

| Parameter | Description |
|---|---|
| IP address | IP address of module |
| Subnet mask | Subnet mask of module |
| Default GW | Default gateway |
| V2P IP address | (not used) |
| FPGA mode | FPGA configuration mode<br>0 – FPGA is not configured after power up<br>1 – FPGA configured from internal Flash memory<br>2 – FPGA is configured from FTP server |
| FTP server | FTP server IP address where configuration bit file resides |
| Username | FTP server username |
| Password | FTP server password |
| FTP Filename | FTP server configuration file name |
| Flash Filename | Configuration file name on internal flash |
| UDP Port | UDP server port for FPGA data access |

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

## Configuring Operating Parameters and Tuning Delay Lines (command t)

The Event Receiver has to be configured for proper event clock rate and the on-board delay lines have to be tuned for the operating conditions. Before setting up the board make sure you have an Event Generator with the correct event clock connected to the Event Receiver. Also, let the EVR reach operating temperature (power on for 10 minutes in crate). The EVR asks for event clock rate, RF pattern which applies only to the RF version and the Micrel SY87739L programming word. The event clock rate is given with 1 MHz precision only, this value is used e.g. for being able to count the heartbeat monitor 1.6 s timeout. The 20 least significant bits of the RF pattern are the bits sent out at the event clock rate through the RF output e.g. 0x000C6318 is in binary 11000110001100011000b which given four clock cycles during one event clock i.e. 500 MHz with an event clock of 125 MHz.

To start configuration and tuning issue command ‘t’ from the telnet prompt:

```
EVR-200 -> t ↵
Event clock    125 MHz ? ↵
RF Pattern    000c6318 ? ↵
SY87739L conf 0c928166 ? ↵
Starting tuning...
Trying with initial delay value of 0
Found edge at 0
Trying with initial delay value of 16
Found edge at 14
Trying with initial delay value of 32
Found edge at 14
Trying with initial delay value of 48
Found edge at 14
Trying with initial delay value of 64
Found edge at 57
Period limits, min: 36, max: 60, measured period 43
Tuned delay value 35
Tuned RF delay value 34
EVR-200 ->
```

After tuning the tuned values have to be stored in non-volatile memory:

```
EVR-200 -> s ↵
Confirm save (yes/no) ? yes ↵
Configuration saved.
EVR-200 ->
```

## Upgrading IP2022 Microprocessor Software (command u)

To upgrade the Ubicom IP2022 microprocessor software download the filesystem image containing the upgrade following the instructions in Upgrading FPGA Configuration File below.

Verifying the downloaded image is very important as trying to upgrade from a faulty image can make the board unusable and requiring a programming cable to get the board up again.

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

```
EVR-200 -> u ↵
Really update firmware (yes/no) ? yes ↵
Self programming triggered.
```

The Event Receiver starts programming the new software and restarts. To re-connect to the board you probably have to ping the board see Changing the IP Address of the Module above.

## *Upgrading FPGA Configuration File*

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

### Linux

In Linux use e.g. interactive tftp:

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

### Windows

In Windows command prompt issue the following command:

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

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

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

### Linux

In Linux use e.g. interactive tftp:

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

If files differ you should get following message:

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

### Windows

In Windows command prompt issue the following command:

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

## Event Receiver TTL Transition Board (EVR-TTB)

The event receiver TTL transition board delivers fourteen programmable width pulse outputs and two level outputs on LEMO connectors. The drivers are TTL level and capable of driving a ground terminated 50 ohm load. The transition module provides a status led to indicate that the board is in operation.


**Figure 10: Event Receiver TTL Transition Board (EVR-TTB) Front Panel**

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

| Connector / Led | Style | Level | Description |
|---|---|---|---|
| POWER | Green Led | | Transition Board Powered Up |
| OTP/DBUS0 | LEMO-EPL | TTL | Programmable width pulse 0 or Distributed Bus Bit 0 |
| OTP/DBUS1 | LEMO-EPL | TTL | Programmable width pulse 1 or Distributed Bus Bit 1 |
| OTP/DBUS2 | LEMO-EPL | TTL | Programmable width pulse 2 or Distributed Bus Bit 2 |
| OTP/DBUS3 | LEMO-EPL | TTL | Programmable width pulse 3 or Distributed Bus Bit 3 |
| OTP/DBUS4 | LEMO-EPL | TTL | Programmable width pulse 4 or Distributed Bus Bit 4 |
| OTP/DBUS5 | LEMO-EPL | TTL | Programmable width pulse 5 or Distributed Bus Bit 5 |
| OTP/DBUS6 | LEMO-EPL | TTL | Programmable width pulse 6 or Distributed Bus Bit 6 |
| OTP/DBUS7 | LEMO-EPL | TTL | Programmable width pulse 7 or Distributed Bus Bit 7 |
| OTP8 | LEMO-EPL | TTL | Programmable width pulse 8 |
| OTP9 | LEMO-EPL | TTL | Programmable width pulse 9 |
| OTP10 | LEMO-EPL | TTL | Programmable width pulse 10 |
| OTP11 | LEMO-EPL | TTL | Programmable width pulse 11 |
| OTP12 | LEMO-EPL | TTL | Programmable width pulse 12 |
| OTP13 | LEMO-EPL | TTL | Programmable width pulse 13 |
| OTL0 | LEMO-EPL | TTL | Level output 0 |
| OTL1 | LEMO-EPL | TTL | Level output 1 |

# Event Receiver Optical Transition Board (EVR-OTB)

The Event Receiver Optical Transition Board delivers fourteen programmable width pulse outputs as Agilent HFBR-1528 Versatile Link optical outputs. The transition module provides a status led to indicate that the board is in operation.

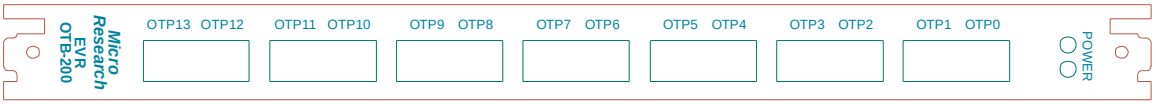


**Figure 11: Event Receiver Optical Transition Board (EVR-OTB) Front Panel**

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

| Connector / Led | Style | Level | Description |
|---|---|---|---|
| POWER | Green Led | | Transition Board Powered Up |
| OTP/DBUS0 | LEMO-EPL | Optical | Programmable width pulse 0 or Distributed Bus Bit 0 |
| OTP/DBUS1 | LEMO-EPL | Optical | Programmable width pulse 1 or Distributed Bus Bit 1 |
| OTP/DBUS2 | LEMO-EPL | Optical | Programmable width pulse 2 or Distributed Bus Bit 2 |
| OTP/DBUS3 | LEMO-EPL | Optical | Programmable width pulse 3 or Distributed Bus Bit 3 |
| OTP/DBUS4 | LEMO-EPL | Optical | Programmable width pulse 4 or Distributed Bus Bit 4 |
| OTP/DBUS5 | LEMO-EPL | Optical | Programmable width pulse 5 or Distributed Bus Bit 5 |
| OTP/DBUS6 | LEMO-EPL | Optical | Programmable width pulse 6 or Distributed Bus Bit 6 |
| OTP/DBUS7 | LEMO-EPL | Optical | Programmable width pulse 7 or Distributed Bus Bit 7 |
| OTP8 | LEMO-EPL | Optical | Programmable width pulse 8 |
| OTP9 | LEMO-EPL | Optical | Programmable width pulse 9 |
| OTP10 | LEMO-EPL | Optical | Programmable width pulse 10 |
| OTP11 | LEMO-EPL | Optical | Programmable width pulse 11 |
| OTP12 | LEMO-EPL | Optical | Programmable width pulse 12 |
| OTP13 | LEMO-EPL | Optical | Programmable width pulse 13 |

# Event Receiver High-Speed Optical Transition Board (EVR-HTB)

The Event Receiver High-Speed Optical Transition Board delivers fourteen programmable width pulse outputs as four Agilent HFBR-1528 Versatile Link optical outputs and ten Agilent HFBR-1414 Transmitters. The transition module provides a status led to indicate that the board is in operation.

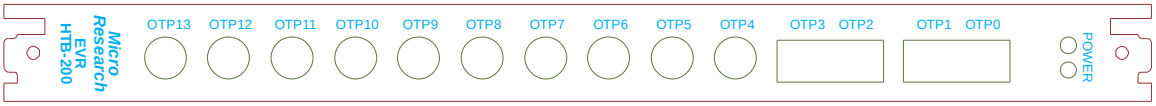


**Figure 12: Event Receiver High-Speed Optical Transition Board (EVR-HTB) Front Panel**

The front panel of the Event Receiver High-Speed Optical Transition Board includes the following connections and status leds:

| Connector / Led | Style | Level | Description |
|---|---|---|---|
| POWER | Green Led | | Transition Board Powered Up |
| OTP/DBUS0 | LEMO-EPL | Optical/ Versatile Link | Programmable width pulse 0 or Distributed Bus Bit 0 |
| OTP/DBUS1 | LEMO-EPL | Optical/ Versatile Link | Programmable width pulse 1 or Distributed Bus Bit 1 |
| OTP/DBUS2 | LEMO-EPL | Optical/ Versatile Link | Programmable width pulse 2 or Distributed Bus Bit 2 |
| OTP/DBUS3 | LEMO-EPL | Optical/ Versatile Link | Programmable width pulse 3 or Distributed Bus Bit 3 |
| OTP/DBUS4 | LEMO-EPL | Optical/ST | Programmable width pulse 4 or Distributed Bus Bit 4 |
| OTP/DBUS5 | LEMO-EPL | Optical/ST | Programmable width pulse 5 or Distributed Bus Bit 5 |
| OTP/DBUS6 | LEMO-EPL | Optical/ST | Programmable width pulse 6 or Distributed Bus Bit 6 |
| OTP/DBUS7 | LEMO-EPL | Optical/ST | Programmable width pulse 7 or Distributed Bus Bit 7 |
| OTP8 | LEMO-EPL | Optical/ST | Programmable width pulse 8 |
| OTP9 | LEMO-EPL | Optical/ST | Programmable width pulse 9 |
| OTP10 | LEMO-EPL | Optical/ST | Programmable width pulse 10 |
| OTP11 | LEMO-EPL | Optical/ST | Programmable width pulse 11 |
| OTP12 | LEMO-EPL | Optical/ST | Programmable width pulse 12 |
| OTP13 | LEMO-EPL | Optical/ST | Programmable width pulse 13 |

# Event Receiver TTL/NIM Transition Board (EVR-NTB)

The event receiver TTL/NIM transition board delivers 32 Event Receiver outputs as 19 TTL and 13 NIM level signals on LEMO connectors. The TTL level drivers are capable of driving a ground terminated 50 ohm load. The transition module provides a status led to indicate that the board is in operation.

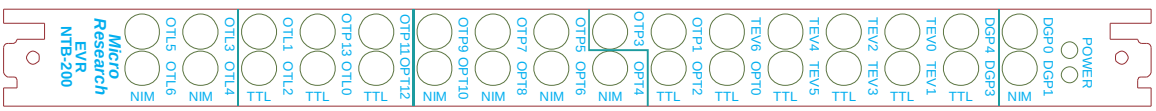


**Figure 13: Event Receiver TTL/NIM Transition Board (EVR-NTB) Front Panel**

The front panel of the Event Receiver TTL/NIM Transition Board includes the following connections and status leds:

| Connector / Led | Style | Level | Description |
|---|---|---|---|

| POWER | Green Led | | Transition Board Powered Up |
|---|---|---|---|
| DGP0 | LEMO-EPY | NIM | Extended Delayed Pulse 0 (DGP0) |
| DGP1 | LEMO-EPY | NIM | Extended Delayed Pulse 1 (DGP1) |
| DGP2 | LEMO-EPY | TTL | Extended Delayed Pulse 2 (DGP2) |
| DGP3 | LEMO-EPY | TTL | Extended Delayed Pulse 3 (DGP3) |
| TEV0 | LEMO-EPY | TTL | Trigger Event 0 (TEV0) |
| TEV1 | LEMO-EPY | TTL | Trigger Event 1 (TEV1) |
| TEV2 | LEMO-EPY | TTL | Trigger Event 2 (TEV2) |
| TEV3 | LEMO-EPY | TTL | Trigger Event 3 (TEV3) |
| TEV4 | LEMO-EPY | TTL | Trigger Event 4 (TEV4) |
| TEV5 | LEMO-EPY | TTL | Trigger Event 5 (TEV5) |
| TEV6 | LEMO-EPY | TTL | Trigger Event 6 (TEV6) |
| OTP/DBUS0 | LEMO-EPY | TTL | Programmable width pulse 0 or Distributed Bus Bit 0 |
| OTP/DBUS1 | LEMO-EPY | TTL | Programmable width pulse 1 or Distributed Bus Bit 1 |
| OTP/DBUS2 | LEMO-EPY | TTL | Programmable width pulse 2 or Distributed Bus Bit 2 |
| OTP/DBUS3 | LEMO-EPY | TTL | Programmable width pulse 3 or Distributed Bus Bit 3 |
| OTP/DBUS4 | LEMO-EPY | NIM | Programmable width pulse 4 or Distributed Bus Bit 4 |
| OTP/DBUS5 | LEMO-EPY | NIM | Programmable width pulse 5 or Distributed Bus Bit 5 |
| OTP/DBUS6 | LEMO-EPY | NIM | Programmable width pulse 6 or Distributed Bus Bit 6 |
| OTP/DBUS7 | LEMO-EPY | NIM | Programmable width pulse 7 or Distributed Bus Bit 7 |
| OTP8 | LEMO-EPY | NIM | Programmable width pulse 8 |
| OTP9 | LEMO-EPY | NIM | Programmable width pulse 9 |
| OTP10 | LEMO-EPY | NIM | Programmable width pulse 10 |
| OTP11 | LEMO-EPY | TTL | Programmable width pulse 11 |
| OTP12 | LEMO-EPY | TTL | Programmable width pulse 12 |
| OTP13 | LEMO-EPY | TTL | Programmable width pulse 13 |
| OTL0 | LEMO-EPY | TTL | Level output 0 |
| OTL1 | LEMO-EPY | TTL | Level output 1 |
| OTL2 | LEMO-EPY | TTL | Level output 2 |
| OTL3 | LEMO-EPY | NIM | Level output 3 |
| OTL4 | LEMO-EPY | NIM | Level output 4 |
| OTL5 | LEMO-EPY | NIM | Level output 5 |
| OTL6 | LEMO-EPY | NIM | Level output 6 |