MRF Timing System with Active Delay Compensation

EPICS Meeting, Oak Ridge, Sep 2016

Jukka Pietarinen

Motivation for Active Delay Compensation

- Requirement for higher event rate
  - SwissFEL event clock rate is 142.9 MHz, 7 ns event clock cycle
- Current fan-out CDR technology limits event clock rate to 135 MHz
- No commercial CDR solution available
- Need to make fan-outs active

- Adds fan-out complexity and cost

+ Compensation for delay and drift

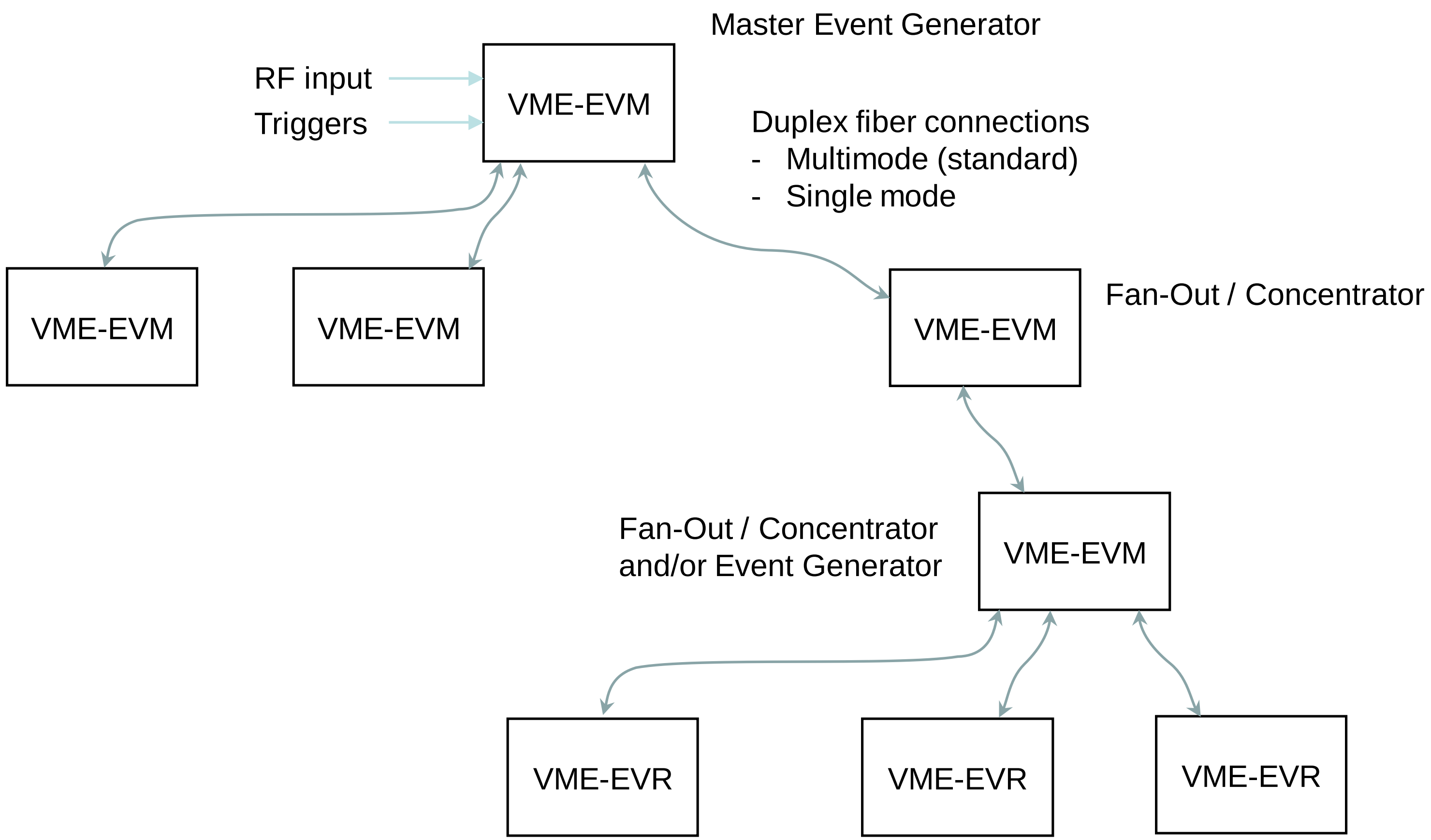
+ Adds possibility to fine tune all triggers of a single receiver

+ Precise synchronization of downstream EVGs

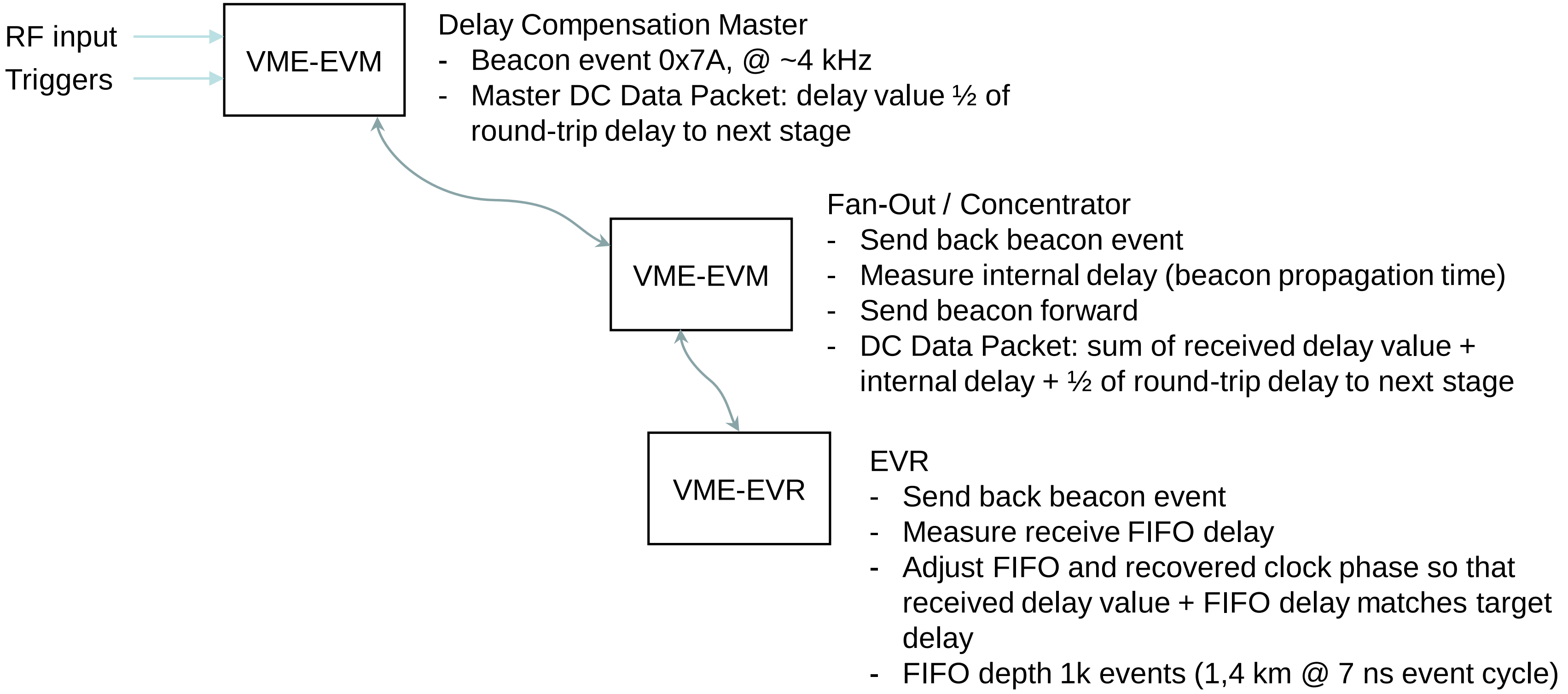
Hardware Compatibility with Delay Compensation

- VME-EVM-300 (new product combines EVG and Fan-Out/Concentrator)
- VME-EVR-300 (new product)
- PCIe-EVR-300DC (new product)
- mTCA-EVR-300 (new product)

Timing System Topology with Delay Compensation



Delay Compensation



RF input

Triggers

VME-EVM

Delay Compensation Master
- Beacon event 0x7A, @ ~4 kHz
- Master DC Data Packet: delay value ½ of round-trip delay to next stage

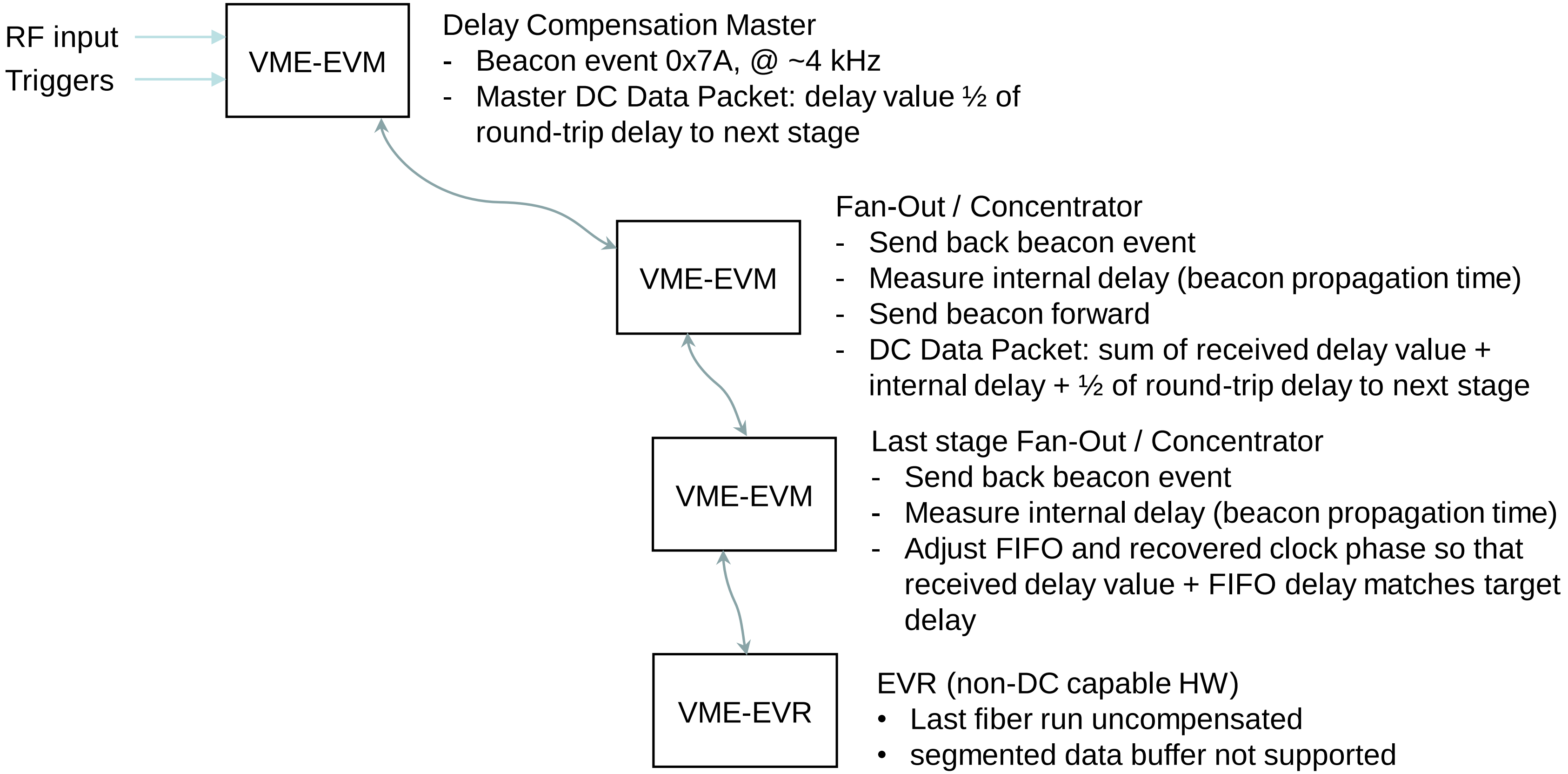
VME-EVM

Fan-Out / Concentrator
- Send back beacon event
- Measure internal delay (beacon propagation time)
- Send beacon forward
- DC Data Packet: sum of received delay value + internal delay + ½ of round-trip delay to next stage

VME-EVR

EVR
- Send back beacon event
- Measure receive FIFO delay
- Adjust FIFO and recovered clock phase so that received delay value + FIFO delay matches target delay
- FIFO depth 1k events (1,4 km @ 7 ns event cycle)

Delay Compensation for non-DC capable hardware



RF input

Triggers

VME-EVM

Delay Compensation Master
- Beacon event 0x7A, @ ~4 kHz
- Master DC Data Packet: delay value ½ of round-trip delay to next stage

VME-EVM

Fan-Out / Concentrator
- Send back beacon event
- Measure internal delay (beacon propagation time)
- Send beacon forward
- DC Data Packet: sum of received delay value + internal delay + ½ of round-trip delay to next stage

VME-EVM

Last stage Fan-Out / Concentrator
- Send back beacon event
- Measure internal delay (beacon propagation time)
- Adjust FIFO and recovered clock phase so that received delay value + FIFO delay matches target delay

VME-EVR

EVR (non-DC capable HW)
- Last fiber run uncompensated
- segmented data buffer not supported

VME-EVM-300

- VME64x
- Event Generator
- 8-Way Fan-Out / Concentrator
- Event clock up to 142,9 MHz (room for improvement left)

- Integrated Fan-Out Concentrator and Event Generator
  - Reference clock from RF input or recovered clock from upstream EVM
  - Allows for fully synchronized downstream EVGs
- Masked events in Sequencer
  - Hardware or software mask/enable
- Segmented Data Buffer
  - 128 segments of 16 bytes each (2 kbyte buffer)

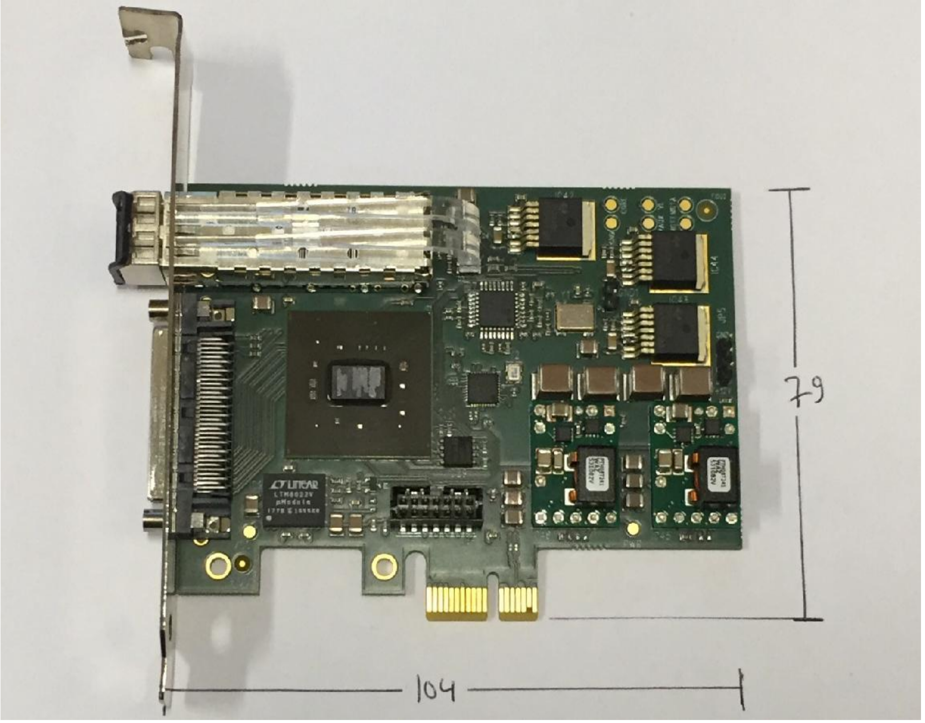
VME-EVR-300

- VME64x
- Event Receiver
- Four Universal I/O Slots (one slot driven by GTX)
- Two differential CML outputs (GTX)
- Two TTL inputs with default state selectable with jumpers
- Optional transition board VME-UNIV-TB64x

- Pulse Generator Triggers from prescalers and DBUS bits
- Gated pulse outputs
- Dual output mapping registers
- Segmented Data Buffer

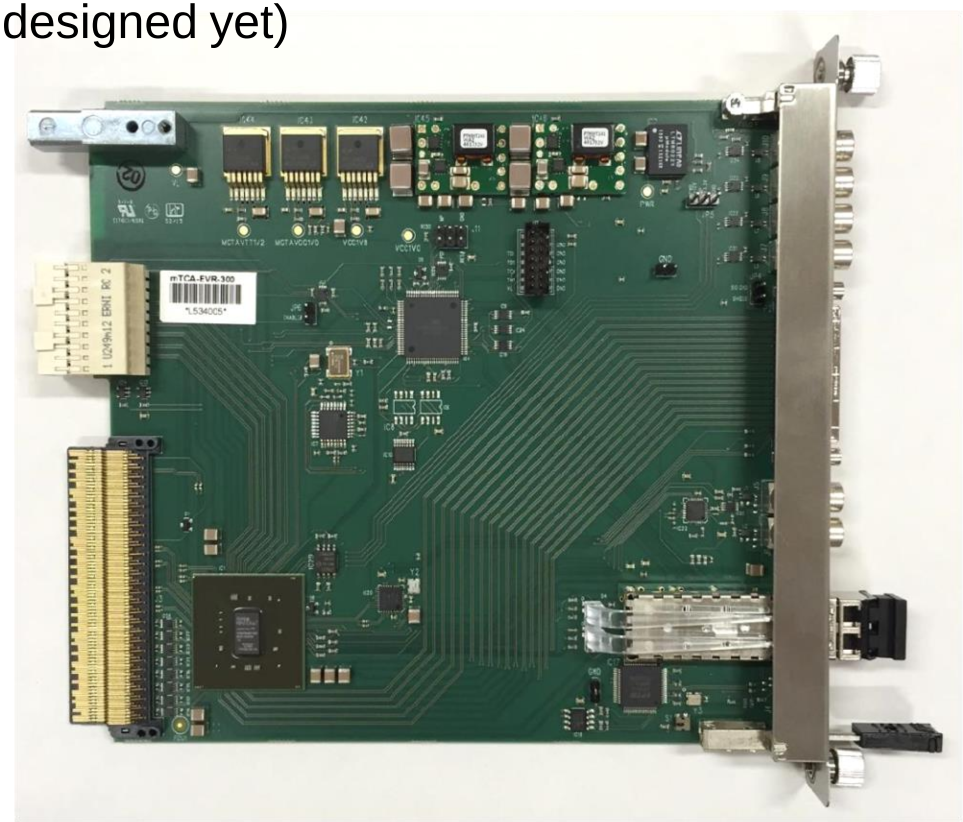
PCI Express Event Receiver PCIe-EVR-300DC

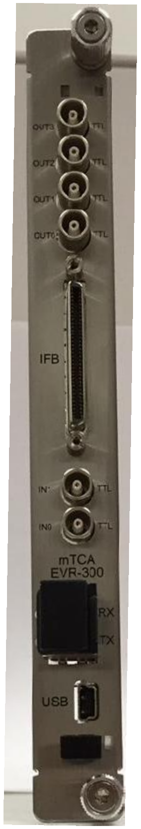
- Redesigned using Xilinx Kintex-7 FPGA
- IFB-300 interface for I/O
- Getting close to 10W limitation



MicroTCA.4 Event Receiver mTCA-EVR-300

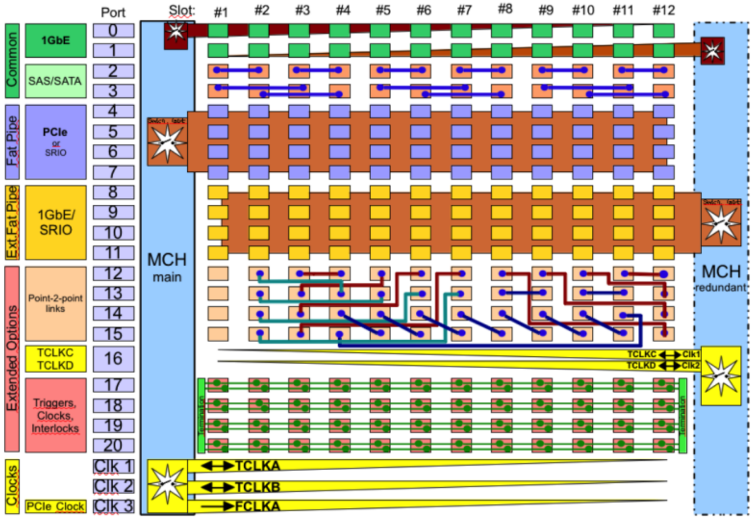
- Four TTL Outputs
- Two TTL Inputs
- IFB-300 interface
- RTM interface (no RTM designed yet)
- TCLKA/TCLKB
- Differential triggers
- DESY MMC





MicroTCA.4 Event Receiver Future Ideas

- mTCA-EVR-300U version?
  - Two/three? Universal I/O slots instead of IFB Connector/LEMOs
  - Requires cutout on main PCB and shorter spacer on Universal I/O module (probably easier to ship products with spacer on the main boards)
- Use point-2-point links 12-15 to share event link to neighboring mTCA.4 modules?



Masked Sequencer Events (EVM)

- New mask field in sequence RAM
- Four enable signals
  - When mask enable bit active ‘1’ enable event transmission only when HW signal active high or software mask enable bit active ‘1’
- Four disable signals (these will probably be changed into enables)
  - When mask disable bit active ‘1’ disable event transmission when HW signal active high or software mask disable bit active ‘1’

ram address	32 bits	8 bits	8 bits	
0	event address / timestamp	mask	code	1. event
1	event address / timestamp	mask	code	2. event
2	event address / timestamp	mask	code	3. event
	⋮			
2047	event address / timestamp	mask	code	2048. event

Segmented Data Buffer

- 128 segments of 16 bytes each (2 kbyte buffer)
- Last segment is reserved for Delay Compensation (delay value transmission)
  - Changed to last segment in FW 0205, before it was first segment
- Transmission
  1. Write data into buffer
  2. Set up transmission: starting segment number, data length (may overlap several segment)
  3. Trigger transmission
- Receiving
  - Each segment has following flags (3 x 128 bits) and register:
    - Receiving complete
    - Receiver overrun (segment overwritten before clearing receiving complete flag)
    - Checksum error
    - Register for number of bytes received
  - Interrupt may be enabled separately for each segment
  - Flags set only for starting segment

- Changes:
  - Receiver is always “armed” – no need to enable after a single data buffer reception
  - Starting with firmware 0206: coexist with previous data buffer and returns compatibility with 230 series data buffer

EVR Gates and Dual Output Mapping Registers

- Up to eight pulse generators operate as gates (masks and pulse enables)
- Gate state can be changed by
  - Events
  - Software
  - External signal by generating an external event
- Gates can be used to enable or disable pulse generators
- Each output mapping register now supports combining two sources (logical OR)
- This allows for:
  - “Switching” pulse generators (or pulse parameters) on event e.g. displacing trigger based on machine state
- **Warning!** The 16-bit mapping register now has two 8-bit mapping registers and an unused register has to read 0x3f for a static ‘0’ output. If e.g. the mapping register reads 0x0028 it will output a signal consisting of pulse generator 0 output or’ed with prescaler 0 output.

EVR New Features 0207

(applies to VME-EVR-300, PCIe-EVR-300DC, mTCA-EVR-300)

- Integrated EVG features:
  - Software Events
  - One Sequencer than can be triggered from any pulse generator, prescaler or distributed bus bit
- Stand-alone operation using internal fractional synthesizer reference clock as event clock
- Can be used without delay compensation. In this case the user may adjust the unit’s timing delay (receiver FIFO depth).
- Old style data buffer and segmented data buffer can coexists in same system without interference: segmented data buffer now uses different protocol which is ignored by 230 series boards

- Plans to port new firmware also to cPCI-EVR-300 just without delay compensation feature but with and adjustable receive FIFO. So in principle we can have delay compensation but without drift compensation (delay adjust steps of one event clock cycle).

Lessons learned during development of Delay Compensation

- The amount of complexity caused by implementing delay compensation was underestimated, debugging is very difficult and getting all the bits and pieces right is very time consuming – first versions took ~1 h to lock, now delay compensation lock time is less than a minute
- Latency behavior of the Kintex 7 transceivers (both RX and TX) took a lot of time to understand
  - Unclear documentation
  - Ended up using undocumented features of the transceivers after talking to a Xilinx expert
- New Xilinx Vivado tool (learning curve especially constraining the design)
- **Always try to maintain compatibility!** In the beginning it was considered not to be possible to maintain compatibility with existing protocol/register map
  - Segmented data buffer was located at the same address location with the earlier data buffer
  - Relocation to a new address whereas the “old” data buffer was re-instantiated for backward compatibility in FW 0205 caused major software issues
- Should have had same hardware/software setup at MRF and PSI
- Issues with Lattice ECP3: transceiver operation beyond 2.5 Gbps the way the EVR requires was unsuccessful, redesign using Xilinx Kintex 7

  → Do not change something that works: going from Xilinx to Lattice

Planned EVM/EVR 02xx Firmware Features

- Integrate Downstream Event Receiver into EVM
    - Decode event into “Internal triggers”
    - Map internal triggers to EVG inputs e.g. Sequencer control
    - Need to distinguish between Fan-Out mode and Downstream EVG mode with different way of handling priorities and event forwarding etc.
- Integrate Upstream Event Receiver into EVM
    - – Forwarding of events/data further upstream
    - – Decode events into “internal trigger”

For discussion

- Feature requests?
  - Programmable delay for pulse set/reset

Ideas (thinking out loud…)

- Include “topology” information in DC frame: use nibble for each fan-out level to indicate port number e.g.
  - 00000001 (Port 1 of f 1st level)
  - 00000123 (Port 3 of 1st level, port 2 of 2nd level, port 1 of 3rd level