

# SLS & Diamond Timing System update

**Timo Korhonen**

**PSI**

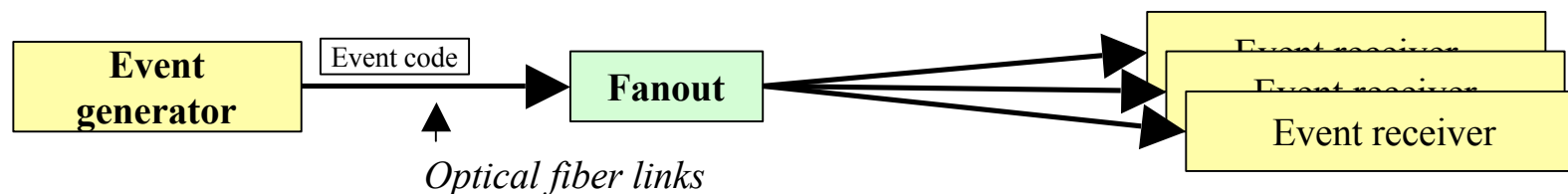
1. Concepts revisited...again
3. New (Diamond) cards features and status
4. EPICS interface
5. Conclusions

## Event System Concepts Recap

-Many **event receivers** (in IOC crates) are listening to an event stream from a single or a few (cascaded) **event generators**

-The event generator transmits bare 16-bit data frames at the hardware frame rate. Frame rate can be internally generated or a subharmonic of RF; for SLS, 50 MHz. Event receivers phase lock to the event stream.

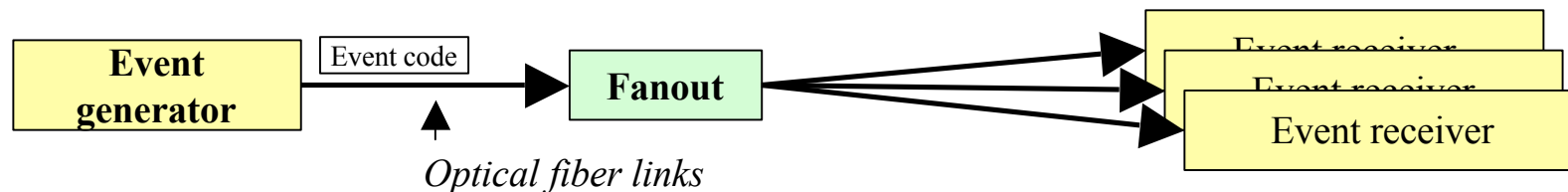
-external stimuli to event generator are encoded to 8-bit event codes; when no stimulus is present, null codes are sent out. The other 8 bits in the frame are “free”; in SLS implementation they are used to distribute clock signals (SR revolution frequency, etc.), but could be used to transmit arbitrary data.



## Event System Concepts Recap

The system provides:

- global distribution of events to all systems that have a receiver
- trigger and gate signals to hardware
- synchronized timestamp facility
- software sequencing by triggering channels to process from events
- software (EPICS records) can be used to send events



The stimulus to send an event can be:

- pulse on a hardware input**
- software event** (write to a register)
- an entry in an **event playback RAM** .

When an event code is received the receiver can:

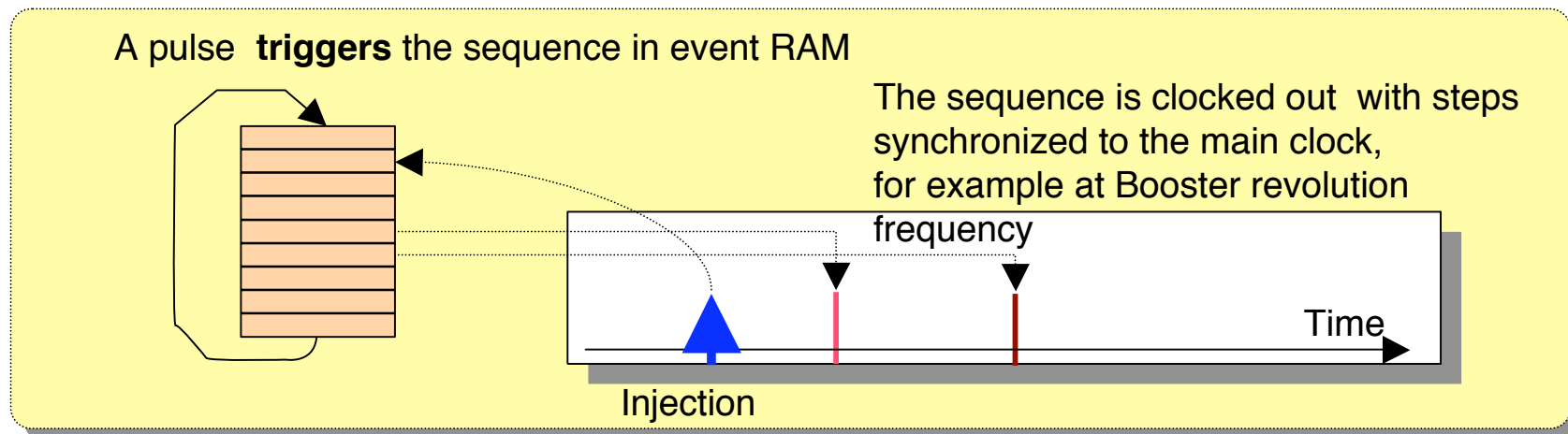
- output a pulse, of specified delay and width**
- trigger a software action (process an EPICS record)**

Each event receiver can be programmed to respond in a different way to the same event code.

## (SLS) Event Generator

3 possible event sources:

- hardware inputs. 8 per card (could be increased)
- event generation by register write
- event RAMs(2) for event storage and playback (fast and deterministic sequencer)



At SLS, the injection sequence is run from the RAM

- the step clock can be set (to multiples of revolution period)
- alternate mode: one RAM runs while the second is loaded.
- any sequence can be programmed through EPICS records.

# SLS Event Generator

RF (500 MHz) input

Fiberoptic Transceiver

AC line input (3-12 V)

However, this is already history (for new designs)....

Gigabit Ethernet  
Transceiver chip  
(ser/par conversion,  
framing, PLL, etc.)

Event RAMs  
(2 x 512kB)

VME interface

Flash memory for FPGA configurations

## Event System Series 200

Adaptation of the SLS philosophy to DIAMOND needs

- use the latest technology (more bandwidth, better frequency coverage)
  - do the things we did not manage (time, money) to do at SLS  
(hot-swap, full VME64x implementation with geographical addressing...)
  - improve some key areas (VME interface – register space, speed)
  - VME master capability (nice to have)
  - better jitter performance (Diamond requirement)
  - new cards (4-ch timer, gun driver) and form factors (PMC)
  - new ideas (not all implemented for Diamond): delay drift compensation
- 
- retain SLS functionality (with some enhancements)
  - compatible with SLS series (accepts 1 Gbit/s stream)

## EVR-200 First Prototype (September '04)

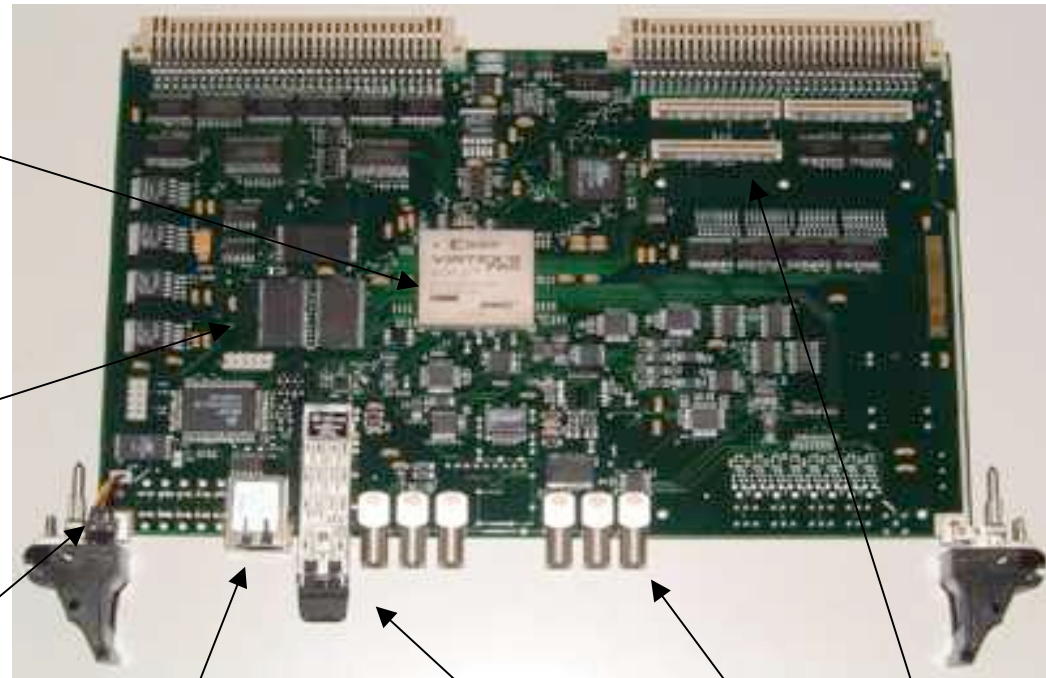
Virtex II-Pro FPGA  
(RocketIO links, PowerPC core)

-link speeds up to 2.5 Gbit/s  
-wide range of frequencies  
RF frequency up to 1 GHz

RAM for program storage  
(embedded PPC)

-careful ground  
balancing for  
improved jitter control

Hot-swap  
switch,  
controller



Ethernet port  
(10BaseT),  
Microcontroller for  
remote access (FPGA  
reprogramming, etc.)

SFP transceiver

PMC slot

Front panel  
outputs

# New Faces



EGUN-Tx (EVR with links to an e-gun trigger card (low jitter, < 1ps)  
4-channel precision timer (10 ps resolution)  
Event Generator  
EVR with RF recovery

Transition modules:  
-optotriggers (for power supplies)  
-4-ch timer card  
-EVG  
-EVR



## EPICS support for the event system

- Dedicated record types to enable software control of
  - mapping hardware input and output signals to events
  - event generation from an EPICS record
  - record processing upon receipt of an event (timing signal)
  - repetitive events and their placement in time (sequence RAM)
- major weak point in current design: loading of sequence RAMs clumsy and time consuming: processing of any of these records causes a RAM rewrite (clear & write sequence).  
Slow!

## EPICS records for the event system

### Event generator

#### **EG** record

-setup of the generator:  
mapping of hardware signals to  
events, RAM mode

#### **EGEVENT** record

-insertion of events into the event RAM.  
One record for one event.

### Event receiver

#### **ER** record

-setup of the receiver: configuration of  
hardware signals (enable, delay  
generator setup, etc.)

#### **EREVENT** record

-event code to output mapping. Specifies  
what is the action upon receipt of an  
event.

For SLS, we took the scheme from APS, with some (quite a few)  
additions/modifications to the original records

- EG record: RAM prescaler & frequency field, max delay (RAM position) field
- EGEVENT: priority handling of events (in RAM)
- ER record: pulse width fields for OTP, prescaler settings, IRQ delay settings, signal polarity selection

Receiver side records are mostly OK, event generator (especially egevent) need  
rework

(not much used at APS?, works OK for SLS/other light sources)

## Injection sequence programming

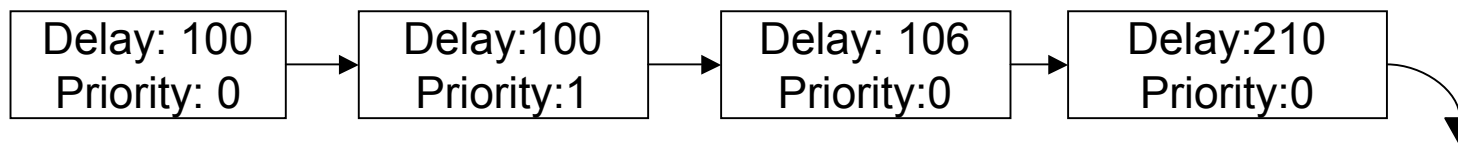
Injection sequence (machine cycle) defined by placing events in the sequence RAM

-event records: one event per record

-linked to a list

-processing of one record in list causes the list to be reloaded to RAM. Inefficient (when multiple delays changed)

-list is always sorted in delay & priority order (to define how to handle events with exactly same delay)



## Injection sequence programming

### Problems with the list:

- how to decide time of reprogramming RAM? No way to decide if still more delays are to be changed.
- current implementation: erase RAM & rewrite all (expensive, prohibitive with even larger RAMs)
- problematic with multiple types of sequences (lots of processing would be required)
- better approach: “event sequence” record?

## Support for new (planned) features

### Real-time data transfer

hardware-assisted data transfer: data from event generator memory transferred to event receivers

Write data at event generator

Unpack at event receivers

Use waveform/SubArray records?

Need to discuss the applications and requirements.

(MPG workshop/discussion)

## Status

Prototype hardware (EVR-200, EVG-200, E-Gun receiver) working.

A few delays in the first series (connector delivery, change of production facility)

First small series arriving in a few weeks, start developing drivers from January (first port existing drivers to new HW)

Cards work with existing event stream, a few firmware issues with bit patterns to be solved (RocketIO vs. earlier scheme)

-first jitter tests with E-gun receiver produced excellent results: first proto 1-2 ps, optimized version below 1 ps jitter.