

LCLS-I/LCLS-II Timing System

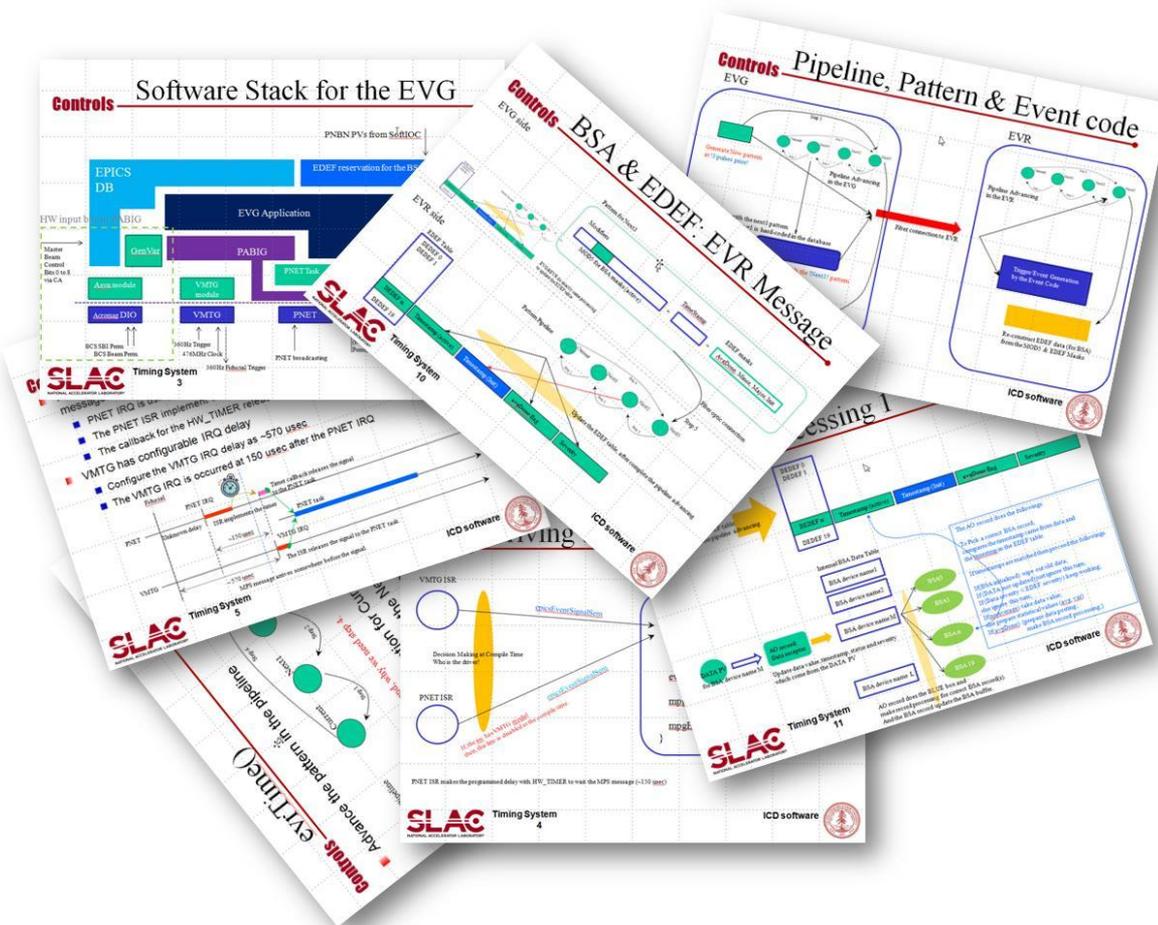
Low Level

Kukhee Kim for LCLS Timing Team

ICD Software, SLAC National Accelerator Laboratory

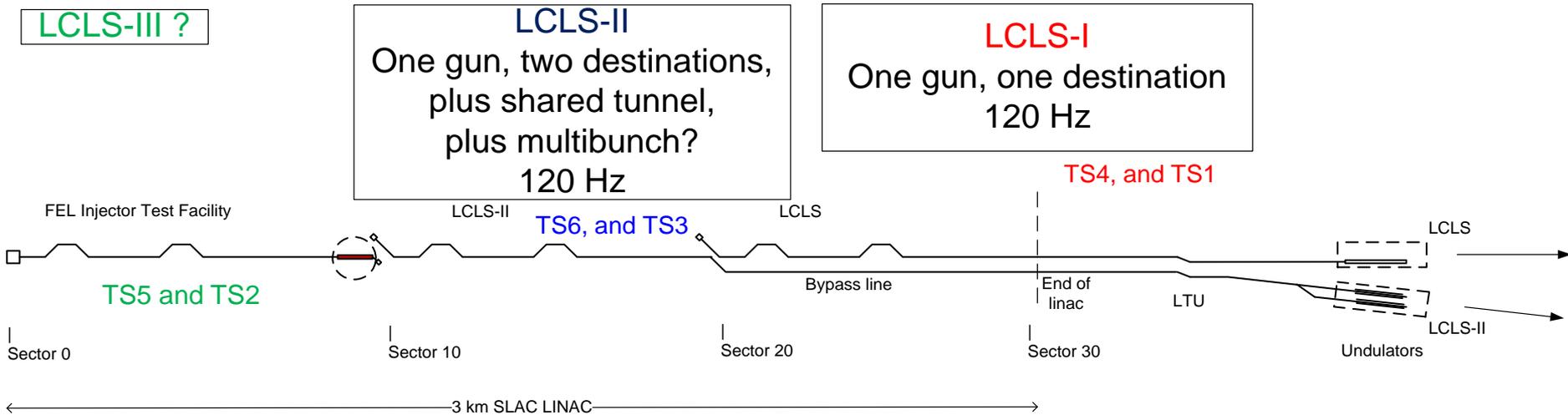
October 22, 2012

Event System is a most complicated system!



Accelerator complex for current and future (LCLS-I, LCLS-II and ...)

Interleaved Accelerator Operation



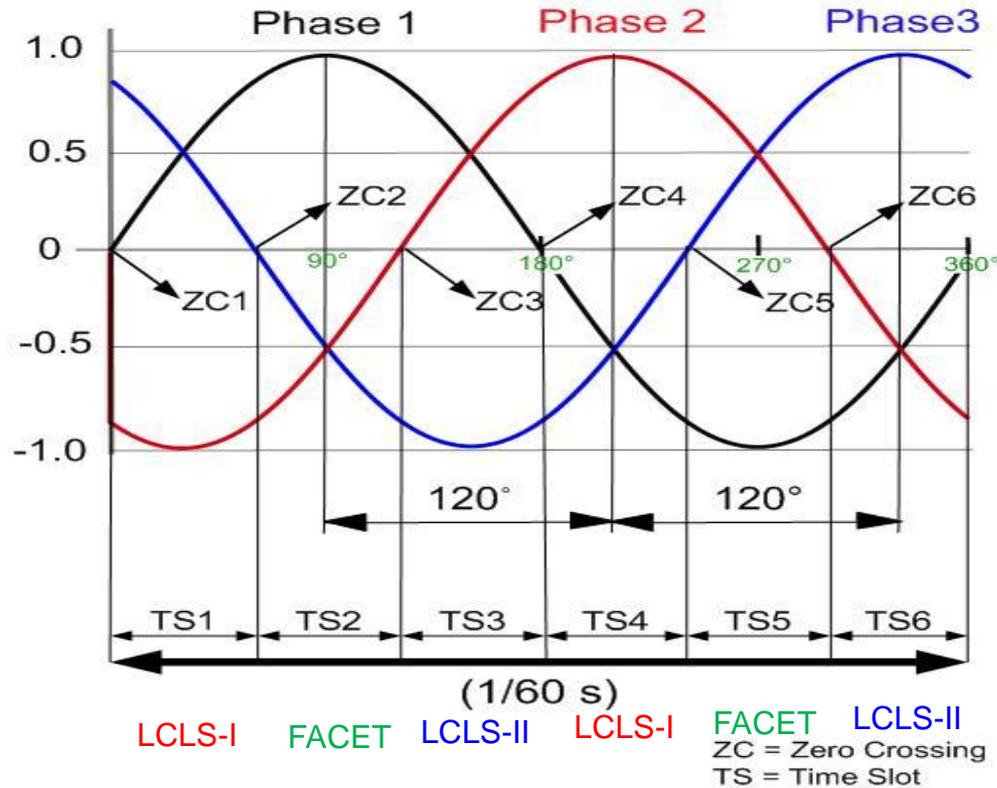
Separate but synchronous timing systems

360 Hz = 6 x 60 Hz power line timeslots

Independent rate control to each destination

MPS-aware timing

360Hz – constraint: What is the timeslots?



Legacy Timing System

- SLC timing system
 - The Linac is a Pulsed Machine (get a packet of beam per pulse) runs at a max of 360Hz (120Hz)
 - Three Main Timing Signals:
 - Hardware
 - 476MHz Master Accelerator Clock (runs down 2mile Heliac Main Drive Line cable)
 - 360Hz Fiducial Trigger / encoded onto the 476MHz master clock
 - Software – MPG (Master Pattern Generator)
 - 128-Bit PNET (Pattern Network) Digital Broadcast (contains time slot, trigger setup, beam type & rate information, beam destination)
 - Provide the timing pattern 3 fiducial prior
- Advantage of the Legacy system
 - Physicists & Operator are used to the old system
 - Provide event patterns 3 fiducial prior
 - give a time margin to slow device, or pre-trigger up to 2 fiducial ahead

- LCLS Timing System
 - Use MRF EVG and EVRs
 - Generate event code at 360 Hz based on:
 - PNET pattern input (beam code + 128 bits pattern that define beam path and other conditions, [scheduled information](#))
 - Receive MPS*) information ([real-time information](#)), and queue it up into the pattern pipeline (pipeline bypass for [quasi real-time](#))
 - Add LCLS information (MPS, BPM calibration, diagnostics and)
 - Extend the pattern to 192 bits
 - Hardwired BCS fault
 - Send out timing pattern, including EPICS timestamp with encoded pulse ID on the timing fiber
 - Manage user-defined Beam Synchronous Acquisition (BSA)

MPS*) Please, see the talk for LCLS Machine Protection System (M. Boyes) in hardware solution session (16:30 – 16:50 Oct 22, 2012)

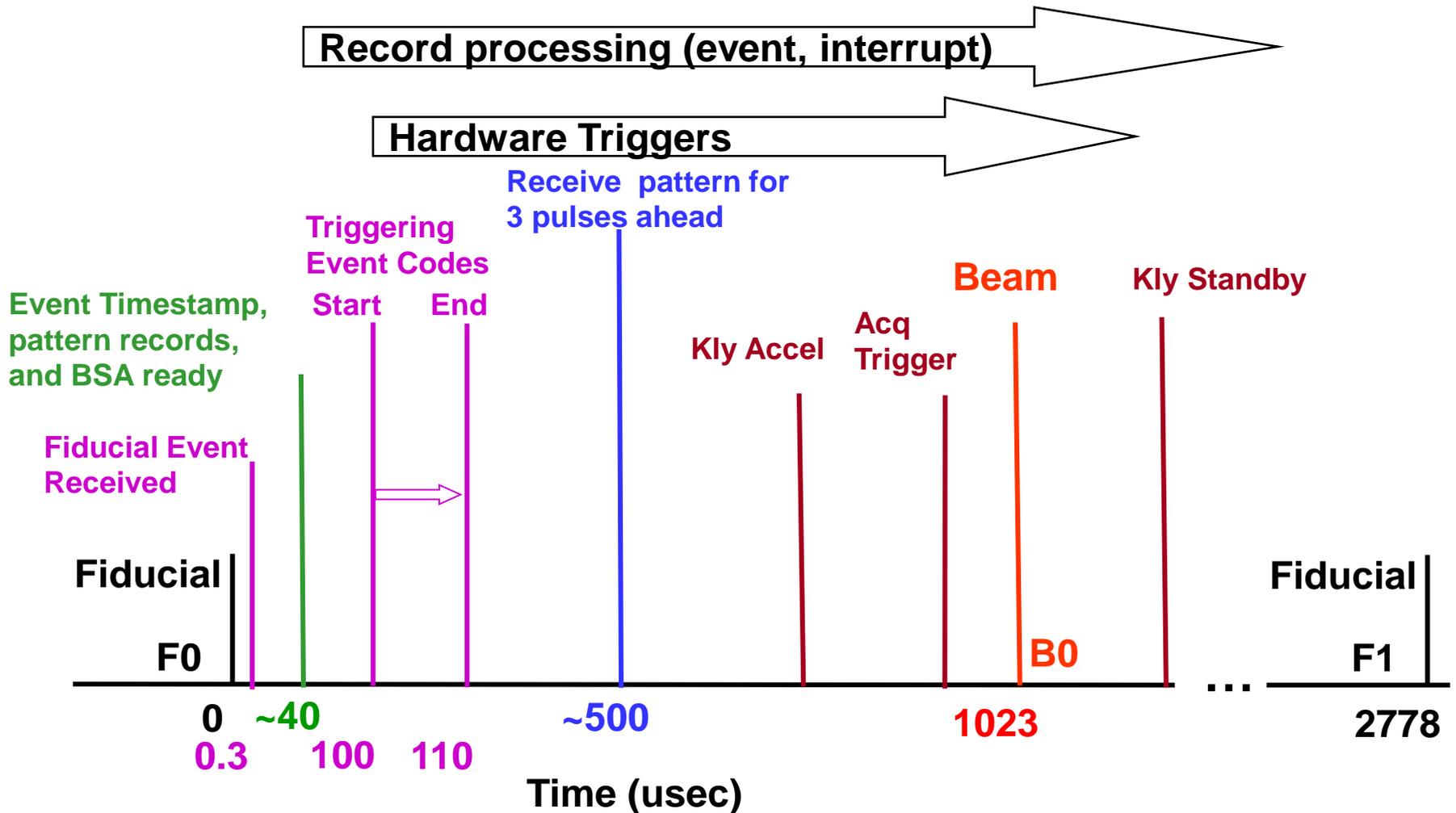
How to work the EVG

- PNET message broadcast from the old timing system (following a fiducial, 360 Hz rate)
- PNET receiver in EVG IOC generates IRQ
- EVG software takes the PNET data and uses it to assign the proper event codes (the PNET data pipelined 3 fiducial ahead)
- Event code setup in EVG one cycle head
- Next fiducial is sent: 360 Hz fiducial signal received by EVG
- The EVG begins to send out the event codes in its Sequence RAM
- The event codes get sent across the timing fiber links to the EVRs
- Inside the EVR, its mapping RAM is set to map a specific HW trigger to an event code
- when the Event Code matches the same value in the mapping RAM, a hit is generated and after a programmed delay, a HW trigger is output from the EVR to the device

How to work the EVR

- Set trigger delays, pulse width, and enable/disable via user requests
- Set event code per trigger
- Receive timing pattern ~ 8.3 msec (3 fiducials prior) before corresponding pulse. Provides EPICS timestamp to record processing
- Perform BSA based on tags which are set by EVG in the timing pattern
- Process pre-defined records when specific event codes are received

Timeline for the EVR



Upgrade Timing system for both LCLS-I and LCLS-II

- Main Goals

- Remove the dependency with the SLC timing system
- Independent EVG for each beam program
 - LCLS-I, LCLS-II, FACET, XTA and future beam programs
 - Promise for independent operation
 - Relieves the pain from limited event number (255 limitation)
- Keep compatibility with old system, keep SLAC specific features (advantage of the legacy system)
 - Beam code and timing pattern pipeline:
 - For pattern aware operation
 - BSA
 - Event Code Sequence for photon group
- Covers various platforms for EVR: RTEMS, linux, and linuxRT
 - LCLS-I, and LCLS-II both are going to use various hardware platforms
 - VME, PMC, Standard PC, uTCA, and COM-X
 - Need to support linux/linuxRT for new hardware platform

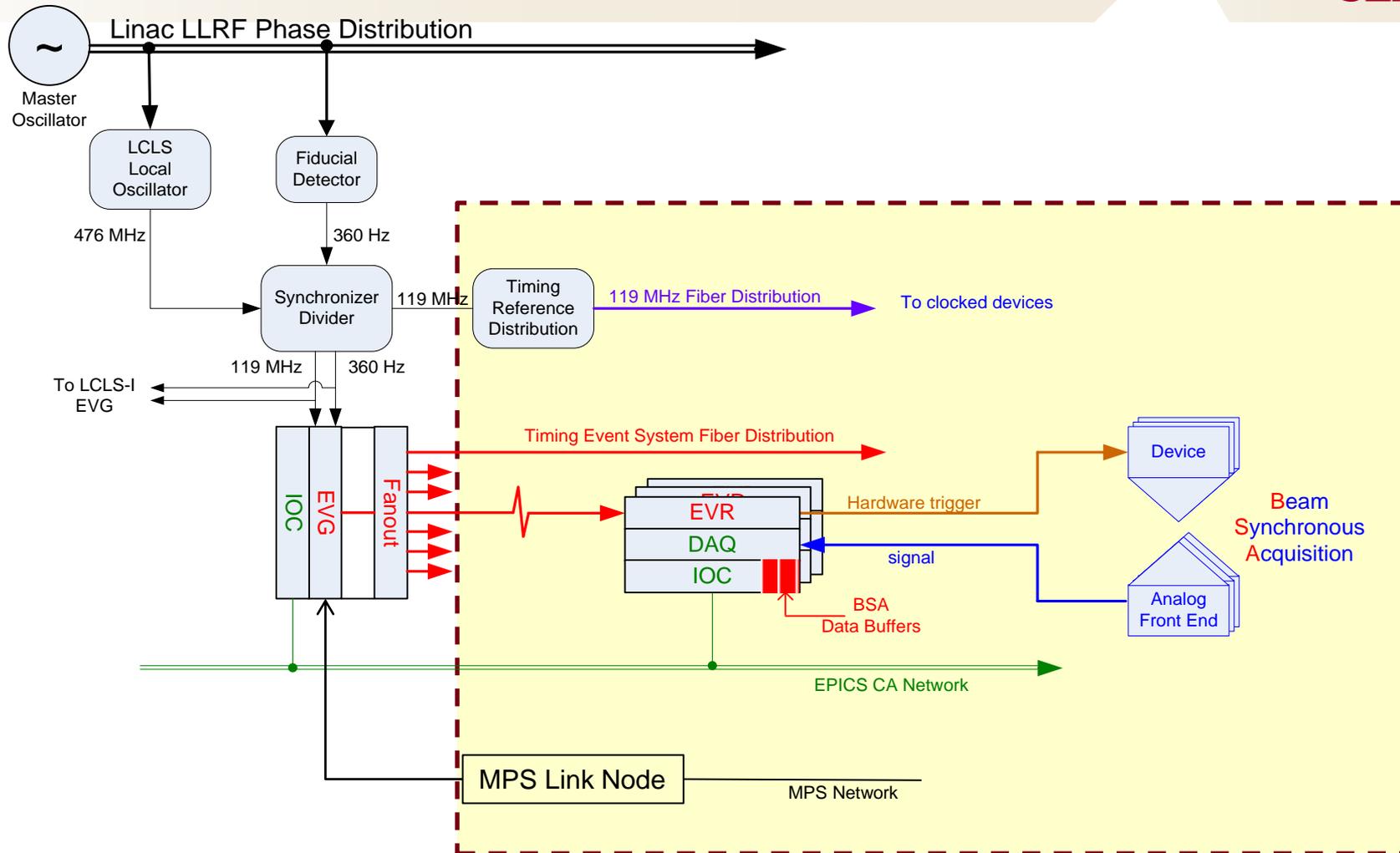
Timing System Upgrade Cont'd

- Upgrade
 - Develop a new hardware called VMTG: VME based Master Trigger Generator which provides 360 Hz IRQ to drive EVG software instead of the PNET receiver (VMTG in EVG side)
 - Develop new software module PABIG: Pattern Bits Generator which mimics the PNET
 - Develop a new software tool to design the event/timing pattern visually: evGUI¹⁾
 - Work on the EVR software to support linux/linuxRT²⁾ for the new hardware platform: uTCA, and COM-X

evGUI¹⁾ Please, see the talk for LCLS timing system for pattern design , evGUI and high level (M. Zelazny) in the timing system session (15:00 – 15:20 Oct 22, 2012)

Support for linux/linuxRT²⁾ Please, see the talk for Real-Time Performance Issues on linux/linuxRT (K. Kim) in the low level application session (14:10 – 14:30 Oct 23, 2012)

New architecture for the timing system



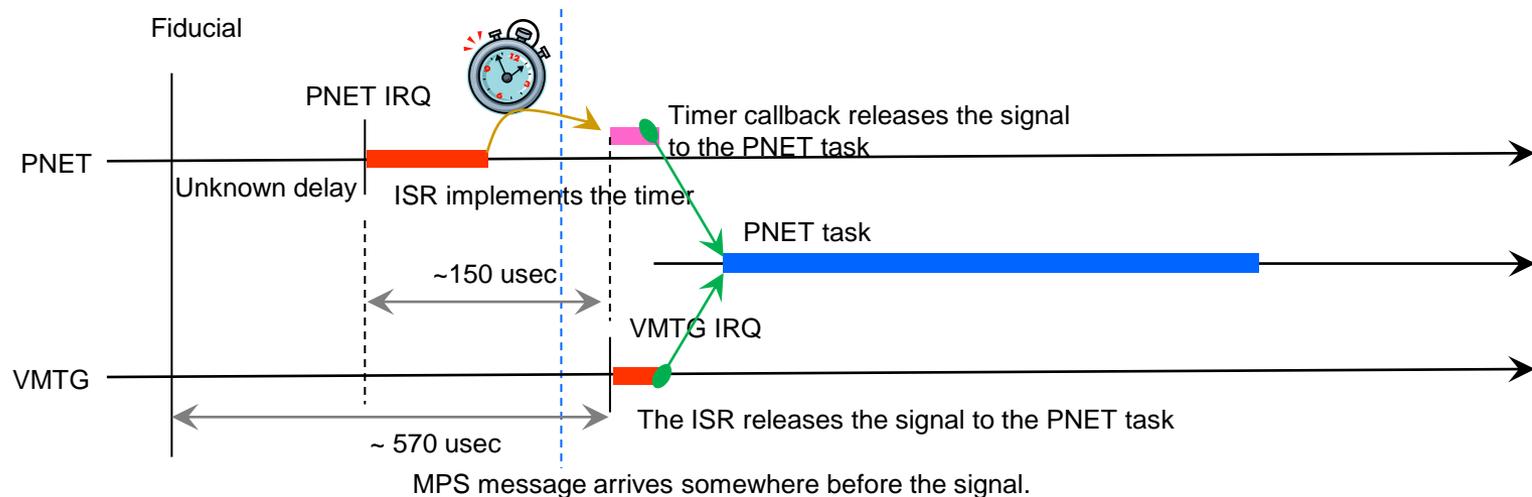
LCLS-II CD-2 DOE Review / Aug 21 - 23, 2012

- Sector 0 (LI00) VMTG – Master Trigger Generator
 - Provides the synchronized 360Hz trigger to the fiducial generator
 - Input: 360Hz AC trigger train + 476 MHz MDL RF
 - Output: 360Hz synchronized trigger
 - Then, the fiducial generator makes synchronization with the Master Oscillator

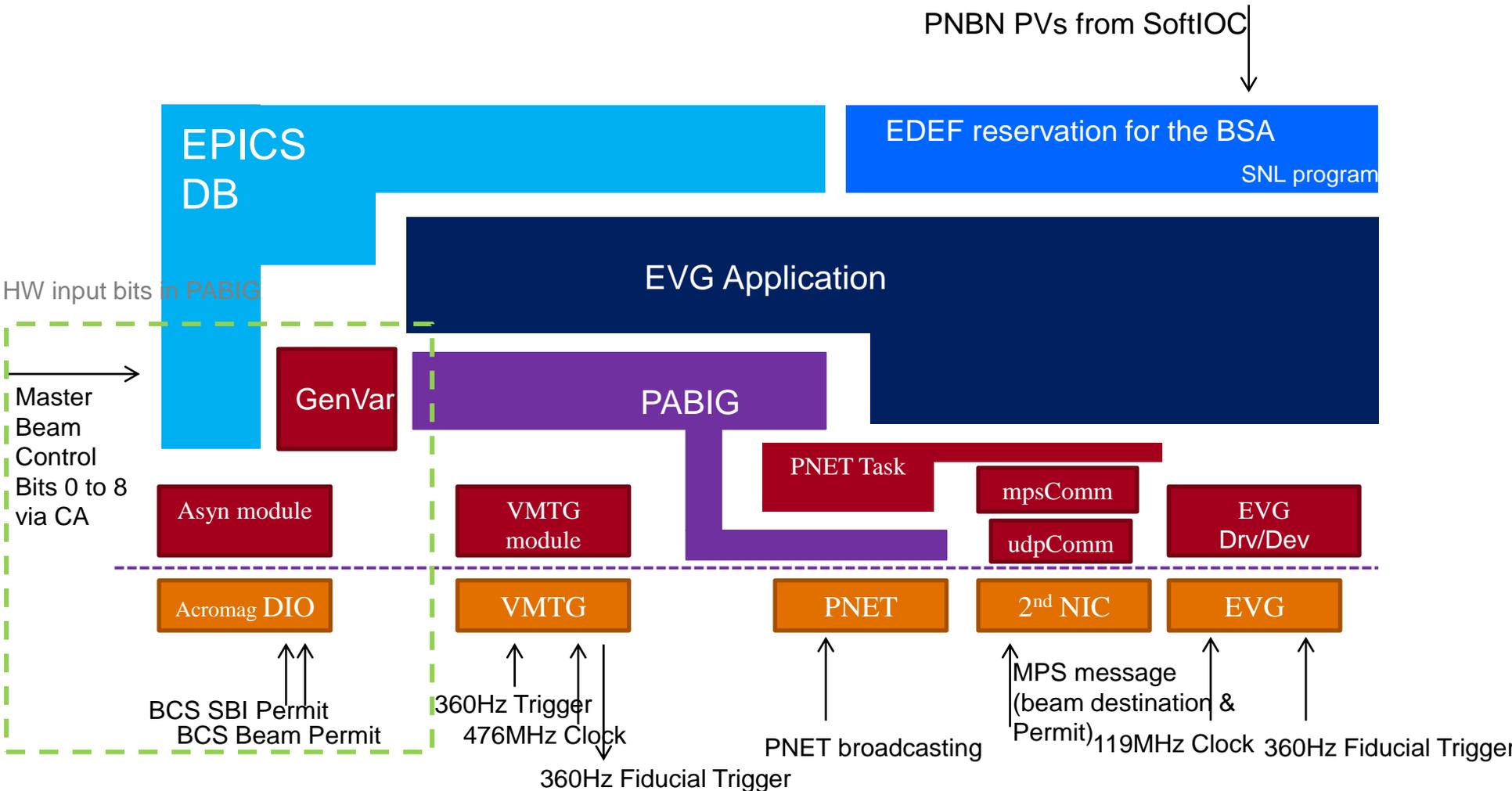
- Sector 20 (IN20) VMTG with the EVG – Master Trigger Generator
 - Provide 360 Hz interrupts with the Timeslot counter
 - Provide Timeslot Synchronization
 - Input: 60 Hz AC trigger (AC phase C) + Synchronized 360 Hz trigger
 - Output: 360 Hz interrupts + timeslot counters

Disabling PNET allows VMTG control

- PNET task needs to be delayed ~ 150 usec after the PNET IRQ to wait the MPS message.
 - PNET IRQ is using HW_TIMER to make the delay
 - The PNET ISR implement 150 usec delay on the HW_TIMER
 - The callback for the HW_TIMER releases signal to execute the PNET Task.
- VMTG has configurable IRQ delay
 - Configure the VMTG IRQ delay as ~570 usec
 - The VMTG IRQ occurs 150 usec after the PNET IRQ

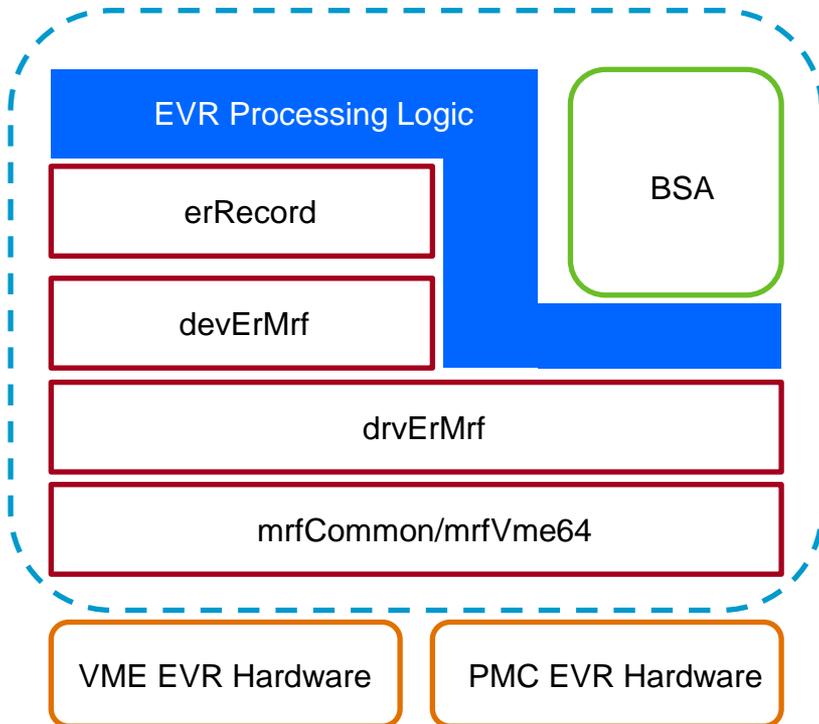


Software Stacks for the new EVG application and the PABIG



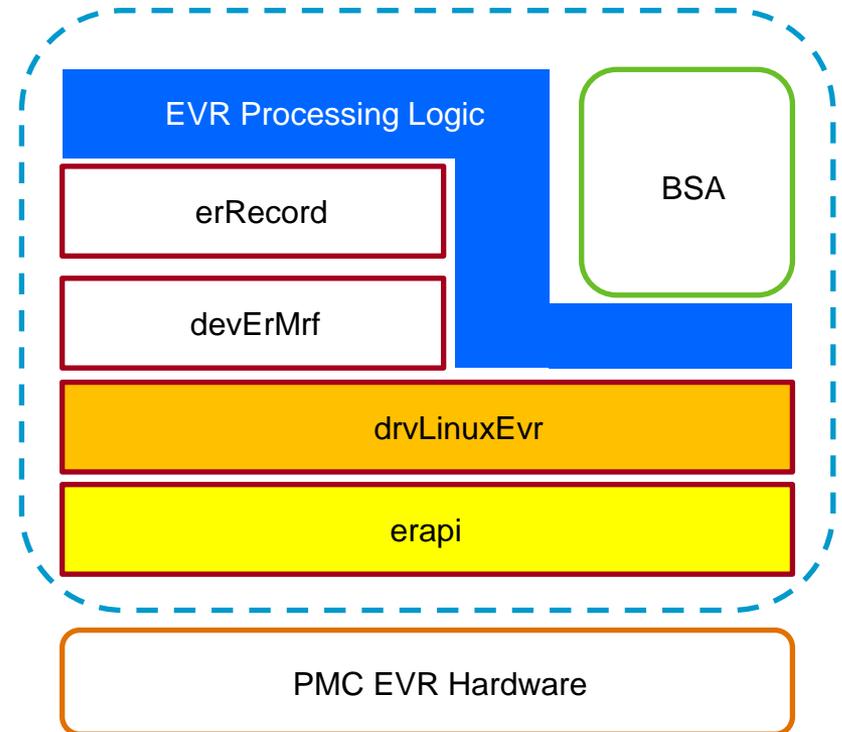
EVR Software Stacks/ Form factor & OS dependency

Event Module for RTEMS/vxWorks



Works with old register map

Event Module for linux/linuxRT



Works with modular register map (new)

- Successfully upgraded timing system
 - Standalone from the SLC system
 - Same code base for LCLS-I, LCLS-II and future projects
 - Backward compatibility, keep the advantages from the legacy system

- Our Team has done the following work:
 - Develop new VMTG hardware: [John Dusatko](#)
 - Develop PABIG software: [Till Straumann](#)
 - Develop evGUI: [Mike Zelazny](#) and [Partha Natampalli](#)
 - EVG software integration,
EVR software support for linux/linuxRT and real-time performance issues: [Kukhee Kim](#), [Till Straumann](#), [Stephanie Allison](#)

Related Talks

- LCLS Machine Protection System (M. Boyes)
in hardware solution session (16:30 – 16:50 Oct 22, 2012)
- LCLS timing system for pattern design , evGUI and high level (M. Zelazny)
in the timing system session (15:00 – 15:20 Oct 22, 2012)
- Real-Time Performance Issues on linux/linuxRT (K. Kim)
in the low level application session (14:10 – 14:30 Oct 23, 2012)

Thank You!