How to use the Timing System as a Client

April 6, 2012

Controls Software
Contents

- EVR Low Level Screens
- How to make triggers with the Low Level Screens
- What is the relationship between Low Level Screen and High Level Screen
- How to set up the EVR in your application
- Form factor & OS dependency on the EVR: go to first place with photo
- What is BSA
- How does BSA work
- How to Setup BSA
Complicated...
EVR Diag. Screen

Controls

(D1) Board information
(D2) Board Control and Monitoring
(D3) Trigger Selection Key for Front Panel
(D4) Extended Delay Front Panel Trigger
(D5) Regular Trigger Control
(D6) VME IRQ delay configuration

Don’t Use It!

IN20 EVR RF01 Diagnostics

Changes on this display are NOT restored on IOC reboot!

Module Description | PMC Card @ 2/4/0, IRQ 0x80 | FPGA Version 0x305
---|---|---

Front Panel Signal

<table>
<thead>
<tr>
<th>Trigger Output</th>
<th>State</th>
<th>Polarity</th>
<th>Width**</th>
<th>Delay*</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Enabled</td>
<td>Normal</td>
<td>12</td>
<td>108980</td>
</tr>
<tr>
<td>1</td>
<td>Enabled</td>
<td>Normal</td>
<td>12</td>
<td>110000</td>
</tr>
<tr>
<td>2</td>
<td>Enabled</td>
<td>Normal</td>
<td>12</td>
<td>110000</td>
</tr>
<tr>
<td>3</td>
<td>Enabled</td>
<td>Normal</td>
<td>12</td>
<td>110000</td>
</tr>
<tr>
<td>4</td>
<td>Enabled</td>
<td>Normal</td>
<td>12</td>
<td>110000</td>
</tr>
<tr>
<td>5</td>
<td>Enabled</td>
<td>Normal</td>
<td>12</td>
<td>110000</td>
</tr>
<tr>
<td>6</td>
<td>Enabled</td>
<td>Normal</td>
<td>12</td>
<td>110000</td>
</tr>
<tr>
<td>7</td>
<td>Enabled</td>
<td>Normal</td>
<td>12</td>
<td>110000</td>
</tr>
<tr>
<td>8</td>
<td>Enabled</td>
<td>Normal</td>
<td>12</td>
<td>107156</td>
</tr>
<tr>
<td>9</td>
<td>Enabled</td>
<td>Normal</td>
<td>12</td>
<td>110648</td>
</tr>
<tr>
<td>10</td>
<td>Enabled</td>
<td>Normal</td>
<td>12</td>
<td>110648</td>
</tr>
<tr>
<td>11</td>
<td>Enabled</td>
<td>Normal</td>
<td>12</td>
<td>110648</td>
</tr>
<tr>
<td>12</td>
<td>Enabled</td>
<td>Normal</td>
<td>12</td>
<td>110648</td>
</tr>
<tr>
<td>13</td>
<td>Enabled</td>
<td>Normal</td>
<td>12</td>
<td>110648</td>
</tr>
</tbody>
</table>

Extended Delay Trigger Output

<table>
<thead>
<tr>
<th>State</th>
<th>Polarity</th>
<th>Width***</th>
<th>Delay* Pre-Scaler</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Enabled</td>
<td>Normal</td>
<td>12</td>
</tr>
<tr>
<td>1</td>
<td>Enabled</td>
<td>Normal</td>
<td>12</td>
</tr>
<tr>
<td>2</td>
<td>Enabled</td>
<td>Normal</td>
<td>12</td>
</tr>
<tr>
<td>3</td>
<td>Enabled</td>
<td>Normal</td>
<td>12</td>
</tr>
</tbody>
</table>

VME IRQ

- Disabled

** Max pulse width is 55535 ticks. Values greater than 55535 will wrap around to a tower number.

*** Max pulse width is 4339 ticks.

* All delays and pulse widths on this panel are in periods of the 119MHz event rate. Max delay is 4339 ticks.

All delays start from the time of receipt of the event code assigned to the channel (the ‘Events’ display has detail). The actual delay is the entered value plus the intrinsic system delay which is different per channel, measured at ~300ns for VME EVHs and ~410ns for PMC EVHs, including cable delays.

Go to IOC-specific trigger panels to enter delays in msec from TREF and to change trigger output states. Changes made on these displays will be permanent and restored on IOC reboot. Changes to settings on this panel that need to be permanent must be sent to controls-software so that the bootup databases are changed.
Cont’d: EVR Diags

Controls

EVR Board
- Board Type/Instance/IRQLevel/FPGA version (see D1)
- Board Enable/Disable (see D2)
- RX link Status (D2)
- Error Counter and Reset (D2)

Extended Trigger/Front Panel Trigger (D3/D4)

Rear Panel Trigger (D5)

VME delayed IRQ (Not Use) (D6)
Controls

Front Panel Trigger
- Physically located on the front panel (D4)
  - VME: 4 Channels
  - PMC: 3 Channels
- Select the Real Trigger Channel and route to Front Panel
  - Configure with the trigger selection key from 11 to 24 (please, see the trigger selection key list in the D3 session)
  - Uses same delay and width as the rear trigger which is selected by the trigger selection key

Extended Delay Trigger
- Physically using same output channel with the front panel trigger
- Shares same event code with the rear transition
- Configure with the trigger selection key from 0 to 3 (please, see the trigger selection key list in the session D3)
- Has its own delay, width and prescaler
Extended Delay and Prescaler

- Max Width: 64k (16 bits)
- Max Delay: 4.3E+9 (32 bits)
- Need more longer delay? Then the Extended Delay
  - Extended Delay = Delay x Prescaler (32 bits)
  - Thus, we have 64 bits wide delay counter for the extended delay
- PMC EVR has only 2 prescalers for the first 2 channels
- Thus, the third channel on the front panel
  - is NOT available as the Extended Trigger
  - is available as the Front Panel Trigger
## Trigger Panel

### Controls

- **(T1) event code** for trigger generator
- **(T2) enable/disable** the event code (only for the trigger generator)
- **(T3) matrix switches** for mapping the events to the trigger channel

### Timing system

- **ICD software meeting**

---

### Table: Event Code Configuration

<table>
<thead>
<tr>
<th>Event Code</th>
<th>0*</th>
<th>1*</th>
<th>2*</th>
<th>3*</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>IRQ*</th>
</tr>
</thead>
<tbody>
<tr>
<td>02</td>
<td>Enabled</td>
<td>Disabled</td>
<td>Disabled</td>
<td>Disabled</td>
<td>Disabled</td>
<td>Disabled</td>
<td>Disabled</td>
<td>Disabled</td>
<td>Disabled</td>
<td>Disabled</td>
<td>Disabled</td>
<td>Disabled</td>
<td>Disabled</td>
<td>Disabled</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>Enabled</td>
<td>Disabled</td>
<td>Disabled</td>
<td>Disabled</td>
<td>Disabled</td>
<td>Disabled</td>
<td>Disabled</td>
<td>Disabled</td>
<td>Disabled</td>
<td>Disabled</td>
<td>Disabled</td>
<td>Disabled</td>
<td>Disabled</td>
<td>Disabled</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>Enabled</td>
<td>Disabled</td>
<td>Disabled</td>
<td>Disabled</td>
<td>Disabled</td>
<td>Disabled</td>
<td>Disabled</td>
<td>Disabled</td>
<td>Disabled</td>
<td>Disabled</td>
<td>Disabled</td>
<td>Disabled</td>
<td>Disabled</td>
<td>Disabled</td>
<td></td>
</tr>
<tr>
<td>34</td>
<td>Enabled</td>
<td>Disabled</td>
<td>Disabled</td>
<td>Disabled</td>
<td>Disabled</td>
<td>Disabled</td>
<td>Disabled</td>
<td>Disabled</td>
<td>Disabled</td>
<td>Disabled</td>
<td>Disabled</td>
<td>Disabled</td>
<td>Disabled</td>
<td>Disabled</td>
<td></td>
</tr>
<tr>
<td>34</td>
<td>Enabled</td>
<td>Disabled</td>
<td>Disabled</td>
<td>Disabled</td>
<td>Disabled</td>
<td>Disabled</td>
<td>Disabled</td>
<td>Disabled</td>
<td>Disabled</td>
<td>Disabled</td>
<td>Disabled</td>
<td>Disabled</td>
<td>Disabled</td>
<td>Disabled</td>
<td></td>
</tr>
<tr>
<td>80</td>
<td>Enabled</td>
<td>Disabled</td>
<td>Disabled</td>
<td>Disabled</td>
<td>Disabled</td>
<td>Disabled</td>
<td>Disabled</td>
<td>Disabled</td>
<td>Disabled</td>
<td>Disabled</td>
<td>Disabled</td>
<td>Disabled</td>
<td>Disabled</td>
<td>Disabled</td>
<td></td>
</tr>
</tbody>
</table>

* Either regular or extended delay trigger. Extended delay triggers are available from front panel only.

** IRQ must be enabled for event record processing and the update of event rate shown on the "Events" display. It must also be enabled for records with the ISE Field set to the event code.
How to control the trigger

Matrix switches on the trigger panel

An Internal Clock

Trigger Generator

Prescaler, Delay Counter, Width Counter

Trigger Event

Event Code PV

Enable/Disable PV

Prescaler PV, Delay PV, Width PV

Status PV, Polarity PV

EVR Diag. Panel PVs

D4 and D5

Triggers Panel PVs

(T1), (T2), (T3)

ICD software meeting
Form Factor/OS dependency

Event Module for RTEMS/vxWorks

- EVR Processing Logic
  - erRecord
  - devErMrf
  - drvErMrf
  - mrfCommon/mrfVme64

- VME EVR Hardware
- PMC EVR Hardware

Works with old register map

Event Module for Linux

- EVR Processing Logic
  - erRecord
  - devErMrf
  - drvLinuxEvr
  - erapi

- PMC EVR Hardware

Works with modular register map (new)
High Level Screen
**Issue 1: FWD/BWD Propagation**

**Controls**

- Low Level PVs on Diagnostics Screen
- High level PVs on Events Screen
- Event Number on Trigger Screen

**Timeline system**

12

**Save/Restore for High Level PVs**

**Hard-coded Event number and Trigger Configuration**

**ICD software meeting**
Issue 2: Event Code Invariant Delay

- Each Event Code has its own offset
- Each event code has to have different offset
- The delay has been hard-coded in the EVG
- EVG assumes there is no duplicated offset

- These offsets are involved in the hardware trigger calculations for trigger delay on EVR side
- But, the offset PV is hard-coded for each trigger channel
- Thus, the changing event code (or, changing trigger selection) makes different delay
Event Code and Delay

**Delay Calculation**

\[ T_{delay} = T_{ref} + T_{des} - \left\{ T_{ec} \cdot \left( \frac{1}{119 \text{ MHz}} \right) \right\} \]

- To make “event code invariant delay”, need to fix the hard-coded part
  - Require to detect changing event code (or, changing trigger selection)
  - Re-calculate the forward propagation
  - Actually, the offset of event code is a function of event code and trigger configuration

Clock Rate

Fiducial to Beam: Constant

Desired Delay

Event Code Offset by EVG

ICD software meeting
Event Module

- Pick a correct event module
  - R3-14-8-2
    - Please, use event-R3-3-1
    - for LCLS, FACET
    - back-propagation, event invariant delay
    - requires the standalone generaltime package
  - R3-14-12
    - Please use event-R3-2-2-br_generaltime-2-1
    - for LCLS, FACET and XTA
    - bundled generaltime in EPICS base
    - back-propagation, event invariant delay
    - mrfApi for linux platform
    - Info(autosaveFields_pass0, “VAL”) + PINI=YES in evrDevTrig.db
    - Additional macros to bsaTOROEdif.substitutions, egu, hopr, lopr, prec, adel

**Notice** The latest update (to fix bugs for the Linux PMC EVR) is still stuck in the HEAD of the branch for the R3-14-12. Thus, there will be another release very soon!
EVR record instance + EVR event instances

- EVR record instance / need to choose one of the followings
  - evr.db
    - Simplest one
    - DEV, CARD, DELAY, WIDTH
  - evrWithDelays.db
    - More degree of freedom for delays
    - DEV, CARD, DELAY[0_9, A_D], WIDTH
  - evrWithExtDelays.db
    - evr.db + more degree of freedom for front panel trigger channels
    - State, Width, Delay, Prescaler, Polarity for each front panel channels
  - evrWithFrontPanel.db
    - Similar with evrWithFronPanel
    - Disable Rear panel trigger
    - Less degree of freedom: DEV, CARD, WIDTH DELAY
Cont’d: EVR Instance

EVR record instance and the EVR Diag Screen
Controls

EVR event instances
- controls the mapping between event code and trigger channel
- Use evrEventCtrl.db
- DEV, CARD, NAME, ID, P0-P13, VME
  - NAME: EVENT1 to EVENT14
  - ID: Event code
  - Pn: 0 to 1 for the matrix switch
  - VME: Enable the softevent

ICD software meeting

Cont’d
Example of *evr.substitutions

file evr\ithDelays.db
{ #
  Device  Card  Pulse  Channel Delays (clock ticks)
  pattern { DEV  ,  CARD  ,  WIDTH ,  DELAY0, DELAY1, DELAY2, DELAY3, DELAY4, DELAY5, DELAY6, DELAY7, DELAY8, DELAY9, DELAYA, DELAYB, DELAYC, DELAYD }
    { EVR:LI24:RF01,  0 ,  12 , 110000, 110000, 110000, 110000, 110000, 110000, 110000, 110000, 110000, 110000, 110000, 110000, 110000, 110000 } }

file evrEventCtrl.db
{ #
  Device  Card  NAME  ID  P0,P1,P2,P3,P4,P5,P6,P7,P8,P9,P10,P11,P12,P13,WME
  pattern { DEV  ,  CARD  ,  NAME ,  ID ,P0,P1,P2,P3,P4,P5,P6,P7,P8,P9,P10,P11,P12,P13,WME }
    { EVR:LI24:RF01,  0 ,  EVENT1 ,  40 ,  0,  1,  0,  1,  0,  1,  0,  0,  0,  0,  0,  0,  0,  0 } }

Timing system

ICD software meeting
*pattern.substitutions*

**Controls**

First Session for evrPattern.db
- Records for the pipeline related PVs
- Just use evrPatternAll.db
- Need to specify ACTIVE TIMESLOT
  - LCLS: 1 and 4
  - FACET: 2 and 5
  - XTA: 3 and 6

Second Session for Events
- for the Event records
- Each beam program has different event sets
- Need to use a proper template
  - LCLS: evrEventAll.db
  - FACET: evrEventFACET.db
  - XTA: evrEventXTA.db
Example of *pattern.substitutions

Specify proper ACTIVE TIMESLOT for each beam program

Choose a proper template for the each beam program
What is the Timeslot

- Zero Crossings at AC 3 phases lead out the 6 time slots
- Same Timeslot in different period shows exactly same AC phase configuration.
- Active Timeslot
  - LCLS: TS1 and TS4
  - FACET: TS2 and TS5
  - XTA: TS3 and TS6
- Primary Timeslot

Timing system

ICD software meeting
*trig.substitutions*

- Make PV for the High Level Screen
- Use, evrDevTrig.db
- Add 2 new macros for new features
  - Backward Propagation
  - Event Code Invariant Delay
  - TOUT
    - Make mapping between logic and output channel
    - OUT0 to OUT9 and OUTA to OUTD
    - For the front panel re-use OUT0 to OUT3
  - ACTV
    - Enable/Disable the event code invariant delay
- Timing Usability project to auto-generate trigger EDM screens
Example of *trig.substitutions

For trigger screen autogeneration:
1) Add comment tags & 2) modify Makefile (next slide)

```c
#include <stdio.h>

int main() {
    // Example of *trig.substitutions
    #CONTROLPV1 SIOC:SYS0:AL00:MODE
    #CONTROLPV2 SIOC:SYS0:AL00:TOD
    #NEW MACROS
    Optional tags for LCLS; necessary for other facilities

    // Example code
    #define PIOV1 SIOC:SYS0:AL00:MODE
    #define PIOV2 SIOC:SYS0:AL00:TOD

    // Example usage
    PIOV1 = 1;
    PIOV2 = 0;

    return 0;
}
```
Example of xApp/Db Makefile

```makefile
TOP=../../..
include $(TOP)/configure/CONFIG

# Create and install (or just install) into <top>/db # databases, templates, substitutions like this
DB += IOC-XT01-IM01-adc.db
DB += IOC-XT01-IM01evr.db
DB += IOC-XT01-IM01pattern.db
DB += IOC-XT01-IM01bsa.db
DB += IOC-XT01-IM01trig.db

NEW target
• Pass in *trig.substitutions file
  • For multiple EVRs, one file contains substitutions for every EVR, separated with appropriate #EVR, #FILE tags
  • edl trigger files specified by #FILE are output to $EDM/install directory

gen_trig_edl:
    create_edm_event_msi.py IOC-XT01-IM01trig.substitutions

include $(TOP)/configure/RULES
#----------------------------------------
# ADD RULES AFTER THIS LINE
```

• Infrastructure will be ready for engineers to build after next week PAMM (4/10)
Bean Synchronous Acquisition (BSA)

- Acquire beam dependent scalar values across multiple IOCs to analyze the correlations among the values which are acquired at the same pulse.
- Maintain the buffer up to 2800 points.
- The buffered values can be averaged up to 1000 samples.
- Up to 20 different BSA requests are available.
- Each BSA request can specify:
  - Beam Code
  - Inclusion/Exclusion Masks for the Event Pattern
  - Measurement Count (number of data points)
  - Average per Measurement
  - Severity Level
How to Setup the BSA request

### EVG side

- Bring up the Event Global
- Choose Event Definition
- Make EDEF reservation with your own name
- Bring up your EDEF slot
- Set up Masks / Beam Code / Measurements/ Average/Severity
- Turn ON to start BSA
- Finally release your EDEF slot
Bring up the EDEF screen

Event Global Screen
Make EDEF Reservation

### Controls

**SYS0 Event Definitions (EDEFs)**

<table>
<thead>
<tr>
<th>App Name</th>
<th>User</th>
<th>Last Active Time</th>
<th>Last Reserved Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>zelazny</td>
<td>OFF 04/03/12 10:22:57</td>
<td>OFF 03/22/12 11:12:20</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>1HZ</td>
<td>ON 03/21/12 16:42:12</td>
<td>ON 03/21/12 16:42:12</td>
</tr>
<tr>
<td>18</td>
<td>10HZ</td>
<td>ON 03/21/12 16:42:12</td>
<td>ON 03/21/12 16:42:12</td>
</tr>
<tr>
<td>19</td>
<td>FULL</td>
<td>ON 03/21/12 16:42:12</td>
<td>ON 03/21/12 16:42:12</td>
</tr>
<tr>
<td>20</td>
<td>FBCK1</td>
<td>ON 03/21/12 16:42:12</td>
<td>ON 03/21/12 16:42:12</td>
</tr>
<tr>
<td>21</td>
<td>FBCK2</td>
<td>ON 03/21/12 16:42:12</td>
<td>ON 03/21/12 16:42:12</td>
</tr>
</tbody>
</table>

**SYS0 Event Definitions (EDEFs)**

<table>
<thead>
<tr>
<th>App Name</th>
<th>User</th>
<th>Last Active Time</th>
<th>Last Reserved Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>OFF 03/03/12 10:42:12</td>
<td>OFF 03/03/12 10:22:12</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>1HZ</td>
<td>ON 03/21/12 16:42:12</td>
<td>ON 03/21/12 16:42:12</td>
</tr>
<tr>
<td>18</td>
<td>10HZ</td>
<td>ON 03/21/12 16:42:12</td>
<td>ON 03/21/12 16:42:12</td>
</tr>
<tr>
<td>19</td>
<td>FULL</td>
<td>ON 03/21/12 16:42:12</td>
<td>ON 03/21/12 16:42:12</td>
</tr>
<tr>
<td>20</td>
<td>FBCK1</td>
<td>ON 03/21/12 16:42:12</td>
<td>ON 03/21/12 16:42:12</td>
</tr>
<tr>
<td>21</td>
<td>FBCK2</td>
<td>ON 03/21/12 16:42:12</td>
<td>ON 03/21/12 16:42:12</td>
</tr>
</tbody>
</table>

Timing system

ICD software meeting
Play with your EDEF slot

EDEF Slot

Mask Setup

ICD software meeting
Pipeline, Pattern & Event code

Controls

EVG

Generate New pattern at 3 pulses prior!

Pipeline Advancing in the EVG

Dealing with the next1 pattern
Pipeline index =1 is hard-coded in the database

Decide event code list with the !Next1! pattern

Construct EDEF data (for BSA) from the MOD5 & EDEF Masks

EVR

Fiber connection to EVR

Pipeline Advancing in the EVR

Trigger/Event Generation by the Event Code

Re-construct EDEF data (for BSA) from the MOD5 & EDEF Masks

ICD software meeting

Timing system 31
BSA & EDEF: EVR Message

EVG side

Pattern for Next3

Modifiers

TimeStamp

EDEF masks

MOD5 for BSA masks (active)

AvgDone, Minor, Major, Init

EDEF masks

EVG&EVR do exactly same processing to update the EDEF table

Fiber optic connection

Step 5

Update the EDEF table, after complete the pipeline advancing

EVR side

Pattern Pipeline

Current

Next1

Next2

Next3

Step 1

Step 2

Step 3

Step 4

DEDEF Table

DEDEF 0

DEDEF 1

DEDEF n

DEDEF 19

Timestamp (active)

Timestamp (Init)

avgDone flag

Severity

Timing system

ICD software meeting
BSA processing

Update EDEF table (DEDEF 0, DEDEF 1, DEDEF n, DEDEF 19) after the pipeline advancing.

Internal BSA Data Table

- BSA device name1
- BSA device name2
- BSA device name M

Update data value, timestamp, status, and severity which come from the DATA PV.

DATA PV for BSA device name M

AO record: Data receptor

AO record does the BLUE box and make record processing for correct BSA record(s). And the BSA record update the BSA buffer.

ICD software meeting

Timing system 33

avgDone flag

Severity

Timestamp (active)

Timestamp (Init)

BSA0

BSA1

BSA n

BSA 19

BSA device name L

BSA device name M

BSA device name 1

BSA device name 2

BSA device name M

BSA device name 1

BSA device name 2
How to setup the BSA in your application

Data Source

BSA

History Buffers
- Compression record: for value
- Compression record: for average
- Compression record: for rms

IOSCAN request to make record processing

INP (INST type)
- Specify secName to make connection with data block

BSA record

OUT link: (INST type)
- Specify secName to make connection with data block

AO record

BSA Device Support

EVR Device Support

Your Application

BSA Package in Event Module
<system> bsa.substitutions

### Controls

- Pick a proper template: `<…>` Edef.db
- Example

```
file bsaPHASEdef.db
{
    # Device , loc'n, IOC, Attribute
    # Future. To standardize for SCORE
    # { TCAV:LI24:800 , LI24 , RF01 , P , TCAV:LI24:800:TC3_PAVG, "" }
    # Future. To standardize for SCORE
    # { TCAV:LI24:800 , LI24 , RF01 , P , TCAV:LI24:800:P , "" }
    # Future. To standardize for SCORE
    # { PCAV:LI25:300 , LI24 , RF01 , P , PCAV:LI25:300:PH3_3_PACT , "" }
    # Future. To standardize for SCORE
    # Future. To standardize for SCORE
    # { PCAV:LI29:100 , LI24 , RF01 , P , PCAV:LI29:100:PH4_3_PACT , "" }
    # Future. To standardize for SCORE
    # { PCAV:LI29:100 , LI24 , RF01 , P , PCAV:LI29:100:P , "" }
}
```

```
file bsaAMPLEdef.db
{
    # Device , loc'n, IOC, Attribute
    # Future. To standardize for SCORE
    # { TCAV:LI24:800 , LI24 , RF01 , A , TCAV:LI24:800:TC3_AAVG, "" }
    # Future. To standardize for SCORE
    # { TCAV:LI24:800 , LI24 , RF01 , A , TCAV:LI24:800:A , "" }
    # Future. To standardize for SCORE
    # Future. To standardize for SCORE
    # Future. To standardize for SCORE
    # { PCAV:LI29:100 , LI24 , RF01 , A , PCAV:LI29:100:PH4_3_AACT , "" }
    # Future. To standardize for SCORE
    # { PCAV:LI29:100 , LI24 , RF01 , A , PCAV:LI29:100:A , "" }
    # Future. To standardize for SCORE
    # { PCAV:LI29:100 , LI24 , RF01 , A , PCAV:LI29:100:PH4_3_AACT , "" }
}
```

### Session for LLRF Phase

- BSA device name
- Location
- Unit
- Attribute
- Data PV name

### Session for LLRF Amplitude

- BSA device name
- Location
- Unit
- Attribute
- Data PV name

---

**ICD software meeting**

**Timing system**

35

---

**SLAC National Accelerator Laboratory**
### BSA templates

<table>
<thead>
<tr>
<th>Template</th>
<th>Description</th>
<th>BSA properties</th>
</tr>
</thead>
<tbody>
<tr>
<td>bsaAMPLEdef.db</td>
<td>RF Amplitude</td>
<td>Amplitude</td>
</tr>
<tr>
<td>bsaPHASEdef.db</td>
<td>RF Phase</td>
<td>Phase</td>
</tr>
<tr>
<td>bsaBLENEdef.db</td>
<td>Bunch Length</td>
<td>RAW, IMAX</td>
</tr>
<tr>
<td>bsaBLENTest.db</td>
<td>Fake for Test</td>
<td></td>
</tr>
<tr>
<td>bsaBPMSCavityEdef.db</td>
<td>Cavity BPMs</td>
<td>Re(Hor), Im(Hor), Re(Ver), Im(Ver), RAW ref</td>
</tr>
<tr>
<td>bsaBPMSEdef.db</td>
<td>BPMs (stripline?)</td>
<td>X, Y, TMIT</td>
</tr>
<tr>
<td>bsaBPMSTest.db</td>
<td>Fake for Test</td>
<td></td>
</tr>
<tr>
<td>bsaEnergyEdef.db</td>
<td>Beam Energy</td>
<td>Single Attribute</td>
</tr>
<tr>
<td>bsaFARCEdef.db</td>
<td>Faraday Cup/Bunch Charge</td>
<td>Charge</td>
</tr>
<tr>
<td>bsaFARCTest.db</td>
<td>Fake for Test</td>
<td></td>
</tr>
<tr>
<td>bsaPMTEdef.db</td>
<td>PMT</td>
<td>QADC Raw</td>
</tr>
<tr>
<td>bsaPMTTest.db</td>
<td>Fake for Test</td>
<td></td>
</tr>
<tr>
<td>bsaPWREdef.db</td>
<td>Beam Power</td>
<td>Power</td>
</tr>
<tr>
<td>bsaTOROEdf.db</td>
<td>Toroids/Beam current</td>
<td>TIMIT</td>
</tr>
<tr>
<td>bsaTOROTest.db</td>
<td>Fake for Test</td>
<td></td>
</tr>
<tr>
<td>bsaWIREEdef.db</td>
<td>Wire Scanner</td>
<td>Position, Mask</td>
</tr>
<tr>
<td>bsaWIRETest.db</td>
<td>Fake for Test</td>
<td></td>
</tr>
</tbody>
</table>
BSA Check Point

- DOL field in data receptor
  - AO record in BSA facility
  - The data receptor gets data, timestamp, and severity from the DOL LINK.
  - DOL should be pointed your data source

- Data Source PV
  - Timestamp (BSA aware)
  - FLNK to the data receptor
BSA aware Timestamp

Basically, BSA facility compares the timestamp from data source and BSA event definition.

Assume, the timestamp from data source reflects ACTIVE_TIMESLOT and Pulse ID:
- ACTIVE_TIMESLOT: TS1 and TS4, event#0 in LCLS Event system
- Pulse ID: lower 17 bits in nano-sec in timestamp

TSE=-2 for your data source PV:
- If, our device support for data source PV, takes care the timestamp
- Somewhere in device support should call the following function: evrTimeGet(&pMyRec->timestamp, 0)

TSE=0, -1 for your data source PV:
- If we can guarantee that record processing must be finished within 8.33 msec (time interval between active timeslot 1/120).

TSE>1 for your data source PV:
- If we can not guarantee, the record processing can be finished within 8.33 msec (time interval between active time slot 1/120)
- Or, data gettering is not synchronized with beam or event system.
BSA aware timestamp and BSA Success/Fail scenario (1)

Success Scenario: if TSE=(0,-1) and Guaranteed everything can be finished before next active time slot (within 8.3 msec)

- Device support gets data for TSn
- When the record instance gets timestamp, it should be still TSn
BSA aware timestamp 
and BSA fail/success scenario (2)

Fail Scenario: if TSE=0, -1
- Device support successfully get data for TS\text{n}
- But, need to assume, there should be undefined delay before start record processing
- And, record processing might be not finished until next active timeslot
- The record (the data source PV) time stamp gets TS\text{n+1}
- Then, BSA could be failed. BSA assumed TS\text{n} timestamp for the data source PV
BSA aware timestamp and BSA fail/success scenario (3)

Success Scenario: if TSE=-2

- Device support provides timestamp TSn to the record instance
- Even, record processing is finished after TSn+1, the timestamp should be TSn.
- BSA is succeeded
BSA aware timestamp and BSA fail/success scenario (4)

**Controls**

- **Device support processing**
- **Record support processing**
- **Undefined delay between Dev. and Rec.**

---

**Timestamps for active time slot**

- **TSn**
- **TSn+1**

8.3 msec interval

---

**Event #** | **Description** | **Timestamp**
--- | --- | ---
0 | Active timeslot | 03/23/2011 14:30:11.1343234234
1 | 360Hz Fiducial | 03/23/2011 14:30:11.12343443343

---

(1) Event system update internal timestamp tables with the given event number

(2) Record gets timestamp from the Event System Internal Table with the given event number

**Success Scenario:** if TSE=m (event for ▲)

- Need to Enable VME IRQ for event m
- Event System maintain internal timestamp table for the events (including event m)
- Record gets latest timestamp from table for event number m

---

**Timing system**

42

---

**ICD software meeting**
Example for Scenario (1)

Guarantee that everything can be finished within 8.3 msec

LLRF PAD reading

HW Trigger

PAD

FNET

udpComm

DevPadDataProc

AI record

SCAN=I/O Intr
TSE=0, -1

DOL

FLNK

AO in BSA
Example for Scenario (3)

BPM PAD reading

Device support handles
Timestamp for data source PVs
Example of Scenario (4): Long Processing Time

Record processing is spilled out to the next active time slot, but is not violated the dead-line for the next event.

HW Trigger with Event code \( n \)

Motor HW

AI record (Data Source)

AO in BSA

SCAN=I/O Intr

TSE=\( n \)

Get timestamp from the latest event \( n \), instead of the current active timeslot
Example of Scenario (4) Asynchronous BSA?

No Sync with BSA

- HW
- Motor Record
  - SCAN=I/O Intr
- AI record (Data Source)
  - SCAN=Event
  - EVENT=n
  - TSE=n
- AO in BSA
  - DOL
  - FLNK

Controls
BSA Check Point AGAIN!

(*1) check up your data source driving mechanism
(*2) check up TSE in your data source, Please remind the success/fail scenario
(*3) (*5) check up if the PV name matched, DOL in data receptor and INP in BSA
(*4) check up the FLNK, your data source should drive the data receptor

(*1) SCAN=I/O Intr? or EVENT?
(*2) TSE=0, -2

(*3)DOL
(*4)FLNK

Data receptor

BSA Record

Device Support

Data Source PV

AO in BSA

INP=@PV name for Data Source

Data receptor

(*1) check up your data source driving mechanism
(*2) check up TSE in your data source, Please remind the success/fail scenario
(*3) (*5) check up if the PV name matched, DOL in data receptor and INP in BSA
(*4) check up the FLNK, your data source should drive the data receptor