Introduction

The purpose of this document is to describe how the new TPR (Timing Pattern Receiver) will replace the existing EVR (EVent Receiver) to do timing in the shared LCLS-I / LCLS-II sections of the beamline for slow devices. While the details of what is provided on the timing fiber applies to other areas, this document is not specifically discussing:

- The LCLS-I-only areas of the beamline (which could either continue to use the existing **event** modules with an EVR or convert to the TPR, as desired), or
- The high-performance systems (HPS) on the LCLS-II beamline, which will have specialized hardware and firmware to do data acquisition, timestamping, and BSA.

The details of what is sent over the timing fibers which are given in the appendices to this document do apply to these regions, though.

The LCLS-I timing system has a few drawbacks that makes it unsuitable for use in LCLS-II, the most important of which is the amount of data being transmitted for each fiducial event. As the event codes transmitted by the EVG (EVent Generator) are used to produce timing signals, each unique event code must occur at a fixed delay from its fiducial, otherwise the timing between the fiducial and the actual trigger pulse will vary from shot to shot. LCLS-I has:

- 7 AC synchronized rates (120 Hz, 60 Hz, 30 Hz, 10 Hz, 5 Hz, 1 Hz, 0.5 Hz) each of which has 6 possible phases, for a total of 42 slots. (Admittedly, the 120 Hz signals are actually a combination of two different 60 Hz phases, so there are really only 3 unique 120 Hz signals. However, the current LCLS-I timing system sends out all 6, and saving three or even six event codes here does not really change the fundamental argument.)
- 7 AC synchronized rates with beam present.
- 80 sequencer event codes.
- 33 time slots for programming the fiducial count into the timestamp register (32 shift events, and 1 load).
- Roughly 17 other events relating to the machine state.
- 5 user-defined events.

This is a total of at least 184 event code slots, which at 119MHz would take over 1.5us to transmit. At a maximum beam rate of 929kHz, LCLS-II has less than 1.08us between pulses.

The LCLS-II solution is to change the serial protocol to be more bandwidth efficient. The TPR is the LCLS-II replacement for the EVR which both operates at a faster rate (two-byte words sent at 186 MHz instead of 119 MHz) and has a more compact frame. It also has an LCLS-I timing fiber input, which allows it to switch operational modes between the two timing sources.

The TPR has a few features that make it superior to the EVR, besides its higher bandwidth operation. To begin with, all of the timing events arrive in a single frame, while in the EVR, different event codes arrive at different offsets from the fiducial. While the **event** module is capable of adjusting for this in software, no such adjustment is required for the TPR. Secondly, the TPR communicates to the software by copying information from selected timing frames into memory. As there is sufficient buffering, there should never be an issue with timing data being lost (as there is with the single data buffer in the EVR). Furthermore, these message buffers can be read by multiple user-space processes without difficulty, unlike the EVR event queue, which cannot be directly shared between processes as each read causes the hardware queue to advance. The LCLS-I timing system also sends most of the timing information two fiducials before the actual event, which has the unfortunate side effect that IOCs need to listen to all fiducial events at 360Hz, not just the ones that correspond to a data acquisition at 120Hz or less.

While it is certainly possible to embed an EVR-compatible subsystem into the TPR, a better alternative is to rewrite the LCLS-I support to make it look more like the LCLS-II message-based approach. The remainder of this document describes TPR messages for both LCLS-I and LCLS-II, how the TPR kernel module will present timing messages to user space processes, and how BSA for slow devices using a TPR will work.

TPR Operation

In LCLS-II mode, the TPR receives timing frames and depending on the device settings, generates triggers and/or sends DMA messages to the host. These messages are at most 128 bytes long and have the following 8-byte header:

Byte Offset	Size (bytes)	Name	Description
0	2	Channels	A channel number if Tag=BSA_EVENT, otherwise a channel mask.
2	1	Tag	Bit 7 previous message(s) dropped Bit 6 LCLS-I message Bits 3:0 Type of message: 0=EVENT, 1=BSA_CONTROL, 2=BSA_EVENT, 15=END
3	1	DMA ownership	Bit 7 new message Bit 6 previous message(s) dropped

The EVENT message contains information about a selected timing event:

Byte Offset	Size (bytes)	Name	Description
4	4	Length	Total number of four-byte words in this message, not including the header.
8	8	PulseID	Monotonically increasing pulse identifier / fiducial.
16	4	Timestamp nanoseconds	Time past 1990 epoch.
20	4	Timestamp seconds	Time past 1990 epoch.
24	2	Rates	bits 15:10 - AC rates (60, 30, 10, 5, 1, 0.5 Hz, MSB to LSB) bits 9:7 - Unused bits 6:0 - Fixed rates (186 MHz divided by one of 1, 13, 91, 910, 9100, 91000,

			910000, MSB to LSB)
26	2	Timeslot	bit 15 - 71 kHz resync marker bits 14:3 - Clocks since timeslot change bits 2:0 - AC timeslot (1-6)
28	4	Beam Req	bits 31:16 charge (pC) bits 7:4 destination bit 0 beam present
32	8	Beam Energy	Four 16-bit estimates of beam energy.
40	4	Photon Wavelen	Two 16-bit estimates of photon wavelength
44	2	Status	Timing, MPS, BCS sync status
46	2	MPS Limit	Machine Protection Limit: 16 bits indicating which beam destinations are limited by MPS
48	8	MPS Class	Machine Protection Allow 4 bits per destination indicating the max allowed “power class”.
56	36	Sequences	Control Sequence bits

The BSA_CONTROL message sends global information about BSA that applies to all channels.

Byte Offset	Size (bytes)	Name	Description
4	8	PulseId	Monotonically increasing pulse identifier / fiducial.
12	4	Timestamp nanoseconds	Time past 1990 epoch.
16	4	Timestamp seconds	Time past 1990 epoch.
20	8	BSA Init Mask	Latch time and clear BSA buffers with bits set.
28	8	BSA Minor Alarm Mask	If BSA Init and this is set, allow acquisition of data with minor alarm.
36	8	BSA Major	If BSA Init and this is set, allow

		Alarm Mask	acquisition of data with major alarm. If minor alarm mask also set, then allow acquisition of data with invalid alarm.

The BSA_EVENT message is sent to a particular channel to indicate that an EVENT for this channel has occurred and should be included in a BSA.

Byte Offset	Size (bytes)	Name	Description
4	8	Active Pulse ID	The pulse ID that triggered this event.
12	8	BSA Active	The mask of BSA for which the data acquired at Pulse ID should be added to the running sum.
20	8	BSA AvgDone	The mask of BSA for which the running sum should be pushed into the buffer and the sum reset.
28	4	Timestamp nanoseconds	Time past 1990 epoch
32	4	Timestamp seconds	Time past 1990 epoch
36	8	BSA Update	The mask of BSA which should push their data to PVs

In LCLS-I mode, it is desirable for the TPR behavior to be as similar to the LCLS-II behavior as possible. While it is certainly possible to simply emulate an EVR, it is likely that better performance can be achieved by pushing some of the routine pattern maintenance tasks into the TPR firmware.

Therefore, in LCLS-I mode, the TPR will aggregate pattern and event code information for a given fiducial event and send a single message for events of interest to software. “Time zero” (aka **TREF**), the time of message transmission and event triggering, must therefore occur after the last event code of interest. Since the pattern itself has sufficient information to decode the AC rate events, this would mean some time after 12116 ticks after the fiducial event. (The event-invariant timing in the current **event** module usually sets **TREF** to coincide with event code 140, which occurs 11900 ticks after the fiducial, so this is not much of an additional delay: < 2μs.) The TPR **Event Select** register will still be used to select events:

- Using “fixed” mode will select nothing.
- Using “AC” mode will compare the rate and timeslot masks in the register to the AC rates in modifier 5 and the timeslot masks in modifier 2.

- Using “seq” mode will select the specified event code.

Therefore, the BSA_CONTROL and BSA_EVENT messages will be unchanged for LCLS-I, except for the source of the information and the reduced number of BSA buffers. The EVENT message will have the following structure:

Byte Offset	Size (bytes)	Name	Description
4	4	Length	Total number of four-byte words in this message, not including the header.
8	8	PulseID	Pulse identifier / fiducial. Note that for LCLS-I this is actually only 17-bits. (Maximum value 0x1fffe0.)
16	4	Timestamp nanoseconds	Time past 1990 epoch.
20	4	Timestamp seconds	Time past 1990 epoch.
24	2	Rates	bits 15:10 - AC rates (60, 30, 10, 5, 1, 0.5 Hz) bits 9:0 - zero
26	2	Timeslot	bit 15:3 - zero bits 2:0 - AC timeslot (1-6)
28	4	Beam Req	bits 31:16 - zero bits 7:4 destination (0=D10DMP, 1=LI25, 2=UND) bit 0 beam present (POCKCELL)
32	24	LCLS-I Modifiers	Modifier words 1-6.
56	32	Event Codes	Little-endian bit vector of event codes received for this fiducial. (Note: the AC rate event codes, 9-16, 20-26, 30-36, 40-46, 50-56, and 60-66 are <u>not</u> indicated, although bits are allocated for them.)
88	4	Reserved	-

For efficiency in the firmware implementation, all messages that originate from the same timing strobe (929kHz or 360Hz) will be concatenated with the message tag headers separating them. All message groups are terminated with the END tag. The order will be preserved.

TPR Kernel Module

The TPR kernel module allows programming of the TPR registers and queueing of received TPR messages on a per-channel basis.

Minor device 14 is the master device that is used to memory map the TprBase registers described above for read/write access. At most one process is permitted to open this minor device. This process will be the IOC that controls and configures the TPR.

Minor devices 0-13 and 15 are devices that permit read-only memory mapping of the message queues. Multiple processes are permitted to open these minor devices. The memory map for all of these devices is identical and described by the following bit of C code:

```
struct TprEntry {
    /* A 128-byte message buffer. */
    uint32 message[32];
};
struct TprQueues {
    /* Shared EVENT/BSA_CONTROL messages */
    struct TprEntry allq[32768];
    /* BSA_CNTL and BSA_EVENT messages */
    struct TprEntry bsaq[1024];
    /* Per channel indices into allq */
    long long allrp[12][32768];
    /* Per channel write pointers into allrp */
    long long allwp[12];
    /* Write pointer into bsaq */
    long long bsawp;
    /* Global write pointer into allq */
    long long gwp;
} *memory_map;
```

While the memory map for these minor devices is identical, the behavior on a `read` is different. If new messages have been received on channel N {N=0..13}, minor device N will read a 1 from an open file. Otherwise, the `read` will block, or if non-blocking, return **EAGAIN**. Similarly, reading minor device N returns 1 when BSA messages have been received else it blocks.

A user-space process that is interested in a particular channel N will do the following:

- Open the corresponding minor device and memory map the queues.
- Open the BSA device (minor device 15) and memory map the queues (same map as above).
- Grab the current values of `allwp[N]` and `bsawp` into `arp` and `brp` respectively.
- Enter the main loop:

- `read` (or `select`/`poll`) the open file(s). This will not return until new messages for channel N (or BSA) have been received.
- While `arp < allwp[N]` (or `brp < bsawp)`,
  - Read and process a message. (Either from `allq[allrp[N][arp & 32767] & 32767]` or `bsaq[brp & 1023]`, respectively.)
  - Increment `arp` or `brp`.
- This can be done in two threads.

Interrupt Handling

The kernel module enables and disables interrupts by writing to the interrupt control register in the TPR card. When a device file is opened, interrupts are enabled. When an interrupt is received by the kernel, the kernel module disables interrupts and calls the interrupt handler. The interrupt handler processes all pending messages, and then enables the interrupts upon its completion if there are still any minor device files open for the device.

Slow BSA Using the TPR

For slow devices, it is very likely that the detector will not have the ability to sample exactly one pulse at 929kHz. Therefore, the TPR introduces the idea of a BSA sensitivity window using the BSADelay and BSAWidth registers. Whenever an EVENT is received on a channel with BSA enabled, if BSA Active or BSA AvgDone events occur within the sensitivity window, a BSA_EVENT will be sent to this channel. If BSA Active is sent but not BSA AvgDone, the TPR will continue to listen outside of the sensitivity window and send the AvgDone when it eventually arrives. (For efficiency, this might be delayed so that several BSA_EVENTs can be aggregated, but in any event it will be sent before the next BSA Update.)

The TPR will see to it that even at slow rates, a consistent BSA picture is presented. If more than one BSA Active or BSA AvgDone event occurs during the sensitivity window, only one will be presented to software, so BSA events never occur faster than data acquisition.

Software BSA processing for a particular buffer is as follows:

- BSA Init is received. (This will never coincide with AvgDone/Active, so LCLS-II will use these bits in the timing frame as the major/minor alarm masks.) The BSA buffer will be cleared, and the timestamp and major/minor alarm masks will be latched.
- One (or more for LCLS-II) update sequences will be received.
    - One or more averaging sequences will be received.
        - A sequence of BSA Active events are received. Data will be added to a running average for each event.
        - A BSA AvgDone event is received. The current running average will be pushed into an internal buffer and the timestamp/pulse ID will be recorded.
    - A BSA Update event is received. The internal buffers will be pushed into PVs. (As LCLS-I does not have a separate “BSA Update” bit, the TPR will generate a BSA Update for each BSA AvgDone.)

Pulse IDs and Timestamps

In LCLS-I, the 17-bit fiducial ID is embedded into the 64-bit timestamp. However, this is not the case for LCLS-II, which has a 64-bit pulse ID as well as a 64-bit timestamp. LCLS-I software often takes advantage of this, as knowing the timestamp means that you know the pulse ID as well. (Both the BLD and BSA modules use this feature.) However, this is not the case for LCLS-II, although there is still a one-to-one mapping between them. The chosen solution to this problem is to assume that the EPICS TPR module will continue to use the timestamps as expected and provide a translation service so that given a timestamp, the matching pulse ID can be found. The LCLS-II DAQ will need to use the 64-bit timestamps and provide its own translation service, if necessary.

Event Sequencer

The timing system for LCLS-II will have 18 groups of programmable control bits (16 bits per group) to be used for generic trigger control. This is analogous to the 80 event codes reserved in LCLS-I for experiment event sequencing. While LCLS-I event codes are set via a software process in the EVG, the LCLS-II control sequences will be set by firmware engines running in the TPG.

Each 16-bit group has its own programmable engine. The engine is programmed with a set of instructions consisting of:

Instruction	Parameters	Range	Description
Fixed Rate Sync	Marker (m), Occurrence(n)	[1Hz-1MHz] [1-2048]	Wait for the nth occurrence of fixed rate marker m.
AC Rate Sync	Marker (m), time slot mask (s), Occurrence(n)	[0.5Hz-60Hz] [OR 1-6] [1-2048]	Wait for the nth occurrence of an AC rate marker m matching the timeslot mask s.
Branch unconditional	Instruction (i)	[0-2047]	Jump to instruction # i.
Branch conditional	Instruction (i), repeat (n), counter (c)	[0-2047] [1-256] [a,b,c,d]	Jump to instruction # i n times. Use different counters for nested loops. Counter resets when branch passes.
Insert	16-bit word (w)	[OR 1-16]	Inserts control word w into timing frame.

Each sequence engine has a cache holding up to 2048 instructions, and each instruction has a channel access representation. Multiple distinct instruction sets can be contained within a cache. Sequence programming is thus achieved by a series of channel access transactions writing an instruction set into the cache. An instruction set is consequently started by a single channel access transaction. Furthermore, multiple engines can be started simultaneously by a single transaction. Beam generation sequences are similarly executed (with the Insert instruction replaced by a beam request instruction), allowing beam generation and control sequences to be synchronized.

Local Sequence Generation

Some sequence groups can be designated as local sequences. In this case, the timing frame can be remastered by dedicated hardware (the DAQ Timing Master, for example) in the hutch to

overwrite these sequence groups with sequences that are only distributed within the hutch. Thus, each hutch can define its own use for these sequence groups. This reduces the need for a larger number of control sequence groups.

Appendix A: LCLS-I Timing Information

For each fiducial event, the timing fiber for LCLS-I transmits:

- A series of 8-bit event codes that should be active at this fiducial time.
- A 52-byte data buffer containing timing and machine information.

Relative to the arrival of the fiducial event code (1), each event code has a fixed time offset. With the exception of the AC rate codes, the largest delay occurs 12116 119MHz ticks after the fiducial event.

The 52-byte pattern contains the following information:

Field	Offset (bytes)	Size (bytes)	Description
Message Type	0	2	Constant, PATTERN=1.
Message Version	2	2	Constant, PATTERN VERSION=1.
Modifier Fields: Modifiers 1-5 are sent 2 fiducials early, Modifier 6 is current.			
Modifier 1	4	4	bits 7:0 - YY bits 12:8 - k bit 14 - MPG IPLing bit 15 - Modulo 720
Modifier 2	8	4	bits 5:0 - AC timeslot mask (LSB=TS1) bit 6 - EVG burst bit 15 - BCS fault bit 20 - No gun perm bit 28 - Kicker LI25 bit 29 - Kicker LTU bit 30 - TCAV3 perm
Modifier 3	12	4	bit 7 - Test burst bit 19 - Pockcel perm (BEAM!) bit 28 - TCAV0 perm
Modifier 4	16	4	bits 31:29 - AC timeslot (0 to 5)
Modifier 5	20	4	bits 19:0 - BSA Active, add this sample to the selected acquisitions. bits 24:20 - AC rates (30, 10, 5, 1, 0.5 Hz, from LSB to MSB).

			bit 27:25 - User bits (bit 25 - profile diagnostic). bit 28 - Beam.
Modifier 6	24	4	MPS Word. bit 0: MPS valid bits 19-1: MPS destination mask bits 31-20: MPS permission mask
Timestamp Fields: sent 2 fiducials early.			
Timestamp seconds	28	4	Time since 1990 epoch.
Timestamp nanoseconds	32	4	Time since 1990 epoch. 17-bit fiducial ID is overlaid into low 17 bits.
BSA Fields			
BSA Avg Done	36	4	bits 19:0 - Append summed entry to buffers and push to PVs.
BSA Minor Mask	40	4	bits 19:0 - Append summed entry even if minor alarm. (Used if BSA Init is active.)
BSA Major Mask	44	4	bits 19:0 - Append summed entry even if major alarm. (Used if BSA Init is active.)
BSA Init	48	4	bits 19:0 - Initialize BSA buffers.

Appendix B: LCLS-II Timing Information

The timing fiber for LCLS-II sends 912 bit (114 byte) frames at a repetition rate of 929 kHz. The frames contain:

- Unique identifiers for each beam pulse and the associated data collected;
- Rate markers for simple synchronization of control and acquisitions;
- Beam presence and destination;
- Meta-data concerning the desired beam charge and energy along the linac (MPS input);
- MPS status;
- BSA commands, and
- Sequenced control words for complex synchronization of control and acquisitions.

The frame information is formatted as follows:

Field	Offset (bits)	Size (bits)	Description
Version	0	16	Frame Format Version
Time and Destination Fields			
Pulse ID	16	64	Unique, monotonic, automatically increments at base rate.
Timestamp nanoseconds	80	32	Time since 1990 epoch.
Timestamp seconds	112	32	Time since 1990 epoch.
Fixed Rates	144	10	Fixed rate markers, 186Mhz divided by one of 1, 13, 91, 910, 9100, 91000, and 910000, LSB to MSB.
AC Rates	154	6	Powerline markers: 60, 30, 10, 5, 1, 0.5 Hz, LSB to MSB.
Timeslot	160	3	360 Hz timeslot: 1-6.
Timeslot Phase	163	12	Base clocks since timeslot change.
Resync	175	1	71 kHz resync marker.
Beam Request	176	1	Beam is requested.
Destination	180	4	One of 16 beam destinations.
Charge	192	16	Requested bunch charge (pC)

Energy @ L0	208	16	Requested bunch energy at L0 (MeV)
Energy @ L1	224	16	Requested bunch energy at L1 (MeV)
Energy @ L2	240	16	Requested bunch energy at L2 (MeV)
Energy @ L3	256	16	Requested bunch energy at L3 (MeV)
HXR Wavelength	272	16	Requested HXR photon wavelength (pm)
SXR Wavelength	288	16	Requested SXR photon wavelength (pm)
Status and Limit Fields			
Gen Status	317	1	Generator status.
MPS Valid	318	1	MPS input link status.
BCS Fault	319	1	BCS fault status.
MPS Limiting	320	16	MPS limiting to destination - one bit for each possible beam destination indicating if MPS is limiting rate to that destination.
MPS Limits	336	64	MPS power limits to each of 16 destinations; 4 bits per destination indicating the max allowed “power class”. Parameter values for the power class calculations are provided by MPS. The power calculation is defined to be max of beam charge times number of bunches within a given time window.
Beam Synchronous Acquisition Fields			
BSA Init	400	64	Initialize BSA buffers. When BSA Init is asserted, BSA Active and BSA AvgDone (LSB to MSB) make a two bit alarm mask (0=None, 1=Minor, 2=Major, 3=Invalid) indicating the maximum allowable alarm level on a measurement in the acquisition to follow.
BSA Active	464	64	Add this sample to the selected acquisitions.
BSA AvgDone	528	64	Append summed entry to buffers.

BSA Update	592	64	Push buffer updates to software.
Sequence Control Fields			
Seq Control[0:17]	656	288	Set of 18 16-bit sequences.

Appendix C: XTPG Timing Information

The XTPG is a module that remasters the timing stream taking either LCLS-I timing input or LCLS-II timing input and forwarding timing information in the LCLS-II format and frequency with an additional segment. The additional segment is ignored by timing receivers that are not configured to parse it. In some locations in the LCLS-II timing distribution, TPR modules will receive the XTPG output on the LCLS-II timing input port. The XTPG encoding of LCLS-I timing input into the LCLS-II output format is described below (from *github:slaclab/l2si/firmware/common/xpm/rtl/XpmStreamFromCu.vhd*). Note that the LCLS-I information is only present in the 929kHz XTPG frames that coincide with the 360Hz fiducial with the exception of the timestamp, which is constant for all frames between 360Hz fiducials.

Field	Offset (bits)	Size (bits)	Source
Version	0	16	LCLS-II Frame Format Version
Time and Destination Fields			
Pulse ID	16	64	(1<<63) | XTPG internal frame counter
Timestamp nanoseconds	80	32	LCLS-I time since 1990 epoch.
Timestamp seconds	112	32	LCLS-I time since 1990 epoch.
Fixed Rates	144	10	XTPG internally generated fixed rate markers.
AC Rates	154	6	LCLS-I AC rates.
Timeslot	160	3	LCLS-I Timeslot.
Timeslot Phase	163	12	Base clocks since timeslot change.
Resync	175	1	71 kHz resync marker.
Beam Request	176	1	XTPG designated LCLS-I Eventcode.
Destination	180	4	Always zero.
Charge	192	16	Always zero.
Energy @ L0	208	16	Always zero.
Energy @ L1	224	16	Always zero.
Energy @ L2	240	16	Always zero.

Energy @ L3	256	16	Always zero.
HXR Wavelength	272	16	Always zero.
SXR Wavelength	288	16	Always zero.
Status and Limit Fields			
Gen Status	317	1	Always zero.
MPS Valid	318	1	Always zero.
BCS Fault	319	1	Always zero.
MPS Limiting	320	16	Always zero.
MPS Limits	336	64	All bits set.
Beam Synchronous Acquisition Fields			
BSA Init	400	64	LCLS-I edefInit. Bits 63:20 are zero.
BSA Active	464	64	LCLS-I ((edefMinor&edefInit) | (dmod(147:128)&~edefInit)). Bits 63:20 are zero.
BSA AvgDone	528	64	LCLS-I ((edefMajor&edefInit) | (edefAvgDn&~edefInit)). Bits 63:20 are zero.
BSA Update	592	64	Bits 63:20 are zero. Bits 19:10 are LCLS-I edefMajor(29:20). Bits 9:0 are LCLS-I edefMinor(29:20).
Sequence Control Fields			
Seq Control[0:17]	656	288	LCLS-I eventcodes. Bits 287:256 are zero.

Appendix D: TPR Register Map

The TPR has several regions of interest:

- 0x10000 - Version, Firmware version identification
- 0x20000 - Boot PROM reprogramming
- 0x30000 - XAdc
- 0x40000 - Transceiver crossbar
- 0x50000 - LED control
- 0x60000 - Interrupt control
- 0x60400 - DmaControl, registers to set the DMA buffers.
- 0x80000 - TprBase, the channel event select and readout controls.
- 0xC0000 - TprCore, receive link diagnostic registers.

Address	Register Name	Description
0x10000	fwVersion	32-bit version identifier
0x10004	scratchPad	Test register for RW accesses
0x10008	upTimeCnt	Seconds since FPGA reset
0x10100	haltReload	reserved
0x10104	fpgaReload	Start FPGA reload from flash
0x10108	fpgaReloadAddr	Flash address
0x1010c	masterReset	Hard reset
0x10300	fdValue	unused
0x10400	userValues	unused
0x10500	DEVICE_ID_G	unused
0x10600-0x10610	BUILD_INFO_C	Git hash of firmware repository at build time
0x10700	dnaValue	FPGA serial id
0x10800-0x108fc	BUILD_STRING	Character array of firmware build date and user

Address	Register Name	Description
0x20000	inputBus	b[31:16] command opcode, b[15:0] data
0x20004	addr	b[31] RNW, b[30:0] flash address
0x20008	outputBus	b[15:0] data
0x2000c	test	reserved
0x20010	addrBus	b[30:0] flash address
0x20014	fastData	b[15:0] fast data
0x20018	bufRead	b[30:0] flash address

The transceiver crossbar registers (Out[2] and Out[3]) set the source of timing for the card - either external or internally generated timing.

Address	Register Name	Description
0x40000	Out[0]	Set source for LCLS timing fiber Tx { 0=LCLS timing Rx, 1=LCLS-II timing input, 2=TPR output }
0x40004	Out[1]	Set source for LCLS-II timing fiber Tx {0=LCLS timing input, 1=LCLS-II timing input, 2=TPR output }
0x40008	Out[2]	Set timing input for TPR LCLS Rx { 0=LCLS, 1=LCLS-II, 2=Near end loopback }
0x4000C	Out[3]	Set timing input for TPR LCLS-II Rx { 0=LCLS, 1=LCLS-II, 3=Near end loopback }

Interrupt control is provided for the kernel module to only receive interrupts when an outstanding message is pending.

Address	Register Name	Description
0x60000	IrqEnable	bit 0 - Interrupt Enable
0x60004	IrqStatus	bit 0 - Interrupt Pending
0x60008	PhyAddr	Location in timing distribution tree
0x6000C	Reserved	

0x60010	CountReset	bit 0 - Clear counters of raw and selected events/frames.
0x60014	Reserved	
0x60018	DmaFullThreshold	bit 23:0 - Free DMA buffer threshold for asserting FULL status.

The DMA engine registers are listed below. Physical addresses are pushed into a queue that holds at most 1023 entries. An interrupt request is generated when the number of queued entries falls below a programmable threshold.

Address	Register Name	Description
0x60400	Queue Push	Queue DMA buffer address (32-bit physical)
0x60500	Queue Control	Bit 31 DMA enable Bit 23:0 Max DMA size (bytes)
0x60504	Irq Threshold	Generate interrupt when # of queue entries falls below threshold.
0x60508	DMA Count	Number of completed DMAs
0x6050C	Last Address	Bit 31:2 Last queued address Bit 0 Queued address repeated

For software purposes, the most important registers are those in TprBase. These are described below. Features marked “L1” are LCLS-I only, and “L2” are LCLS-II only.

Address	Register Name	Description
Channel configurations (N=0-13)		
0x80000+0x1000*N	ChN Control	bit 0 - enable, bit 1 - enable BSA DMA, bit 2 - enable event DMA
0x80004+0x1000*N	ChN EventSelect	bits 12:0 - rate select bits 12:11 - selection type (fixed=0, AC=1, seq=2, reserved=3) fixed (L2): bits 3:0 indicate marker: 0: 929kHz / 1 1: 929kHz / 13 2: 929kHz / 91 3: 929kHz / 910 4: 929kHz / 9100

		5: 929kHz / 91000 6: 929kHz / 910000 AC: bits 2:0 indicate marker: 0: 60 Hz 1: 30 Hz 2: 10 Hz 3: 5 Hz 4: 1 Hz 5: 0.5 Hz bits 8:3 - timeslot mask seq: bits 10:0 indicate sequence bit. bits 31:13 - destination select bit 31 - reserved bits 30:29 - inclusive=0,exclusive=1, dont_care=2, reserved=3 bits 28:13 - mask of destinations (L1: 0=D10Dump, 1=LI25, 2=UND; L2: 0=BSYDUMP, 1=DIAG0, 2=SXU, 3=HXU. Further note that for LCLS-I, the existing event code 140 is the inclusive mask of all three destinations, even though beam is only delivered to the hutch if the destination is UND.)
0x80008+0x1000*N	EventCount	Count of frames matching selection criteria
0x8000C+0x1000*N	BSADelay	bits 19:0 - Delay until beginning of BSA sensitivity window in 929kHz/360Hz pulses. bits 26:20 - Negative delay (L2).
0x80010+0x1000*N	BSAWidth	bits 19:0 - Width of BSA sensitivity window in 929kHz/360Hz pulses
0x8E008	FrameCount	Raw frame count
Trigger Control (N=0-11)		
0xA0000+0x1000*N	Control	bit 31 - enable bit 16 - polarity (0=pos, 1=neg) bit 3:0 - input channel
0xA0004+0x1000*N	Delay	bit 27:0 - trigger delay in ticks (L1@119 MHz, L2@185.7 MHz.)
0xA0008+0x1000*N	Width	bit 27:0 - trigger width in ticks (L1@119 MHz, L2@185.7 MHz.)

0xA000C+0x1000*N	FineDelay	bit 5:0 - Fine delay. If v<31, v*82.24ps, if 32<v<64, (v-1)*82.24ps, otherwise 31*82.24ps (L2).

The receiver core statistics from the transceiver module are described below.

Address	Register Name	Description
0xC0000	SOFcnt	Start of Frame counts
0xC0004	EOFcnt	End of Frame counts
0xC0008	FiducialCnt	Valid frame counts
0xC000C	CRCErrCnt	CRC error counts
0xC0010	RecClkCnt	Recover clock counter (div 16)
0xC0014	ResetDoneCnt	Count of reset completions
0xC0018	DecErrCnt	8b/10b decode error counts
0xC001C	DspErrCnt	8b/10b disparity error counts
0xC0020	CSR	B0 - counter reset B1 - Rx link is up B2 - Rx polarity setting B3 - Rx reset control B4 - Clock select (LCLS=0, LCLS-II=1) B5 - Rx link went down (latched, clear on write) B6 - Rx buffer bypass reset control B7 - Rx PLL reset B24 - Rx data buffer no delay (set to zero)
0xC0024	msgDelay	Frame delay control in recovered clocks
0xC0028	TxClkCnt	Tx reference clock counts (div 16)
0xC002C	BBCnt	B[31:16] Rx buffer bypass error counts B[15:0] Rx buffer bypass done counts

Appendix E: Trigger Pipeline Behavior

The TPR supports several channels of TTL output each operating at up to 1 MHz. Each channel has a pipeline allowing up to 128 triggers to be queued for the output before their delay setting has expired. This design was chosen to match the fact that the LCLS-II beam lags the timing frame information by approximately 100 microseconds. So, delays can be used which exceed the time between consecutive triggers; the triggers simply enter the pipeline and march towards the TTL driver.

A consequence of allowing delays to exceed the inter-trigger spacing is that a reduction in the trigger delay setting can lead to unexpected behavior. Consider the case of triggers being generated at 10kHz and with delay of 1 ms. This would place 10 triggers in the pipeline at any given time. If the delay setting were suddenly changed to 0.5 ms, then new triggers would be targeted to occur before the last 5 triggers with the old delay setting had made their way out of the pipeline. How should this case be handled? Triggers must be dropped - either the ones already in the pipeline or the ones that precede triggers already in the pipeline. The current firmware implementation does not handle this case explicitly, and defaults to withholding triggers until the pipeline is empty.