LASER INTERFEROMETER GRAVITATIONAL WAVE OBSERVATORY

LIGO Laboratory / LIGO Scientific Collaboration

LIGO-T2000406-v6 Advanced LIGO 7 July 2023

Software Interface to the PCIe Timing Board

Marc Pirello, Daniel Sigg

Distribution of this document:
LIGO Scientific Collaboration

This is an internal working note
of the LIGO Laboratory.

California Institute of Technology
LIGO Project – MS 18-34
1200 E. California Blvd.
Pasadena, CA 91125
Phone (626) 395-2129
Fax (626) 304-9834
E-mail: info@ligo.caltech.edu

Massachusetts Institute of Technology
LIGO Project – NW22-295
185 Albany St
Cambridge, MA 02139
Phone (617) 253-4824
Fax (617) 253-7014
E-mail: info@ligo.mit.edu

LIGO Hanford Observatory
P.O. Box 1970
Richland WA 99352
Phone 509-372-8106
Fax 509-372-8137

LIGO Livingston Observatory
P.O. Box 940
Livingston, LA 70754
Phone 225-686-3100
Fax 225-686-7189

http://www.ligo.caltech.edu/

1 Overview

The PCIe timing interface board is a PCI Express board that interfaces the computer software through a memory mapped interface. As part of the PCI bus enumeration the operating system assign the board a unique memory region in the physical address space. This document describes the register assignment within this region.

The board either uses an 8 kB address region divided into 2 blocks of 4 kB (earlier versions), or a 16 kB address region with the last 8 kB used to reprogram the flash (later versions). The first block is used for the control registers, whereas the second block is used to read the status of the LIGO timing interface.

Address	Description	Size
0x0000	Control and monitor registers for on-board features, backplane and expansion modules	4096
0x1000	Timing Diagnostics Information	4096
0x2000	Program SPI flash memory	8192

Table 1: Memory map of PCIe timing interface.

The PCIe timing board has the following features:

- Part of the Advanced LIGO optical timing distribution system which is based on distributed synchronized 2^{26} Hz crystal oscillators driven by a master GPS clock.
- Interfaces the IO interface backplane and provides clocking signals for converters.
- Supports an optional master clock expansion module with GPS capabilities
- Supports an optional fanout expansion module with 12 fiber downlinks.

2 Applicable Documents

The applicable documents can be located through the following dcc pages:

- E2000328: PCIe Timing Interface (DCC top node)
- E2000337: PCIe Timing Interface Software Release
- D2000329: PCIe Timing Interface Board
- D2000297: IO Interface Backplane (LVDS)
- D2000301: Master Clock Expansion Option (GPS)
- D2000302: Fanout Expansion Option
- E0900036: Diagnostics Information for the Advanced LIGO Optical Timing Distribution

3 Control Registers

Registers are organized in double words (32 bit). It is envisioned that they are read and written as double words through the PCI Express bus. Reading from unassigned memory addresses within the assign memory region results in zeros. Similarly, writing to an empty address or a read-only address is ignored but treated as successful.

This PCI board supports 4 message-signaled interrupts (MSI). Interrupts can be globally enabled and disabled with the interrupt control register. However, individual interrupts must be setup through the interrupt configuration registers located at 0x00C0 to 0x00FC. Typically, the global interrupt register is controlled by the device driver.

3.1 Current Time

There are two registers to read the current time. The first register needs to be read first. It contains the fractional seconds of the GPS time and it will latch the second register which contains the GPS seconds. It is recommended to read the full time as an atomic 64-bit value.

Address	Description	RW	Size
0x0000	Fractional second in units of 2^{-32} s Reading this register will latch the GPS seconds.	R	4
0x0004	GPS seconds Reads the GPS seconds associated with the last reading of the fractional second register. Writing has no effect.	R	4

Table 2: Registers to read the current time.

3.2 Node Address

The node address is accessible in the register at address 0x0134. Generally, the first nibble (hex digit) is the nesting level, and each following nibble represents a leaf in the timing distribution system with the master always set to 0x0. The first fanout port of a master gets assigned 0x1000000, the second port is 0x11000000, and so forth. For the node address to be valid the board must be synchronized. This can be check with the OK bit in the global status register.

3.3 Board and Firmware ID

The board ID and revision can be read from register 0x0140, whereas register 0x0144 contains the software ID and revision.

The board ID is an 8-digit BCD number, where the first 7 digits represent the dcc number (D) of the converter PCB board, whereas the last digit represents the revision number. Similarly, the software ID is an 8-digit BCD number, where the first 7 digits represent the dcc number (E) of the converter FPGA code with the last digit representing the revision number. For the timing board the board ID is 0x20003291 linking to D2000329-v1, whereas the firmware ID currently is 0x20003372 linking to D2000337.

3.4 Status and Interrupt Control

This register is shared between the global status flags and interrupt control. Status flags are read only. The OK flag can be used to check that the timing system is running normally. However, one should also check that the reported GPS seconds are reasonable, i.e., GPS seconds > 1e9.

Address	Description	RW	Size
0x0008	Bit 31: OK flag (timing system is locked) Bit 30: This is a root node Bit 29: Supports fanout ports Bit 28: Uplink is up and working Bit 27: Loss-of-signal on uplink Bit 26: OCXO is locked Bit 25: GPS is locked Bit 24: VCXO control voltage out-of-range Bit 23: UTC time mode enabled Bit 22: Leap seconds decoded Bit 21: Subtraction of one leap second pending Bit 20: Addition of one leap second pending Bits 19...16: Reserved Bits 15..8: Leap seconds Bits 7…4: Reserved for interrupt control Bit 3…0: Enable for MSI interrupts 3 down to 0	R RW	4
0x000C	Firmware revision of VHDL code	R	4

Table 3: Status and interrupt control register and firmware revision.

3.5 Firmware Release

The firmware release version is accessible at address 0x000C.

3.6 Converter Backplane Interface

The PCIe timing board interfaces a 10-slot backplane which supports interface cards for the digital converters LIGO is using. Each slot has its own dedicated clock line that can be configured individually. For each clock line the frequency and phase shift can be specified. One can also ask that the clock starts at the next 1 second boundary.

There are 3 options for providing clock signals to the 10 slots:

1. Old type TTL ADC/DAC clock (1 per slot)
2. Fast LVDS clock (1 per slot)
3. Fast LVDS clock and sync lines (shared on neighboring odd/even slots)

Additionally, there a few bits of binary IO that can be controlled for each slot. This is typically used to add the analog DuoTone signal to slot 1, and to route another DuoTone signal from slot 1 to 2. However, these bits are available on all slots, and can be used for other purposes.

3.6.1 Common Functions

The table below shows the common functions for the converter backplane interface. Depending on bit 0 a write to register 0x0018 will either reset the watchdog timer or initialize the SPI devices.

Address	Description	RW	Size
0x0010	Backplane configuration Bit 4: Start clock at next transition from idle value (OR with slots) Bit 3: Start on next second boundary for all slots (OR with slots) Bit 2: Global enable for all slots (AND with sots) Bit 1: Reset watchdog timer every time the current time is read Bit 0: Disable DuoTone output	RW	4
0x0014	Watchdog reset or SPI initialization Bit 0: Init SPI bus when 1, Reset watchdog when 0	W	4
0x0018	Backplane status Bits 31…10: Reserved Bit 9: Backplane is present Bit 8: X5 input Bit 7: X3 input Bit 6: X1 input Bit 5: Temperature alarm Bits 4…3: Backplane revision Bit 2: Watchdog monitor Bit 1: Indicates that all clocks are running Bit 0: Indicates that all clocks are active	R	4
0x001C	Reserved		4

Table 4: Common registers for the backplane interface.

The analog DuoTone signal is only routed to slots 1 and 2.

3.6.2 Slot Functions

The table below lists the configuration and monitor registers for each slot of the back plane. There are 10 slots, with 2 10 configuration registers and 1 monitor register each. Slot 1 starts at address 0x0020, slot 2 at 0x0030 and so on. All registers are initialized to zero at the beginning and no clock signals are sent to the backplane. Timing diagrams can be found in Section 5.

Address	Description	RW	Size
0x0020	Slot 1 (odd) configuration Bits 31…23: Reserved Bit 22: Bit 2 output high Bit 21: Bit 2 is binary output (shared on neighboring odd/even slots) Bit 20: Bit 1 output high Bit 19: Bit 1 is binary output Bit 18: Enable DAC DuoTone on 2nd to last ADC channel Bit 17: Enable Duotone on last ADC channel Bit 16: Use LVDS clock lines Bits 15…12: Reserved Bit 13: Use timing signal as clock (Bits 12…0 except 9 are ignored) Bit 12: Pull idle clock line high Bit 11: Start clock at next transition from idle value Bit 10: Start clock at next second boundary after a 0.25 s countdown Bit 9: Inverted clock signal Bit 8: Enable clock signal Bits 7…0: Log2 frequency, 2'complements, 2^{26} Hz is the highest allowed frequency, 2^{-8} Hz is the lowest frequency	RW	4
0x0024	Slot 1 (odd) phase shift (fixed point, decimal point prior to Bit 31) Bit 31…0: Phase shift in fractions of the clock period.	RW	4
0x0028	Slot 1 (odd) status Bits 31…23: Reserved Bit 22: Monitor of bit 2 (shared on neighboring even slot) Bit 21: Undefined Bit 20: Monitor of bit 1 Bit 19: Monitor of DAC DuoTone select on last DAC channel Bit 18: Monitor of DAC DuoTone select on 2nd to last ADC channel Bit 17: Monitor of DAC DuoTone select on last ADC channel Bit 16: Undefined Bits 15…2: Reserved Bit 1: Indicates that clock is running Bit 0: Indicates that clock is active	R	4
0x002C	Reserved		4

0x0030	Slot 2 (even) configuration Bits 31…23: Reserved Bit 22…21: Ignored Bit 20: Bit 1 output high Bit 19: Bit 1 is binary output Bit 18: Select DAC DuoTone on 2nd to last ADC channel Bit 17: Select Duotone on last ADC channel Bit 16: Use LVDS clock lines Bits 15…12: Reserved Bit 13: Use timing signal as clock (Bits 12…0 except 9 are ignored) Bit 12: Pull idle clock line high Bit 11: Start clock at next transition from idle value Bit 10: Start clock at next second boundary after a 0.25 s countdown Bit 9: Inverted clock signal Bit 8: Enable clock signal Bits 7…0: Log2 frequency, 2’s complements, –8 to 26	RW	
0x0034	Slot 2 (even) phase shift (fixed point, decimal point prior to Bit 31) Bit 31…0: Phase shift in fractions of the clock period.	RW	4
	Slot 2 (even) status Bits 31…23: Reserved Bit 22...21: Undefined Bit 20: Monitor of bit 1 Bit 19: Monitor of DAC DuoTone select on last DAC channel Bit 18: Monitor of DAC DuoTone select on 2nd to last ADC channel Bit 17: Monitor of DuoTone select on last ADC channel Bit 16: Undefined Bits 15…2: Reserved Bit 1: Indicates that clock is running Bit 0: Indicates that clock is active	R	4
0x003C	Reserved		4
0x0040 … 0x00BC	Configuration and status registers for slots 2 to 10.		128

Table 5: Registers to setup clock signals for the ADCs and DACs.

An 8-bit value, $N \leq 26$, is used to select the frequency with the equation 2^N Hz. The period is then 2^{-N} s. To get the clock signal applied one must enable it for the slot and globally. If the clock is disabled, the clock output is set to the idle value. If the clock must start at a 1 second boundary, one can choose to delay the enable to the next 1 second boundary after a fixed minimum delay of 0.25 sec has passed. A global start on next second boundary can be used to start all clocks simultaneously on a second boundary. A logical AND is performed between the slot and the global enables and start bits. Also, the clock can be enabled immediately, or on the next transition away from the idle value. The two options can be combined: wait for the 1PPS boundary and then wait again for the next transition away from idle. If the clock is enabled immediately, it may result in a very short half cycle at the beginning; it could be as short as 2^{-27} sec which may be too short for the converters of the data acquisition system.

An optional phase shift can be applied and is specified in units of 2^{-32} sec. The least significant bits below the clock's resolution are ignored, as well as the most significant bits that would indicate a phase shift larger than a clock cycle. Additionally, there is a bit to invert the final output clock signal. This is equivalent to a 180º phase shift.

Alternatively, one can select the timing signal as the clock. This will allow boards that support it to synchronize to GPS independently (may take 1-3 sec). Once selected the timing signal will be on and phase, enable and start conditions are ignored. Typically, this clock is provided through LVDS.

There are two status readbacks for the clock output: active and running. The difference is that running will wait for the first transition away from idle, when this option is selected, whereas active will go high immediately after the optional wait for the 1PPS.

Several binary IO channels are available too. The binary IO channels associated with the DuoTone were previously only available on slots 9 and 10 but were physically the same signals. In this implementation, separate binary IO channels are routed to each slot. If they are not used for DuoTone, they can be used for other purposes—together with two additional bits. Bit 1 and 2 have configurable direction. Bit 1 is a per slot channel, whereas bit 2 is shared among two neighboring odd/even slots.

3.7 Interrupt Configuration

The board supports 4 message-based interrupts (MSI) over the PCI Express bus. These interrupts can be configured to be issued at regular intervals. The setup is the same as a converter clock with the rising edge of the configured clock signal being used to initiate the interrupt. PCI Express interrupts must be configured and enabled by the root complex during the initial setup process.

Address	Description	RW	Size
0x00C0	MSI 0 configuration Bit 12: Pull idle clock line high Bit 11: Start clock at next transition from idle value Bit 10: Start clock at next second boundary after a 0.25 s countdown Bit 9: Inverted clock signal Bit 8: Enable interrupt signal Bits 7…0: Log2 frequency, 2’s complements, –8 to 25	RW	4
0x00C4	MSI 0 phase shift Bit 31…0: Phase shift in units of 2^{-32} s.	RW	4
0x00C8	MSI 0 status Bit 1: Indicates that interrupts are issued at regular intervals Bit 0: Indicates that the interrupt is configured	R	4
0x00CC	Reserved		4
0x00D0 … 0x00FC	Configuration of interrupts MSI 1 through 3.		24

Table 6: Registers to configure interrupts over the PCI Express bus.

The interrupt processing is typically done by the device driver, which is loaded by the operating system. Interrupt configuration registers for the first MSI are located at 0x00C0. The second, third and fourth MSI are at 0x00D0, 0x00E0 and 0x00F0, respectively.

3.8 GPS Expansion Module

It is possible to connect an expansion module to the DB25 and SMA connectors of the PCIe timing board. This expansion module implements an OCXO and a GPS receiver, an RS422 port, an additional SFP, both a 1 PPS input and output, as well as an external frequency input. If both GPS and OCXO are present, the PCIe interface board will default to be a timing root node (master clock). The OCXO is connected to the SMA on the PCI board and its frequency must be 2^{25} Hz. For a timing root node, it serves as the main clock source for the entire timing distribution system. The GPS receiver provides a time stamp that relates to UTC through a 1 PPS signal and a serial interface. If the expansion board is not equipped with a GPS or an OCXO, the PCI board acts as a normal downstream timing interface.

Address	Description	RW	Size
0x0100	GPS expansion configuration Bits 31…13: Reserved Bit 13…10: Gain of XO locking loop, in units of 2^N, N from −8 to +7 Bit 9: Sign of XO locking loop Bit 8: Enable XO locking Bit 7: Use reference input for OCXO (only possible if not a root node) Bit 6: Output an IRIG-B (2nd channel) rather than an 1 PPS Bit 5: Disable binary output (1 PPS normally) Bit 4: Disable RS422 Bits 3…2: 1 PPS control (for a timing root node) b00: Use external 1 PPS, if present, GPS 1 PPS else b01: Always use the GPS 1 PPS b10 or b11: Always use the external 1 PPS Bits 1…0: Timing root node control b00: If both GPS and OCXO are present, default to root b01: Prevent this to be the timing root node b10 or b11: Set this to be the timing root node	RW	4
0x0104	Preset frequency Bits 31…0: Preset frequency in Hz for OCXO locking	RW	4
0x0108	GPS expansion status Bits 31…5: Reserved Bit 7: OCXO clock present Bit 6: Reference clock present Bit 5: GPS present Bit 4: GPS is locked Bit 3: OCXO is locked Bit 2: External frequency input is present Bit 1: External 1 PPS is used Bit 0: Root node flag	R	4
0x010C	Frequency counter Bits 31…0: External frequency in Hz	R	4

Table 7: Registers for the GPS expansion module.

The GPS expansion module implements an external frequency input. Normally, this input is just used as a frequency counter. However, it can also be used to lock the frequency of an external OCXO. The desired frequency must be set through the PCI. Since this OCXO will only be locked to the 1 PPS, its RF phase error can be relatively large.

The expansion module also has an external 1 PPS input which can be used to synchronize the master with an external frequency standard. A digital output is available which is normally configured as a 1 PPS output but can also output an IRIB-B signal instead.

An RS422 port writes the timing status information once a second and can be read by the slow controls system to provide diagnostics o the timing system. Since fanout modules collect the diagnostics information of their direct downstream partners, all diagnostics information is available by monitoring the fanout chassis.

The master clock expansion module also implements a fiber port. This fiber port can be used for either an uplink or a downlink. The logic is as follows: if the PCIe timing interface acts as a timing master, the fiber port is a fanout port. If this is a standard downstream timing interface, the fiber port is an uplink port. It can be used instead of the port mounted on the board. If both ports are connected, the on-board port has priority. However, the port on the expansion module can be forced to be a fanout port through software.

3.9 Fanout Expansion Module

The fanout expansion module requires a PCIe daughter board to provide the additional signal lines. It provides up to 16 fiber downlink ports and makes the PCIe timing interface a fanout timing module. The daughter board implements 4 fiber ports, and it can be used stand-alone in systems that do not need the full number of fanout ports.

One fiber port is implemented on the main timing interface board. Typically, this will be used for the uplink to a timing fanout. It is possible to add expansion boards which implement additional fiber ports. This can make the PCIe timing interface a master clock with fanout. There are 12 fiber ports located on the fanout expansion board. 4 fiber ports are located on the daughter board and one extra fiber port is located on the GPS expansion board. The GPS expansion board is connected through a DB25 to the main timing interface. The daughter board connects to the timing interface through the Samtec connectors and makes the PCIe board double width. The fanout expansion board connects to the daughter board through a SCSI-2 type high density cable.

The fiber ports on the expansion boards can be enabled or disabled by setting the corresponding bits in the configuration register. Furthermore, the last port on an expansion board can be reassigned to be the uplink. This is useful in situations where the fiber port on the timing interface is not easily accessible. If the uplink has been reassigned, the port on the timing interface can optionally be reused as a fanout port.

Since the numbering of the fanout ports can be confusing, a port identification option is available to enable double blinking of the LED on a selected fanout or uplink port.

The fanout expansion module also implements two BNC connectors that are used for the input and output of an IRIG-B signal. The IRIG-B signals are TTL based square wave signals and not analog sine waves. Alternatively, the IRIG-B input can also be reassigned for an additional 1 PPS input, whereas the IRIG-B output can be used to output a square wave at a fixed frequency of 2^N Hz.

Address	Description	RW	Size
0x0110	Fanout expansion configuration Bits 31…12: Reserved Bits 11…7: Port number (1-16) for identification (Use 0 for uplink) Bit 6: Enable port identification Bit 5: Reuse FPGA board SFP for fanout (bit 4..3 must be non-zero) Bit 4…3: Uplink SFP selection b00: Use SFP on main FPGA board b01: Use last SFP on the fanout expansion board for uplink b10: Use last SFP on the daughter board for uplink b11: Use SFP on the GPS expansion board for uplink Bit 2: Disable the SFP on the GPS expansion board Bit 1: Disable the 4 SFPs on the daughter board Bit 0: Disable the 12 SFPs on the fanout expansions board	RW	4
0x0114	IRIG-B configuration Bits 31…20: Leap seconds Bit 19: Delete rather than add a leap second Bit 18: Append a leap second (at end of next quarter) Bit 17: Daylight saving time Bit 16…11: Time zone (signed value in units of 0.5h) Bits 10…9: IRIG-B selection (2nd channel) b00: GPS based (no leap seconds) b01: UTC B10 or b11: Local time Bits 8…7: IRIG-B selection b00: GPS based (no leap seconds) b01: UTC B10 or b11: Local time Bit 6: Reuse IRIG-B input for an additional 1 PPS input Bit 5: Output a 2^N clock rather than an IRIG-B Bits 4…0: Log2 frequency, 2^{25} Hz is the highest allowed frequency	RW	4
0x0118	Fanout expansion status Bits 31…17: Reserved Bits 16…15: DIP switch position (SW12/SW11) Bits 14...10: Position of reused uplink port Bits 9…5: Number of enabled fanout ports Bits 4…0: Number of configured fanout ports	R	4
0x011C	Time delay between internal and the external PPS/IRIG-B signals, 2's complement in units of 2^{-32} s	R	4
0x0120	Time delay between internal and GPS PPS signals, 2's complement in units of 1 s	R	4
0x0124 to 0x012C	Decoded bits of the received IRIG-B signal 89 bits in little endian format, bits 89 to 95 are used for an error counter. These bits are reported on the next second after the IRIG-B signal has been decoded, so it is 2 seconds behind	R	12

Table 8: Registers for the fanout expansion module.

3.10 Advanced Timing Features

The advanced timing features describe the available features.

Address	Description	RW	Size
0x0130	Advanced timing configuration Bits 31…0: Reserved	RW	4
0x0134	Node address Writing the value 0xFEEDC0DE to this register will write the 1 kbyte located at 0x1C00 back to the EEPROM on the daughter board.	R W	4
0x0138	Advanced timing status Bit 31…27: Unused Bit 26..24: Timing link version Bit 23: Analog output for XO locking enabled Bit 22: BRAM option enabled for timing diagnostics Bit 21: PCIE option enabled Bit 20: IRIG-B option enabled Bit 19: RS422 option enabled Bit 18: PPS option enabled Bit 17: OCXO option enabled Bit 16: GPS option enabled Bit 15: EEPROM on daughterboard has been read Bits 14...3: Future expansion options Bit 2: Fanout expansion is present Bit 1: GPS expansion is present Bit 0: Daughter board is present	R	4
0x013C	Shift of DuoTone signal relative to 1PPS Bits 31…0: Signed shift in units of 2^{-26} sec clock cycles		4
0x0140	Board ID and revision	R	4
0x0144	Software ID and revision	R	4
0x0148	VCXO control voltage. Bits 15…0: unipolar from 0V to 3.3V	R	4

Table 9: Registers for advanced timing features.

The DuoTone shift is 0 for boards with older firmware, and 476 for newer firmware. The new firmware has the DuoTone aligned to 1 PPS, whereas for the older firmware the DuoTone is late by 7.09 µs. This means the actual zero crossing is at (476 – specified shift) * 2^{-26} sec.

3.11 On Board Features

Several on board diagnostics features are available. They are mostly related to the power supply.

Address	Description	RW	Size
0x0180	Board configuration: Bits 31…20: Reserved Bits 19…4: Divider (M) for the external synchronization frequency, default is 19. Bits 3…0: Exponent (N) for the external synchronization frequency, default is 0.	RW	4
0x0184	XADC configuration: Currently not used	RW	4
0x0188	Board and power supply status: Bits 31…16: DIP switches (switch off is ‘1’, on is ‘0’) Bits 15…9: Reserved Bits 8: Switching regulator applied interrupt Bits 7…2: Interrupt flags of switching supply Temperature, low input voltage, power good for supplies 4 to 1 Bit 1: Power good signal from Gigabit transceivers supplies Bit 0: Power good signal from switching supply (ADP5050)	R	4
0x018C	XADC status: Bits 31…6: Reserved Bit 5: XADC enabled Bit 4: VCCAUX alarm (>3% deviation) Bit 3: VCCINT alarm (>3% deviation) Bit 2: User temperature alarm (>75º) Bit 1: Over temperature alarm (>95º) Bit 0: Any alarm	R	4
0x0190	Temperature and internal power supply 1 Bits 31…16: VCCINT (gain 1/3, nominally 1.0V) Bits 15…0: Current internal chip temperature (T=503.975*V-273.15º)	R	4
0x0194	Internal power supply 2 Bits 31…16: VCCBRAM (gain 1/3, nominally 1.0V) Bits 15…0: VCCAUX (gain 1/3, nominally 1.8V)	R	4
0x0198	External power supply 1 Bits 31…16: Current of VCCINT (gain 1Ω) Bits 15…0: Current of 3.3V (gain 1Ω)	R	4
0x019C	External power supply 2 Bits 31…16: Current of 2.5V (gain 0.333Ω) Bits 15…0: Current of VCCAUX (gain 1Ω)	R	4

0x01A0	External power supply 3 Bits 31…16: VDD (gain 1/4, nominally 2.5) Bits 15…0: VREG (gain 1/6, nominally 5.1V)	R	4
0x01A4	External power supply 4 Bits 31…16: AVTT (gain 2/3, gigabit transceivers, nominally 1.2V) Bits 15…0: AVCC (gain 2/3, gigabit transceivers, nominally 1.0V)	R	4
0x01A8	External power supply 5 Bits 31…16: N5 (N5=6*V-6.25V, analog supply, nominally –5V) Bits 15…0: P5 (gain 1/6, analog supply, nominally +5V)	R	4
0x01AC	External power supply 6 Bits 31…16: N12 (N12=15*V-17.5V, analog supply, nominally –12V) Bits 15…0: VCC (gain 1/5, nominally +3.3V)	R	4
0x01B0	External power supply 7 Bits 31…16: P10 (gain 1/11, analog supply, nominally +10V) Bits 15…0: VADC (gain ½, XADC supply voltage, nominally 1.8V)	R	4
0x01B4	External power supply 8 Bits 31…16: Current of V12 (gain 1Ω) Bits 15…0: V12 (gain 1/15, main supply voltage, nominally +12V)	R	4

Table 10: Registers for board diagnostics.

The external synchronization frequency can be used to synchronize an external switching regulator and is accessible through a header connector. The board implements a divide-by-8 linear-feedback shift register to generate different clock edges that are spaces 45 degrees apart. Its input frequency is selected by two configuration values *M* and *N*. If *N* is nonzero, the input frequency is using the following equation: 2^{N+10} Hz, where *N* is a 4-bit number ranging from 1 to 15. This supports frequencies from ~2 kHz up to ~33 MHz before the divider. The divider is a linear feedback shift register that divides by 8 (nominal) or 4, depending how it is configured.

N=0 turns this feature off and enables the divider value *M*. If *M* is nonzero, the input frequency is determined by the following equation: $2^{26} / (M+1)$ Hz. If both *M* and *N* are zero, the frequency output is disabled. An odd *M* will result in a duty cycle that is slightly different from 50%. The default values of *N=0* and *M=19* result in an input frequency of 3.3554432 MHz, and 419.4304 kHz after the divider. With a sampling rate of 16384 Hz, this will alias to a frequency of 6.5536 kHz.

The ADC code is unipolar and consists of 16 bits. So, the ADC code should be divided by 65536 to get the input voltage of the ADC. Since most measured voltages are larger, they have a gain smaller than one. The gain is listed in the table and the ADC voltage needs to be divided by it. The temperature readout and negative voltages need an offset correction, which is also listed in the table above. All current readbacks but for the 12V supply need an offset correction of -0.05V.

4 Timing Diagnostics Information

The available timing information is described in detail in E0900036. The amount of available information depends whether we are configured as an end point in the timing distribution, or whether we are master or fanout module. The PCIe timing interface can support master and fanout functions through expansion modules. Every timing interface module sends some information back to the upstream fanout module through the uplink fiber port. Fanout modules collect this information and make it available through a RS422 port, which in turn is read by the slow controls system.

The same is true for the PCIe timing interface with the addition that this information is mapped into the timing diagnostics block and available through the PCIe interface. The PCIe interface implements version of the protocol which allocates an additional 64 bytes of data before the information form the downstream ports.

The memory layout is as follows:

Address	Description	Size
0x1000	Status & Configuration Information	64
0x1040	Records from up to 16 downlink records	128
0x10C0	GPS status	32
0x10E0	CRC error counters from fanout ports	16
0x10F0	Reserved	16
0x1100(v2)	Additional information from the PCIe board (version 2 only)	64

0x1100(v1)
0x1140(v2)
	Information sent back from the downlink nodes, each has
	32 bytes of status information
	96 bytes of node specific information
		2048

0x1900(v1)
0x1940(v2)
	CRC	4

0x1904(v1)
0x1944(v2)
	Reserved	764(v1)
		700(v2)

0x1C00	Data from EEPROM on daughter board (if present)	1024

Table 11: Memory map of timing diagnostics information.

We explicitly list the first 16 double words and refer to E0900036 for any additional information.

Address	Description	Size
0x1000	Board ID and revision First 7 digits are the dcc number (D), last digit is the revision. Set to 0x2000329R with R the revision	4
0x1004	Board serial number. Currently used to indicate if this is a root or fanout node (0x0000) or a normal timing interface (0x0001)	4
0x1008	Software ID and revision First 7 digits are the dcc number (E), last digit is the revision. Set to 0x2000337R with R the revision R >= 2: Duotone is aligned with 1PPS	4
0x100C	Software revision number This is the subversion revision of the loaded FPGA program	4
0x1010	GPS time in seconds	4
0x1014	Module address. The first nibble (hex digit) is the nesting level and each following nibble represents a leaf in the timing distribution system with the master always set to 0x0. The first fanout port of the master gets assigned 0x1000000, the second port is 0x11000000, etc.	4
0x1018	Status Bits 31…16: Control voltage applied to the VCXO Bits 15…8: Status of the DIP switches Bits 7…6: Status of SW10 and SW9 Bit 5: Loss of signal (LOS) at the uplink port Bits 4…1: Number of consecutive seconds with missing 1PPS Bit 0: Uplink is up and running normally	4
0x101C	Configuration (master and fanout) Bits 31…17: Unused Bit 16: External OCXO present (any frequency) Bit 15: Reference OCXO is present (2^25 Hz) Bit 14: External 1PPS present Bit 13: A GPS module is present Bit 12: GPS is locked Bit 11: OCXO is locked Bit 10: External 1PPS is used for timing reference Bit 9: GPS 1PPS is used for timing reference Bit 8: Uplink 1PPS is used for timing reference Bit 7: Uplink up and running Bit 6: Uplink loss of signal Bits 5…2: Number of fanout ports (0 represents 16 ports) Bit 1: This is a fanout module Bit 0: This is a master module	4

0x1020	OCXO Controls Bits 31…16: Unused Bits 15…0: OCXO control voltage	4
0x1024	OCXO Error Time difference between the internal 1PPS obtained from the OCXO clock signal, and the external 1PPS, 2's complement in units of 2^{-32} s	4
0x1028	Uplink 1PPS delay Time difference between the uplink 1PPS input and the internal 1PPS	4
0x102C	Ext 1PPS delay Time difference between the external 1PPS and the internal 1PPS.	4
0x1030	GPS 1PPS delay Time difference between the GPS 1PPS and the internal 1PPS.	4
0x1034	Fanout Up/LOS Bits 31…16: Fanout ports are up (LSB is port 0) Bits 15…0: Loss of signal at fanout port	4
0x1038	Fanout Missing return 1PPS and delay error Bits 31…16: Missing 1PPS in downlink return Bits 15…0: Fiber delay is out of range	4
0x103C	Leap Seconds & Error Count Bit 31: UTC time mode enabled Bit 30: Leap seconds decoded Bit 29: Subtraction of one leap second pending Bit 28: Addition of one leap second pending Bits 27...24: Reserved Bits 23…16: Leap seconds Bits 15…12: Unused Bit 11: GPS error (if master) Bit 10…8: GPS error count (if master) Bits: 7…0: CRC error count	4

Table 12: Memory map of the status information from the basic timing diagnostics.

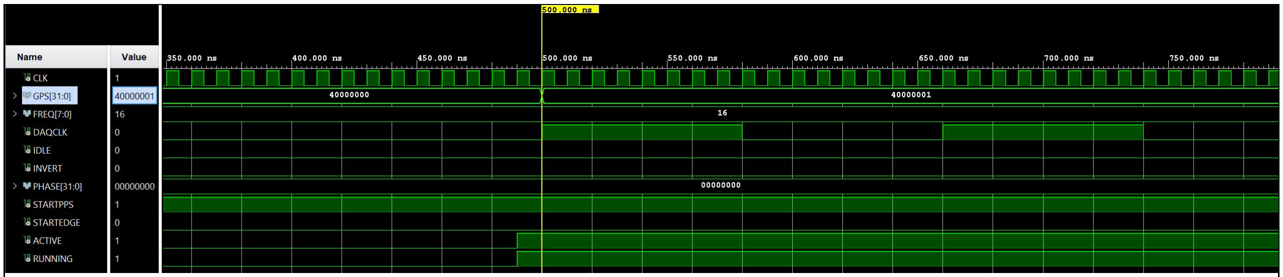
An extended set of timing information comprising of another 16 double is sent back through the fiber. For version 2 diagnostics information this is mapped starting at address 0x1100.

Address	Description	Size
0x1100	Board configuration from 0x0180.	4
0x1104	Board and power supply status from 0x0188.	4
0x1108	XADC status from 0x018C.	4
0x110C	12V current & internal chip temperature from 0x01B4 MSW and 0x0190 LSW.	4
0x1110	Backplane configuration from 0x0010.	4
0x1114	Backplane status from 0x0018.	4
0x1118	Not used.	4
0x111C	Not used.	4
0x1120	GPS expansion configuration from 0x0100.	4
0x1124	GPS expansion status from 0x0108.	4
0x1128	Fanout expansion configuration from 0x0110.	4
0x112C	Fanout expansion status from 0x0118.	4
0x1130	Advanced timing status from 0x0138.	4
0x1134	Advanced timing status from 0x013C	4
0x1138	Not used.	4
0x113C	CRC in fiber uplink, otherwise not used	4

Table 13: Memory map of the status information from the extended timing diagnostics.

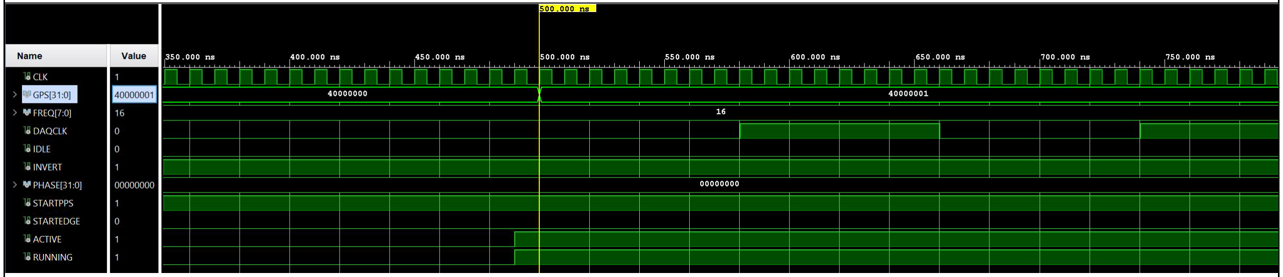
5 Timing Diagrams

The following timing diagrams have been generated with different configurations of the inverted and idle bits to produce clocks with rising or falling edges at the 1 PPS or half a cycle later. If a small phase shift is implemented as seen in diagrams 5 and 6, one must be careful to avoid a false transition at the 1 PPS. The "start clock at next transition from idle value" bit can be used to suppress it. All timing diagrams set the "start clock at next second boundary" bit.



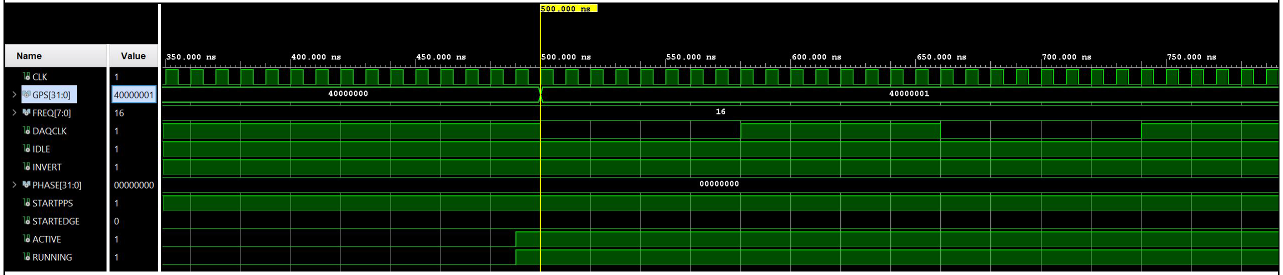
Name	Value
CLK	1
GPS[31:0]	40000001
FREQ[7:0]	16
DAQCLK	0
IDLE	0
INVERT	0
PHASE[31:0]	00000000
STARTPPS	1
STARTEDGE	0
ACTIVE	1
RUNNING	1

Rising edge on 1 PPS



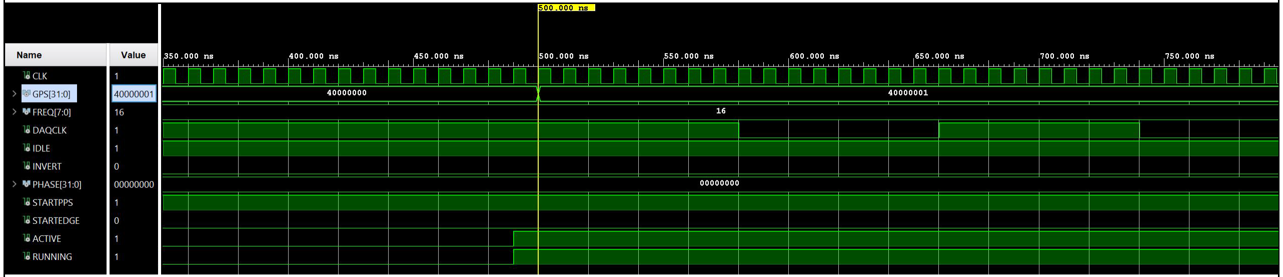
Name	Value
CLK	1
GPS[31:0]	40000001
FREQ[7:0]	16
DAQCLK	0
IDLE	0
INVERT	1
PHASE[31:0]	00000000
STARTPPS	1
STARTEDGE	0
ACTIVE	1
RUNNING	1

Rising edge on half cycle after 1 PPS



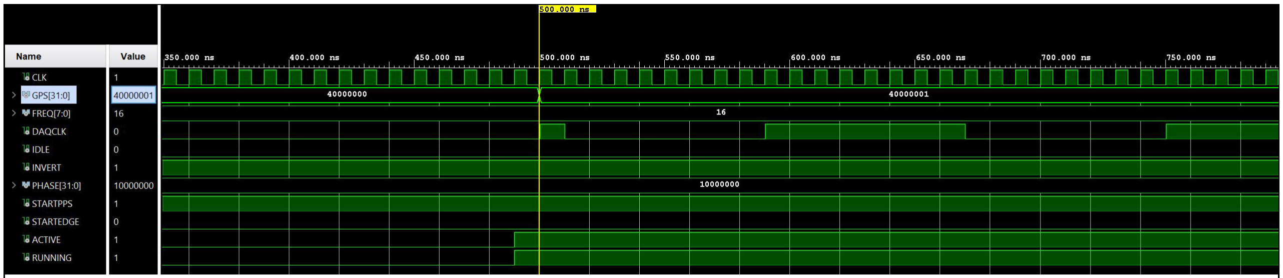
Name	Value
CLK	1
GPS[31:0]	40000001
FREQ[7:0]	16
DAQCLK	1
IDLE	1
INVERT	1
PHASE[31:0]	00000000
STARTPPS	1
STARTEDGE	0
ACTIVE	1
RUNNING	1

Falling edge on 1 PPS



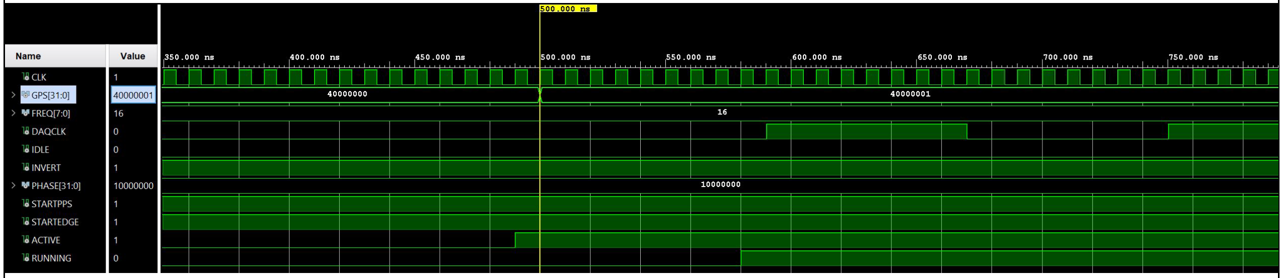
Name	Value
CLK	1
GPS[31:0]	40000001
FREQ[7:0]	16
DAQCLK	1
IDLE	1
INVERT	0
PHASE[31:0]	00000000
STARTPPS	1
STARTEDGE	0
ACTIVE	1
RUNNING	1

Falling edge on half cycle after 1 PPS



Name	Value
CLK	1
GPS[31:0]	40000001
FREQ[7:0]	16
DAQCLK	0
IDLE	0
INVERT	1
PHASE[31:0]	10000000
STARTPPS	1
STARTEDGE	0
ACTIVE	1
RUNNING	1

Rising edge with a phase shift of 22.5º. Has a false transition at 1 PPS



Name	Value
CLK	1
GPS[31:0]	40000001
FREQ[7:0]	16
DAQCLK	0
IDLE	0
INVERT	1
PHASE[31:0]	10000000
STARTPPS	1
STARTEDGE	1
ACTIVE	1
RUNNING	0

Rising edge with a phase shift of 22.5º and no false transition at 1PPS

6 SPI Flash Programming

The SPI flash PROM typically contains 2 configuration files for the FPGA. The “golden” image which is read only, and a multiboot image which can be reprogrammed. A programming sequence would typically consist of the following steps:

1. Activate the golden image (erase second page in flash).
2. Erase all pages of the multiboot image.
3. Program new image into multiboot image.
4. Verify that new image has been programmed correctly.
5. Program a jump command at the beginning of the second page, pointing at the multiboot image.
6. Reboot the device.

Activating the golden image first will make sure that the FPGA will be able to boot even if reprogramming the multiboot image has failed.

Programming the SPI flash is accomplished through the 8 kB memory region mapped into the PCI space. This region is organized as follows:

Address	Description	Size
0x2000	Control, status and command registers	256
0x2100	First data block	3840
0x3000	Second data block	4096

Table 14: Memory map of SPI flash memory program interface.

The first and second data block can be used interchangeably to reprogram the flash. In particular, if one block is in use by the flash programmer, new data can be loaded into the other block through the PCI interface. Once the flash programmer is finished with its block, it can be switched to the other block, and new data can be loaded into the former. This sequence can be repeated until the full image has been programmed.

6.1 Supported Flash Devices

The following flash devices are supported:

Infineon: S25FL064L, S25FL128L, S25FL256L, S25FL128Sxxxxxx0, S25FL128Sxxxxxx1, S25FL256Sxxxxxx0, S25FL256Sxxxxxx1, and S25FL512S.

Micron: MT25QU and MT25QL families.

ISSI: IS25WP and IS25LP families.

The timing board is using a S25FL128L, whereas the converters are using a MT25QU512.

6.2 Register Map

The register set is organized as follows:

Address	Description	RW	Size
0x2000	Status Bit 31: SPI x8 is used Bits 30…5: Reserved Bit 5: Length is ok (poll and check after updating data length) Bit 4: Address is ok (poll and check after updating data address) Bit 3: Timeout during command execution Bit 2: Command returned with error Bit 1: Command completed successful Bit 0: Command is in process (poll after issuing a command)	R	4
0x2004	Data address	RW	4
0x2008	Data length	RW	4
0x200C	Command 1: Select golden image 2: Select multiboot image 3: Erase flash pages covered by address/length 4: Erase flash subpages covered by address/length 5: Program flash at address/length using first data block 6: Program flash at address/length using second data block 7: Read flash from address/length into first data block 8: Read flash from address/length into second data block 9: Restart FPGA all others: NOP (will always complete with error)	RW	4
0x2010	Flash ID after a flash ID read command has been executed Examples: 0x00016018 for the Infineon S25FL128L flash memory 0x0020BB20 for the Micron MT25QU512 flash memory	R	4
0x2014	Flash ID used by the programmer Bits 31…24: Set to 0 Bits 23…16: Manufacturer ID Bits 15…8: Memory type ID Bits 7…0: Capacity ID	R	4
0x018	Size of memory and reserved space Bits 31…24: Log2 of page size (Example: 8 for most devices) Bits 23…16: Log2 of sector size (Example: 16 for most devices) Bits 15…8: Log2 of memory size (Example: 13 for 8 kB) Bits 7…0: Log2 of reserved space (Example: 8 for 256 B)	R	4

0x201C	FPGA image side bytes Examples: 2,192,012 for an XC7A35T or XC7A35T 3,825,788 for an XC7A75T or XC7A100T 9,730,652 for an XC7A200T	R	4
0x2020	Image size in bytes (multiple of the sector size)	R	4
0x2024	Start address for multiboot image (multiple of the sector size)	R	4
0x2028	Start address of user data (multiple of the sector size)	R	4
0x202C	End address of flash (multiple of the sector size)	R	4
0x2030 … 0x203C	ASCII string of flash device family	R	16
0x2040	IDCODE from device	R	4
0x2044	BOOTSTS register from device	R	4
0x2048	AXSS register from device	R	4
0x204C	Unique ID 0x8AA26F33	R	4

Table 1: Registers to program the SPI flash PROM over the PCI Express bus.

6.3 Flash ID

If a read command with 0 data length is requested, a flash ID read will be executed instead. This will set register 0x02010, when finished. This command should be executed first to make sure the actual flash ID is equal to the one used by the controller. The later can be accessed through register 0x2014.

6.4 Command Sequence

A command sequence consists of the following steps:

1. Check that no previous command is still in progress by checking bit 0 in the status register.
2. Erase, program and read commands require an address and length, both are in bytes. Write the values to the corresponding address and length registers.
3. Make sure the register values have "settled" by checking bits 4 and 5 in the status register.
4. Write the command into the command register.
5. Make sure the command has completed by continuously reading bit 0 in the status register until it no longer indicates that it is busy.
6. Once the command is completed check if it was successful or not by inspecting bits 1-3 in the status register. Bit 1 indicates a successful operation, whereas bit 2 indicates a failed command with bit 3 indicating a timeout.

The commands for selecting the golden or user images and the restart command do not require an address or length, so steps 2 and 3 in the command sequence can be skipped.

The erase command takes the starting address of the first sector to be erased as the address and the number of sectors that need to be erased as the length. Any erase command trying to erase any sector before the multiboot image will be ignored. Meaning, the boot sectors and the golden image cannot be erased using the erase command.

The read command can access the entire flash device, so is able to verify that the boot sectors is written correctly, or that the device supports multiboot.

Like the erase command the program command cannot access the boot sector or the golden image. Only the multiboot image and the user area following the multiboot area can be programmed. The flash device organizes writes in pages, so the program command should be issued for 1 to multiple pages at the time.

6.5 Verification Steps

The following verification steps should be performed before programming a new multiboot image:

1. Check that there is a flash controller present by reading the unique ID register.
2. Issue a flash ID command and compare the flash ID against the ID the programmer is assuming. They should match.
3. Check that the current flash image supports multiboot by reading the data between addresses 0x00000100 and 0x0001FE00. All bytes should read 0xFF. This isn't bullet proof but it will catch almost all instances where just a single image has been loaded to the flash device.
4. Check that the FPGA of the new multiboot image is the same as the actual FPGA device on the timing board. This requires to extract the ID code from a bitstream and compare it to the one read from the IDCODE register.
5. After the new multiboot has been written to the flash device and before the select multiboot command is issued read back the entire multiboot image and compare to the one of file.

6.6 Code

A C++ class library is available to reprogram the flash device. It is located in the following subversion archive: https://svn.ligo.caltech.edu/svn/Altium-D2D/Software/Windows. The applicable files are "Common/PciInterface/FlashInterface.h" and "Common/PciInterface/FlashInterface.cpp". After creating an instance of the FlashInterface class, the method ProgramMultiboot can be used for loading a new multiboot image.

6.7 Rebooting

The easiest way to activate the new multiboot image is to power cycled the computer and IO chassis. The restart FPGA command can be used to reboot the FPGA, but this will make the timing board inaccessible to the computer until the PCI bus is rescanned.