LASER INTERFEROMETER GRAVITATIONAL WAVE OBSERVATORY

LIGO Laboratory / LIGO Scientific Collaboration

LIGO-T2200212-v1 Advanced LIGO November 2022

Software Interface to the Converter Prototype Board

Marc Pirello, Daniel Sigg

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/

Table of Contents

1 Overview

The PCIe converter board is a PCI Express board (Vendor ID 10EE, Device ID 7024, Revision 01) that interfaces the computer software through a memory mapped interface (BAR 0, BARs are 64 bit) as well as through a DMA engine with a memory mapped configuration space (BAR 2). As part of the PCI bus enumeration the operating system assign the board unique memory regions in the physical address space. This document describes the register assignment within the memory mapped interface region. The DMA interface is described in Xilinx PG195.

Vendor ID	Device ID	Revision	Sub Vendor ID	Subsystem ID
0x10EE	0x7024	01	0x10EE	0xD8C7

Table 1: PCI interface identification.

Access to the registers and memory located in BAR0 is through single double word (32-bit) read and write packets on the PCI bus, and through 32-bit wide memory read and writes in the driver software.

The board uses a 32 kB address region divided into 4 blocks: two 4kB and two 8kB, respectively. The first block is used for the control registers, whereas the second block is used to read the status of the converters. The third block is used to access the flash PROM, and the fourth block is used for the filter coefficients.

Address	Description	Size
0x0000	Control and monitor registers for on-board features, backplane and expansion modules	4096
0x1000	Converter Diagnostics Information	4096
0x2000	Flash Programmer	8192
0x4000	Filter Coefficient Memory	8192

Table 2: Memory map of PCIe converter interface.

2 Applicable Documents

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

- E2300321: PCIe LIGO Converter (DCC top node)
- E2200440: PCIe LIGO 32 Bit Digital to Analog Converter
- E2300322: PCIe LIGO 24 Bit Analog to Digital Converter
- D2000329: PCIe Timing Interface Board

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. Only 32-bit access is supported on all registers.

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 3: Registers to read the current time.

3.2 Slot Assignment

When operated in conjunction with a LIGO timing board and a LIGO IO chassis, the clocking is provided through the backplane slots using the same signal as the timing distribution system. Each slot acts like a fanout port of a timing master with port address ranging from 1 through 10. This 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 (indicating slot 1 for a converter), the third one is 0x12000000 (indicating slot 2) and so forth. The last node address used by the back plane is 0x1A000000 indicating slot 10. 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 Firmware Release

The firmware release version is accessible at address 0x000C. Any officially released version of the FPGA code will have a non-zero value.

3.5 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 (always 0) Bit 29: Supports fanout ports (always 0) Bit 28: Uplink is up and working Bit 27: Loss-of-signal on uplink (always 0) Bit 26: OCXO is locked (always 0) Bit 25: GPS is locked (always 0) Bit 24: VCXO control voltage out-of-range Bit 23: UTC time mode enabled (always 0) Bit 22: Leap seconds decoded Bit 21: Subtraction of one leap second pending Bit 20: Addition of one leap second pending Bits 19...17: Reserved Bit 16: Watchdog monitor 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 4: Status and interrupt control register and firmware revision.

For a converter board to be ready to start DMA transfers it has to be full synchronized, meaning the OK flags is set and indicating that the clock is locked to the timing signal, and that the GPS time has been obtained.

3.6 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.

Different converter boards if build with the same timing and DMA core, will typically have the same vendor and device numbers, so they can be served by a single unified software driver. Board and firmware ID can then be used to determine the exact configuration of ADCs and DACs, speed, as well as the channel count. For example, the LIGO DAC32 is represented by 0x22003681 linking to D2200368-v1, and by 0x23002291 linking to E2300229. Revision numbers are typically only updated by a major release. Use the firmware release number to identify smaller updates.

3.7 Converter Sampling Interface

3.7.1 Sampling Configuration

The table below shows the common functions for the prototype DMA interface. The error bits are sticky and are getting cleared when read.

Address	Description	RW	Size
0x0010	Status: Bits 31…30: Unused Bit 29: DAC timestamp error channel 1 Bit 28: DAC timestamp error channel 0 Bit 27: DAC DMA missing data error channel 1 Bit 26: DAC DMA missing data error channel 0 Bit 25: DAC DMA ready error channel 1 Bit 24: DAC DMA ready error channel 0 Bit 23: Watchdog monitor (same as bit 1 in register 0x01FC) Bit 18: Valid DAC sampling configuration Bit 17: DAC converter is running Bit 16: DAC DMA is running Bits 15…12: Unused Bit 11: ADC DMA missing data error channel 1 Bit 10: ADC DMA missing data error channel 0 Bit 9: ADC DMA ready error channel 1 Bit 8: ADC DMA ready error channel 0 Bit 7: Timing OK flag (same as bit 31 in register 0x0008) Bit 2: Valid ADC sampling configuration Bit 1: ADC converter is running Bit 0: ADC DMA is running	R	4
0x0014	Configuration: Bits 31...24: Log2 of number of buffers per DAC DMA channel Bit 23: Use interrupt to signal DAC DMA done Bits 22…20: Unused Bit 19: Ignore DAC timestamp errors Bit 18: Disable DAC timestamp Bit 17: Disable DAC conversion Bit 16: Enable DAC DMA Bits 15...8: Log2 of number of buffers per ADC DMA channel Bit 7: Use interrupt to signal ADC DMA done Bits 6…3: Unused Bit 2: Disable ADC timestamp Bit 1: Disable ADC conversion Bit 0: Enable ADC DMA	RW	4
0x0018	ADC late/missing DMA error counter; write any value to reset	RW	4
0x001C	DAC late/missing DMA error counter; write any value to reset	RW	4

Table 5: Common registers for the prototype DMA interface.

3.7.2 Sampling Setup

Sampling and DMA rate can be set up for ADCs and DACs separately, if both are supported. Only rates which are a power of 2 are supported. The ADC and DAC conversion rates needs to be equal or larger than the corresponding DMA rates.

Address	Description	RW	Size
0x0020	ADC DMA period (in units of 2^{-32} sec) – 1	RW	4
0x0024	ADC DMA delay (in units of 2^{-32} sec) – 1	RW	4
0x0028	ADC sampling delay (in units of 2^{-32} sec) – 1	RW	4
0x002C	ADC sampling period (in units of 2^{-32} sec) – 1	RW	4
0x0030	DAC DMA period (in units of 2^{-32} sec) – 1	RW	4
0x0034	DAC DMA delay (in units of 2^{-32} sec) – 1	RW	4
0x0028	DAC sampling delay (in units of 2^{-32} sec) – 1, bit 0 for extra delay	RW	4
0x003C	DAC sampling period (in units of 2^{-32} sec) – 1	RW	4

Table 6: Registers to setup clock signals for the ADCs and DACs.

For example, if the desired DMA and ADC rates are both 2^{16} Hz, the registers for ADC and DMA periods must be set to 0x0000FFFF. The ADC sampling delay can be set to 0 to start sampling exactly at the 1 sec mark. The register for the ADC sampling delay should then be set to 0xFFFFFFFF. The DMA delay needs to take the processing delay into account and is typically larger than the ADC sampling delay by a fixed offset which is determined by the specific ADC board. However, it can be made longer to spread the DMA transfers among multiple boards.

The largest recognized DMA delay is one DMA period. All bits potentially indicating a larger delay are ignored. The same is true for the sampling delay which cannot be longer than the DMA period. Hence, all delay specifications are modulo the DMA period. However, if bit 0 of the DAC sampling delay is set to zero, an additional DMA period is added between the start of the DMA transfer and the start of the conversion.

For a DAC it is important to account for the time it takes to transfer the data from the host to the board. A good choice for the DMA delay is 0 and half the DMA period for the sampling delay, i.e., 0xFFFFFFFF and 0x00007FFF, respectively.

Both ADC and DAC data can use timestamps to prevent stale data from being processed. Data timestamps only uses bits larger than the sample period. All bits modulo the sampling period are either set or expected to be zero. If the DMA period and sampling period are identical, the data timestamp is sent or received with each set of samples describing the sampling time. In case of oversampling where multiple sets of samples are transferred with a single DMA transfer, only one data timestamp is needed describing the first set of samples.

3.7.3 Converter Configuration

These parameters are configured by the board and its application. The maximum channel count can be used to estimate the required buffer sizes, whereas the actual channel count should be used to configure the DMA engine. Each ADC or DAC sample is fit into a 32-bit signed integer requiring 4 bytes. Nominally, each converter value is shifted so that the largest value is $2^{28} - 1$, and the lowest value -2^{28}. However, since digital filters are used for down sampling, actual ADC values can become slightly larger than this range.

Address	Description	RW	Size
0x0040	Maximum number of ADC channels	R	4
0x0044	Maximum number of DAC channels	R	4
0x0048	Actual number of ADC channels	R	4
0x004C	Actual number of DAC channels	R	4
0x0050	Bits 31…24: Maximum ADC DMA rate (log2) Bits 23…16: Minimum ADC DMA rate (log2) Bits 15…8: Maximum ADC rate (log2) Bits 7…0: Minimum ADC rate (log2)	R	4
0x0054	Bits 31…24: Maximum DAC DMA rate (log2) Bits 23…16: Minimum DAC DMA rate (log2) Bits 15…8: Maximum DAC rate (log2) Bits 7…0: Minimum DAC rate (log2)	R	4
0x0058	Bits 31…16: ADC processing delay Bits 15…8: ADC native rate (log2) Bits 7…0: Maximum ADC oversampling (log2)	R	4
0x005C	Bits 31…16: DAC processing delay Bits 15…8: DAC native rate (log2) Bits 7…0: Maximum DAC oversampling (log2)	R	4
0x0060	ADC DMA buffer size in bytes	R	4
0x0064	DAC DMA buffer size in bytes	R	4
0x0068	Bits 31…16: Number of loopback channels Bits 15…5 Unused Bit 4: Loopback enabled Bit 3: DAC filter interface present Bit 2: DACs interface present Bit 1: ADC filter interface present Bit 0: ADCs interface present	R	4

0x006C	Bits 31…16: Unused Bits 15…8: Log2 of maximum number of buffers per DMA channel Bits 7…0: Log2 of clock rate	R	4
0x0070	Bits 31…24: Log2 of DAC DMA buffer transfer bus width in bytes Bits 23…16: Log2 of DAC DMA buffer sampling width in bytes Bits 15…8: Log2 of ADC DMA buffer transfer bus width in bytes Bits 7…0: Log2 of ADC DMA buffer sampling width in bytes	R	4
0x0078	Frequency of the AXI clock (in Hz) (derived from the PCIe clock)	R	4
0x007C	DAC data delay in units of the clock rate	R	4

Table 1: Registers describing the converter setup.

To estimate the required DMA transfer length, the following steps need to be considered:

- There is a dual ported buffer between the converter and PCI bus that can have a different width on the sampling side and the transfer side. Typically, the sampling side works with 4 converter values in parallel and is 16 bytes wide. The transfer side is either 8 bytes or 16 bytes wide.
- When ADC values are loaded into the DMA buffer, dummy values are added if the number of channels is not a multiple of the buffer width. This will typically not be an issue for boards that have a multiple of 4 sampling channels. The same is true for DAC values that are read from the buffer with possible dummy values ignored.
- If oversampling has been configured the number of converter values rounded up to the width of the DMA buffer access is multiplied by the oversampling value. All these values must fit into the DMA buffer which has a finite length. This will typically limit the maximum oversampling. Timestamps and status values do not reside in this buffer and don't contribute to this count.
- The DMA buffer is transferred to and from the PCI bus at the bus transfer width. If the buffer size for the converter values isn't a multiple of this bus transfer width, it again must be rounded up. Typically, this isn't an issue since the sampling width is equal or larger than the bus transfer width.
- If enabled, a 16-byte time stamp and status information block is added to this buffer.
- For ADC values the DMA length is required to be a multiple of the cache line size, typically 64 bytes.
- The final DMA length needs to be a multiple of the bus transfer width. Since the bus transfer width is 16 bytes or smaller, no further rounding should be required at this stage.
- This constitutes the final DMA length that needs to be written into the appropriate DMA channel configuration registers. The DMA engine works with a 28-bit value representing the number of bytes in the DMA transfer. However, the converter interface will ignore any bits beyond twice the buffer size. It will also ignore the lowest 4 or 3 bits depending on whether the bus transfer width is 16 or 8 bytes, respectively.

3.7.4 Filter Setup

Converter board typically implement IIR filters to smooth decimate and up-sampling. IIR filters are organized into banks. All samples passing through a bank will use the same filter. Depending on the sampling rate and the clock rate, this filter has a maximum cycle count. Typically, 4 cycles are used for applying a gain factor and for waiting out the pipeline delay. The remaining cycles are divided into groups of 4 cycles, each group implementing a second-order-section. All but the 3 cycles of the pipeline delay require a filter coefficient that is loaded from memory. This coefficient is 64-bits wide. Hence, the required memory for the filter coefficients is the number of cycles times 8 bytes.

However, the memory block allocated for the filter can be larger than what is needed for a single filter. In this case, it is possible to select from multiple available filters by changing the filter selection bits in register 0x0090. The base-2 logarithm of the cycle count and of the number of available filters can be read from the filter configuration register at address 0x0080, using the lower and upper nibble of a byte, respectively. Up to 4 filters can be configured with a single register. If a configuration byte is set to zero, no filter is available.

Address	Description	RW	Size
0x0080	Filter Configuration: Bits 31…24: Filter Configuration 4 Bits 23…16: Filter Configuration 3 Bits 15…8: Filter Configuration 2 Bits 7…0: Filter Configuration 1 Lower nibble: Log2 of cycle count Upper nibble: Log2 of number of available filters	R	4
0x0090	Filter Selection: Bits 31…24: Filter Selection 4 Bits 23…16: Filter Selection 3 Bits 15…8: Filter Selection 2 Bits 7…0: Filter Selection 1	RW	4

Table 1: Registers describing the filter setup.

For example, if the converter clock is 2^{20} Hz and the FPGA clock runs at 2^{26} Hz, a filter can have up to 64 cycles. This requires 512 bytes of memory. If the allocated memory block is 8 Kbytes, there is space for 16 different filters that can be selected by setting the corresponding filter selection byte in the register. Unused bits in the filter selection byte are ignored. In the example above, only the lowest 4 bits are used.

The filter coefficients are available at offset 0x4000 of the first BAR. The memory starts with filter 1 and continues with filter 2, etc. However, there may be gaps in memory, since it each filter has to be aligned by its block size. For instance, an 8 Kbytes filter memory block has to start at an address which is a multiple of 0x2000.

3.8 DMA Channel Configuration for Input

The 2 DMA channels transferring data from the converter board to the computer memory are configured starting at address 0x00C0 with the second channel starting at address 0x00D0.

Address	Description	RW	Size
0x00C0	Destination address (64b aligned, lower 32 bits, even ADC samples)	RW	4
0x00C4	Destination address (64b aligned, upper 32 bits, even ADC samples)	RW	4
0x00C8	Length in bytes (must be a multiple of 64 bytes, only 28 bits used)	RW	4
0x00CC	Buffer offset in bytes (must be a multiple of 64 bytes)	RW	4
0x00D0	Destination address (64b aligned, lower 32 bits, odd ADC samples)	RW	4
0x00D4	Destination address (64b aligned, upper 32 bits, odd ADC samples)	RW	4
0x00D8	Length in bytes (must be a multiple of 64 bytes, only 28 bits used)	RW	4
0x00DC	Buffer offset in bytes (must be a multiple of 64 bytes)	RW	4

3.9 DMA Channel Configuration for Output

The 2 DMA channels transferring data from the computer memory to the converter board are configured starting at address 0x00E0 with the second channel starting at address 0x00F0.

0x00E0	Source address (64b aligned, lower 32 bits, even DAC samples)	RW	4
0x00E4	Source address (64b aligned, upper 32 bits, even DAC samples)	RW	4
0x00E8	Buffer length in bytes (only 28 bits used)	RW	4
0x00EC	Buffer offset in bytes (must be a multiple of 64 bytes)	RW	4
0x00F0	Source address (64b aligned, lower 32 bits, odd DAC samples)	RW	4
0x00F4	Source address (64b aligned, upper 32 bits, odd DAC samples)	RW	4
0x00F8	Buffer length in bytes (only 28 bits used)	RW	4
0x00FC	Buffer offset in bytes (must be a multiple of 64 bytes)	RW	4

Table 2: Registers to configure interrupts over the PCI Express bus.

The 2 DMA channels for each transfer direction are used in a double buffer arrangement, where the first channel is used for even sample and the second for odd samples. The two corresponding source or destination addresses can be set to the same address if double buffering is not required. The destination buffer in the host memory for ADC transfers, must be a multiple of 64 bytes and aligned

to 64 bytes (cache line). Setting up only one channel is a mistake. The number of buffers per DMA channel is set in the configuration register at 0x0014. This register also includes interrupt enables bits that allow rising an interrupt when a DMA transfer is finished.

It is important that the DMA transfers are not enabled before both the DMA channel configuration and the sampling configuration have been set up. If the log2 number of buffers per DMA channel is not zero, its value should be set up at least one full cycle through all the DMA buffers before the DMA is enabled. This ensures that all DMA transfers will use the correct buffer address.

3.10 Advanced Timing Features

Each converter implements a timing interface with limited options. Options such as the GPS module, the fanout SFP ports, the daughter card EEPROM, an OCXO, the PPS signals, the RS422 interface, the IRIG-B outputs, or the timing diagnostics information are not available.

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	R	4
0x0138	Advanced timing status Bit 31…27: Unused Bit 26..24: Timing link version Bit 23: Analog output for XO locking enabled (always 0) Bit 22: BRAM option enabled for timing diagnostics (always 0) Bit 21: PCIE option enabled (always 1) Bit 20: IRIG-B option enabled (always 0) Bit 19: RS422 option enabled (always 0) Bit 18: PPS option enabled (always 0) Bit 17: OCXO option enabled (always 0) Bit 16: GPS option enabled (always 0) Bit 15: EEPROM on daughterboard has been read (always 0) Bits 14...3: Future expansion options (always 0) Bit 2: Fanout expansion is present (always 0) Bit 1: GPS expansion is present (always 0) Bit 0: Daughter board is present (always 0)	R	4
0x013C	Unused		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 3: Registers for advanced timing features.

3.11 On Board Features

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

3.11.1 Common Readouts

Address	Description	RW	Size
0x0180	Board configuration: Bits 31…0: Reserved	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

Table 4: Registers for board diagnostics (common).

The XADC values are unipolar and consists of 16 bits. So, the ADC value 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 tables above and below.

3.11.2 LIGO DAC32

The measurements below are implemented on the LIGO DAC32 board.

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: Current of +7V (gain 0.3Ω) Bits 15…0: +7V (gain 1/10, nominally +7V)	R	4
0x01A4	External power supply 4 Bits 31…16: Current of –7V (gain 0.3Ω) Bits 15…0: +7V (gain 1/10, nominally –7V)	R	4
0x01A8	External power supply 5 Bits 31…16: Current of AVCC (gain 0.6Ω) Bits 15…0: Current of AVTT (gain 0.6Ω)	R	4
0x01AC	External power supply 6 Bits 31…16: Current of VCCA (gain 0.6Ω) Bits 15…0: Current of DVDD (gain 0.3Ω)	R	4
0x01B0	External power supply 7 Bits 31…16: Current of AVCC3.3 (gain 0.3Ω) Bits 15…0: AVCC3.3 (gain 1/5, nominally +3.3V)	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
0x01FC	Watchdog (write any value to toggle watchdog trigger) Bit 1: Watchdog monitor Bit 0: Readback of watchdog trigger	RW	4

Table 5: Registers for board diagnostics (LIGO DAC32).

3.11.3 LIGO ADC32

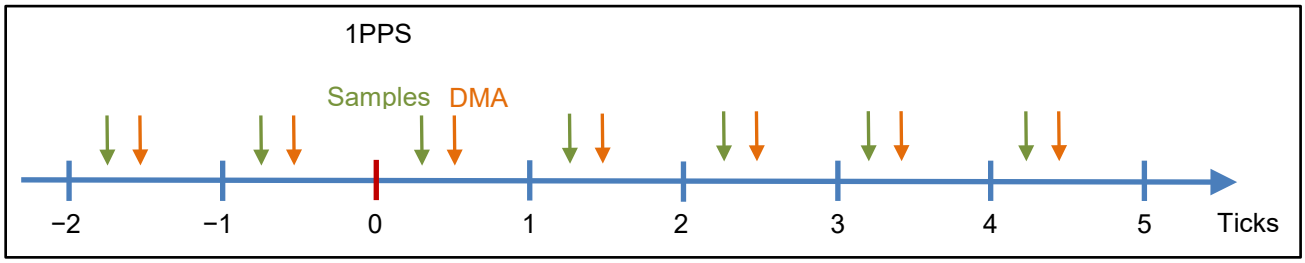
The measurements below are implemented on the LIGO DAC32 board.

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: Current of +7V (gain 0.3Ω) Bits 15…0: +7V (gain 1/10, nominally +7V)	R	4
0x01A4	External power supply 4 Bits 31…16: Current of –7V (gain 0.3Ω) Bits 15…0: +7V (gain 1/10, nominally –7V)	R	4
0x01A8	External power supply 5 Bits 31…16: Current of AVCC (gain 0.6Ω) Bits 15…0: Current of AVTT (gain 0.6Ω)	R	4
0x01AC	External power supply 6 Bits 31…16: Current of VCCA (gain 0.6Ω) Bits 15…0: Current of DVDD (gain 0.3Ω)	R	4
0x01B0	External power supply 7 Bits 31…16: Current of AVCC3.3 (gain 0.3Ω) Bits 15…0: AVCC3.3 (gain 1/5, nominally +3.3V)	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
0x01F8	Control voltage for the ADC VCXO Bits 15…0: unipolar from 0V to 3.3V	R	4
0x01FC	Watchdog (write any value to toggle watchdog trigger) Bit 1: Watchdog monitor Bit 0: Readback of watchdog trigger	RW	4

Table 6: Registers for board diagnostics (LIGO ADC32).

4 Converter Configuration

The picture below shows a horizontal time axis in units of clock ticks. Tick 0 is aligned with the 1 PPS. In the example below the sampling is delayed relative to the clock ticks and a DMA transfer is initiated after each sample. To specify this sequence of events one has to specify the period and delay of both sampling and DMA. This values are specified in units of 2^{-32} sec minus 1.



4.1 Example 1: ADC Sampling and DMA at 65536 Hz

The values for both the sampling and DMA period are 0x0000FFFF. With the sample aligned with the 1 PPS the delay is 0xFFFFFFF. There is a minimum time delay between the sampling clock and when the data is available for DMA. This delay is board specific, for example 15 clock cycles, where the clock runs at 2^{26} Hz. The DMA delay then needs to be set to 0x000003FF or larger.

4.2 Example 2: ADC Sampling at 524288 Hz and DMA at 65536 Hz

The DMA rate is again specified by 0x0000FFFF. If we set the DMA delay to the center between the DMA ticks, its value becomes 0x00007FFF. The ADC sampling period is 0x00001FFF and we start the 8 samples belonging to a DMA cycle 4 cycles before the 1 PPS. The ADC samples are again aligned with the 1 PPS, so the ADC delay is 0xFFFFFFFF. This setup will result in 8 ADC samples for each DMA cycle, starting 4 ADC cycles before the 1 PPS.

4.3 Example 3: DAC Sampling and DMA at 65536 Hz

We start the DMA cycle aligned with the 1 PPS and start the DAC conversion in the middle of the DMA cycle. This requires the values for both the sampling and DMA period to be 0x0000FFFF, the DMA delay to be 0xFFFFFFFF, and the sample delay to be 0x00007FFF.

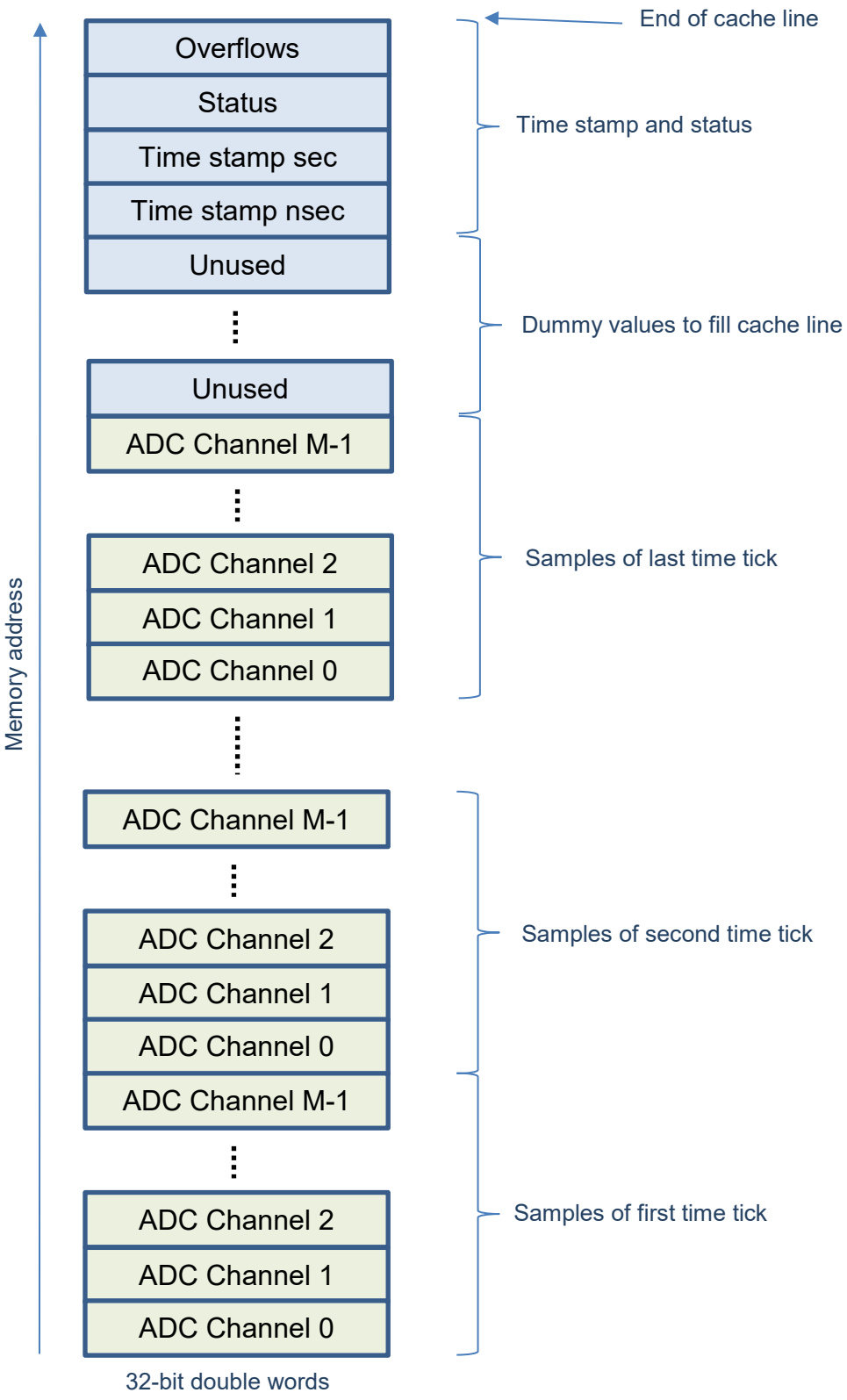
5 DMA Buffer Configuration

Input and output require 2 DMA channels each that work in a double buffer configuration. Hence, a total of 4 DMA channels need to be configured, it both inputs and outputs are needed. Furthermore, each DMA channel can cycle through 2^N buffers individually, with N = 0, 1, 2, etc.

Each DMA channel requires a 64-bit destination or source address pointing to the first DMA buffer, a 28-bit DMA buffer length, the value N to specify the number of buffers per DMA channel, and a 32-bit DMA buffer offset.

DMA buffers need to align with the cache line of the processor, typically 64 bytes. The ADC DMA buffer length must be a multiple of the cache line.

5.1 DMA Buffer Layout



The above memory layout of a DMA buffer assumes an ADC board with M input channels. If oversampling is enabled, the data from multiple time ticks is put into the same DMA buffer. The ADC data is followed by some dummy words, a 64-bit time stamp, 4 byte of status indicators, and a 32-bit reserved word. The ADC buffer is patched by the dummy words, so the entire buffer is a multiple of the cache line size. No dummy words are expected for DAC buffers.

ADC and DAC values are stored in 32-bit two's complement words, where only 28-bits are used. The first 4 bits are signed extended and are typically all zeros or ones. The time stamp is a 64-bit fixed point value representing GPS seconds with the decimal point after 32 bits. The time stamp represents a time tick, meaning the resolution is given by the inverse of the sampling rate.

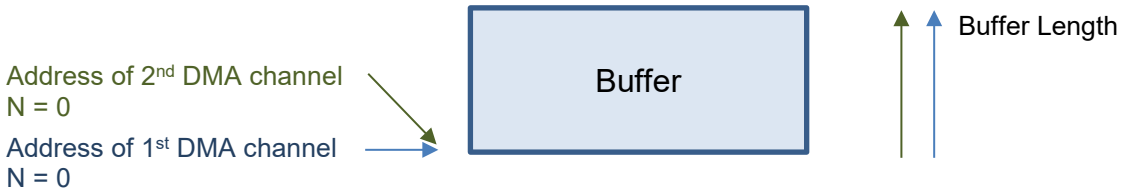
The 4 status bytes represent the ADC state machine status, the ADC DMA errors, the DAC state machine status, and the DAC DMA errors:

Byte	Description		
0	ADC state machine status Bit 7: Timing OK flag (same as bit 31 in register 0x0008) Bits 6…3: Reserved Bit 2: ADC data valid Bit 1: DMA for ADCs is running Bit 0: ADCs are enabled and running		
1	ADC DMA errors Bits 7…4: Reserved Bit 3: DMA data in channel 1 didn't arrive in time Bit 2: DMA data in channel 0 didn't arrive in time Bit 1: DMA channel 1 isn't ready Bit 0: DMA channel 0 isn't ready		
2	DAC state machine status Bit 7: Watchdog monitor (same as bit 1 in register 0x01FC) Bits 6…3: Reserved Bit 2: DAC data valid Bit 1: DMA for DACs is running Bit 0: DACs are enabled and running		
3	DAC DMA errors Bits 7…6: Reserved Bit 5: DMA timestamp error in channel 1 Bit 4: DMA timestamp error in channel 0 Bit 3: DMA data in channel 1 didn't arrive in time Bit 2: DMA data in channel 0 didn't arrive in time Bit 1: DMA channel 1 isn't ready Bit 0: DMA channel 0 isn't ready		

The DMA errors are sticky, meaning if an error is raised it stays until a reset is issued through reading the DMA Status register at address 0x0010.

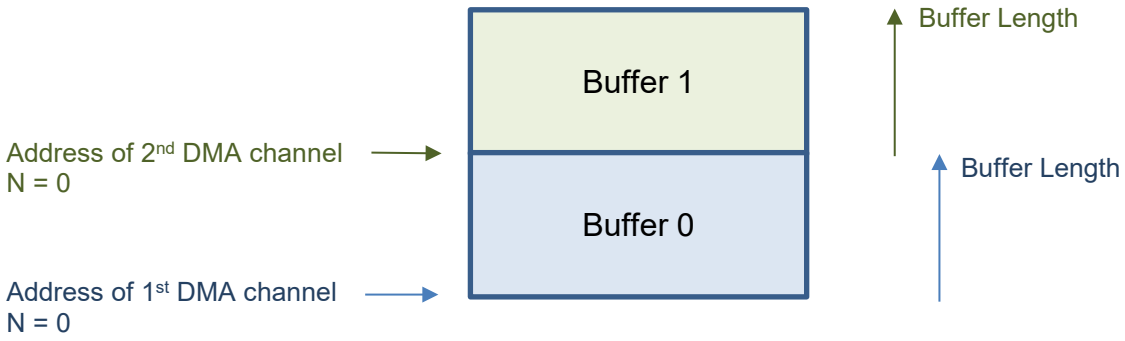
5.2 Example 1: ADC Input with a Single Buffer

The destination addresses of the 2 DMA channels need to be identical, N and the DMA buffer offset are set to 0. The DMA buffer needs to be long enough to hold all ADC values as well as a time stamp. The ADC values are 32 bit, whereas the time stamp is 128 bit. The required buffer length then needs to be rounded up to the next cache line multiple. For example, a 32 channel ADC board that gets read out after each sample clock requires 144 bytes. The DMA buffer size then needs to be rounded up to 192 bytes. Similar, if we require an 8x over sampling, the DMA buffer size need to be 1088 bytes.



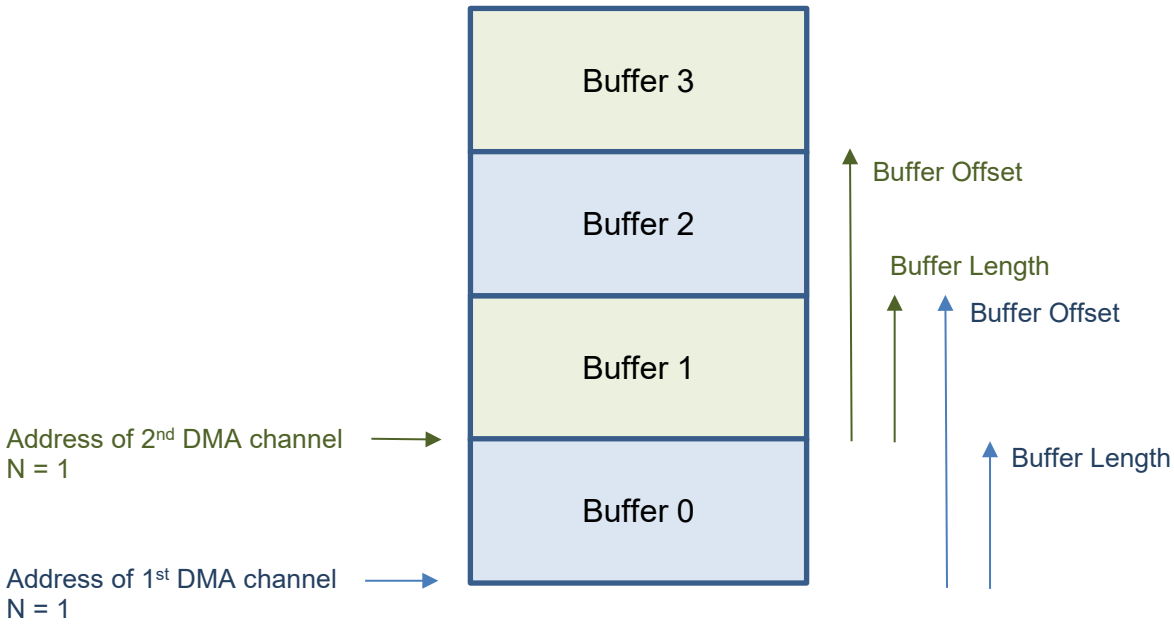
5.3 Example 2: ADC Input with Double Buffering

The only difference here is that the destination address of the second DMA channel needs to be set to a different value. For instance, it could be higher by the DMA buffer size. The samples associated with the 1 PPS tick are read into the first buffer (even), whereas the next set of samples are read into the second buffer (odd).



5.4 Example 2: ADC Input with 4 Buffers

Here we set N to 1, which results in 2 buffers per DMA channel. Again, the first DMA channel processes the samples from the even ticks, and the second DMA channel processes the samples of the odd ticks. To obtain 4 DMA buffers in series arranged in a ring buffer, the destination address of the second DMA channel should be again higher by the DMA buffer size, and the DMA buffer offset should be set to twice the DMA buffer size, see illustration below.



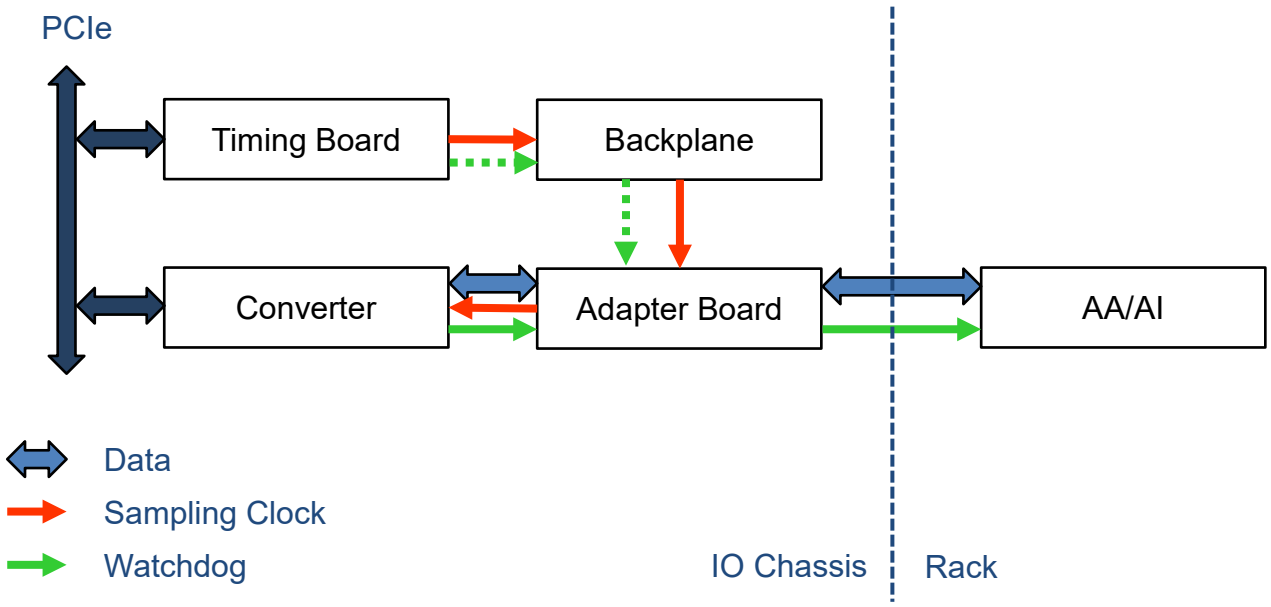
6 Data Format

The ADC and DAC data values are represented by 32-bit signed integers using 2's complement. Although, only 28 bits are significant, meaning the maximum full scale readout corresponds to $2^{27}-1$, whereas the minimum corresponds to -2^{27}. For a DAC output there is no strict overflow at 28-bits before the filter processing, i.e., the filters will process the full 32-bits. The only strict 28-bit overflow is at the DAC itself. For example, one could send signals that use more than 28-bits, and as along as the filter reduces it to a signal below 28-bits it will not overflow at the DAC. In reverse, the ADC output is adjusted for a full range at 28 bits, but the following filter processing may result in a number that is slightly larger than full scale.

The board does report on overflows by looking at the filter output and limit it to 32-bits (it uses a few bits more internally) and by looking at the DAC inputs and making sure they fit into 28-bits. Overflow bits are reported in the status word which is part of the of the readback data.

7 Startup Procedure

The diagram below shows a typical setup of a converter and a timing board inside a LIGO IO chassis. The timing board is synchronized to GPS and feeds clocking signals to the converters via a backplane and an adapter board. For the LIGO converters the clocking signal is the same timing signal used by the timing distribution signal which implements a 2^{23} Hz clock. The positive edges are used to synchronize a local 2^{26} Hz VCXO on the converter, and the negative edges are shifted inwards or outwards to transfer timing data, such as the 1 PPS, the GPS time, the leap second information, and the slot number (see section 3.2).



The startup procedure has the following steps:

1. Setup pointers to the PCI register space for converters and timing boards.
2. Wait for the timing board to lock to GPS (see section 3.4 in T2000406).
3. Set bit 13 and 16 in the slot configuration of the timing board (see section 3.6.2 in T2000406). The slot number is determined by the position of the adapter board in the backplane.
4. Wait for the converter board to lock to GPS (see section 3.5).
5. Read the board id, firmware id and revision as well as the slot id to identify the converter board if needed (see sections 3.2 to 3.6).
6. Calculate the size for the DMA buffers and allocate them.
7. Set the DMA address, length and offset registers (see section 3.8 and 3.9).
8. Calculate the sampling parameters and set the corresponding registers (see section 3.7.2).
9. Load filter coefficients if needed, and select the desired filter (see section 3.7.4).
10. Start the IOP process that fills the DMA output buffers and reads the DMA input buffers.
11. Once everything is running stably, reset the error flags by reading the sampling status register. Also, reset the error counters if they are used (see section 3.7.1).

8 Filter Coefficient Memory

Address	Description	Size
0x1000		

Table 7: Memory map of timing diagnostics information.