User Manual

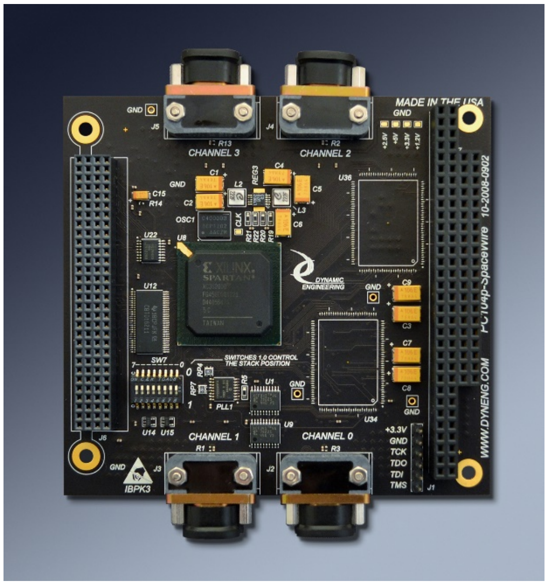
Hardware and Software

SpaceWire Monitor

Manual Revision 01p6
Revision Date 01/02/2023

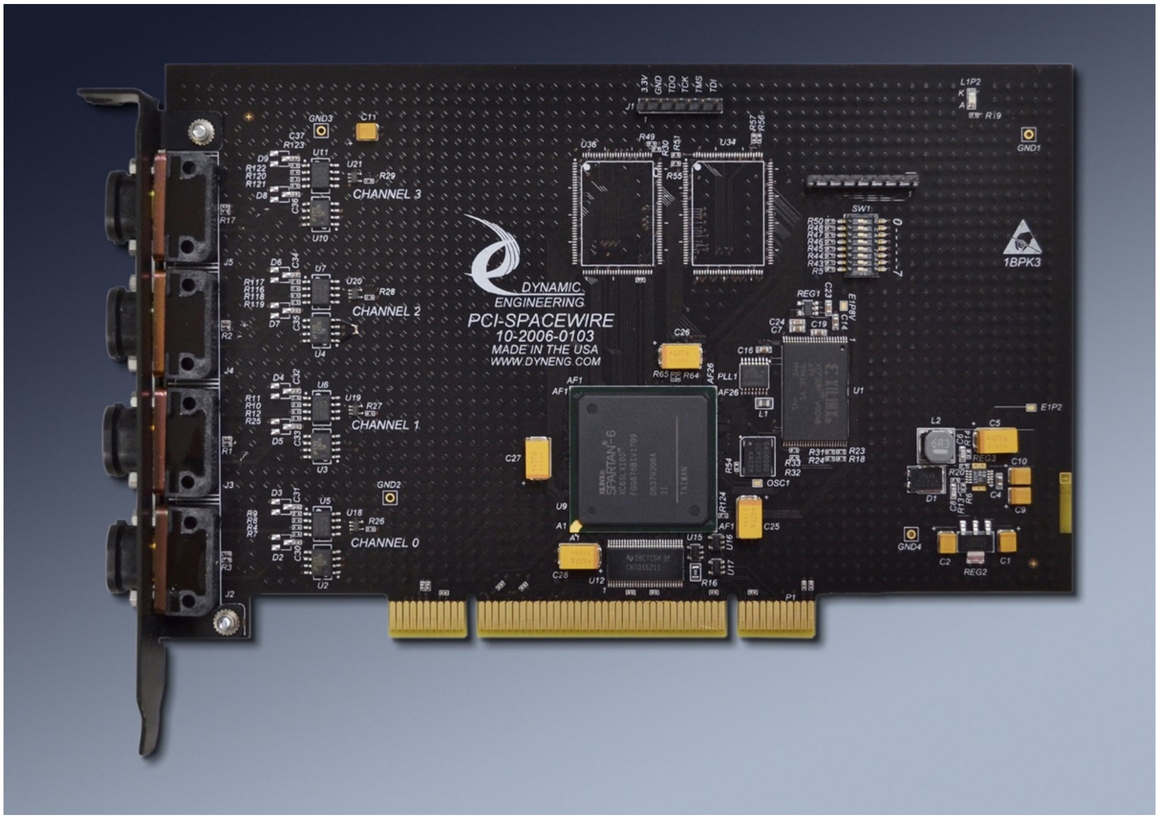
Dynamic Engineering
150 DuBois St. Suite B/C
Santa Cruz, CA 95060
(831) 457-8891
www.dyneng.com
sales@dyneng.com
Est. 1988

PC104p-SpaceWire



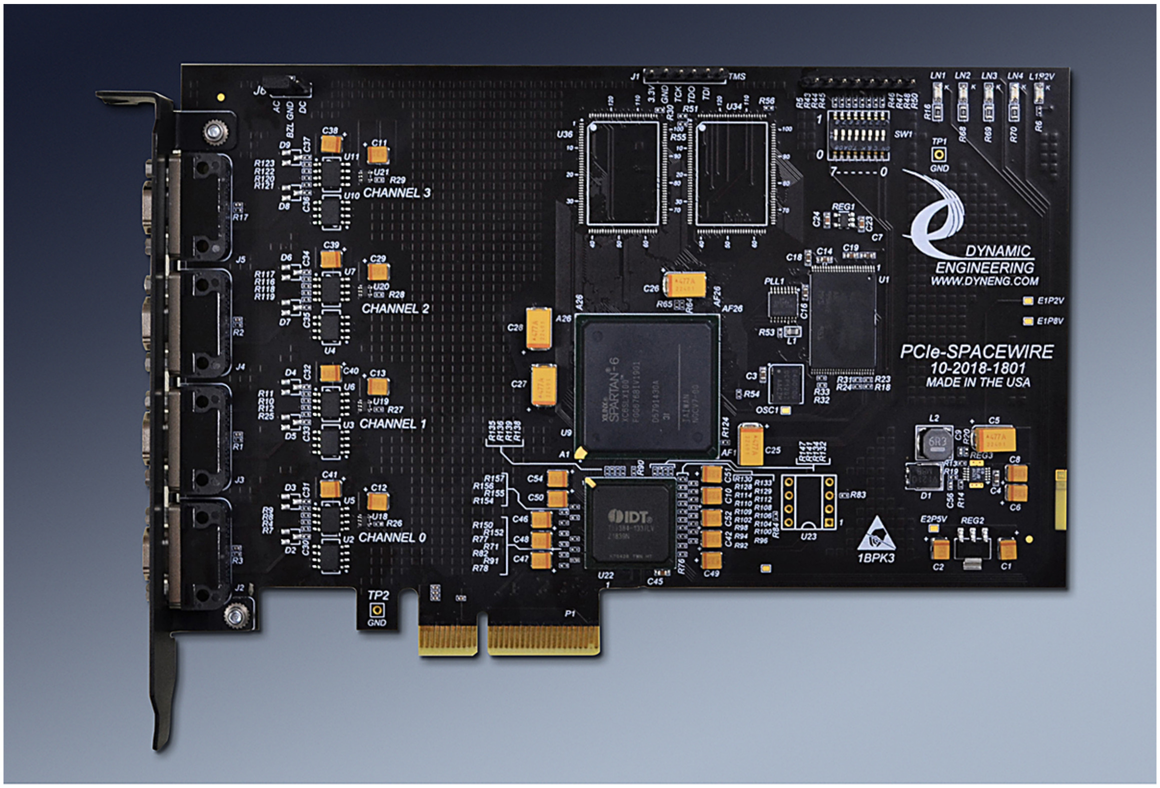
Corresponding Hardware: 10-2008-0903

PCI-SpaceWire



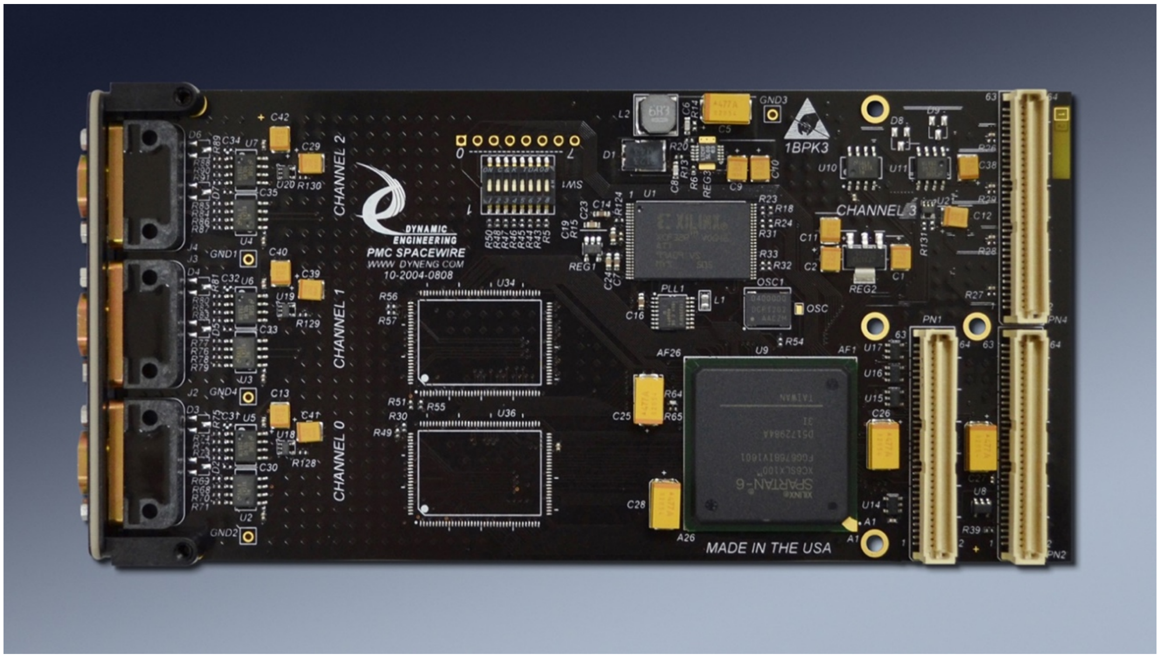
Corresponding Hardware: 10-2006-01(04,05)

PCIe-SpaceWire



Corresponding Hardware: 10-2018-18(02,03)

PMC-SpaceWire



Corresponding Hardware: 10-2004-08(10,11)

NOTE: This manual includes:

1. PC104p-SpaceWire-Monitor
2. PCI-SpaceWire-Monitor
3. PCIe-SpaceWire-Monitor
4. PMC-SpaceWire-Monitor

which are collectively referred to as SpaceWire Monitor henceforth.

NOTE: This manual provides design and usage details of the SpaceWire Monitor. Since the SpaceWire Monitor was created by modifying a SpaceWire-RX card and its FPGA design, please see the SpaceWire User Manual for extensive details of the Monitor board and design not covered in this manual. Changes to the SpaceWire design to create the SpaceWire Monitor are detailed in this Manual.

Cautions and Warnings

The electronic equipment described herein generates, uses, and can radiate radio frequency energy. Operation of this equipment in a residential area is likely to cause radio interference, in which case the user, at their own expense, will be required to take whatever measures may be required to correct the interference.

Dynamic Engineering's products are not authorized for use as critical components in life support devices or systems without express written approval from the president of Dynamic Engineering.

Connection of incompatible hardware is likely to cause serious damage.

Table of Contents

Tables

Figures

Design Revision History

Table 1: Flash and Software Revision History

Revision	Date	Description
		Flash
2.1	7/2/2021	Initial release
		Linux
1	7/2/2021	Initial release
NOTE: For additional PCB-related revision information, refer to the SpaceWire User Manual.		

Manual Revision History

Table 2: Manual Revision History

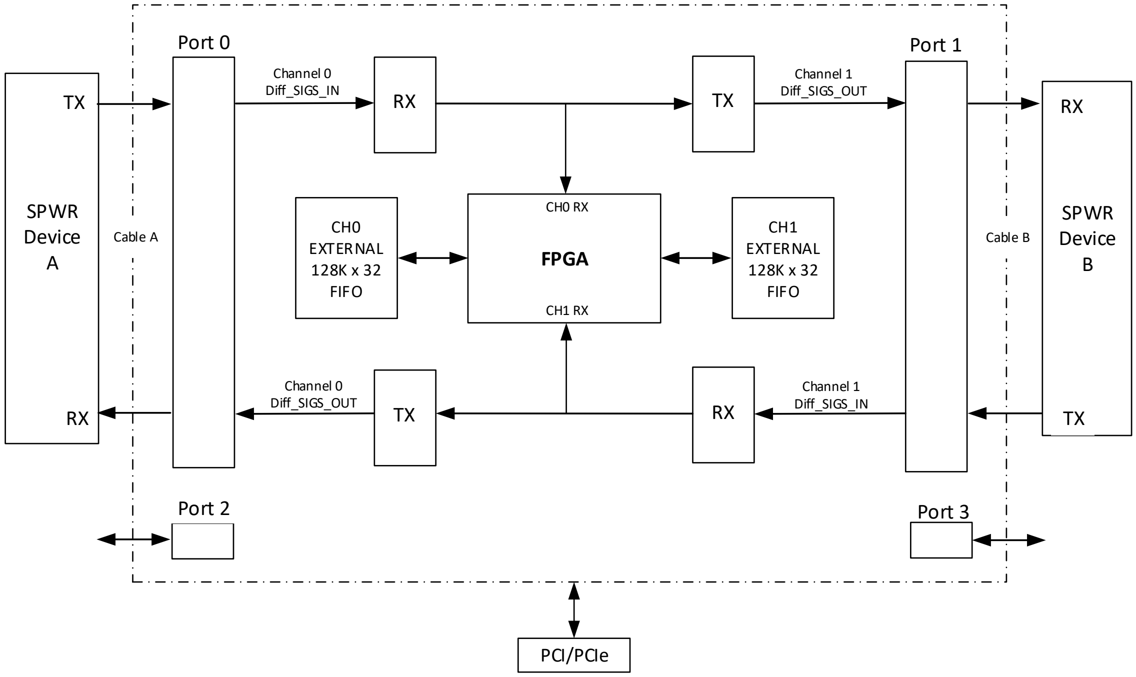
Revision	Date	Description
01p1	7/27/21	Initial release of design and manual
01p4	8/23/21	Minor clean-up revs through p4
01p5	9/24/21	Updated due to support of packet mode disabled
1p6	01/02/23	Updated to new manual format, misc. clean-up. Removal of some items [included in SW manuals]

NOTE: Dynamic Engineering has made every effort to ensure that this manual is accurate and complete; that being said, the company reserves the right to make improvements or changes to the product described in this document at any time and without notice. Furthermore, Dynamic Engineering assumes no liability arising out of the application or use of the device described herein.

Key Product Features

Seamlessly monitor SpaceWire traffic between any two devices using two standard SpaceWire cables. SpaceWire Monitor captures and stores full duplex traffic at link speeds up to 200 MHz. and packet sizes up to 512K bytes.

Figure 1: SpaceWire Monitor Connectivity (block diagram)



Product Description

Frequently, it is essential to be able to look at the history of a communication link in order to determine where some unexpected action was triggered. SpaceWire Monitor allows the user to capture both sides of the communication between two SpaceWire nodes. The timing between the nodes being captured is not affected. The decoded data can be stored to system memory, disk, or another device.

SpaceWire-Monitor leverages Dynamic Engineering's previous design experience with the SpaceWire interface series of modular IO. SpaceWire-BK-128RX was chosen as the starting point due to the added memory designed into ports 0 & 1.

SpaceWire SIN/DIN signals are received from each port (0,1) and retransmitted through the opposite port to complete the path for the nodes being captured. In addition, the received data from both ports is decoded and stored to local memory. Local TX functions are disabled – the cross coupled nodes provide the flow control in this configuration.

Each port has 64K of internal FIFO plus 512K of external FIFO. Unlike a standard SpaceWire node the SpaceWire Monitor does not have flow control on the connected SpaceWire ports. All data decoded must be moved to host memory before the local memory fills to prevent overflow.

Each SpaceWire Monitor port has an independent DMA engine to support the application/driver moving data from the on-card storage to the destination of choice. *It is recommended to use a hard drive with cache memory to allow for the burst nature of DMA transfers, for high bandwidth* *applications an NVMe drive is recommended, please refer to the SW section of this manual for further usage details.*

SpaceWire Monitor incorporates logic to allow starting the monitoring process with the link up and already transferring as well as link down situations. The hardware will find the end of the current transfer and start capture with the new data.

Data is time tagged to allow the two port streams to be compared and manipulated after capture.

Product Specifications

Table 3: Product Specifications

Specification	Description
Memory	576 KB data storage/port
Ports	Two ports to capture data stream from both devices
DMA	Separate DMA engines to move data to host memory
Frequency	Up to 200 MHz. auto-frequency Rx
FLASH	Upgradable as new feature released
Temperature	Industrial temperature components
Connectors	Standard MDM connectors and pinout.
IO	SpaceWire specification number: compliant

Construction and Reliability

Dynamic Engineering Modules are conceived and engineered for rugged industrial environments. The SpaceWire family is constructed out of 0.062-inch thick High-Temp RoHS-compliant FR4 material.

RoHS and standard processing are available options.

Through-hole and surface-mount components are used. PMC connectors are rated at 1 Amp per pin, 100 insertion cycles minimum. These connectors make consistent, correct insertion easy and reliable. The PCI and PCIe gold fingers are gold over nickel for high reliability and long-lasting connections. PC104p stacking connectors are mounted in accordance with the manufacturers specifications and using gold plated mounting holes for reliable connections.

PMCs are secured against the carrier with four screws attached to the two standoffs and two locations on the front panel. The four screws provide significant protection against shock, vibration, and incomplete insertion for PMC. ccPMC has additional thermal rail mounting points, which also enhance operation in high-vibration environments. PC104p are stacked and retained with inter-module standoffs and mounting hardware. PCI/PCIe cards are retained with bezel mounting screws.

The PCB provides a (typical based on PMC) low temperature coefficient of 2.17 W/°C for uniform heat. This is based upon the temperature coefficient of the base FR4 material of 0.31 W/m-°C, and taking into account the thickness and area of the board. The coefficient means that if 2.17 Watts are applied uniformly on the component side, the temperature difference between the component side and solder side is one degree Celsius.

The PC104p version of the design has the ground plane tied in with the mounting hardware to allow for inter-stack cooling in conduction cooled environments. Air cooling is a viable method also.

Installation and Interfacing Guidelines

Some general interfacing guidelines are presented below. If you need more assistance, contact Dynamic Engineering. Also refer to the base HW manuals for each SpaceWire board type.

Installation

Warning: Connection of incompatible hardware is likely to cause serious damage.

ESD

Proper ESD handling procedures must be followed when handling the SpaceWire boards. The cards are shipped in anti-static shielded bags. The cards should remain in their bags until ready to use. When installing the card, the installer must be properly grounded and the hardware should be on an anti-static workstation.

Start-Up

Guidelines

Grounds

All electrically grounded equipment should have a fail-safe common ground that is large enough to handle all current loads without affecting noise immunity. Power supplies and power consuming loads should all have their own ground wires back to a common point.

Power Supply

Inputs can be damaged by static discharge or by applying voltage outside of the device-rated voltages.

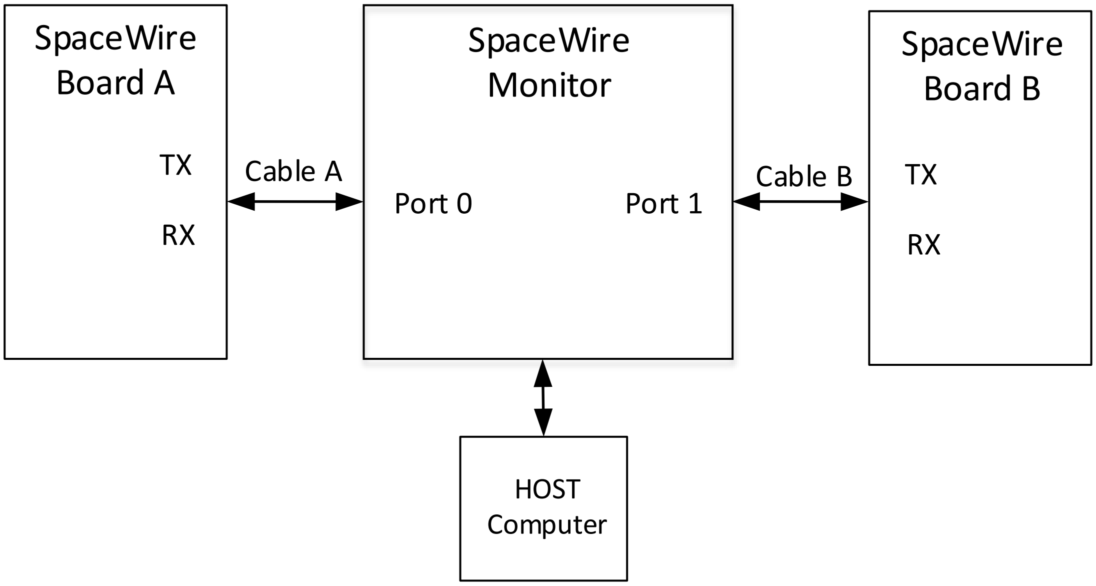
Thermal Considerations

The SpaceWire design consists of CMOS circuits. The power dissipation due to internal circuitry is very low. It is possible to create higher power dissipation with the externally connected logic. If more than one Watt is required to be dissipated due to external loading; forced-air cooling is recommended. With the one-degree differential temperature to the solder side of the board, external cooling is easily accomplished.

Theory of Operation

To use the SpaceWire Monitor, replace the SpaceWire cable between two devices with two cables. Install the Monitor into the appropriate location (PMC, PCI, PCIe, or PCI-104), install the driver, and run the application. Please see the SW section for more information on installing the driver and running the Monitor application.

Figure 2: SpaceWire Monitor Usage Diagram



When the Monitor application & Hardware are in an operational state, the data transferred over the link is captured by the SpaceWire Monitor hardware. (FPGA) In parallel, the SpaceWire driver waits for the FPGA to assert an interrupt indicating data has been captured. The driver initiates a DMA transfer to move the captured data to disc storage. The captured data is logged into output files and is referred to as A-side and B-side data, which is data received (seen) by Port 0 and Port 1 respectively. See the Software Description section later in this manual.

Two capture modes have been designed into the hardware and can be initiated when invoking the application: Link-Down-Start and Link-Up-Start. When Link-Down-Start is used, the Monitor detects when the link comes up and is able to capture the data immediately. When Link-Up-Start is used, the Monitor detects when an EOP occurs, syncs up to the SpaceWire link protocol, and is then able to capture data as it passes through the link.

Address Maps and Register Definitions

This section documents the register addresses, and register bit maps. SpaceWire Monitor register bits are all initialized to ‘0’ on reset.

Because the SpaceWire Monitor channel is a receive (RX) only channel, the majority of transmit (TX) logic that existed in the SpaceWire channel has been disabled. Some logic and state machines in the SpaceWire design contain both RX and TX logic, for that logic, the TX-related signals have been hardwired such that only the RX logic is operational.

All registers, their address decodes, and bits in the SpaceWire design exist in the SpaceWire Monitor design. All TX registers and TX control/status bits in the SpaceWire design exist in the SpaceWire Monitor design but are not used by the driver or the application. Some changes to the functionality, usage, or purpose of several Command and Status register bits have been implemented for each Monitor channel (channels 0 and 1 - Ports 0 and 1 respectively) and are **bolded** in the bit descriptions to indicate where the SpaceWire-Monitor bit definitions differ from standard SpaceWire. For registers and bits in registers that are no longer applicable an NA is appended and the text lightened.

Table 4:SpaceWire Address Map

Register Name	Offset	Description
SPWR_BASE_CNTRL	0x0000	0 Base control register
SPWR_USER_SWITCH	0x0004	User switch & status read port
SPWR_PLL_FIFO	0x0010	4 Write to PLL programming FIFO, Read PLL read-back FIFO
SPWR_PLL_STATUS	0x0014	5 Status associated with PLL programming
SPWR_CHAN_CNTRL_0	0x0050	20 Channel 0 Control register
SPWR_CHAN_STATUS_0	0x0054	21 Channel 0 Status register
SPWR_CHAN_FIFO_0	0x0058	22 Channel 0 RX FIFOs single word access
SPWR_CHAN_RD_DMA_PNTR_0	0x0060	24 Channel 0 read DMA physical PCI address
SPWR_CHAN_RX_FIFO_COUNT_0	0x0060	24 Channel 0 receive FIFO data count
SPWR_CHAN_TX/RX_PKT_LEN_0	0x0064	25 Channel 0 Write TX/Read RX packet-length
SPWR_CHAN_RX_AFL_0	0x006C	27 Channel 0 RX almost full level
SPWR_CHAN_RX_PKT_FF_FULL_CNTL_0	0x0074	29 Channel 0 RX Packet FIFO Full Control register
SPWR_CHAN_MONITOR_CNTL_0	0x0078	30 Channel 0 Monitor Control register
SPWR_CHAN_MONITOR_STATUS_0	0x007C	31 Channel 0 Monitor Status register
SPWR_CHAN_MONITOR_RD_SIZE_0	0x0080	32 Channel 0 Monitor Read Size register
SPWR_CHAN_*_1	0x00A0 to 0x00C4	40 to Channel 1 registers - same as Channel 0 49
SPWR_CHAN_MONITOR_CNTL_1	0x00C8	50 Channel 1 Monitor Control register
SPWR_CHAN_MONITOR_STATUS_1	0x00CC	51 Channel 1 Monitor Status register
SPWR_CHAN_MONITOR_RD_SIZE_1	0x00D0	52 Channel 1 Monitor Read Size register

Register Definitions

SpaceWire Base Control Register

Table 5: SpaceWire Base Control Register

SPWR_BASE_CNTL	
[0x0000] Base Control Register (read/write)	
Data Bit	Description
31	BigEndianDma
30	Spare
27-25	Spare
24	PLL USE ALT
23	PLL CHK
22	PLL RD
21	PLL RST
20	PLL Enable

31-24, 23-16, 15-8, 7-0 ⇔ 7-0, 15-8, 23-16, 31-24-byte swapping pattern implemented.

All bits are active high and are reset on system power-up or reset; except PLL enable, which defaults to enabled (high) on power-up or reset.

BigEndianDma: (Bit 31) ‘0’ disables this option. ‘1’ enables this option. When operating with a BigEndian platform and using PCI accesses, DMA can have challenges. The register accesses directly over the PCI bus are usually automatically taken care of with byte swapping within the CPU or PCI interface on the CPU. DMA data is written-to or read-from the local memory and is not swapped. The direct read/write from memory ends up with scrambled data [relative to SpaceWire little endian definitions]. Setting this bit will byte reverse the data for the DMA path into the Tx and out of the Rx FIFO’s only. Register accesses are not affected.

PLL USE ALT: (Bit 24) When set, selects the Alternate PLL address. 0 → x69, 1 → x6A

PLL_CHK: (Bit 23) Set to check PLL address.

PLL RD: (Bit 22) when set, selects reading the PLL. When cleared, selects writing to the PLL registers.

PLL RST: (Bit 21) When set, ‘1’ causes a reset to the PLL programming hardware.

PLL Enable: (Bit 20) When this bit is set to a one, the signals used to program and read the PLL are enabled.

By default the driver programs the external PLL to provide IO Clock(s) A-D to the SpaceWire Monitor design. IO Clocks provided by the PLL are required for the design to function properly.

SpaceWire User Switch Port

Table 6: SpaceWire User Switch Port

SPWR_USER_SWITCH	
[0x0004] User Switch Port (read only)	
Data Bit	**Description**
31-28	Spare
25	Channel 1 Interrupt Active
24	Channel 0 Interrupt Active
23-20	Xilinx Design Revision Minor
19-16	Xilinx Design Configuration Type
15-8	Xilinx Design Revision Number Major
7-0	Switch Setting

Channel 0-1 Interrupt Active: When a one is read, it indicates that the corresponding channel's interrupt is active. When a zero is read, that interrupt is inactive.

Xilinx Design Configuration Type Major and Minor and Xilinx Design Revision Number: These values describe the channel configuration and revision of the Xilinx design.

Currently there are 3 configurations for the hardware with the following definitions:

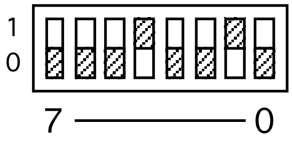
Table 7: SpaceWire Configurations

SpaceWire Model Number	Description
0	Spare
1	S6 with internal 64 Kbyte data FIFOs and two Gbyte maximum packet-lengths for all channels
2	S6 with internal 64 Kbyte data FIFOs and two Gbyte maximum packet-lengths for all channels Plus, external 128Kx32 FIFOs for channel 0 Rx and Tx
3	S6 with internal 64 Kbyte data FIFOs and two Gbyte maximum packet-lengths for all channels Plus, external 128Kx32 FIFOs for channel 0 Rx and Tx
4-F	Spare

NOTE: The Major Revision field is the released name for the particular revision. The Minor Revision field is for Dynamic Engineering revision tracking during development, and for minor released updates between Major Updates. Monitor uses Configuration "3".

Switch 7-0: The user switch is read through this port. The bits are read as the lowest byte. Access the read-only port as a long word and mask off the undefined bits. The DipSwitch positions are defined in the silkscreen. For example, the switch figure below indicates a 0x12.

Figure 3: DipSwitch Silkscreen Position Definition Example



SpaceWire PLL Data FIFO

Table 8: SpaceWire PLL Data FIFO

SPWR_PLL_FIFO	
[0x0010] PLL Data FIFO (read/write)	
Data Bit	**Description**
31-0	Data to PLL or Data from PLL

SpaceWire has an improved I2C interface for programming the PLL. Dynamic Engineering driver support packages include utilities to take the .jed file from the Cypress CyberClocks program, parse and load into the FIFO with the proper sequence of controls via Base Control Register. Please see the reference code for the sequence. Linux, VxWorks, Win7 packages.

The data to program the PLL is written to this address. The hardware has a state-machine to read the data from the FIFO and load into the PLL. Similarly, the state-machine can read the data from the PLL and write it to the read side FIFO.

The IO clock is used in the design and the PLLA, PLLB frequencies must be set for proper operation. Operational SW automatically programs the PLL.

SpaceWire PLL Status Register

Table 9: SpaceWire PLL Status Register

SPWR_PLL_STATUS	
[0x0014] PLL Status (read/write)	
Data Bit	**Description**
31-11	Spare
10	PLL Error Latched
9	PLL Done Latched
8	PLL Ready
7	Spare
6	PLL FIFO RX Data Valid
5	PLL FIFO RX Full
4	PLL FIFO RX Empty
3	Spare
2	PLL FIFO TX Data Valid
1	PLL FIFO TX Full
0	PLL FIFO TX MT

The PLL Status bits are used as feed-back to control the transfer of data to and from the PLL FIFO. TX refers to programming the PLL and RX refers to reading back from the PLL.

The Latched Bits {10,9} are held until cleared by writing back with the bit position(s) set.   Usually these bits are cleared before starting an operation.

PLL Error Latched: (Bit 10) is set when an error is detected in the I2C transfer. The main purpose for this bit is in discovery for the address of the PLL. The Address can be x6A or x69. Once the correct address is known, this bit should be checked but not set. Sticky bit, write with bit position set to clear.

PLL Done Latched: (Bit 9) is set when the transfer is completed. This bit can be polled to know when the PLL has been programmed or when the PLL has been read.

NOTE: The PLL settling time is in addition to the transfer time. Several mS should be delayed after programming the PLL to make sure the specified frequencies are within range. 10 mS is recommended.

<u>PLL FIFO RX Data Valid:</u> (Bit 6) is set when data is Valid in the output port for the PLL read path. Data is pre-read from the FIFO and held in the FIFO holding register. The FIFO can be Empty and still have 1 word left in the holding register if Valid is still set.

<u>PLL FIFO RX FULL:</u> (Bit 5) is set when the read-back FIFO for the PLL is full.

<u>PLL FIFO RX Empty:</u> (Bit 4) is set when the read-back FIFO for the PLL is empty.

<u>PLL FIFO TX Data Valid:</u> (Bit 2) is set when data is valid in the pipeline between the FIFO and the State –Machine. The bit is cleared each time the data is read. During operation, this bit will toggle to provide some indication that the transfer is occurring.

<u>PLL FIFO TX FULL:</u> (Bit 1) is set when the programming FIFO for the PLL is full.

<u>PLL FIFO TX MT</u> (Bit 0) is set when the programming FIFO for the PLL is empty.

SpaceWire Channel Control Register

Table 10: SpaceWire Channel Control Register

SPWR_CHAN_CNTRL_0-1	
[0x0050, 0x00A0] Channel Control Register (read/write)	
Data Bit	**Description**
31	Read DMA Ready (read only)
29-28	Time-Code Flags (read only)
27-26	Spare
25	Return Valid Packet-Lengths Only Enable
23	Receive FIFO Programmable Level Load
21	Read DMA Arbitration Priority Enable
19	Read DMA Interrupt Enable
17	Force Interrupt
16	Master Interrupt Enable
15	**Tick Received Interrupt Enable**
14	Packet Received Interrupt Enable
13	**RX Error Interrupt Enable**
12	RX Almost Full Interrupt Enable
10	Packet Disable
9	Link Auto-Start
8	Link Start
7	**Link Enable**
6	FIFO Write Control
5	**Receive FIFO Reset**
4	**Transmit FIFO Reset**

All bits are active high and are reset on system power-up or reset.

Read DMA Ready: (Bit 31) These two read-only bits report the DMA state-machine status. If they are read as a one, the corresponding DMA state-machine is idle and available to start a transfer. If the bits are read as a zero, the corresponding DMA state-machine is processing a data transfer.

Time-Code Flags: (Bit 29-28) The time-code flags have been moved to the control register to make room for the latched almost empty/full status bits that were added to the status register. These two read-only bits are currently undefined in the SpaceWire specification and will most likely always be seen as zeros.

Return Valid Packet-Lengths Only Enable: (Bit 25) When this bit is set to a one, only valid packet-lengths will be returned. If no new packet has been received since the packet-length FIFO was read, the packet-length will be returned as zero. When a zero is written to this bit and no new packet has been received since the packet-length FIFO was read, the packet-length from the last packet received will be returned. When enabled, this control allows packet-lengths to be confidently read without first checking the Receive Packet Length Valid status bit in the channel status register. This control bit was added starting with rev. J.

Receive FIFO Programmable Level Load: (Bit 23) These bits are only valid for channels with external data FIFOs. The load bits must be active during FIFO reset to select the programmable level feature. Once selected, these bits must be set to zero for normal FIFO operation. When set to one, data accesses are instead directed to the almost empty and almost full level registers (See FIFO data sheet for details).

Read DMA Arbitration Priority Enable: (Bit 21) These two bits, when set to one, enable the DMA arbiter to use the TX almost empty and/or RX almost full status to give priority to a channel that is approaching the limits of its FIFOs. The levels written to the TX almost empty and RX almost full registers are used to determine these status values. When these bits are zero, normal round-robin arbitration is used to determine access to the PCI bus for DMA transfers.

Read DMA Interrupt Enable: (Bit 19) These two bits, when set to a one, enable the interrupts for DMA write and read completion for the referenced channel. These two interrupts cannot be disabled by the master interrupt enable.

Force Interrupt: (Bit 17) When this bit is set to one, a system interrupt will occur provided the channel master interrupt enable is set. This is useful to test interrupt service routines.

Master Interrupt Enable: (Bit 16) When this bit is set to a one, all enabled interrupts for the referenced channel (except the DMA interrupts) will be gated through to the PCI host; when this bit is a zero, the interrupts can be used for status without interrupting the host.

Tick Received Interrupt Enable: Channel Control Register bit (Bit 15): This bit is now hard-wired to ‘0’ as generating interrupts from receiving time codes is not relevant to the SpaceWire Monitor.

Packet Received Interrupt Enable: (Bit 14) When this bit is set to a one, an interrupt will be generated when a complete packet is received, provided the channel master interrupt enable is asserted. When a zero is written to this bit, an interrupt will not be generated, but the latched status can still be read from the channel status register.

RX Error Interrupt Enable: Channel Control Register (Bit 13): This bit is still read/writeable but inside the SpaceWire Monitor design, RX Errors are blocked from generating interrupts. Sources of

RX interrupts are bits [12:8] in the SPWR_CHAN_STATUS register and include bits such as Receive FIFO Overflow, which is now handled by the Monitor Status bits, and Credit Error Detected, which would be detected by either of the devices being monitored.

RX Almost Full Interrupt Enable: (Bit 12) When this bit is set to a one, an interrupt will be generated when the receive FIFO level becomes equal or greater to the value specified in the SPWR_CHAN_RX_AFL register, provided the channel master interrupt enable is asserted. When this bit is zero, an interrupt will not be generated, but the status can still be read from the channel status register.

Packet Disable: (Bit 10) When this bit is set to a one, data is transferred without being separated into packets. No end-of-packet characters are generated or received and the packet-length FIFOs are not used. As soon as data is written to the transmit FIFO it will be sent out, provided all other conditions allow this. When this bit is zero, the data will be sent in packets. Data must be written to the transmit data FIFO and packet lengths must be written to the TX packet-length FIFO, before data can be transferred.

Link Auto-Start: (Bit 9) The behavior of this bit is similar to Link start, however, when this bit is set and Link start is not set, the state-machine will not proceed to the Started state unless a Null character has been seen, which indicates that the other node is attempting to establish a connection. This bit allows the connection process to be cleanly initiated from one side of the link only.

Link Start: (Bit 8) When this bit is set to a one, the link state-machine will move from the Ready state to the Started state and will attempt to establish a connection with another node. When this bit is zero, the state-machine will remain in the Ready state, provided it has already achieved this state. Once the state-machine has left the Ready state, this bit has no effect.

Link Enable: Channel Control Register (Bit 7): This bit is still read/writable but inside the SpaceWire Monitor, it has been replaced by the decoded monitor mode bits [1:0] in the SPWR_CHAN_MONITOR_CONTROL register.

FIFO Write Control: (Bit 6) When this bit is set to a one, any data written to the FIFO will be written to the receive FIFO. This allows for fully testing the data FIFO path without connecting to another SpaceWire node. When this bit is zero, normal operation is enabled.

Transmit/Receive FIFO Reset: Channel Control Register: (Bit 4 and 5) These bits are active when the Monitor is disabled. When the Monitor is enabled, FIFO resets asserted by the driver in response to the monitored link going down/up are ignored and allow the Monitor to continue capturing data once the link is re-established.

SpaceWire Channel Status Register

Table 11: SpaceWire Channel Status Register

SPWR_CHAN_STATUS_0-1	
[0x0054, 0x00A4] (status read/latch clear write)	
Data Bit	**Description**
31	Latched Receive FIFO Almost Full
29-24	Time-Code Data
23	Interrupt Active
22	Receive Packet Length Valid
20	**SpaceWire Link Established**
19	Read DMA Error
17	Read DMA List Complete
15	**TICK_OUT Received**
14	Packet Received
13	**Receive Error**
12	**Receive FIFO Overflow**
11	**Credit Error Detected**
10	**Escape Error Detected**
9	**Disconnect Error Detected**
8	**Parity Error Detected**
7	Receive Data Valid
6	Receive FIFO Full
5	Receive FIFO Almost Full
4	Receive FIFO Empty

Latched Receive FIFO Almost Full: (Bit 31) When a one is read, it indicates that the receive FIFO data count has become greater than or equal to the value in the SPWR_CHAN_RX_AFL register. A zero indicates that the FIFO has not become almost full. This bit is latched and can be cleared by writing back to the Status register with a one in this bit position.

Time-Code Data: The last time-code value received can be read from this six-bit data-field. The TICK_OUT received status bit will indicate if the data is a new valid time-code value. A time-code is considered valid if it is one more than the previous stored value. If the time-code is the same as the stored value, it is assumed to be a duplicate resulting from a cycle in the SpaceWire network and is therefore ignored. If the time-code meets neither of these conditions, it is stored, but the TICK_OUT signal is not asserted until the next time-code is received and is one more than that last stored value. At this point, the time-code is deemed to be re-synchronized.

Interrupt Active: When a one is read, it indicates that an enabled interrupt condition (other than the DMA interrupts) is active for the referenced channel. A zero indicates that no enabled interrupt condition is active.

Receive Packet Length Valid: When a one is read, there is at least one valid receive packet-length value available. When this bit is a zero, it indicates that there are no valid receive packet-length values.

SpaceWire Link Established: Channel Status Register bit [20] – In a SpaceWire design, this bit indicates a link has been established. In the SpaceWire Monitor design, it indicates the link-up, active, and being monitored.

Read DMA Error: When a one is read, it indicates that an error has occurred while the corresponding DMA was in progress. This could be a target or master abort or an incorrect direction bit in one of the DMA descriptors. These bits are latched and must be cleared by writing the same bit back to the channel status port. A zero indicates that no DMA error has occurred.

Read DMA List Complete: When a one is read, it indicates that the corresponding DMA has completed. These bits are latched and must be cleared by writing the same bit back to the channel status port. A zero indicates that the corresponding DMA has not completed.

Tick-Out Received: Channel Status Register bit [15] – This bit is now hard-wired to '0' as TICK_OUT Received interrupts are not processed; however, the SpaceWire Monitor does decode and provide the Time-Code value(s) received in the Time-Code Data field (Channel Status Register bits [29:24]).

Packet Received: When a one is read, it indicates that a packet has been received since this bit was last cleared. This bit is latched and must be cleared by writing the same bit back to the channel status port. A zero indicates that a packet has not been received.

Receive Error: Channel Status Register bit [13] – This bit is now hard-wired to '0' as the five error sources for it (Channel Status Register bits [12:8]) are all hard-wired to '0'.

Receive FIFO Overflow: Channel Status Register bit [12] – This bit is now hard-wired to '0' as Receive FIFO Overflows are handled using the SPWR_CHAN_MONITOR_STATUS register.

Credit Error Detected: Channel Status Register bit [11] – this bit is now hard-wired to '0' as Credit Error detection is handled by the devices being monitored.

Escape Error Detected: Channel Status Register bit [10] – This bit is now hard-wired to '0'. Disabling this Escape Error Detection allows the monitor driver to continue running and capturing data when/if the monitored link goes down and comes back up.

Disconnect Error Detected: Channel Status Register bit [9] – This bit is now hard-wired to '0'. Disabling this Disconnect Error Detection allows the monitor driver to continue running and capturing data when/if the monitored link goes down and comes back up.

Parity Error Detected: Channel Status Register bit [8] – This bit is now hard-wired to '0' as the monitor logic/driver/application is not currently configured to test/log/report parity errors and it is expected the detection of parity errors will be handled by the devices being monitored.

Receive Data Valid: When a one is read, there is at least one word of valid receive data. When data is written to the receive FIFO, the first four words are read and held in batches to be ready for a PCI read DMA or single-word read. Therefore, although the FIFO is empty if this bit is set, there are as many as four additional long-words of receive data. A zero indicates that there is no valid receive data.

Receive FIFO Full: When a one is read, the receive data FIFO for the corresponding channel is full; when a zero is read, there is room for at least one more data-word in the FIFO.

Receive FIFO Almost Full: When a one is read, the number of data-words in the receive data FIFO for the corresponding channel is greater or equal to the value written to the SPWR_CHAN_RX_AFL register for that channel; when a zero is read, the level is less than that value.

Receive FIFO Empty: When a one is read, the receive data FIFO for the corresponding channel contains no data; when a zero is read, there is at least one data-word in the FIFO.

SpaceWire Channel Read DMA Pointer Port

Table 12: SpaceWire Channel Read DMA Pointer Port

SPWR_CHAN_RD_DMA_PNTR_0-1	
[0x0060, 0x00B0] (write only)	
Data Bit	**Description**
31-0	First Chaining Descriptor Physical Address

This write-only port is used to initiate a scatter-gather read [RX] DMA. When the address of the first chaining descriptor is written to this port, the DMA engine reads three successive long words beginning at that address. Essentially, this data acts like a chaining descriptor value pointing to the next value in the chain.

The 1st read retrieve the address of the first memory block (DMA buffer) to write to, the 2nd is the length in bytes of that block, and 3rd is the address of the next chaining descriptor in the linked list (buffer memory blocks). This process is continued until the end-of-chain bit of the next pointer value read indicates it is the last chaining descriptor in the list.

All three values are on LW boundaries and LW in size. Addresses for successive parameters are incremented. The addresses are physical addresses the HW will use on the PCI bus to access the Host memory. In most operating systems you will need to convert from virtual to physical. The length parameter is a number of bytes, and must be on a LW divisible number of bytes.

Status for the DMA activity can be found in the channel control register and channel status register.

NOTES:

1. Writing a zero to this port will abort a write DMA in progress.
2. End of chain should not be set for the address written to the DMA Pointer Address Register. End of chain should be set when the descriptor follows the last length parameter.
3. The Direction should be set to ‘1’ for Burst Out DMA in all chaining descriptor locations.
4. The segment size is 31-0 however the byte count is shifted down two positions to create a LW count in memory. This means the actual count is 1G-1 [4Gbytes minus 1 LW] or xFFFFFFFC for the max byte count.

NOTE: The direction bit (bit 1) must be set when the physical address of the first chaining descriptor is written to this register or read DMA error will result.

SpaceWire Channel RX FIFO Data Count Port

Table 13: SpaceWire Channel RX FIFO Data Count Port

SPWR_CHAN_RX_FIFO_COUNT_0-1	
[0x0060, 0x00B0] RX FIFO Data Count (read only)	
Data Bit	**Description**
Channels with External RX Data FIFOs	
31-20	Number of RX Packet-length Values Stored
19-0	Number of RX Data-Words Stored
Channels with Internal RX Data FIFOs	
31-20	Number of RX Packet-Length Values Stored
19-14	Spare
13-0	Number of RX Data-Words Stored

These read-only register ports report the number of 32-bit data words in the corresponding receive FIFO. There are four additional latches in the read DMA pipeline that may contain data, which allows this value to be a maximum of 0x4003 for channels with internal receive data FIFO's or 0x24002 for a channel with an external receive data FIFO.
NOTE: the counts are LW based. x4 for byte counts.

The receive packet-length count field is also in this register. This allows the user to know how many packet-length values are stored in the packet-length FIFO. The first receive packet-length written to the packet-length FIFO is read as soon as it is written to be immediately available. This causes the receive packet-length valid status bit to be set and the packet-length count to become one. As subsequent packets are received, the count will increment and as lengths are read, the count will decrement.

SpaceWire RX Packet-Length FIFO Ports

Table 14: SpaceWire RX Packet-Length FIFO Ports

SPWR_CHAN_RX_PKT_LEN_0-1	
[0x0064, 0x00B4] RX Packet-Length FIFO Ports (write only)	
Channels with 2 Gbyte Maximum Packet-Lengths	
Data Bit	**Description**
31	Terminate Packet with an Error-End-of-Packet
30-0	Packet Length (31 bits)

These ports access the write transmit packet-length FIFO and the read receive packet-length FIFO ports for the respective channels. These FIFOs are used to store packet lengths for sending transmit packets and reading received packets.

The bit above the packet-length field is an error flag. If this bit is read as a one, it indicates that either the packet was terminated with an EEP (Error-End-of-Packet) or an error condition occurred while the packet was being received. If this bit is written as a one, it indicates that the transmit packet should be terminated with an EEP rather than an EOP.

SpaceWire Channel RX Almost Full Level Register

Table 15: SpaceWire Channel RX Almost Full Level Register

SPWR_CHAN_RX_AFL_0-1	
[0x006C. 0x00BC] RX Almost Full Level (read/write)	
Data Bit	**Description**
31-0	RX FIFO Almost Full Level

These read/write ports access the receiver almost-full level registers for the respective channels. When the number of data words in the receive data FIFO is equal or greater than this value, the almost full status bit is set and an interrupt may be generated if it is enabled. While the port is defined larger than the FIFO size in LW, the comparison should be set to a value within the range of the FIFO for proper operation.

SpaceWire Channel RX Packet FIFO Full Control Register

Table 16: SpaceWire Channel RX Packet FIFO Full Control Register

SPWR_CHAN_RX_PKT_FF_FULL_CNTRL_0-1	
[0x0074, 0x00C4] RX Packet FIFO Full Control Register (read/write)	
Data Bit	**Description**
31-10	Spare
9-0	RX Packet FIFO Full Level

These bits are set to 0x3DF on system power-up or reset.

RX Packet FIFO Full Level: These bits set the RX Packet FIFO Full Threshold level. By default, these bits are set to 0x3DF. The RX Packet FIFO has 1024 locations (0x3FF) therefore, by default, the logic will see the FIFO as full once 960 or more locations are occupied. Once the threshold is reached (or passed), the logic will back pressure the SpaceWire link by not providing additional FCT's until the number of FIFO locations drops below the threshold.

Per the SpaceWire specification the maximum outstanding FCT's is 56. If single byte transfers are being used and the maximum FCT's are outstanding when the FIFO goes full, 56 additional packets maybe received; 0x3DF was chosen to be the default value to cover this case.

Setting this register to either 0x3FF or 0x000 puts this logic in legacy mode. Other values can be used to retain protection and optimize for your system based on the minimum packet size [vs. FCT's].

Table 17: SpaceWire Monitor Control Register

SPWR_CHAN_MONITOR_CONTROL	
[0x0078, 0x00C8] Channel Monitor Control Register (read/write)	
SPWR Monitor Control Register	
Data Bit	**Description**
31-2	Spare
1-0	Monitor Mode

Monitor Mode: these two bits control the Monitor Mode of operation for the channel.

Monitor Mode	Name	Definition
"00"	Disabled:	Monitor is disabled
"01"	Link-Down-Start:	Configures Monitor to capture data after link comes up
"10"	Link-Up-Start:	Configures Monitor to sync to an active/running link then capture data
"11"	Reserved:	Same as "01" Link-Down-Start

Table 18: SpaceWire Monitor Status Register

SPWR_CHAN_MONITOR_STATUS	
[0x007C, 0x00CC] Channel Monitor Status Register (read/write)	
SPWR Monitor Status Register	
Data Bit	**Description**
31-16	Packet Overflow Count
15-1	Spare
0	Monitor Pause Activated

Monitor Pause Activated: This write '1' to clear status bit is set to '1' when the number of packets stored in the Packet FIFO reaches or exceeds 4,063 (0xFDF).

Packet Overflow Count: These bits indicate the number of packets that were not captured and stored in the Packet FIFO.

Under normal operation conditions, the Monitor Pause Activated should not be set and the Packet Overflow Count should remain 0x0000. The Monitor Pause Activated status bit is set when the internal pause monitor (pause_mon) signal is asserted and the hardware attempts to write the packet FIFO. The pause monitor signal is asserted when the Packet FIFO reaches or exceeds 4,063 (0xFDF). Once the Monitor Pause Activated status bit is set, it stays set until a logic reset of the hardware occurs or a '1' is written to bit [0] of the SPWR Monitor Status register.

The Packet Overflow Count increments each time the hardware attempts to write the packet FIFO while the pause monitor signal is asserted. The Packet Overflow Counter is cleared by disabling the Monitor (Monitor Mode = 00) or by a logic reset.

Table 19: SpaceWire Monitor Read Length Size Register

SPWR_CHAN_MONITOR_RD_SIZE	
[0x0080, 0x00D0] Channel Monitor Read Size Register (read/write)	
SPWR Monitor Read Size Register	
Data Bit	**Description**
31	Spare
30-0	Packet Read Length Size

Packet Read Length Size: This register displays the output of the Received Packet FIFO, which provides the received packet read length size for the current packet being processed or last packet processed and written out to external storage using DMA.

Warranty and Repair

Please refer to the warranty page on our website for the warranty and options that are currently offered.

www.dyneng.com/warranty

Service Policy

Before returning a product for repair, verify to the best of your ability, that the suspected unit is as fault. Then call the Dynamic Engineering Customer Service Department for a Return Material Authorization (RMA) number. Carefully package the product, in the original packaging if possible, and ship prepaid and insured with the RMA number clearly written on the outside of the package. Include a return address and the telephone number of a technical contact. For out-of-warranty repairs, a purchase order for repair charges must accompany the return. Dynamic Engineering will not be responsible for damages due to improper packaging of returned items. For service on Dynamic Engineering products not purchased directly from Dynamic Engineering, contact your reseller. Products returned to Dynamic Engineering for repair by anyone other than the original customer will be treated as out-of-warranty.

Out-of-Warranty Repairs

Out-of-warranty repairs will be billed on a material and labor basis. Customer approval will be obtained before repairing any item if the repair charges will exceed one half of the list price for one of that kind of unit. Return transportation and insurance will be billed as part of the repair in addition to the minimum RMA charge.

Contact:

Customer Service Department
Dynamic Engineering
150 DuBois St. Suite B/C
Santa Cruz, CA 95005
(831) 457-8891
support@dyneng.com

Ordering Information

Industrial Temperature Rated Components: -40 - 85°C

Table 20: Ordering Information

Product	Description
	SpaceWire Monitor: combination of Interface Module and Software. Capture data from 2 links cross connected through Monitor. DMA transfer to host. Up to 200 MHz link rate. Up to 512K Byte packet size. Continuous capture with up to 256K byte packets.
	Models: add as prefix PCI, PCIe, PMC, PCI-104
	"PCIe-SpaceWire-Monitor"
	Options: SpaceWire Cables type AL – 50-2004-0801-XX.YY.ZZ XX.YY.ZZ Major.Minor.Units SpaceWire Cables type A – 50-2020-0801-XX.YY.ZZ XX.YY.ZZ Major.Minor.Units -ROHS, switch to ROHS soldering. Standard is Non-ROHS. -CC Conformal Coating

Glossary

Baud	Used as the bit period when talking about UARTs; Not strictly correct, but is the common usage when talking about UARTs.
CardID	Unique number assigned to a design to distinguish between all designs of a particular vendor
CFM	Cubic feet per minute
FIFO	First In First Out memory
Flash	Non-volatile memory used on Dynamic Engineering boards to store FPGA configurations or BIOS
JTAG	Joint Test Action Group – a standard used to control serial data transfer for test and programming operations.
LFM	Linear feet per minute
LVDS	Low Voltage Differential Signaling
MUX	Multiplexor – multiple signals multiplexed to one with a selection mechanism to control which path is active.
Packed	When UART characters are always sent/received in groups of four, allowing full use of host bus/FIFO bandwidth.
Packet	Group of characters transferred. When the characteristics of the group of characters is known, the data can be stored in packets and transferred as such; the system is optimized as a result. Any number of characters can be transferred.
PCI	Peripheral Component Interconnect – parallel bus from host to this device
PIM	PMC Interface Module (PIM). Provides rear I/O in cPCI systems. Mounts to PIM Carrier
PIM Carrier	PIM Mounting Device. Mounts on rear of cPCI backplane.
PMC	PCI Mezzanine Card – establishes common connectors, connections, size and other mechanical features.
TAP	Test Access Port – basically a multi-state port that can be controlled with JTAG [TMS, TDI, TDO, TCK]. The TAP States are the states in the State Machine that are controlled by the commands received over the JTAG link.
TCK	Test Clock provides synchronization for the TDI, TDO, and TMS signals

TDI Test Data in – this serial line provides the data input to the device controlled by the TMS commands. For example, the data to program the FLASH comes on the TDI line while the commands to the state machine to move through the necessary states comes over TMS. Rising edge of TCK valid.

TDO Test Data Out is the shifted data out. Valid on the falling edge of the TCK. Not all states output data.

TMS Test Mode State – this serial line provides the state switching controls. ‘1’ indicates to move to the next state, ‘0’ means stay put in cases where delays can happen; otherwise, 0,2 are used to choose which branch to take. Due to the complexity of state manipulation, the instructions are usually precompiled. Rising edge of TCK valid.

UART Universal Asynchronous Receiver Transmitter. Common serialized data transfer with start bit, stop bit, optional parity, optional 7/8 bit data. Can be over any electrical interface. RS232 and RS422 are most common.

Unpacked When UART characters are sent on an unknown basis requiring single character storage and transfer over the host bus

VendorID Manufacturers number for PCI/PCIe boards. DCBA is Dynamic Engineering’s VendorID

VME Versa Module European

VPX Family of standards based on the VITA 46.0

XMC Switched mezzanine card (PMC with PCIe)