User Manual

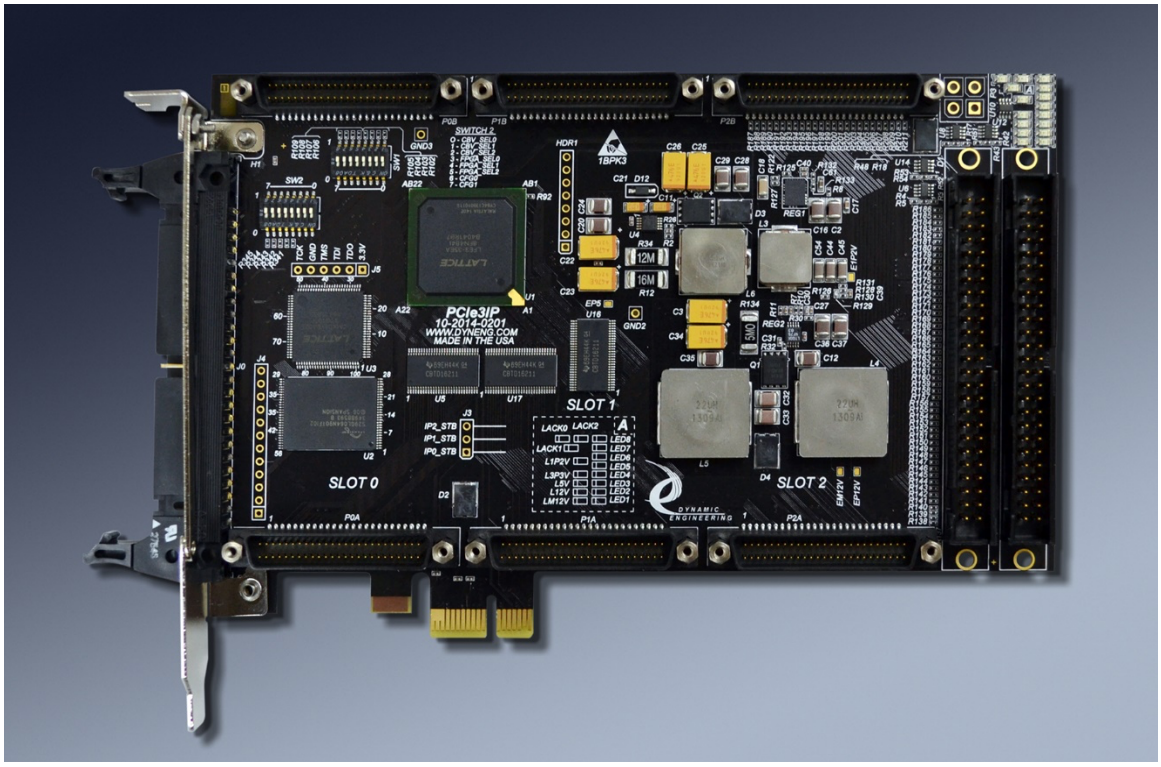
PCIeIP Carrier Series

PCI Express X1 to Industry Pack (IP) Bridge

Manual Revision 01p05
Revision Date 7/18/22

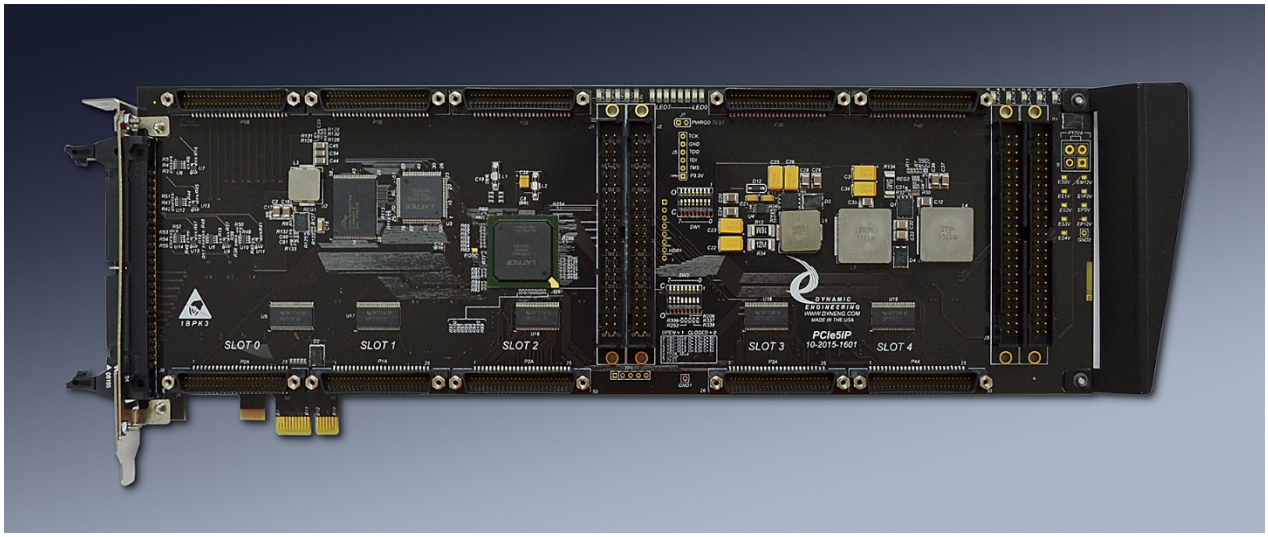
Dynamic Engineering
150 DuBois St. Suite B/C
Santa Cruz, CA 95060
(831) 457-8891
www.dyneng.com
sales@dyneng.com
Est. 1988

PCIe3IP

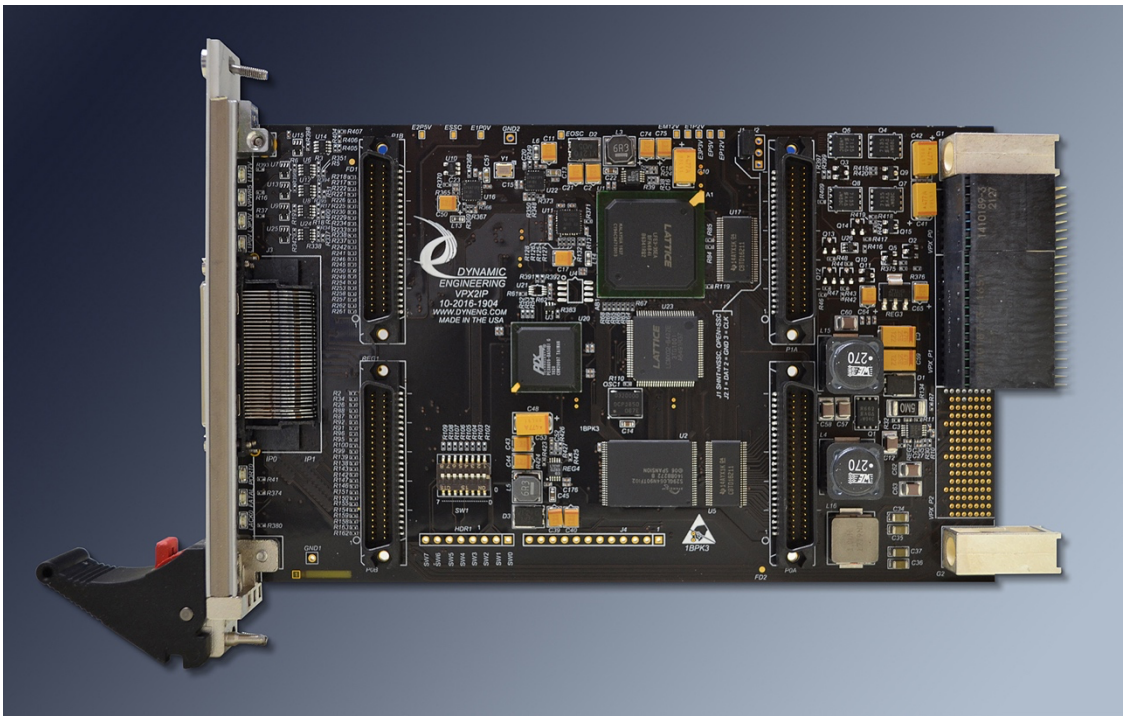


Corresponding Hardware: 10-2014-02(02-04)

PCIe5IP



Corresponding Hardware: 10-2015-020(02-03)



Corresponding Hardware: 10-2016-19(03-04)

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

Figures

Tables

Design Revision History

Table 1: Design Revision History

Revision	Date	Description
		PCIe3IP
A	06/24/2014	Released new design based on previous IP carriers
B	12/08/2014	Corrected voltage on CPLD, misc. silk issues cleaned up
C	12/02/2015	Mechanical clearance clean-up Added 3.3V EPAD
04(D)	02/27/2019	Added VPWR select and 5/3V terminations, reference to IPs
		PCIe5IP
A	06/30/2015	New design based on PCIe3IP Rev B
02p1	02/16/2017	Corrected minor silk issues, reversed LED order
02p2	02/27/2019	Added VPWR select and 5/3V terminations, reference to IPs
		VPX2IP
A	05/31/2015	Ported from PCIe3IP rev B
02p1	04/01/2019	Updated clock reference circuit for better jitter performance Added VPWR option Switched to VHDCI connectors
03, 04	3/11/22	Add switch to support fixed and spread clocking options

Manual Revision History

Table 2: Manual Revision History

Revision	Date	Description
NOTE: Revisions released prior to August 2020 may have incomplete data		
A1p3	12/01/2016	Combined PCIe3IP, PCIe5IP, VPX2IP, and VPX4IP manuals
01p04	08/17/2020	Updated formatting
01p05	3/11/22	Update VPX2IP data

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.

Product Description

The PCIeIP Carrier Series is part of Dynamic Engineering's IP Compatible family of modular I/O components. The PCIeIP Carrier Series uses a single PCI Express (PCIe) slot or lane (VPX). Products in the PCIeIP Carrier Series covered by this manual include:

- PCIe3IP: a half-length card providing three IndustryPack (IP) compatible sites.
- PCIe5IP: a full-length card providing five IP-compatible sites.
- VPX2IP: a 3U 4HP with bezel or rear I/O card providing two IP-compatible sites.

IndustryPack ID, IO, INT, and MEM access types are supported for read and write cycles. The full 8 Mbytes of address space is allocated to each of the MEM spaces.

Low impedance Quick Switch devices provide a signaling bridge between the 3.3V FPGA signaling and 5V IP signaling environments. A user switch setting can select the IP bus to operate at 3.3V. Users can take advantage of the 3.3V signaling by removing their level shifting devices. Module positions are independent for reference voltage selection.

The PCI Express link is a by one (x1) link that is fully compliant to PCI Express 1.1 revision of the PCI-SIG specification and as such can operate in any compliant PCIe Gen1, Gen2, Gen3, or Gen4 slot. Each IP position supports 8- & 16-bit IP devices and is fully compliant to the Vita 4-1995 specification.

Each IndustryPack module position has a completely separate IP control bus connection to the FPGA. Within the FPGA, each of the IP control buses are separately controlled. Packets received from the PCIe link are routed through steering logic to the proper control bus interface. Within each control interface, FIFOs are used to store multiple packets. With this architecture, multiple IndustryPack data transfers can be queued for execution as the IP is ready. One advantage of this architecture is that a slower IP does not hold up a faster IP from being serviced. Also, a loop to load a FIFO or fill RAM etc. can potentially be completely adsorbed by PCIeIP allowing the CPU to move onto other tasks much faster.

Since some situations require deterministic execution and coordination between installed IndustryPack modules, design features are provided to control the execution of commands in a specific sequence between multiple IP modules. Because all instructions are executed in order, this mode is not needed for single IP deterministic operation.
Per the PCIe specification, every access is at least a one long word. Byte, word, and 3-byte accesses are supported utilizing PCIe byte enables. Any combination of byte enables and starting address as defined by the PCIe specification is supported. The PCIeIP supports PCIe transfer sizes of 1 and 2 long words (2Lwords = 1 quad word).

PCIe accesses are automatically converted into IP accesses, and it may range from a single IP access up to 4 back-to-back IP accesses with the IP address incrementing between cycles unless the address increment disable function is selected. For a read, the IP read data is assembled, and a PCIe read completion packet is returned. The automatic generation of IP accesses greatly enhances the overall throughput when transfers are > 2 bytes. Additionally, based on the PCIe byte enables, the PCIeIP determines when only a single 16-bit IP access needs to be performed for word or byte transfers. In all cases, the appropriate IP byte lane enables are applied as necessary.

Each IP clock is independently programmable for 8 or 32 MHz operation via a bit in its control register. By default, each IP CLK is 8 MHz after power up and/or reset. The clock frequency may be changed at any time without consequence. Regardless of the frequency of each clock, the IP clock outputs are designed to be phase stepped in relation with one another to reduce simultaneous switching noise. For the PCIe3IP and PCIe5IP, the rising edge of IP1 clock is 8 ns and IP2 clock is 16 ns after the rising edge of IP0's clock. The PCIe5IP IP3/4's clock is in phase with IP0/1's clock. For the VPX2IP, the rising edge of IP1 clock is 8 ns after the rising edge of IP0's clock.

In normal operation IP access latency and performance is substantially better and the IP logic runs 4 times faster when the IP CLK is 32 MHz versus 8 MHz.

PCIeIP has a programmable watchdog timer function, which completes the IP access if the IP does not respond within the required amount of clock cycles. The watchdog timer has a status bit and an optional Bus Error interrupt output.

PCIeIP supports interrupts from each IP slot with separate mask bits. Two interrupts from each IP slot are supported. An interrupt force bit is available to aid in software development in addition to the IP required 5V Power Good interrupt. All the interrupts are maskable. The masked interrupt output signals are tied together, and if asserted, they will generate either MSI or INTA#.

PCIeIP has several programmable interrupt features to control when an interrupt is generated. Programmable bits select behavior such as edge or level, or aggregation timer values to pace the rate at which interrupts are generated (see Interrupt section for details).

5 LED's are used with 5 independent voltage monitoring circuits to accurately detect if any of the board's voltages are out of range. One LED for each power monitoring circuit, when the LED is on the voltage is in range, if the LED is off the voltage is out of range. Eight user controllable LED's are supplied. Each LED is programmable with one of sixteen possible sources to provide a variety of status. The 4-bit LED select field in the Switch and LED control register is used to determine each LED's meaning. One of the selections allows the user to directly control the LED's. The default selection uses 4 LEDs to reflect PCIe link status. When all four of the lower four LEDs are on, they indicate a working link.

Table 3: LEDs

	Board Name		
	VPX2IP	**PCIe3IP**	**PCIe5IP**
Number of LEDs	15	16	18
LED That Indicates ACK* Activity	2 (on the 2 channel)	3 (on the 3 channel)	5 (on the 5 channel)

Two 8-bit DipSwitches are provided on the PCIeIP. One 8-bit DipSwitch is for user configuration and is readable via the Switch and LED control register. The other 8-bit DipSwitch is for board configuration and test purposes (see Board Features section for details).

Power-on PCIe PERST# reset is used to reset the entire PCIeIP. Each IP Reset* is asserted as long as PERST# is asserted. Once PERST# is de-asserted each IP's clock starts toggling, and each IP Reset* will remain asserted until a 256 ms timer expires. Once the timer expires, IP Reset* de-asserts synchronously with that IP's CLK. Two separate control register reset bits are provided for each IP. One only asserts IP Reset*, and one asserts IP Reset* and resets that PCIeIP's IP channel/FIFO.

To meet the PCI Express specification requirement for PCIe core initialization within 100 mS from PERST# de-assertion, PCIeIP implements a 16-bit wide 90 ns Flash in conjunction with a CPLD to configure the FPGA via its parallel configuration port. With this architecture, PCIeIP beats this aggressive specification by a comfortable margin.

Regarding the PCIe3IP and the PCIe5IP, the I/Os for each IP are brought out to their own 50 pin headers. For the VPX2IP, stuffing options route the I/O to either the Condo header or the VPX rear connector. All I/O signals for each PCIeIP board are routed carefully with matched length and impedance control. Differential routing techniques are used to support operation with LVDS, RS485 and other differential electrical standards as well as single ended systems – analog, TTL I/O etc. **NOTE:** see the pin-out tables later in this manual for the mapping of IP I/O to header.

For the PCIe3IP/PCIe5IP, the 50-pin header in the first position is mounted [right angle header] to be accessible through the bezel. The second, third, (PCIe3IP) fourth, and fifth positions (PCIe5IP) have traditional vertical headers.

The bezel for the PCIe3IP is a special design with accommodation for the right-angle header and cable routing for the other two. All of the I/O can come through the bezel without wasting another I/O position. The bezel incorporates an arm with hinge to allow the side of the bezel to be rotated out of the way to aide in threading the rear I/O through the bezel. Install the I/O into the rear headers, lift the bezel arm, place the cabling onto the bezel, rotate the arm back into position, and mount into the system.

PCIeIP conforms to the VITA standard for IndustryPack Carriers. This guarantees compatibility with multiple IndustryPack compatible modules.

Dynamic Engineering provides Windows, Linux, and VxWorks drivers for the PCIeIP. The drivers detect the carrier card and communicate with the OS to get the memory, interrupts etc. assigned to the installed carrier. The driver interrogates the IP positions on the carrier and when an IP is located, determines if a corresponding driver is installed. If not, the IP-Generic driver is installed to allow any third-party IP module to be used with the carrier. When a recognized module is detected the driver for that module is installed automatically.

The PCIeIP Architecture is the foundation for all the devices in the PCIeIP Carrier Series. The PCIe3IP Block Diagram (Figure 1) and PCIe3IP FPGA Block Diagram (Figure 2) illustrate a 3IP channel design using the PCIeIP Architecture. To create the PCIe5IP two additional IP channels (IP3 & IP4 – not shown) are implemented. To create the VPX2IP the IP2 channel is removed.

Figure 1: PCIe3IP Block Diagram

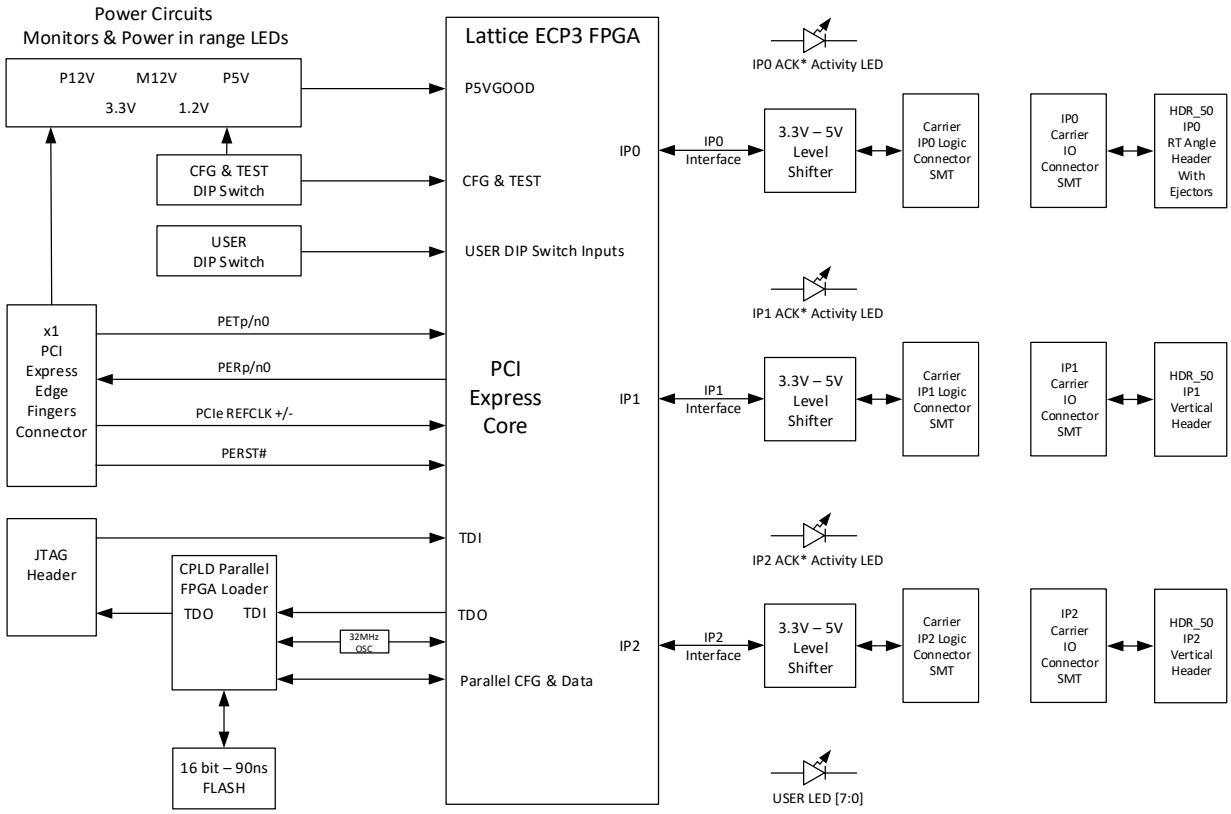
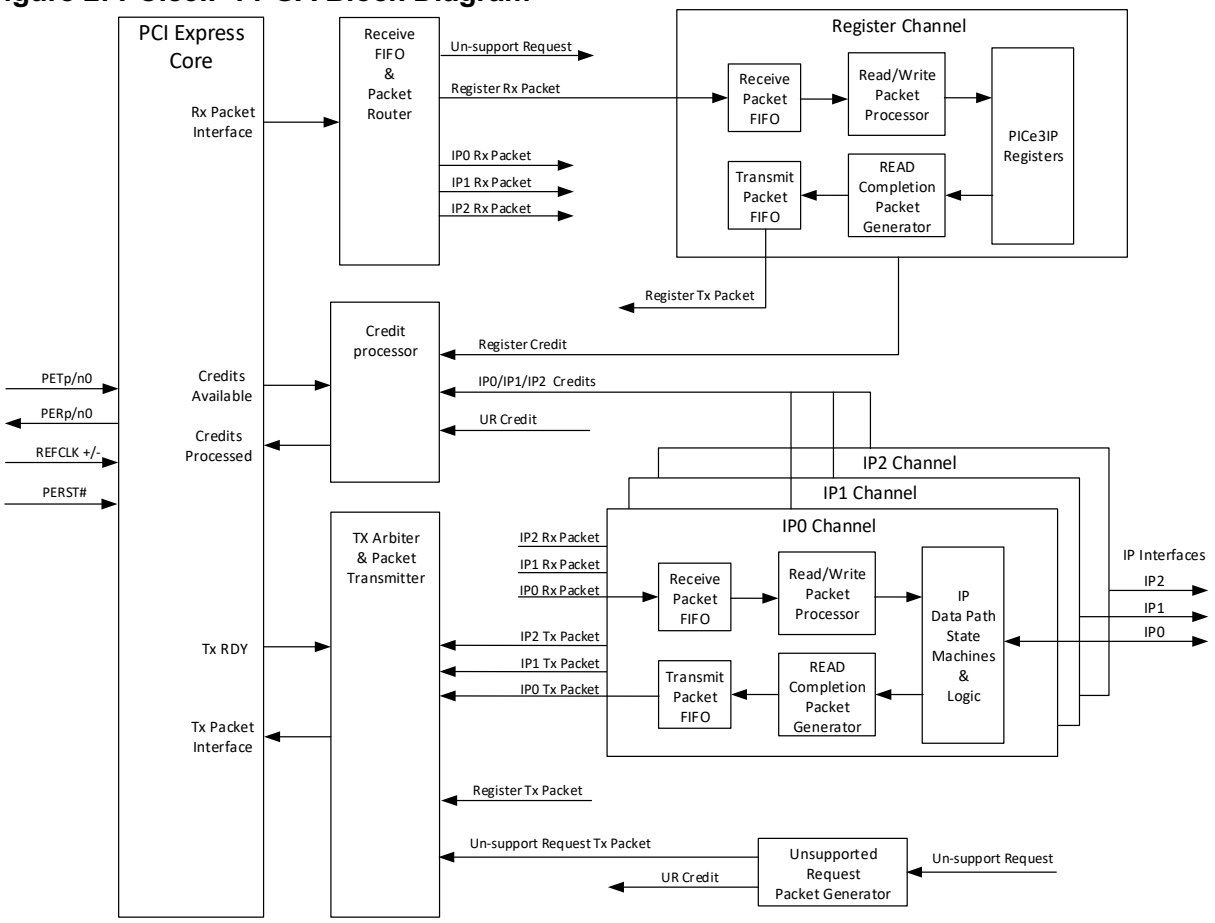


Figure 2: PCIe3IP FPGA Block Diagram



Key Product Features

Table 4: Key Product Features

Feature	Description
IP-Compatible Slots	VPX2IP: 2 slots PCIe3IP: 3 slots PCIe5IP: 5 slots
Operating Speed	8 or 32 MHz operation in each slot Independently programmable
IP CLK Phase Stepper	Reduces simultaneous switching noise
PCI Express compliance	Gen1, Gen2, Gen3, or Gen4 slot operable and compliant
Non-blocking IP FIFO architecture	Prevents IP versus IP packet congestion
Interrupt Support	For MSI and INTA#
Interrupt Pacing/Aggression	Programmable with edge or level detection
IP Channel Activity Monitors	ACK* counters and logic enable sequenced IP transfers

Word Transfer Capability	Byte, word, long-word, and quad word transfer capable
Address Access	Incrementing or static address access of each IP slot
IP Word and/or Byte Swapping	Programmable IP Word and/or Byte Swapping
Bus Error Abort Response Times	Programmable Bus error abort response times
Independent Voltage Monitoring Circuits	5 independent voltage monitoring circuits to detect if a board voltage is out of range
Headers	1:1 50-pin headers with matched 5 mill trace and spacing widths between I/O and header
IP Resets	Individual IP resets with options for IP only or IP and local control
DipSwitches	Two 8-position DipSwitches One for users and one for configuration and test
IP Bus Termination	Configurable Select 5.0V or 3.3V termination via CFG switches
LEDs	8 user LEDs 5 indicate power is good ACK* activity LED for each IP
Power	Fused filtered power with resettable fuses for each position
Drivers	Windows®, Linux, and VxWorks drivers

NOTE: As Dynamic Engineering adds features to the hardware, they will update the PCIeIP page on the Dynamic Engineering website. If you want some of the new features and have already purchased hardware, you can download the Flash update and use the tools provided to update the Flash via the JTAG port. Or you may send the board back and Dynamic Engineering will update the Flash (this service requires a nominal fee).

The basic PCIe identifying information will not change with the updates, including the PCIe Revision ID field which is set to 0x01. To allow configuration control, a Version ID register is provided at offset 0x01C to indicate the current code revision of both the FPGA and CPLD. Each device's revision ID has a major and minor field. The major field is updated with new client releases. The minor is used internally for development purposes and in some cases with specialized client releases. Our drivers make the revision information available as part of the board info.

If your project can benefit from a "non-standard" implementation, or features that we have not thought of or implemented yet, please let Dynamic Engineering know. For example, if your project has IPs that can operate at 64 MHz instead of 32 MHz, Dynamic Engineering could modify the design the meet 64MHz timing.

Product Specifications

Table 5: Product Specifications

Specification	Description
PCIe Interfaces	PCIe 1 lane (x1) interface
Access Types	Configuration and memory space utilized
CLK Rates Supported	Standard 100 MHz PCI Express differential reference clock for PCIe interface. IP interface clock run at either 8 MHz or 32 MHz.
Memory	Multiple FIFOs are implemented to support multiple parallel processes at any one time within the Lattice FPGAs. Parallel processing is achieved with IP accesses. Controllable for sequenced IO across multiple IP's.
I/O	Level shifters are used to shift 3.3V FPGA signaling to 5.0V IP signaling.
Interface	50-pin header. Right angle through the bezel for position 0 and vertical headers for positions 1 and 2 for PCIe3IP or positions 1, 2, 3, 4 for PCIe5IP. Bezel has special features for routing rear connector cables through the bezel. VPX2IP - 3U 4HP with bezel or VPX connector/rear I/O with blank bezel. Comes with alignment pins, and mounting screws.
Software Interface	Control Registers within Lattice FPGA. Drivers provide generic calls for GPB access to allow any user modification to be programmed with the standard driver.
Initialization	Programming procedure documented in this manual.
Access Modes	Registers on long/double word boundaries. Standard target access read and write to registers and memory.
Access Time	Programmable time-out for IP Bus Error situations.
Interrupt	1 interrupt is supported with multiple sources. The interrupts are maskable and are supported with a status and control registers.
Onboard Options	Selectable shunt for 3.3V or 5V reference Bus Termination
Board Stuffing Options	PCIe3IP only - Resistor stuffing options to reconfigure the connectivity between IP1 and IP2's carrier I/O SMT and 50-pin headers VPX2IP only - Resistor stuffing options to reconfigure the connectivity between IP0 and IP1's carrier I/O SMT and 50-pin Condo header or VPX connector.
Dimensions	VPX2IP - 3U 4HP with bezel or VPX connector/rear I/O with blank bezel. PCIe3IP - half-length PCIe board. PCIe5IP - full-length PCIe board.
Construction	High Temp FR4 Multi-Layer Printed Circuit, Surface Mount Components.

Power	12V and 3.3V from PCIe bus. Local 5V, 3.3V, 1.2V, and -12V created with on-board power supplies.
User	8 position software readable switch 8 software controllable LED's, 5 Power Supply indicator LED's. IP activity LED's, one for each IP, 2 for VPX2IP, 3 for PCIe3IP, 5 for the PCIe5IP

Construction and Reliability

PCIe Modules, while commercial in nature, can be conceived and engineered for rugged industrial environments. PCIeIP is constructed out of 0.062-inch-thick High Temp FR4 material. Surface mount components are used. Most devices are high-pin count compared to mass of the device. For high vibration environments, inductors and other higher mass per joint components can be glued down.

Installation and Interfacing Guidelines

Some general interfacing guidelines are presented below. If you need more assistance, contact Dynamic Engineering.

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

Make sure that the system can see your hardware before trying to access it. Many BIOS will display the PCI Express devices found at boot up on a "splash screen" with the VendorID and CardID and an interrupt level. Look quickly. If the information is not available from the BIOS then a third-party PCI device cataloging tool will be helpful. In Windows systems, the device manager can be used.

Guidelines

Grounds – Watch the system grounds. All electrically connected 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 have all their own ground wires back to a common point.

Thermal Considerations

The PCIeIP design consists of CMOS circuits. The power dissipation due to internal circuitry is very low. With the one-degree differential temperature to the solder side of the board, external cooling is easily accomplished.

Theory of Operation

PCIeIP functions as a bridge between PCI Express and IP bus devices that adhere to their respective specifications. Since there are no additional PCIe virtual buses or PCIe ports downstream from the PCIeIP's PCI Express port, the PCIeIP is defined to be a PCIe Endpoint. As an Endpoint, the PCIeIP is downstream from the Host/Root Complex, which detects it and configures it during the enumeration process using PCIe configuration read and write packets. The VPX2IP/PCIe3IP requests a total of 32 MB, and the PCIe5IP requests 64 MB from the Host who provides the requested memory space via BAR0. Each IP's ID, IO, INT, and MEM spaces are mapped inside the 32/64 MB space. When the PCIe Host transmits either a memory read or write packet, which contains an address within the PCIeIP's BAR0, it is routed to either the PCIeIP register block or one of the IPs to be further decoded to generate an ID, IO, INT, or MEM space IP access.

The PCIeIP handles all accesses within its 32/64 MB of space. Writes to reserved registers or memory are dropped. For reserved register reads, 0x0000_0000 is returned. For reads of undefined memory spaces, 0xFFFF_FFFF is returned. In all cases credits are updated.

If a packet is received with an address outside the PCIeIP's BAR0 space it is discarded, credits are updated and an unsupported request PCIe packet is returned to complete the transaction.

PCIeIP uses a Lattice semiconductor ECP3 Family FPGA and utilizes Lattice's PCI Express Endpoint Core IP, which is compliant to PCI-SIG PCI Express 1.1 Base Specifications. The core handles the PCIe Physical and Data Link Layer requirements and provides an extensive interface which the PCIeIP interacts with to receive or transmit Transaction Layer Packets (TLPs) and communicate and/or update available flow control credits.

All Transaction Layer Packets coming from the PCIe Core that are within BAR0 space are written into a receive packet FIFO. Upon detecting a packet is in the receive FIFO it is immediately read out, routed, and written into the target IP's or registers receive FIFO.

The FIFOs are sized and PCIeIP advertises credits such that no access packet has to wait behind a different IP channels access packet, as such this creates a Non-blocking IP FIFO architecture. However, since the IP interfaces run much slower than the rate at which incoming receive FIFO Packets can be received, packets can back up inside a particular IP's receive FIFO, but will never backup into the main receive FIFO.

For example, if 100 reads are in one IP's receive FIFO and a new read is received for a different IP, it will be routed into that IP's receive FIFO.

The header credits are set to the maximum allowed by the PCIe core, which is 127, so up to 127 combined IP read and write requests may be pending at any one time inside the PCIeIP.

Once a register or IP access has completed processing a write or read, the appropriate credit updates will be accumulated and sent to the core, which will create Flow Control DLL packet(s) to inform/update the Host that more space/credit is available inside the PCIeIP. For the VPX2IP/PCIe3IP/PCIe5IP, there are three/four/six possible read targets respectively and/or sources for read completion packets, they are: IP[1:0]/IP[2:0]/IP[4:0], and a register access. When the read target has assembled the read completion packet, it stores it in its transmit FIFO as it must arbitrate for and then write the completion into the transmit packet port of the PCIe core. The transmit packet

arbiter is a round robin arbiter. The transmit packet port won't grant access until the core informs it is ready to receive a packet.

Table 6: Possible Read Targets

	Board Name		
	VPX2IP	**PCIe3IP**	**PCIe5IP**
Number of Possible Read Targets	3	4	6
Read Targets	IP [1:0]	IP [2:0]	IP [4:0]

Programming

PCIeIP is tested in a Windows environment using the Dynamic Engineering Driver to do the low-level accesses to the hardware. Dynamic Engineering uses MS Visual C++ in conjunction with the driver to write our test software. Please consider purchasing the engineering kit for the VPX2IP, PCIe3IP, or PCIe5IP; the software kit includes our test suite. In addition, Linux and VxWorks drivers and reference suites are available.

The drivers take care of discovery and the UserAp allows the client to select which installed board is selected for use.

If you are writing your own driver, it is suggested to get the engineering kit and the Linux version of the software.

Address Maps

VPX2IP Address Map

Table 7: VPX2IP Address Map

Function	Offset	Size
Registers – IP[2:0]	0x000 – 0x3FF	1K Bytes
ID Space – IP0	0x400 to 0x47F	128 Bytes
ID Space – IP1	0x480 to 0x4FF	128 Bytes
Reserved	0x500 to 0x7FF	768 Bytes
IO Space – IP0	0x800 to 0x87F	128 Bytes
IO Space – IP1	0x880 to 0x8FF	128 Bytes
Reserved	0x900 to 0xBFF	768 Bytes
INT Space – IP0	0xC00 to 0xC7F	128 Bytes
INT Space – IP1	0xC80 to 0xCFF	128 Bytes
Reserved	0xD00 to 0xFFF	768 Bytes
Reserved	0x1000 to 0x7F_FFFF	8 MB-4KB
MEM Space – IP0	0x080_0000 to 0x0FF_FFFF	8 MB
MEM Space – IP1	0x100_0000 to 0x17F_FFFF	8 MB
Reserved	0x180_0000 to 0x1FF_FFFF	8 MB

PCIe3IP Address Map

Table 8: PCIe3IP Address Map

Function	Offset	Size
Registers – IP[2:0]	0x000 – 0x3FF	1K Bytes
ID Space – IP0	0x400 to 0x47F	128 Bytes
ID Space – IP1	0x480 to 0x4FF	128 Bytes
ID Space – IP2	0x500 to 0x57F	128 Bytes
Reserved	0x580 to 0x7FF	640 Bytes
IO Space – IP0	0x800 to 0x87F	128 Bytes
IO Space – IP1	0x880 to 0x8FF	128 Bytes
IO Space – IP2	0x900 to 0x97F	128 Bytes
Reserved	0x980 to 0xBFF	640 Bytes
INT Space – IP0	0xC00 to 0xC7F	128 Bytes
INT Space – IP1	0xC80 to 0xCFF	128 Bytes
INT Space – IP2	0xD00 to 0xD7F	128 Bytes
Reserved	0xD80 to 0xFFF	640 Bytes
Reserved	0x1000 to 0x7F_FFFF	8 MB-4KB
MEM Space – IP0	0x080_0000 to 0x0FF_FFFF	8 MB
MEM Space – IP1	0x100_0000 to 0x17F_FFFF	8 MB
MEM Space – IP2	0x180_0000 to 0x1FF_FFFF	8 MB

PCIe5IP Address Map

Table 9: PCIe5IP Address Map

Function	Offset	Size
Registers – IP[2:0]	0x000 – 0x3FF	1K Bytes
ID Space – IP0	0x400 to 0x47F	128 Bytes
ID Space – IP1	0x480 to 0x4FF	128 Bytes
ID Space – IP2	0x500 to 0x57F	128 Bytes
ID Space – IP3	0x580 to 0x5FF	128 Bytes
ID Space – IP4	0x600 to 0x67F	128 Bytes
Reserved	0x680 to 0x7FF	384 Bytes
IO Space – IP0	0x800 to 0x87F	128 Bytes
IO Space – IP1	0x880 to 0x8FF	128 Bytes
IO Space – IP2	0x900 to 0x97F	128 Bytes
IO Space – IP3	0x980 to 0x9FF	128 Bytes
IO Space – IP4	0xA00 to 0xA7F	128 Bytes
Reserved	0xA80 to 0xBFF	384 Bytes
INT Space – IP0	0xC00 to 0xC7F	128 Bytes
INT Space – IP1	0xC80 to 0xCFF	128 Bytes
INT Space – IP2	0xD00 to 0xD7F	128 Bytes
INT Space – IP3	0xD80 to 0xDFF	128 Bytes
INT Space – IP4	0xE00 to 0xE7F	128 Bytes
Reserved	0xE80 to 0xFFF	384 Bytes
Reserved	0x1000 to 0x7F_FFFF	8 MB-4KB
MEM Space – IP0	0x080_0000 to 0x0FF_FFFF	8 MB
MEM Space – IP1	0x100_0000 to 0x17F_FFFF	8 MB
MEM Space – IP2	0x180_0000 to 0x1FF_FFFF	8 MB
MEM Space – IP3	0x200_0000 to 0x27F_FFFF	8 MB
MEM Space – IP4	0x280_0000 to 0x2FF_FFFF	8 MB
Reserved	0x300_0000 to 0x3FF_FFFF	16 MB

The address maps provided are for the local decoding performed within PCIeIP. The addresses are all offsets from a base address. The base address, BAR0 is provided by the host in which the PCIeIP is installed.

The host system will enumerate to find the assets installed during power-on initialization

Table 10: Vendor and Device IDs

	Vendor ID	Device ID
PCIe3IP	0xDCBA	0x0051
PCIe5IP	0xDCBA	0x005C
VPX2IP	0xDCBA	0x0060

Third party utilities can be useful to see how your system Memory Map is configured. The interrupt level expected and style is also set in the registry. Dynamic Engineering recommends using the Dynamic Engineering Driver to take care of initialization and device registration.

Once the initialization process has occurred and the system has assigned an address range to the PCIeIP card, the software will need to determine what the address space is. Dynamic Engineering refers to this address as base in their software.

The next step is to initialize the PCIeIP by configuring the IP(s) control and interrupt registers. Optionally, registers, such as the Switch and LED control, may also be configured at this point.

Dynamic Drivers provide all of the above functionality and a generic IP driver for use when a specific IP driver is not available for a particular IP.

Dynamic Engineering drivers are supported with UserAp. UserAp provides a mini-application to perform tasks on the IP and PCIeIP. UserAp is provided with the source code to allow users to have a running start at integration. The UserAp code is developed to do ATP testing on the hardware and is complete from a feature development point of view. UserAp has functionality in main.c to call the driver and get the handle to the IP. In addition, the functionality to discover multiple cards on multiple carriers is provided. The handle is used along with the specific offsets for the carrier or IP to access the hardware.

The user switch can be used to deterministically select the intended IP module when multiple carriers with multiple modules are in a system. Setting the switches to different known values and reading the switch values from the carriers, then remapping the handles for the carriers can provide an automatic mechanism to be sure your application software is communicating with the correct IP when multiple IPs of the same type are available within the system.

PCIeIP Resets

PCI Express Reset (PERST#) Summary

There are two types of resets in PCIe, both of which are supported by the PCIeIP, they are:

1. Fundamental Reset (cold or warm) – assertion of PERST#.
    a. Cold – Power applied to a cold (non-powered) system.
    b. Warm – Power is up/good before and during assertion of PERST#.
2. Hot Reset – TS1 Ordered-Sets sent with bit [0] of symbol 5 asserted for 2 MS.

Per the PCIe Specification:

- The minimum PERST# assertion time is 100 MS from the time power is stable.
    - In order to be configured and ready for enumeration before PERST# is de-asserted, PCIeIP downloads the FPGA configuration file from a parallel flash.
    - PERST# signal is asserted and de-asserted asynchronously.

IP Reset Summary

Reset* (IPx_RESETN) signal rules from Vita Spec:

- When asserted, Reset* must be asserted for a minimum of 200 MS, there is no maximum.
- Can be asserted asynchronously, and must be de-asserted synchronously.
- +5V must be monitored and reset asserted if power falls below minimum spec
- Reset* is the logical OR of power monitoring reset and the system reset.
- IP module documents must clearly state the time needed from Reset* de-assertion until the IP module is initialized.
- When Reset* is asserted IPs must terminate any cycle, interrupts, DMA requests or future requests.

The PCIeIP supports all carrier card rules for *Reset.

PCIeIP Reset* Functional Behavior

PCI Express PERST# assertion response: Each IP's Reset* is asserted as long as the PERST# is asserted. Once PERST# is de-asserted, each channel starts its own 256 millisecond counter, which when it expires, causes that channel's Reset* to be synchronously de-asserted relative to its clock.

User Control bit response - Each IP channel's Reset* in the PCIeIP can be asserted independently. The PCIeIP provides the following three register bits in each IPx's control0 register (i.e. offset 0x80 for IP0):

1. Reset IP - IPx Control0 Register bit [16]
    a. Reset* is driven low as long this bit is set to 1. Upon writing this bit back to zero, Reset* is de-asserted. Meeting the minimum assertion time of the IP is the responsibility of the software.
2. Reset IP and Channel – IPx Control0 Register bit [17]
    a. Reset* is driven low as long this bit is set to 1. Upon writing this bit back to zero, Reset* is de-asserted after the expiration of that channels 256 millisecond reset counter to ensure Reset* meets the 200 MS minimum assertion time. Assertion of this bit also resets all channel related logic and FIFO, for predictable behavior assertion of this bit should occur after R/W requests to the channel have been completed and no further accesses are submitted until the Reset* pin has been de-asserted.

3. IP Reset Pin Status – IPx Control0 Register bit [18]
    a. Provides real time value of the IP's Reset* pin state. If 0, the IP Reset* pin is de-asserted (1). If 1, the IP Reset* pin is asserted (0).

Since the PCI Express interface and register logic inside the PCIeIP are clocked independent of the IP logic, the PCIeIP can be configured by the software between the time PERST# de-asserts and Reset* de-asserts. Accesses to each IP should be delayed until after the IP Reset* timers have expired 256 milliseconds after PERST# de-assertion.

Asserting the Reset IP - IPx Control0 Register bit [16] has no effect the channels IP clock frequency as set by the Clock SEL bit. Assertion of the Reset IP and channel - IPx Control0 Register bit [17], or if P5VGOOD is 0, will force the Clock SEL for that channel to be 0 (8MHz) for the duration of the assertion plus 256ms at which time the frequency will revert back to 32MHz if the Clock SEL bit is set to 1.

P5VGOOD pin response: If the PCIeIP voltage monitoring circuit detects 5-volt power is out of specification, it asserts the P5VGOOD pin to 0. Since the P5VGOOD pin assertion is asynchronous, the PCIeIP synchronizes it and ensures it has been asserted for at least two clock cycles before detecting it as asserted. Internally, if the P5VGOOD pin value is detected to be 0, the PCIeIP will assert each channel's Reset* for the duration P5VGOOD is 0 + 256 milliseconds. The PCIeIP will also set each channels Force/P5VGOODn Interrupt
NOTE: see Interrupt section for details.
P5VGOOD pin = 0 would not be expected even under extreme operating conditions.

PCIeIP Clocks

Each IP channel on the PCIeIP has its own independent clock (CLK). IP CLKs are held low until PERST# is de-asserted. Upon de-assertion of PERST#, each IP CLK becomes active and runs at 8 MHz. Each IP channel's CLK frequency can be set independently to 32 MHz or back to 8 MHz at any time without restriction by setting or clearing that channels Clock SEL bit - IPx Control0 Register bit [8].

Each channel's CLK can optionally be turned off and set to a steady state of high (disabled) if the user knows that the channel will not be used and/or has EMI concerns. To disable or enable an IP channel's CLK, users may set or clear that channels Clock disable bit - IPx Control0 Register bit [9]. As long as there are no transfers in process when the clock is disabled it may be re-enable and the channel will process read write requests as normal.

***Warning**: If the channel is disabled and reads/writes continue to be sent to that channel the FIFO's in the PCIeIP will back up, credits won't be replenished, and the PCIeIP will become congested and unable to communicate with enabled/working IP's*.

Each IP CLK clock edge is relative to the 32MHz oscillator clock on the PCIeIP board. In order to reduce Simultaneous Switching Outputs (SSO) i.e. switching noise, The PCIeIP has an IP Clock Phase Stepper circuit which phase shifts IP1's clock rising edge by 8 ns, and IP2's clock rising edge by 16 ns relative to IP0's clock rising edge. For the PCIe5IP, IP3/IP4's clock is in phase with IP0/IP1's clock.

Bus Error Processing

In regards to Bus Error the vita spec states the following:

- In the 8 MHz Data Transfer Cycle Details section: “There is no explicit provision for cycle timeout. Even if an IP Module supports a specific select line it is not required to always assert ACK* … However, the IP Module literature must clearly state access time, or “wait states.” … RECOMMENDATION 10-26: IP Modules’ maximum ACK* time should generally be not be more than a few microseconds.”
- In the 32 MHz Operation section – “A maximum number of permitted wait states by an IP Module at 32 MHz is 127 clock cycles.”

Each IP channel in the PCIeIP has its own logic to monitor how long the channel waits for ACK* before terminating the cycle as a Bus Error. The PCIeIP provides some flexibility for users to choose how long the wait should be using the Bus Error timeout select (IPx Control0 Register bit [12]). Its settings are:

- 0 = 127 IP Clocks – 4us at 32 MHz or 63 IP Clocks – 8us at 8 MHz (0 is the default)
- 1 = 255 IP Clocks – 8us at 32 MHz or 127 IP Clocks – 16us at 8 MHz.

A Bus Error event occurs when the PCIeIP initiates an R/W access to an IP and the IP doesn’t assert its ACK* within the predefined number of IP clocks. The number of clocks the PCIeIP waits is determined by the frequency of operation and the setting of the Bus Error timeout select bit. Upon the occurrence of a Bus Error, that IP channel sets its Bus Error Interrupt bit (IPx ISR bit [2] = 1) and an interrupt can be generated.

In normal operation the PCIeIP receives PCI Express packets to write or read up to 64bits on the IP bus. Byte enables are also received in the packet which the PCIeIP uses to determine if a byte, word, Dword, two Dword, or four Dword R/W transaction will be generated on the IP bus. If ACK* is not received for a write, the write transaction is terminated, data is discarded, and credits are updated accordingly.

If ACK* is not received for a read, then depending upon the transfer size and/or byte enables, a read will return different completion data. For a 64-bit transfer, the PCIeIP generates two separate 32-bit accesses; as such the data returned when ACK* isn’t provided, is the same as when an ACK* isn’t provided during two 32-bit transfers back to back. A 32-bit transfer causes the PCIeIP to do two 16-bit IP transfers. If ACK* isn’t received for the first 16-bit IP transaction, 0xFFFF_FFFF will be provided and the second 16-bit IP transaction will not be generated on the IP bus. If the ACK* is received for the first IP transaction but not the second, the read will have 0xFFFF in the upper word and valid data in the lower word of the 32-bit PCIe read completion data field.

For non-32-bit reads where an ACK* does not occur, 0xFFFF will be returned for a Dword access that has a valid byte enable(s). A word in the 32-bit completion data field, where the byte enables are not asserted, will have 0x0000 in them regardless if there is a Bus Error or not. Therefore, the completion data returned depends upon each IP transfer’s byte enables and which transfer the IP doesn’t provide an ACK*.

When a Bus Error occurs, the IP state machines will return to their idle state so that they are able to process the next transaction. For all possible read combinations with bus errors credits are updated accordingly.

Bus Error interrupts can be, and are, used during initialization to scan the IPx ports to detect the presence of an IPx installed in a slot. Typically, a driver will do a 32-bit (or 16-bit) access of the slot's/IP's ID space, and if it detects either 0xFFFF_FFFF (or 0xFFFF) it knows an IP is not installed in that slot. Once a system is initialized and running, a Bus Error interrupt is normally considered a serious system error.

IP Channel Transfer Activity Monitor Logic

To provide the ability to check an IP channel's transfer activity status, health, or to enable sequenced IP transfers, the PCIeIP contains an IPx Channel Transfer Monitor Register (IPx CTM) for each IP channel. Below is a summary the IPx CTM register bits and logical behavior:

- An IPx Channel Busy (IPx CTM bit [28]) indicator
  - This register bit provides a way to ascertain the activity status of that channel.
  - When the IPx Channel Busy status indicator is 0, it indicates there are no transfers in process or pending for that channel.
  - When the IPx Channel Busy status indicator is 1, it indicates at least one transfer is still pending in the channel's FIFO or the IP interface is still processing a transfer.
  - The IPx Channel Busy status changes from busy to not busy [1 to 0] when the respective channel FIFO is empty, and all IP interface write/read cycles have completed, which can be several ACK cycles when performing multi-word accesses. **NOTE:** In the case of a Read Instruction, the busy signal may go to the not busy state before the read data reaches the CPU since arbitration and core processing may not have completed.
- A 16-bit IPx ACK* counter whose count can be read by reading that channels IPx ACK* count (IPx CTM bits [15:0]) to determine if all the required ACK*'s for the cycles submitted to that IP have been received. This counter is enabled when both the ACK* Count Enable (IPx Control0 register bit [13]) and IPx ACK* Counter Clear#/Enable (IPxCTM bit [24]) are enabled (set to 1).
- An IPx ACK* Counter Clear#/Enable (IPx CTM bit [24]) register bit which clears and holds the ACK* Counter to 0x0000h when 0. When 1, it enables the ACK* counter to count ACK*'s.

The IPx Channel Busy indicator and ACK* counter logic are independent of each other. Users may use either or both together if they choose. For accurate transfer monitoring that all transfers have completed, it is recommended that the IPx counter and IPx ACK* Counter Clear#/Enable logic be used.

One possible usage example would be as follows:

A system engineer wants to issue and ensure completion of 20 Dword transfers (composed of Reads and Writes) to IP0, then do and ensure completion of 20 Dword transfers to IP1, then do and ensure completion of 20 Dword transfers to IP2. To do this with the PCIeIP the engineer could do the following:

1. Enable each channel's ACK* counter by setting each Ips ACK* Count Enable (IPx Control0 register bit [13]) bit = 1.
2. Ensure each channel's IP counter is clear, by clearing IP0, IP1, and IP2, ACK* Clear#/Enable bit by writing each channel's IPx CTM bit [24] register bit = 0.
3. Instruct the software to stop issuing IP channel Read/Writes and then read each IPx Channel Busy (IPx CTM bit [28]) indicator bit until each bit is 0.
4. Enable IP0, IP1, and IP2 ACK* counters to count their channel's ACK*s by writing each channel's IPx Clear#/Enable (IPx CTM bit [24]) register bit = 1.

5. Submit the 20 Dword transfers to IP0, then read IP0's ACK* count value. When the ACK* count is equal to 40 decimal all AKC* for the 20 Dword IP transfers that have been received.
6. Repeat four for IP1 then IP2.

PCIeIP Interrupts

This sections purpose is to summarize the PCIeIP Interrupts, Interrupt registers, programmable features usage, and behavior. Here's a brief description of what's covered:

- Summary of PCIe specification Interrupt configuration registers
- Summary of interrupt registers/bits contained in the PCIeIP
- Summary of the PCIeIP's Interrupt specific registers and bits
- Detailed description of PCIeIP's interrupt functionality and features
- Detailed summary of the PCIeIP's interrupt specific registers and bits

PCIeIP Interrupt Registers Summary

PCI Express has two possible types of Interrupt packets that can be generated, they are:

1. INTx# - Legacy PCI Interrupt virtualization packets
2. MSI – Message Signal Interrupt packets

PCIeIP supports both MSI and INTx# interrupt packet generation.

There are eight PCI Express configuration (type 0) registers contained in the PCIeIP, which are involved in the generation of interrupt packets.

The next section summarizes the four registers with INTx# related bits, followed by a section that summarizes the four registers with MSI related bits. The summary of these registers is provided to ease users understanding of PCI Express Interrupts.

NOTE: Users need not be concerned with setting or using the register values written by the host as they are used by the PCIeIP PCI Express core when generating INTx# or MSI PCI Express packets.

PCIeIP INTx# Interrupt Configuration Registers Summary

Command Register (offset 0x04)

- Interrupt Disable (bit [10]) R/W
  - 0 = INTx# enabled (default)
  - 1 = INTx# disabled.

Status Register (offset 0x06)

- Interrupt Status (bit [3]) RO
  - 0 = no Interrupt pending (default)
  - 1 = Interrupt pending. Only valid in INTx# Mode.
  - Disabled, always zero in MSI mode.
- Capabilities List (bit [4]) RO
  - 0 = no capabilities registers
  - 1 = capabilities registers exist. This bit is hardwired to 1 in PCIeIP - MSI registers are capability registers.

Interrupt Line Register (offset 0x3C)

- Interrupt Line [7:0] R/W register written by the Host to distinguish device interrupt.

Interrupt Pin Register (offset 0x3D)

- Interrupt Pin [7:0] RO register, hardwired to 0x01 in PCIeIP to indicate INTA#.

PCIeIP MSI Interrupt Configuration Registers Summary

Message Control Register (offset 0x72)

- MSI Enable (bit [0]) R/W
  - 0 = MSI is disabled (default)
  - 1 = MSI is enabled.
- Multiple Message Capable (bits [3:1]) RO
  - 0x0 = PCIeIP requests 1 MSI.
- Multiple Message Enable (bits [6:4]) R/W
  - 0x0 (default) Host writes to allocate MSI's.
- 64-bit Address Capable (bit [7]) RO
  - 1 = PCIeIP is 64-bit MSI address capable.

Message Address Register (offset 0x74)

- Message Address (bits [31:2]) R/W
  - Host provides - bits [31:2] of D-Word aligned MSI address.

Message Upper Address Register (offset 0x78)

- Message Address (bits [63:32]) R/W
  - Host provides - upper 32bits for 64bit MSI addressing.

Message Data Register (offset 0x7C)

- Message Data (bits [15:0]) R/W
  - Host provides - value used to distinguish PCIeIP interrupt.

PCIeIP Interrupts and Registers Summary

For each IP channel there are four possible interrupt sources.

- IPx Force/P5VGOODn Interrupt
  - Dual source, either user programmable interrupt via register bit or interrupt set due to detection of the 5-volt power source being out of range.
- IPx Bus Error Interrupt
  - Occurs when an IPx does not respond to an access.
- IPx IntReq[1:0]* Interrupts
  - Occurs when an IPx asserts one of its IntReq[1:0]* pins.

PCIeIP has two global interrupt registers related to all IPs and two registers for each IP; registers specific to an IP are prefixed with IPx. The offsets noted below are relative to BAR0. Here's a brief summary of each register:

- Interrupt Status Register (ISR) at offset 0x008
  - Contains all PCIeIP Interrupt bits, each bit in this register can cause the generation of an Interrupt packet.
- Interrupt Control Register (ICR) at offset 0x00C

- Contains bits that control the timing of when an interrupt packet is generated and sent to the Host.

- IPx Interrupt Status Register (IPx ISR)
  - Bits in these registers either contain or reflect the value of the Interrupt source that can cause the generation of an Interrupt packet if it is allowed to progress to the ISR (see spec for offsets, i.e. IP0 ISR offset is 0x088)
- IPx Interrupt Control Register (IPx ICR)
  - This register contains the Enable/Mask bits for each IPx Interrupt. Mask bits control whether or not an interrupt progresses to the ISR.
  - The IPx ICR also contains a Level/Edge Interrupt control bit for each IPx Interrupt. A Level/Edge control bit changes the interrupt packet generation behavior for that interrupt.
    - In Level mode multiple interrupts may be sent to the Host per a single interrupt.
    - In Edge mode only one interrupt will be sent per interrupt assertion.
      NOTE: See spec for offsets, i.e. IP0 ICR offset is 0x08C

PCIeIP Interrupt Functional Operation

The descriptions of the PCIeIP interrupt behavior to follow refer to the PCIe compliant configuration registers and PCIeIP specific Interrupt registers. This was done for the convenience of the reader and should be sufficient for understanding the remainder of this document. For PCI Configuration register details, please refer to the relevant PCI and/or PCIe specification.

When PCIeIP comes out of reset, it is in INTA# mode as both the Interrupt Disable and MSI Enable bits are 0 by default. It is up to the Host to determine if all devices in the system can support MSI before configuring the system to use MSI Interrupts. This section will first describe how PCIeIP INTA# virtualized signaling works then describe the setup, functional, and behavioral differences for MSI Interrupts.

PCIeIP INTA# Interrupt Functional Operation

Each IP channel has 4 possible interrupt sources: Force/P5VGOODn, Bus Error, and IntReq[1:0]*. The state of a channel's interrupt sources is in its IPx ISR bits [3:0]. When asserted, these bits are true high. Each channel also has an IPx ICR register, which contains an Interrupt enable bit for each source interrupt. When disabled, the interrupt source is Masked (i.e. blocked) from generating an interrupt to the Host, each Interrupt Enable/Mask bit is 0 by default (0 = disabled/masked). The Host may enable any IP channel source interrupt independently by setting any of the IPx ICR [3:0] bits = 1.

If an Interrupt source is asserted and enabled (not Masked), it will be registered in the ISR (0x008). The ISR has a register bit for every possible interrupt. When an ISR bit is set, it causes PCIeIP to generate and send a PCI Express INTA# Message packet that contains an Assert INTA message code to the Host. Since PCIeIP only has one interrupt for multiple possible interrupt sources, the Host will have to read the ISR to determine which Interrupt or Interrupts are asserted. To clear an Interrupt/bit, the Host must Write 1 to Clear (W1C) the interrupt bit in the ISR. When the Host executes a W1C of an ISR bit or bits, the bit(s) will be cleared and PCIeIP will generate and send another INTA# Message packet, this time the packet will contain a Deassert_INTA message code.

Once a W1C of any ISR bit occurs, the PCIeIP will wait until the Interrupt de-assert timer expires before checking to see if an ISR bit is set. If an ISR bit is set when or after the timer expires, PCIeIP will generate another assert INTA# packet.

PCIeIP Interrupt Level/Edge Functionality

PCIeIP IPx ICR [11:8] Level/Edge bits individually configure each of a channel's four IP interrupts to be either Level or Edge. In Level mode, if an IPx ISR bit (i.e. source) is asserted and enabled when a W1C for that ISR bit occurs, it triggers the de-assert INTA# packet to be generated. Since the source is still active the interrupt will remain set in the ISR. Another INTA# assert packet will be generated once the de-assert timer expires. In other words, the PCIeIP can generate and send multiple interrupt assert packets to the Host for a single interrupt source assertion. This will continue as long as the interrupt source (in IPx ISR) remains asserted and enabled regardless of when or how many W1C of the ISR occurs. If the source in the IPx ISR is cleared before the W1C of the ISR bit it will not be the cause of an additional Interrupt assertion when the de-assertion timer expires (unless it is reasserted before the timer expires). ICR bits [2:0] select the Interrupt de-assertion time which ranges from 96 nanoseconds to 33 microseconds. In most cases, the edge mode will be used. Level mode is useful when a device needs to be read or loaded whenever above/below a certain level [FIFO etc.] Just because the SW loads data, does not mean the condition has been removed.

In Edge Mode when an interrupt source is set and enabled, its respective ISR bit will be set and an INTA# interrupt assert packet will be generated and sent to the Host. When the Host performs a W1C of that ISR bit, it will clear the ISR bit and PCIeIP will send a de-assert INTA# packet to the Host. That ISR bit will remain clear and no further INTA# packets will be generated [due to that specific bit] until the source interrupt is de-asserted and reasserted. This means that only one Interrupt will be sent to the Host per assertion of the source interrupt regardless of how long the source interrupt is asserted. In this mode, the interrupt can be detected, cleared, and a flag set in software to cause the higher-level application to take action which will result in clearing the interrupt. During this sequence additional interrupts are blocked due to the requirement for the interrupt source to be cleared before it can be set again.

PCIeIP Interrupt Aggregation

PCIeIP contains an Interrupt Aggregation timer and logic that enables the user to program the rate at which an interrupt can be generated. The Interrupt Aggregation enable bit (ICR bit [13]) is = 0 off/disabled by default. When enabled (ICR bit [13] = 1), the Interrupt Aggregation timer starts counting and provides a pulse when the user selected time delay is reached. ICR bits [11:8] select the Interrupt Aggregation time, which ranges from 32 microseconds to 1 second. When the Interrupt Aggregation pulse occurs, it is used to generate an interrupt assertion packet if an Interrupt is set in the ISR. Once the pulse is generated, the Aggregation Logic timer is reset and starts counting again until the Aggregation delay is reached and a pulse is generated again. This fixed cyclical timing (or pace) repeats until disabled. If your system has multiple interrupt sources and the interrupts can be processed with some delay, using the timer will group the interrupts together and reduce interrupt processing. The programmable time allows the user to tune the accumulation time to match system requirements.

PCIeIP Interrupt Timing, Behavior, and W1C

The number of interrupt sources that can cause PCIeIP to generate an interrupt packet to the Host depends upon how many IP's are connected. Since there are 4 sources per IP, the number of possible interrupt sources is: 4 x IPs. All the interrupt sources are synchronized internally before being used and setting their respective ISR bits. However, only under controlled circumstances can one accurately predict when interrupt sources will assert or de-assert. In normal (non-controlled) operating environments, interrupts may assert on any clock cycle, including before, after, or even on the same clock cycle of key interrupt related events such as a read of the ISR or a W1C cycle. For example, you may read the ISR and not "see" an Interrupt set, then read it immediately again and "see" several interrupts set. You may also process an interrupt W1C it then read the ISR to "see" the same interrupt set again.

Because PCIeIP handles interrupts internally in a synchronous fashion and registers them in the ISR, the system software only needs to follow one simple rule to avoid "losing" interrupts.

Interrupt W1C Rule: Only Write 1 to Clear (W1C) the ISR bits (i.e. interrupts) that you've processed.

The reason for this is that new/additional interrupts may be asserted and logged in the ISR since the original reading of the ISR so if the software were to do a W1C of all the ISR bits the new/additional interrupt(s) could be lost.

PCIeIP IP Interrupts

When asserted each of the IP interrupt sources: Force/P5VGOODn, Bus Error, and IntReq[1:0]* if enabled will set their respective bit in the ISR. The current status of the source value can be read at any time by reading the respective IP's IPx ISR. If the source "goes away" the respective ISR bit will remain set until a W1C of the ISR bit is executed. Below are additional details about each IPx Interrupt source register and/or signal.

IPx Force/P5VGOODn Interrupt – The Force/P5VGOODn Interrupt has two possible sources, they are:

1. The Force Interrupt source bit (IPx ISR bit [3]) is a register bit under user control and may be read or written at any time. This bit's purpose is to provide a way for the user to assert/de-assert an interrupt in a controlled fashion for development or design purposes. **NOTE:** Since all incoming register R/W packets flow into a register accesses FIFO they are processed in the order they are received, therefore if there are other register accesses ahead of the Force interrupt R/W, it will be delayed proportionately.
2. Even under extreme operation conditions, the P5VGOOD pin should be at a logic 1. However, if the PCIeIP voltage monitoring circuit detects 5-volt power is out of specification, it asserts the P5VGOOD pin to 0. Internally PCIeIP synchronizes the P5VGOOD pin value, ensures it has been asserted for at least two clock cycles, and inverts its value to create the P5VGOODn interrupt signal. When P5VGOODn goes high it sets each IPx's Force/P5VGOODn bit in the ISR register, these bits will stay set until cleared with a W1C. The IPx Force/P5VGOOD bit in the IPx ISR reflects the inverted value of P5VGOOD pin, as such it is possible the source could be de-asserted/clear when read and yet the IPx Force/P5VGOODn bits in the Interrupt Status Register (ISR) at offset 0x008 are set. When clearing the P5VGOODn interrupt it is recommended that all Force/P5VGOODn

bits in the ISR be W1C at the same time, otherwise additional interrupt may be generated even if P5VGOOD pin is back to true (1).

When the internal P5VGOODn goes high, it also causes all the IPx Resets* to be asserted for at least 256 milliseconds.
NOTE: see reset section for further details.

IPx Bus Error Interrupt: The Bus Error interrupt source bit (IPx ISR bit [2]) is a W1C register that captures the Bus Error event. Because Bus Error is an event, when in Level mode a W1C of the ISR's relevant IP Bus Error bit also clears the IPx Bus Error source register. In either Level or Edge mode, a W1C of the IPx ISR bit [2] does not clear its corresponding ISR Bus Error bit.

IPx Bus Error event: When PCIeIP initiates a R/W access to an IPx, it waits for the IP to respond with ACK*; if ACK* is not asserted within a predefined number of IP clocks, this is a Bus Error and the logic sets IPx ISR bit [2] = 1. The number of clocks is determined by the frequency of operation and the setting of the Bus Error timeout select bit (IPx Control0 Register bit [12]). The default/program values are 127/255 IP clocks at 32 MHz and 63/127 IP clocks at 8 MHz.

IPx Bus Error function and purpose: Upon the occurrence of a Bus Error, PCIeIP will either drop the write or complete the read using all F's for the data, and then return the IPx state machines to their idle state so that they are able to process the next transaction. Bus Error interrupts are used during initialization to scan the IPx ports to detect if an IP is present on the IPx port. Once a system is initialized and running a Bus Error interrupt would normally be considered a serious system error.

IPx Interrupt Requests (IntReq[1:0]*): IPx Interrupt Requests source bits (IPx ISR bits [1:0]) are read only status bits that reflect the IPx's IntReq[1:0]* pin states after being synchronized, inverted and filtered for 2 clocks.
NOTE: Since the IPx Interrupt Request source bits (IPx ISR bit [1:0]) essentially reflect the IntReq[1:0]* pins it is possible they can set their respective ISR bit, but the IPx ISR bit [1:0] source could be de-asserted/cleared when read if the IP toggles the IntReq[1:0]* pin(s). This would be considered unusual behavior. Dynamic Engineering IP Modules do not operate in this manner.

PCIeIP MSI Interrupts

Per the PCI Express specification when a PCI Express device supports both virtualized INTx messages and MSI, only one of the mechanisms will be enabled at any given time. When PCIeIP comes out of reset, it is in INTA# mode as both the Interrupt Disable and MSI Enable bits are 0 by default. Also, per the specification, setting the MSI Enable bit in the Message Control Register (offset 0x72) disables the PCIeIP ability to use INTx messages and enables MSI.

Before enabling MSI, the Host needs to write valid values in the Message Address and Message Data registers. It should also leave the Multiple Message Enable field at the default value of 0x0h as PCIeIP only requests and indicates support for one MSI via the Multiple Message Capable bits = 0x0h.

Once enabled, the key difference between PCIeIP's MSI and INTA# interrupt generation is that only one MSI packet will be sent to the Host to indicate an Interrupt assertion. No MSI packet is generated or sent when the ISR interrupt bit or bits are cleared using a W1C. The timing of when an MSI packet

is sent in response to an assertion is identical to when an INTA# assert packet is sent as described in previous sections.

When the PCIeIP generates and sends the MSI packet, it uses the Message Address and Data register values provided by the Host to create the MSI (Memory Write) packet.

Important Interrupt Notes

For predictable behavior, users should only change the Level/Edge mode, Aggregation enable, Aggregation, and Interrupt de-assert timers when the system is in a quiescent state or if all ISR Interrupts have been processed and all Interrupts are disabled (Masked) by clearing all the IPx's ICR Interrupt Enable [3:0] bits. Not doing so can lead to Interrupts being lost when switching from Level to Edge mode in addition to other unpredictable behavior.

When in INTA# mode, any W1C to any bit in the ISR, whether it is set or not, will cause PCIeIP to generate and send an INTA# Message packet that contains a Deassert_INTA message code.

When the MSI Enable bit is set, the INTA# functionality is disabled regardless of the state of the Interrupt Disable bit. Additionally, the Interrupt Status bit is no longer relevant and remains 0.

Assertion/de-assertion of the IPx ICR Interrupt enable bits [3:0] (i.e. Mask bits) when their respective source bits in the IPx ISR are asserted can be used to mimic assertion/de-assertion of the source interrupt as seen by the ISR.

PCIe3IP Registers

Table 11: PCIeIP register map – Offset 0x000 to 0x3FF

128 bytes for Global/Carrier Card Registers		
Offset	**Register**	**Description**
0x00	**Switch and LED Control**	Switch status and LED control
0x04	Reserved	-
0x08	**Interrupt Status**	Contains Interrupt bits for up to 8 IP's
0x0C	**Interrupt Control**	Interrupt assertion/de-assertion control
0x10	Reserved	-
0x14	**Scratch 0**	User scratch register 0
0x18	**Scratch 1**	User scratch register 1
0x1C	**Version ID**	Contains FPGA & CPLD code version ID's
0x20-7F	Reserved	-
96 bytes for each set of IP Registers		
Offset	**Register**	**Description**
0x80	**IP0 Control0**	IP0 Channel behavior & data flow control
0x84	**IP0 Control1**	IP0 Channel behavior & data flow control
0x88	**IP0 Interrupt Status**	IP0 Interrupts
0x8C	**IP0 Interrupt Control**	IP0 Interrupt Masks
0x90	Reserved	Reserved for IP0
0x94	**IP0 Transfer Monitor**	IP0 ACK*/Channel Transfer & Activity Monitor
0x98-DF	Reserved	Reserved for IP0
Offset	**Register**	**Description**
0xE0	**IP1 Control0**	IP1 Channel behavior & data flow control

0xE4	**IP1 Control1**	IP1 Channel behavior & data flow control
0xE8	**IP1 Interrupt Status**	IP1 Interrupts
0xEC	**IP1 Interrupt Control**	IP1 Interrupt Masks
0xF0	Reserved	Reserved for IP1
0xF4	**IP1 Transfer Monitor**	IP1 ACK*/Channel Transfer & Activity Monitor
0xF8-13F	Reserved	Reserved for IP1
Offset	**Register**	**Description**
0x140	**IP2 Control0**	IP2 Channel behavior & data flow control
0x144	**IP2 Control1**	IP2 Channel behavior & data flow control
0x148	**IP2 Interrupt Status**	IP2 Interrupts
0x14C	**IP2 Interrupt Control**	IP2 Interrupt Masks
0x150	Reserved	Reserved for IP2
0x154	**IP2 Transfer Monitor**	IP2 ACK*/Channel Transfer & Activity Monitor
0x158-19F	Reserved	Reserved for IP2
96 bytes for each set of IP Registers		
Offset	**Register**	**Description**
0x1A0	**IP3 Control0**	IP3 Channel behavior & data flow control
0x1A4	**IP3 Control1**	IP3 Channel behavior & data flow control
0x1A8	**IP3 Interrupt Status**	IP3 Interrupts
0x1AC	**IP3 Interrupt Control**	IP3 Interrupt Masks
0x1B0	Reserved	Reserved for IP3
0x1B4	**IP3 Transfer Monitor**	IP3 ACK*/Channel Transfer & Activity Monitor
0x1B8-1FF	Reserved	Reserved for IP3
Offset	**Register**	**Description**
0x200	**IP4 Control0**	IP4 Channel behavior & data flow control
0x204	**IP4 Control1**	IP4 Channel behavior & data flow control
0x208	**IP4 Interrupt Status**	IP4 Interrupts
0x20C	**IP4 Interrupt Control**	IP4 Interrupt Masks
0x210	Reserved	Reserved for IP4
0x214	**IP4 Transfer Monitor**	IP4 ACK*/Channel Transfer & Activity Monitor
0x218-25F	Reserved	Reserved for IP4
416 reserved bytes		
Offset	**Register**	**Description**
0x260-3FF	Reserved	Reserved

NOTE: Global and IP register locations are the same for the VPX2IP, PCIe3IP and PCIe5IP. Registers for IP's that are not implemented are reserved. Writes to reserved registers are dropped and reads return 0x0000_0000's. For the VPX2IP register locations between 0x140 to 0x25F are reserved. For the PCIe3IP register locations between 0x1A0 to 0x25F are reserved.

Global/Board Level Registers

Offsets 0x00 to 0x7F

Table 12: Switch and LED register (SLR) – Offset 0x000

Bit(s)	Description		Attribute	Default
31:28	**Reserved**		RO	0h
27:24	**LED Select** The decode selects summarize the LED settings. See the LED Decode table for details of signals/status bits that drive the PCIeIP LED's for each setting.		R/W	0h
	0000	Link and board status		
	0001	User LED control [7:0] – bits [23:16] of this register		
	0010	User switch settings – '1' on pin turns on LED		
	0011	FPGA/FLASH switch settings – '1' turns on LED		
	0100	IP0 status		
	0101	IP1 status		
	0110	IP2 status		
	0111	Reserved		
	1000	Posted header credits – available from host		
	1001	Non-posted header credits – available from host		
	1010	Completion header credits – available from host		
	1011	Posted data credits – available from host		
	1100	Non-posted data credits – available from host		
	1101	Completion data credits – available from host		
	1110	Scratch0 register value		
	1111	Scratch1 register value		
23:16	**User LED Control**		R/W	0h
	0	Off		
	1	On		
15:14	**Reserved**		RO	0h
13:12	**FLASH Select** [1:0] – Read Only bits that reflect SW2 [7:6] switch settings. Since the factory default switch setting is on/'1" these pins are pulled-up and seen as binary 2'b11. The intent of these bits (switch settings) is to select PCIeIP FPGA configuration/bit-map at boot-up. SW2 [7:6] = FLASH Select [1:0] decodes. Currently only the 11 setting is implemented and all switch settings load 11 map.		RO	3h
	00	4K space with 1 PH/PD & 1 NPH/NPD credit		
	01	4K space with 127 PH/PD & 32 NPH/NPD credits		
	10	32 MB space 1 PH/PD & 1 NPH/NPD credit		
	11	32 MB space with 127 PH/PD & 32 NPH/NPD credits		
11	**Reserved**		RO	0h
10:8	**FPGA Select** [2:0] - TBD (No logic implemented at this time). VPX2IP/PCIe3IP – Value read = SW2 [7:5] switch values. PCIe5IP – Value read: bits[10:9] = 00, bit[8] = SW2 [5]		RO	xh
7:0	**User Switch Values/Settings**		RO	xh

Table 13: Reserved Register - Offsets 0x004

Bit(s)	Description	Attribute	Default
31:0	-	RO	0h

Interrupt Status Register (ISR)

Offset 0x008

When any one or more of these interrupt bits are set, an MSI or INTA# interrupt packet will be generated and sent to the Host. The pre-masked source of these bits is contained in the relative IPx Interrupt Status Register (i.e. IP0 ISR = 0x088). The Mask bits for these bits are contained in the relative IPx Interrupt Control Register (i.e. IP0 ICR = 0x08C).

Table 14: Interrupt Status Register (ISR)

Bit(s)	Description		Attribute	Default
If any of these bits/interrupts are asserted, an MSI will be sent to the system.				
31:20	**Reserved**		RO	0h
19:16	**IP4 Interrupts same as IP0's**		RW1C	0h
15:12	**IP3 Interrupts same as IP0's**		RW1C	0h
11:8	**IP2 Interrupts same as IP0's**		RW1C	0h
7:4	**IP1 Interrupts same as IP0's**		RW1C	0h
3	**IP0 Force/P5VGOODn Interrupt Status**		RW1C	0h
	0	IP0 Force/P5VGOODn Interrupt bit is not asserted or is disabled/masked		
	1	IP0 Force/P5VGOODn Interrupt bit is asserted and is enabled/unmasked		
2	**IP0 Bus Error Interrupt Status**		RW1C	0h
	0	IP0 bus error interrupt not asserted or is disabled/masked		
	1	IP0 bus error interrupt asserted and enabled/unmasked		
1	**IP0 IntReq1* Interrupt Status**		RW1C	0h
	0	IP0 Interrupt pin not asserted or is disabled/masked		
	1	IP0 Interrupt pin asserted and enabled/unmasked		
	NOTE: pin assertion is true low, bit value is the pin state inverted.			
0	**IP0 IntReq0* Interrupt Status**		RW1C	0h
	0	IP0 Interrupt pin not asserted or is disabled/masked		
	1	IP0 Interrupt pin asserted and enabled/unmasked		
	NOTE: pin assertion is true low, bit value is the pin state inverted.			

Interrupt Control Register (ICR)

Offset 0x00C

Table 15: Interrupt Control Register

Bit(s)	Description	Attribute	Default
31:14	**Reserved**	RO	0h
13	**Interrupt Aggregation Enable** 0: Interrupt Aggregation off, Interrupt Aggregation timer & logic disabled. Interrupts will be generated upon occurrence or after the interrupt de-assert time has expired (time relative to last de-assertion). 1: Interrupt Aggregation on, Interrupt Aggregation timer & logic enabled.	R/W	0h
12	**Reserved**	RO	0h
11:8	**Interrupt Aggregation Timer** These bits set the Interrupt Aggregation timer, i.e. the rate or pace at which an interrupt (INTA# or MSI) may generated and sent to the Host. An MSI or INTA# assert packet is sent if an interrupt in the ISR (0x008) is set and the Interrupt Aggregation logic is enabled when this timer expires. INTA# de-assert packets are sent upon W1C of ISR occurrence. 0000: 32 us; 0001: 64 us; 0010: 128 us; 0011: 256 us; 0100: 512 us; 0101: 1 MS; 0110: 2 MS; 0111: 4 MS; 1000: 8 MS; 1001: 16 MS; 1010: 32 MS; 1011: 64 MS; 1100: 128 MS; 1101: 256 MS; 1110: 512 MS; 1111: 1 Sec.	R/W	0h
7:3	**Reserved**	RO	0h
2:0	**Interrupt De-assert Time** Minimum time delay from an interrupt (INTA# or MSI) de-assertion to interrupt assertion. An interrupt assertion following a de-assertion in the minimum time may be caused by: • The interrupt source for the previous interrupt is still asserted when the W1C of the ISR, to clear it, is executed. • Multiple Interrupts are asserted, but not all are W1C. • A new/different interrupt occurs while (or after) the W1C Interrupt Status register (0x008) bit or bits are cleared. **NOTE:** Delays are approximate due to clock synchronization delays. 000: 96 nanoseconds (IPCLK = 32 MHz) or 288 nanoseconds (IPCLK = 8 MHz) 001: 544 nanoseconds 010: 1.06 microseconds 011: 2.08 microseconds 100: 4.13 microseconds 101: 8.22 microseconds 110: 16.42 microseconds 111: 32.67 microseconds	R/W	0h

Table 16: Reserved Register - Offset 0x010

Bit(s)	Description	Attribute	Default
31:0	-	RO	0h

Table 17: Scratch0 Register (SRH0) - Offset 0x014

Bit(s)	Description	Attribute	Default
31:0	**Scratch0 register** General purpose Read/Writeable register for programming use. Bits [7:0] may also be used to turn on LED's by setting their values to logic '1' and programming Switch and LED (offset 0x000) register **LED Select** bits [27:24] = 1110.	R/W	0h

Table 18: Scrach1 Register (SRH1) - Offset 0x018

Bit(s)	Description	Attribute	Default
31:0	Scratch1 register General purpose Read/Writeable register for programming use. Bits [7:0] may also be used to turn on LED's by setting their values to logic '1' and programming Switch and LED (offset 0x000) register LED Select bits [27:24] = 1111.	R/W	0h

Table 19: Version ID Register (VID) - Offset 0x01C

Bit(s)	Description		Attribute	Default
31:20	**Reserved**		RO	0h
19:16	**IP Slots Available** Indicates number of slots available on carrier		RO	2h or 3h or 5h
	0x2h	VPX2IP		
	0x3h	PCIe3IP		
	0x5h	PCIe5IP		
15:8	**CPLD Version ID** [15:12] Major Rev ID [11:8] Minor Rev ID		RO	10h
7:0	**FPGA Version ID** [7:4] Major Rev ID [3:0] Minor Rev ID		RO	10h
	0x10h	VPX2IP		
	0x10h	PCIe3IP		
	0x10h	PCIe5IP		

Table 20: Reserved Registers - Offsets 0x020 - 0x07F

Bit(s)	Description	Attribute	Default
31:0	-	RO	0h

IP Control0 Register (IPx CR0)

Table 21: IP Control0 Register
IP0/IP1/IP2/IP3/IP4 – Offset 0x0080/0x0E0/0x140/0x1A0

Bit(s)	Description		Attribute	Default
31:19	**Reserved**		RO	0h
18	**IP Reset* Pin Status**		RO	0h
	0	IP Reset* is de-asserted (1)		
	1	IP Reset* is asserted (0)		
17	**Reset IP and IP Channel**		R/W	0h
	0	Normal IP Reset* Operation		
	1	IP Reset* is driven low and all IP channel related logic in the FPGA is held in reset when this bit is set to 1. Upon writing this bit back to zero, IP Reset* is de-asserted after the expiration of that channel's 256 millisecond reset counter to ensure Reset* meets the 200 MS minimum assertion time. IP Reset* is always driven low when PCIe reset asserted.		
16	**Reset IP**		R/W	0h
	0	Normal IP Reset* Operation		
	1	Reset* is driven low when this bit is set to 1, upon writing this bit back to zero, IP Reset* is de-asserted. Meeting the minimum assertion time is the responsibility of the software. IP Reset* is always driven low when PCIe reset asserted.		
15:14	**Reserved**		RO	0h
13	**ACK* Activity Count Enable**		R/W	0h
	0	ACK* Activity Counter disabled		
	1	ACK* Activity Counter enabled		
12	**Bus Error Timeout Select**		R/W	0h
	0	127 IP Clocks – 4 microseconds at 32 MHz		
	0	63 IP Clocks – 8 microseconds at 8 MHz		
	1	255 IP Clocks – 8 microseconds at 32MHz		
	1	127 IP Clocks – 16 microseconds at 8 MHz		
11:10	**Reserved**		RO	0h
9	**IP Clock Disable**		R/W	0h
	0	IP Clock enabled		
	1	IP Clock disabled		
8	**Clock SEL**		R/W	0h
	0	8 MHz IP Clock		
	1	32 MHz IP Clock		
7	**Reserved**		RO	0h
6:5	**Increment Write Disable – Word Address Offset** Selects which 16-bit word is accessed on a 32-bit or 64-bit write relative to the accesses address. Only has an effect when the Address Increment Write Disable bit is asserted.		R/W	0h
	For 64-Bit Accesses			
	11	Word3 is accessed four times		
	10	Word2 is accessed four times		

	01	Word1 is accessed four times		
	00	Word0 is accessed four times		
	For 32-Bit Accesses			
	X1	Word1 is accessed twice		
	X0	Word0 is accessed twice		
	32-bit data accesses are required to be 32-bit address aligned			
	64-bit data accesses are required to be 64-bit address aligned			
4	**Address Increment Write Disable** For a 32- or 64-bit PCI Express write access, each IP Write access uses the same Word aligned address. The Word address used is specified with the Increment Write Disable – Word Address Offset bits (bits 6:5 above).		R/W	0h
3:2	**Reserved**		RO	0h
1	**Word Swap**		R/W	0h
	0	Normal Operation		
	1	Upper word is swapped with lower word for each 32-bit (Dword) transfer.		
0	**Byte Swap**		R/W	0h
	0	Normal Operation		
	1	Byte 0 is swapped with Byte 1 for the first word & Byte 2 is swapped with byte 3 for the second word of a 32-bit transfer		

IP Control1 Register (IPx CR1)

Table 22: IP Control1 Register
IP0/IP1/IP2/IP3/IP4 – Offset 0x084/0x0E4/0x144/0x1A4/0x204

Bit(s)	Description		Attribute	Default
31:29	**IPx Data-in Timing Mux**		R/W	0h
	000	Normal Timing Mode: **WARNING:** non-zero values not intended for normal use. Errors will occur with non-zero values. Multiplexors added the PCIeIP Data and ACK* paths to enable the IP interfaces to meet/exceed the 0 ns setup timing.		
28:7	**Reserved**		RO	0h
6:5	**Increment Read Disable – Word Address Offset** Selects which 16-bit word is accessed on a 32-bit or 64-bit read relative to the accesses address. Only has an effect when the Address Increment Read Disable bit is asserted.		R/W	0h
	For 64-Bit Accesses			
	11	Word3 is accessed four times		
	10	Word2 is accessed four times		
	01	Word1 is accessed four times		
	00	Word0 is accessed four times		
	For 32-Bit Accesses			
	X1	Word1 is accessed twice		
	X0	Word0 is accessed twice		
	32-bit data accesses are required to be 32-bit address aligned			
	64-bit data accesses are required to be 64-bit address aligned			
4	**Address Increment Read Disabled** For a 32- or 64-bit PCI Express read access each IP Read access uses the same Word aligned address. The Word address used is specified with the Increment Read Disable – Word Address Offset bits (bits 6:5 above). **NOTE:** See Write version in IPx CR0		R/W	0h
3:0	**User Control/Status Bits** Read/Writable bits for user use		R/W	0h

IP Interrupt Status Register (IPx ISR)

These bits contain the Interrupt value/setting before the Mask is applied. See PCIeIP Interrupts section for additional details regarding these interrupts.

Table 23: IP Interrupt Status Register
IP0/IP1/IP2/IP3/IP4 - Offset 0x088/0x0E8/0x148/0x1A8/0x208

Bit(s)	Description		Attribute	Default
31:6	**Reserved for additional IPx Interrupts**		RO	0h
5	**IPx Bus Error Interrupt Read**		RW1C	0h
	0	Indicates a Bus Error has not occurred on a read		
	1	Indicates a Bus Error has occurred on a read		
4	**IPx Bus Error Interrupt Write**		RW1C	0h
	0	Indicates a Bus Error has not occurred on a write		
	1	Indicates a Bus Error has occurred on a write.		
3	**IPx Force/P5VGOODn Interrupt**		R/W	0h
	0	Interrupt not asserted		
	1	Interrupt asserted		
	Interrupt that may be set from the following two sources: 1. User programmable interrupt set via this bit “IPx Force” (IPx ISR bit [3]), which provides the user a way to assert/de-assert an interrupt in a controlled fashion for development or design purposes 2. Is set due to detection of the 5-vold power source being out of range			
2	**IPx Bus Error Interrupt**		RW1C	0h
	0	No Bus Error has occurred		
	1	Bus Error has occurred since reset or last W1C		
	May also be cleared by W1C of ISR (0x008) bit [2] (IP0), bit [6] (IP1), bit [10] (IP2) when in Level mode IPx ICR (0x08C/0x0BC/0x0EC) bit [10] = 1. This prevents more than one interrupt being generated per Bus Error in Level Mode. **NOTE:** a hardware Bus Error is an event which is captured and held in this register and forwarded on to ISR if not masked			
1	**IPx IntReq1* Signal Status/Value**		RO	-
	0	Pin value is high/1/not asserted		
	1	Pin value is low/0/asserted		
0	**IPx IntReq0* Signal Status/Value**		RO	-
	0	Pin value is high/1/not asserted		
	1	Pin value is low/0/asserted		

IP Interrupt Control Register (IPx ICR)

Mask bits either mask or allow the Interrupt to progress to the Interrupt Status Register located at offset 0x008. If an Interrupt occurs and is not masked by this register, either a legacy INTA# or MSI Interrupt will be generated.

Edge/Level behavior bits – An interrupt in Level mode may generate additional interrupts (after the de-assert time expires) if its ISR bit is cleared, but it's (this) source interrupt is still set. In Edge mode, only one interrupt is generated per source interrupt assertion. **NOTE:** See Interrupts section of the data sheet for further details.

Table 24: IP Interrupt Control Register
IP0/IP1/IP2/IP3/IP4 – Offset 0x08C/0x0EC/0x14C/0x1AC/0x20C

Bit(s)	Description		Attribute	Default
31:12	**Reserved**		RO	0h
11	**IPx Force/P5VGOODn Interrupt Edge/Level Select**		RW	0h
	0	Edge Interrupt		
	1	Level Interrupt		
10	**IPx Bus Error Interrupt Edge/Level Select**		RW	0h 0 1
	0	Edge Interrupt		
	1	Level Interrupt		
9	**IPx IntReq1* Interrupt Edge/Level Select**		RW	0h 0 1
	0	Edge Interrupt		
	1	Level Interrupt		
8	**IPx IntReq0* Interrupt Edge/Level Select**		RW	0h
	0	Edge Interrupt		
	1	Level Interrupt		
7:4	**Reserved**		RO	0h
3	**IPx Force/P5VGOODn Interrupt Enable**		RW	0h
	0	Interrupt Disabled/Masked		
	1	Interrupt Enabled/Unmasked		
2	**IPx Bus Error Interrupt Enable**		RW	0h
	0	Interrupt Disabled/Masked		
	1	Interrupt Enabled/Unmasked		
1	**IPx IntReq1* Enable**		RW	0h
	0	Interrupt Disabled/Masked		
	1	Interrupt Enabled/Unmasked		
0	**IPx IntReq0* Enable**		RW	0h
	0	Interrupt Disabled/Masked		
	1	Interrupt Enabled/Unmasked		

IPx Channel Transfer Monitor Register (IPx CTM)

This register provides control of and access to IPx's channel ACK* counter; additionally, it provides channel activity status.

NOTE: See IP Channel Transfer Activity Monitor and Logic section for further application/usage details.

Table 25: IPx Channel Transfer Monitor Register (IPx CTM)
IP0/IP1/IP2/IP3/IP4 - Offset 0x094/0x0F4/0x154/0x1B4/0x214

Bit(s)	Description		Attribute	Default
31:29	**Reserved**		RO	0h
28	**IPx Channel Busy**		RO	0h
	0	No transfers pending in IPx FIFO or channel		
	1	Indicates at least one transfer is still pending in the channel's FIFO or the IP interface is still processing an IP read or write cycle		
27:15	**Reserved**		RO	0h
24	**IPx ACK* Counter Clear#/Enable**		R/W	0h
	0	Clears and holds IPx ACK* Counter at 0x0000h		
	1	Enables IPx ACK* Counter to count ACK*s		
23:16	**Reserved**		RO	0h
15:0	**IPx ACK* Count** Once enabled by setting IPx ACK* Counter Clear#/Enable:		RO	0h
	1	Provides a count of how many ACK* have been returned by the IP in response to an ID, IO, INT, or MEM cycle		

LED Decode Tables

Table 26: LED Link and Board Status

LED Select = 0000 - Link and board status		
LED	**Signal Name**	**Description**
LED [7:4]	-	Reserved – 0x0 - LED's are off
LED [3]	dl_up	Data Link Layer is up
LED [2]	l0	L0 – L0 state has been reached
LED [1]	poll	Polling – PCIe core is in the polling state
LED [0]	pll_lk	PLL Lock – PCIe SERDES clock is locked to PCIe ref clock

Table 27: USER LED Control [7:0] Register Bits

LED Select = 0001 - USER LED Control [7:0] register bits		
LED	**Signal Name**	**Description**
LED [7:0]	USER LEDs [7:0]	**USER LED Control [7:0]** register bits control LED [7:0]

Table 28: DipSwitch Setting

LED Select = 0010 – DipSwitch Setting		
LED [7:0]	dip_switch [7:0]	LED is on when switch pin is logic high '1'

Table 29: FPGA and Flash Switch Setting

LED Select = 0011 – FPGA and Flash Switch Setting		
LED	**Signal Name**	**Description**
LED [7:6]	-	Always off
LED [5:4]	flash_sel [1:0]	LED is on when switch pin is logic high '1'
LED [3]	-	Always off
LED [2:0]	fpga_sel [2:0]	LED is on when switch pin is logic high '1'

Table 30: LED Select (0100, 0101, & 0110)

LED Select = 0100 - IP0 Status LED Select = 0101 - IP1 Status LED Select = 0110 - IP2 Status		
LED	**Signal Name**	**Description**
LED [7]	IPx Force Interrupt	Force Interrupt asserted – ISR (offset 0x008) bit[3]
LED [6]	IPx Bus Error	Bus Error Interrupt asserted – ISR (offset 0x008) bit[2]
LED [5]	IPx Intreq1*	IRQN[1] asserted – ISR (offset 0x008) bit[1]
LED [4]	IPx Intreq0*	IRQN[0] asserted – ISR (offset 0x008) bit[0]
LED [3]	IPx_MemSpace	Memory access is in process or was the last IP access
LED [2]	IPx_IntSpace	INT access is in process or was the last IP access
LED [1]	IPx_IOSpace	IO access is in process or was the last IP access
LED [0]	IPx_IDSpace	ID access is in process or was the last IP access

Table 31: LED Select (0111) Reserved

LED Select = 0111 - Reserved		
LED [7:0]	Reserved	All LED's off if selected

Table 32: Posted Header Credits

LED Select = 1000 – Posted Header Credits		
LED	**Signal Name**	**Description**
LED [7:0]	tx_ca_ph[7:0]	**Posted Header Credits** – Host Posted header credits available, i.e. provided from/by host. Actual Internal bus is [8:0]. When [8] = 1 infinite credits provided and all LED's turned on.

Table 33: Non-Posted Header Credits

LED Select = 1001 – Non-Posted Header Credits		
LED	**Signal Name**	**Description**
LED [7:0]	tx_ca_nph[7:0]	**Non Posted Header Credits** – Host Non Posted header credits available, i.e. provided from/by host. Actual Internal bus is [8:0]. When [8] = 1 infinite credits provided and all LED's turned on.

Table 34: Completion Header Credits

LED Select = 1010 – Completion Header Credits		
LED	**Signal Name**	**Description**
LED [7:0]	tx_ca_cplh[7:0]	**Completion Header Credits** – Host Completion header credits available, i.e. provided from/by host. Actual Internal bus is [8:0]. When [8] = 1 infinite credits provided and all LED's turned on.

Table 35: Posted Data Credits

LED Select = 1011 – Posted Data Credits		
LED	**Signal Name**	**Description**
LED [7:0]	tx_ca_pd[7:0]	**Posted Data Credits** – Lower 8 LSBs of Posted Data Credits available - provided by host. Internal bus is [12:0]. When [12] =1 infinite credits provided and all LED's turned on. When [12] = 0 and any [11:8] = 1 all LED's are turned on except LED [0].

Table 36: Non-Posted Data Credits

LED Select = 1100 – Non-Posted Data Credits		
LED	**Signal Name**	**Description**
LED [7:0]	tx_ca_npd[7:0]	**Non Posted Data Credits** – Lower 8 LSBs of Non Posted Data Credits available - provided by host. Internal bus is [12:0]. When [12] =1 infinite credits provided and all LED's turned on. When [12] = 0 and any [11:8] = 1 all LED's are turned on except LED [0].

Table 37: Completion Data Credits

LED Select = 1101 – Completion Data Credits		
LED	**Signal Name**	**Description**
LED [7:0]	tx_ca_cpld[7:0]	**Completion Data Credits** – Lower 8 LSBs of Completion Data Credits available - provided by host. Internal bus is [12:0]. When [12] =1 infinite credits provided and all LED's turned on. When [12] = 0 and any [11:8] = 1 all LED's are turned on except LED [0].

Table 38: Scratch0 Register

LED Select = 1110 – Scratch0 Register (offset 0x014) value		
LED	**Signal Name**	**Description**
LED [7:0]	Scratch0[7:0]	LED is on when Scratch0 register bit is logic high '1'

Table 39: Scratch1 Register

LED Select = 1111 – Scratch1 Register (offset 0x018) value		
LED	**Signal Name**	**Description**
LED [7:0]	Scratch1[7:0]	LED is on when Scratch1 register bit is logic high '1'

Pin Assignments

PCIeIP Carrier IP Logic Connector Pin Assignment

The figure below gives the pin assignments for each IP's Module Logic Interface on the PCIeIP. Pins marked n/c below are defined by the specification, but not used on the PCIeIP.

Table 40: PCIeIP Carrier IP Logic Connector Pin Assignment

GND	GND	1	26
CLK	+5V	2	27
Reset*	R/W*	3	28
D0	IDSel*	4	29
D1	n/c (DMAReq0*)	5	30
D2	MemSel*	6	31
D3	n/c (DMAReq1*)	7	32
D4	IntSel*	8	33
D5	n/c (DMACK*)	9	34
D6	IOSEL*	10	35
D7	n/c (Reserved)	11	36
D8	A1	12	37
D9	n/c (DMAEnd*)	13	38
D10	A2	14	39
D11	n/c (Error*)	15	40
D12	A3	16	41
D13	IntReq0*	17	42
D14	A4	18	43
D15	IntReq1*	19	44
BS0*	A5	20	45
BS1*	n/c (Strobe)	21	46
-12V	A6	22	47
+12V	ACK*	23	48
+5V	n/c (Reserved)	24	49
GND	GND	25	50

NOTE: The no-connect (n/s) signals above are defined by the IP Module Logic Interface Specification, but not used by this IP, specification signals are noted in ( ) parenthesis for convenience.

The layout of the pin numbers in this table corresponds to the physical placement of pins on the IP connector. Thus, the above table may be used to easily locate the physical pin corresponding to a desired signal. Pin 1 is marked with a square pad on the IP Module.

PCIeIP IP Carrier I/O Connector to 50-Pin Header Assignment

The Figure 5 below gives the default pin assignments for each IP's Module I/O Interface to 50 pin Header for each of the PCIeIP IP channels (PCIe3IP illustrated).

Table 41: PCIeIP IP Carrier I/O Connector to 50-Pin Header Assignment

PCIe3IP IP Carrier I/O SMT to 50 pin Header Connections		
IP[2:0] Connector/Pins		Pin Signal Names are named differentially as each pair is routed to within .002" to enable up to 25 differential pairs
P[2:0]b	J[2:0]	
1 / 2	A1 / A2	IO[2:0]_0P / IO[2:0]_0N
3 / 4	A3 / A4	IO[2:0]_1P / IO[2:0]_1N
5 / 6	A5 / A6	IO[2:0]_2P / IO[2:0]_2N
7 / 8	A7 / A8	IO[2:0]_3P / IO[2:0]_3N
9 / 10	A9 / A10	IO[2:0]_4P / IO[2:0]_4N
11 / 12	A11 / A12	IO[2:0]_5P / IO[2:0]_5N
13 / 14	A13 / A14	IO[2:0]_6P / IO[2:0]_6N
15 / 16	A15 / A16	IO[2:0]_7P / IO[2:0]_7N
17 / 18	A17 / A18	IO[2:0]_8P / IO[2:0]_8N
19 / 20	A19 / A20	IO[2:0]_9P / IO[2:0]_9N
21 / 22	A21 / A22	IO[2:0]_10P / IO[2:0]_10N
23 / 24	A23 / A24	IO[2:0]_11P / IO[2:0]_11N
25 / 50*	A25 / A50*	IO[2:0]_12P / IO[2:0]_12N
26 / 27	A26 / A27	IO[2:0]_13P / IO[2:0]_13N
28 / 29	A28 / A29	IO[2:0]_14P / IO[2:0]_14N
30 / 31	A30 / A31	IO[2:0]_15P / IO[2:0]_15N
32 / 33	A32 / A33	IO[2:0]_16P / IO[2:0]_16N
34 / 35	A34 / A35	IO[2:0]_17P / IO[2:0]_17N
36 / 37	A36 / A37	IO[2:0]_18P / IO[2:0]_18N
38 / 39	A38 / A39	IO[2:0]_19P / IO[2:0]_19N
40 / 41	A40 / A41	IO[2:0]_20P / IO[2:0]_20N
42 / 43	A42 / A43	IO[2:0]_21P / IO[2:0]_21N
44 / 45	A44 / A45	IO[2:0]_22P / IO[2:0]_22N
46 / 47	A46 / A47	IO[2:0]_23P / IO[2:0]_23N
48 / 49	A48 / A49	IO[2:0]_24P / IO[2:0]_24N
***To meet differential routing tolerance IO[2:0]_12P/N are not consecutive**		

To enable differential signaling on a per IP (port) basis, each IP channel's PCB traces are matched to within .002" pin to pin (i.e. P[x]b to J[x]). Additionally, each named differential pair is routed with controlled impedance other than the 25,50. Non-differential signaling maybe used which provides 50 single ended signals for use.

VPX2IP IP Carrier VHDCI Connector Assignment: Option 1

Table 42 illustrates the standard VPX2IP IP Carrier I/O SMT to Condo Header IO Connections. J3A/J3B connect IP0/IP1 respectively. Note: VHDCI IO naming is slightly different than header carrier types

Table 42: VPX2IP IP Carrier Bezel 68-Pin Header Assignment

VPX2IP IP Carrier I/O SMT to 68 pin VHDCI Connections		
IP[1:0] Connector/Pins		Pin Signal Names are named differentially as each pair is routed to within .002" to enable up to 25 differential pairs
P[0:1]b	J3[A:B]	
1 / 2	1 / 35	IO[2:0]_0P / IO[2:0]_0N
26 / 27	2 / 36	IO[2:0]_1P / IO[2:0]_1N
3 / 4	3 / 37	IO[2:0]_2P / IO[2:0]_2N
28 / 29	4 / 38	IO[2:0]_3P / IO[2:0]_3N
5 / 6	5 / 39	IO[2:0]_4P / IO[2:0]_4N
30 / 31	6 / 40	IO[2:0]_5P / IO[2:0]_5N
7 / 8	7 / 41	IO[2:0]_6P / IO[2:0]_6N
32 / 33	8 / 42	IO[2:0]_7P / IO[2:0]_7N
9 / 10	9 / 43	IO[2:0]_8P / IO[2:0]_8N
34/ 35	10 / 44	IO[2:0]_9P / IO[2:0]_9N
11 / 12	11 / 45	IO[2:0]_10P / IO[2:0]_10N
36 / 37	12 / 46	IO[2:0]_11P / IO[2:0]_11N
13 / 14	13 / 47	IO[2:0]_12P / IO[2:0]_12N
38 /39	14 / 48	IO[2:0]_13P / IO[2:0]_13N
15 / 16	15 / 49	IO[2:0]_14P / IO[2:0]_14N
40 / 41	16 / 50	IO[2:0]_15P / IO[2:0]_15N
17 / 18	17 / 51	IO[2:0]_16P / IO[2:0]_16N
42 / 43	18 / 52	IO[2:0]_17P / IO[2:0]_17N
19 / 20	19 / 53	IO[2:0]_18P / IO[2:0]_18N
44 / 45	20 / 54	IO[2:0]_19P / IO[2:0]_19N
21 / 22	21 / 55	IO[2:0]_20P / IO[2:0]_20N
46 / 47	22 / 56	IO[2:0]_21P / IO[2:0]_21N
23 / 24	23 / 57	IO[2:0]_22P / IO[2:0]_22N
48 / 49	24 / 58	IO[2:0]_23P / IO[2:0]_23N
25* / 50*	25 / 59	IO[2:0]_24P / IO[2:0]_24N
***To meet differential routing tolerance IO 24P/N are not consecutive unused upper VHDCI pins are grounded**		

VPX2IP IP Carrier Rear IO Connector Assignment: Option 2

Table 43: VPX2IP IP Carrier Rear IO Connector Assignment: Option 2

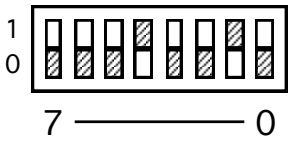
VPX2IP IP Carrier I/O SMT to Rear IO Connections – Option 2 provides 32 differential pairs			
IP[0] Connector/Pins			Pin Signal Names are named differentially as each pair is routed to within .002" to enable up to 32 differential pairs
P0B	VPX_P2	VPX_BP	
1 / 2	A1 / B1	A1 / B1	IO0_0P / IO0_0N
3 / 4	D1 / E1	E1 / F1	IO0_1P / IO0_1N
5 / 6	B2 / C2	C2 / D2	IO0_2P / IO0_2N
7 / 8	E2 / F2	G2 / H2	IO0_3P / IO0_3N
9 / 10	A3 / B3	A3 / B3	IO0_4P / IO0_4N
11 / 12	D3 / E3	E3 / F3	IO0_5P / IO0_5N
13 / 14	B4 / C4	C4 / D4	IO0_6P / IO0_6N
15 / 16	E4 / F4	G4 / H4	IO0_7P / IO0_7N
17 / 18	A5 / B5	A5 / B5	IO0_8P / IO0_8N
19 / 20	D5 / E5	E5 / F5	IO0_9P / IO0_9N
21 / 22	B6 / C6	C6 / D6	IO0_10P / IO0_10N
23 / 24	E6 / F6	G6 / H6	IO0_11P / IO0_11N
25 / 50*	A7 / B7*	A7 / B7*	IO0_12P / IO0_12N
26 / 27	D7 / E7	E7 / F7	IO0_13P / IO0_13N
28 / 29	B8 / C8	C8 / D8	IO0_14P / IO0_14N
30 / 31	E8 / F8	G8 / H8	IO0_15P / IO0_15N
32 / 33	A9 / B9	A9 / B9	IO0_16P / IO0_16N
34 / 35	D9 / E9	E9 / F9	IO0_17P / IO0_17N
36 / 37	B10 / C10	C10 / D10	IO0_18P / IO0_18N
38 / 39	E10 / F10	G10 / H10	IO0_19P / IO0_19N
40 / 41	A11 / B11	A11 / B11	IO0_20P / IO0_20N
42 / 43	D11 / E11	E11 / F11	IO0_21P / IO0_21N
44 / 45	B12 / C12	C12 / D12	IO0_22P / IO0_22N
46 / 47	E12 / F12	G12 / H12	IO0_23P / IO0_23N
48 / 49	A13 / B13	A13 / B13	IO0_24P / IO0_24N
*To meet differential routing tolerance IO[0]_12P/N are not consecutive			
IP[1] Connector/Pins			Pin Signal Names are named differentially as each pair is routed to within .002" to enable up to 32differential pairs
P1B	VPX_P2	VPX_BP	
1 / 2	D13 / E13	E13 / F13	IO1_0P / IO1_0N
3 / 4	B14 / C14	C14 / D14	IO1_1P / IO1_1N
5 / 6	E14 / F14	G14 / H14	IO1_2P / IO1_2N
7 / 8	A15 / B15	A15 / B15	IO1_3P / IO1_3N
9 / 10	D15 / E15	E15 / F15	IO1_4P / IO1_4N
11 / 12	B16 / C16	C16 / D16	IO1_5P / IO1_5N
13 / 14	E16 / F16	G16 / H16	IO1_6P / IO1_6N
NOTE: VPX_BP = VPX Back Plane side connectivity provided for user convenience; Module 1 I/O can be swapped in place of I/O from module 0 to balance signaling. See VPX2IP IP0/IP1 connectivity options section/diagram and/or contact Dynamic Engineering for customization			

PCIe3IP Board Features

PCIe3IP DipSwitches

There are two DIP Switches on PCIe3IP, each with 8 switches. They are labeled SW1 and SW2 with bit numbers 7 to 0 and 1/0 for on/off in the silk screen.

Figure 3: PCIe3IP DipSwitch Settings



SW1 is for user purposes. The settings of SW1 can be seen/used by reading the Switch and LED register [7:0] bits which correspond to SW1 [7:0] bit positions. When a switch is in the 0 (off) position, the bit value read is 0. When in the 1 (on) position, the value read is 1. Switch values can also be displayed on LED [7:0] by setting the LED Select value (Switch and LED register bits [27:24]) = 0010.

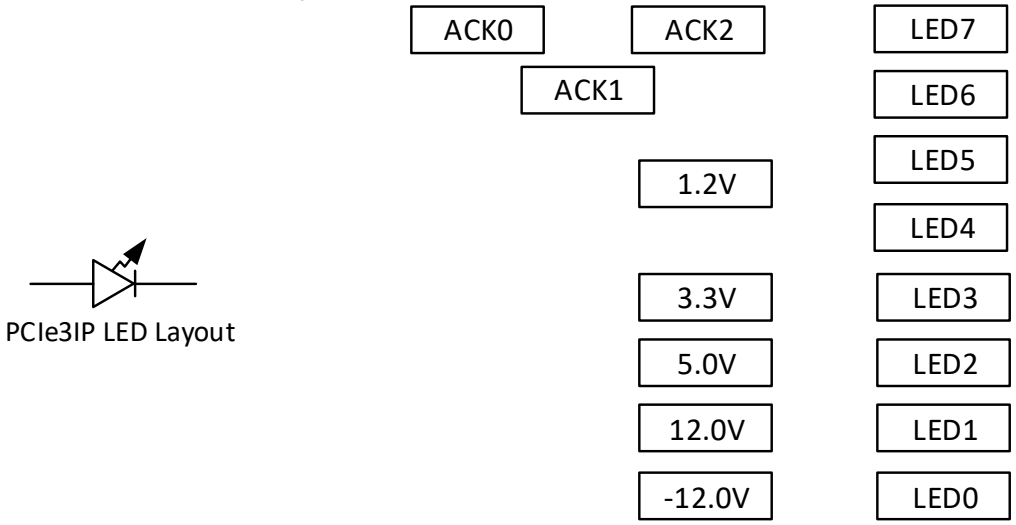
SW2 bits configure PCIe3IP (see Switch and LED register) as follows:

- SW2 [7:6] bit positions: select the Flash image to be used to configure the FPGA at power up and can be read by reading the Switch and LED register [13:12] bits.
- SW2 [5:3]: are used to configure logic inside the FPGA.
  - These switch settings can be seen by reading the Switch and LED register [10:8] bits.
- SW2 [2:0]: selects either 5.0V or 3.3V to be the Bus Termination voltage for each IP.
  - SW2 [2:0] corresponds one to one with the selection for to IP [2:0].
  - When the switch is in the closed position “0” a FET is turned on resulting in 5V referenced logic.
  - When the switch is ‘1’ position the FET is disabled and the logic reference is 3.3V.
  - The ES[2:0]V pads on the board can be checked with a multi-meter to make sure you have the expected bus reference/termination voltage.
  - The PCIe3IP ships with SW2[2:0] = 000, all IP set to operate with 5V signaling/bus termination.
    NOTE: The 3.3V reference is provided through a blocking diode which results in a small voltage drop.

PCIe3IP LEDs

The PCIe3IP has sixteen LED's. All the LED's are in the upper right corner of the PCIe3IP board and are oriented as follows:

Figure 4: PCIe3IP LED Layout



The ACK[2:0] LED's assert when an ACK[2:0]* is asserted on that channels IP bus. Blinking logic is used to allow users to see when a single ACK* assertion occurs.

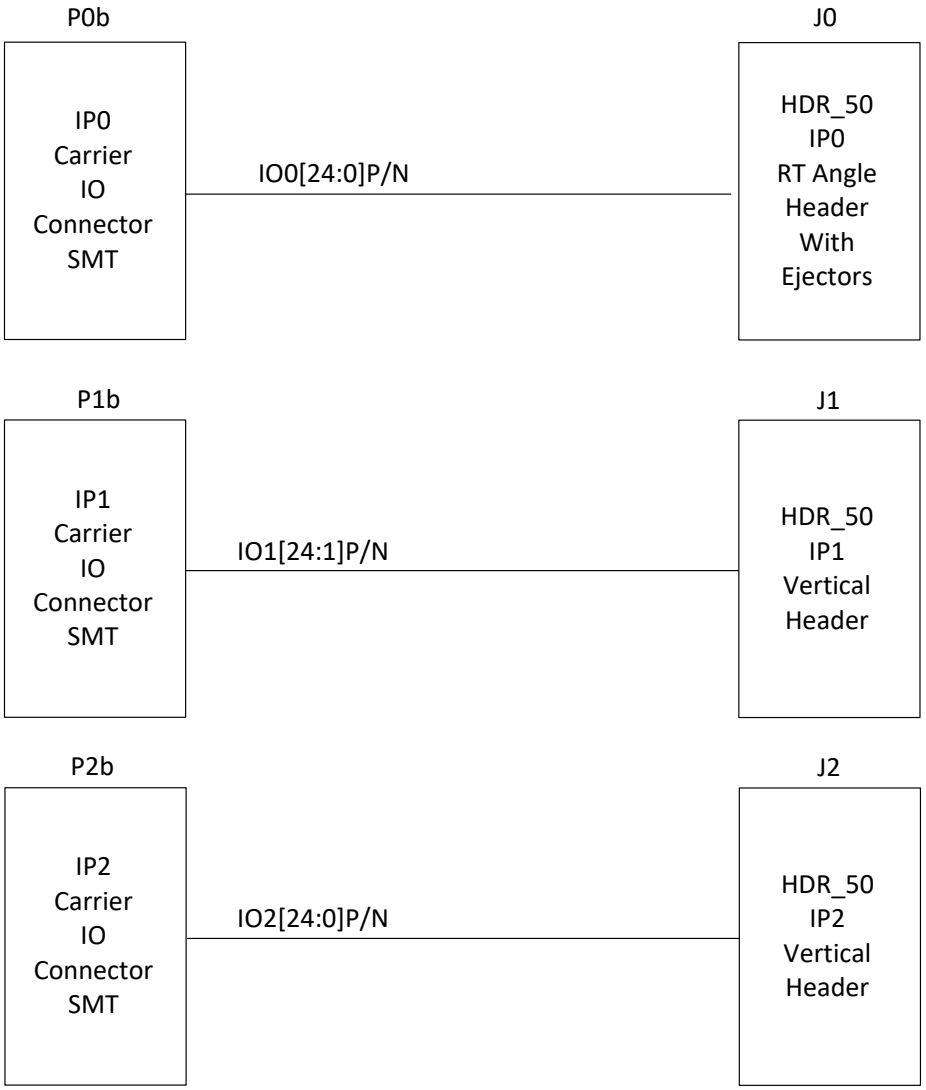
The 1.2V, 3.3V, 5.0V, 12.0V, and -12.0V LED's are associated with their own independent voltage monitoring circuits which accurately detect if any of the board's voltages are out of range. One LED for each power monitoring circuit, if the LED is on the voltage is in range, if the LED is off the voltage is out of range.

The eight (8) user controllable LED's are user programmable with one of sixteen possible sources to provide a variety of status. The default setting provides link status at power up. When link-up occurs properly after power up LED3 through LED0 will light.

PCIe3IP IP1/IP2 Connectivity Options

To provide local control and enable maximum designer flexibility, Dynamic Engineering allows engineers to specify resistor stuffing options to reconfigure the connectivity between IP1 and IP2 carrier I/O and 50 pin headers. See Table 11 for an illustration. IP0's P0b to J0 channel traces have a direct connection. IP1's (P1b to J1) and IP2's (P2b to J2) channel traces are also directly connected by default by stuffing or not stuffing resistor pads. Please contact Dynamic Engineering if you desire a non-default trace/stuffing configuration.

Figure 5: PCIe3IP IP[2:0] I/O to 50-Pin J[2:0] Header



Notes:

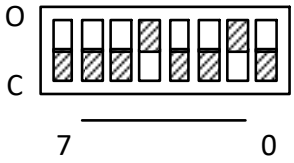
1) By default, R0 & R3 are 0 ohm and R1 & R2 are open.
2) R3, R2, R1, R0 are 0402, 1/16W pads – contact Dynamic Engineering for stuffing options
3) When not stuffed, R1 & R2 leave virtually no-stub.

PCIe5IP Board Features

PCIe5IP DipSwitches

There are two DIP Switches on PCIe5IP, each with 8 switches. They are labeled SW1 and SW2 with bit numbers 7 to 0 and O/C for Open/Closed in the silk screen.

Figure 6: PCIe5IP DipSwitches



SW1 is for user purposes. The settings of SW1 can be seen/used by reading the Switch and LED register [7:0] bits which correspond to SW1 [7:0] bit positions. When a switch is in the C/Closed position the bit value read is 0. when in the O/Open position the value read is 1. Switch values can also be displayed on LED[7:0] by setting the LED Select value (Switch and LED register bits [27:24]) = 0010.

SW2 bits configure PCIe5IP (see Switch and LED register for details) as follows:

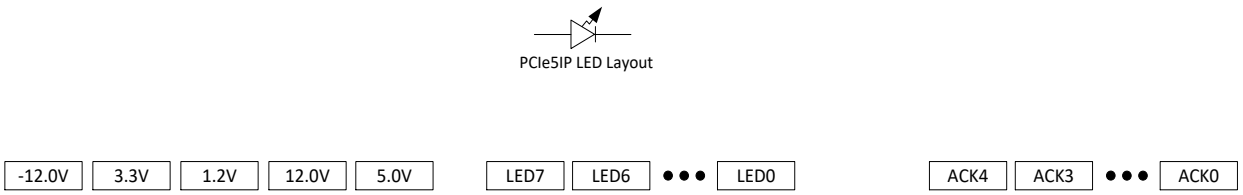
- SW2 [7:6] bit positions: select the Flash image to be used to configure the FPGA at power up and can be read by reading the Switch and LED register [13:12] bits.
- SW2 [5]: is used to configure logic inside the FPGA.
    - This switch setting can be seen by reading the Switch and LED register bit [8].
- SW2 [4:0]: selects either 5.0V or 3.3V to be the Bus Termination voltage for each IP.
    - SW2 [4:0] corresponds one to one with IP [4:0] Bus Termination voltage selection.
    - When the switch is in the closed position (logic 0) a FET is turned on resulting in 5V referenced logic.
    - When the switch is in the open position (logic 1) the FET is disabled and the logic reference is 3.3V.
    - The ES[4:0]V pads on the board can be checked with a multi-meter to make sure you have the expected bus reference/termination voltage.
    - The PCIe5IP ships with SW2[4:0] = 00000, all IP set to operate with 5V signaling/bus termination.

      NOTE: The 3.3V reference is provided through a blocking diode which results in a small voltage drop.

PCIe5IP LEDs

The PCIe5IP has eighteen LEDs. All the LEDs are on the upper/top edge of the PCIe5IP board. The voltage within range monitor LEDs and USER LEDs are located at the top center of the board and the ACK[4:0] LED's are located on the top right corner. Relative to the front top of the board the LED's are oriented as follows:

Figure 7: PCIe5IP LED Layout



The ACK[4:0] LED's assert when an ACK[4:0]* is asserted on that channels IP bus. "Blinking logic" is used to allow users to see when a single ACK* assertion occurs.

The 1.2V, 3.3V, 5.0V, 12.0V, and -12.0V LEDs are associated with their own independent voltage monitoring circuits which accurately detect if any of the board's voltages are out of range. One LED for each power monitoring circuit. If the LED is on the voltage is in range, if the LED is off the voltage is out of range.

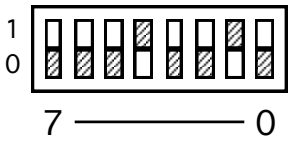
The eight (8) user controllable LEDs are user programmable with one of sixteen possible sources to provide a variety of status. The default setting provides link status at power up. When link-up occurs properly after power up LED3 through LED0 will light.

VPX2IP Board Features

VPX2IP DipSwitches

There are two DIP Switches on VPX2IP, each with 8 switches. They are labeled SW1 and SW2 with bit numbers 7 to 0 and 1/0 for on/off in the silk screen.

Figure 8: VPX2IP DipSwitch



SW1 is for user purposes. The settings of SW1 can be seen/used by reading the Switch and LED register [7:0] bits which correspond to SW1 [7:0] bit positions. When a switch is in the 0 (off) position the bit value read is 0. When in the 1 (on) position the value read is 1. Switch values can also be displayed on LED[7:0] by setting the LED Select value (Switch and LED register bits [27:24]) = 0010.

SW2 bits configure VPX2IP (see Switch and LED register) as follows:

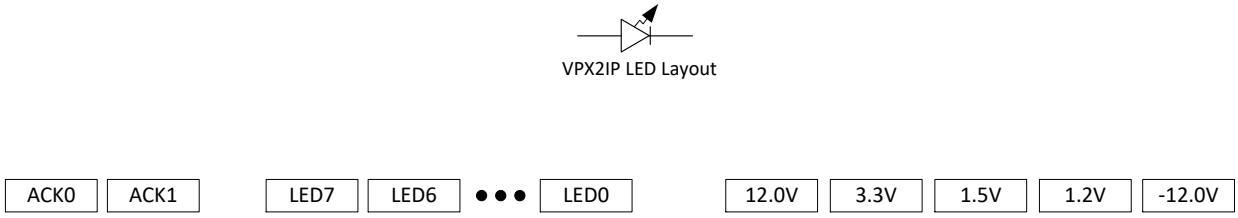
- SW2 [7:6] bit positions: select the Flash image to be used to configure the FPGA at power up and can be read by reading the Switch and LED register [13:12] bits.
- SW2 [7:6] bit positions: select the Flash image to be used to configure the FPGA at power up and can be read by reading the Switch and LED register [13:12] bits.
- SW2 [2]: is not used.
- SW2 [1:0]: selects either 5.0V or 3.3V to be the Bus Termination voltage for each IP.
    - SW2 [1:0] corresponds one to one with the selection for to IP [1:0].
    - When the switch is in the closed position “0” a FET is turned on resulting in 5V referenced logic.
    - When the switch is in the ‘1’ position the FET is disabled and the logic reference is 3.3V.
    - The ES[1:0]V pads on the board can be checked with a multi-meter to make sure you have the expected bus reference/termination voltage.
    - The VPX2IP ships with SW2[1:0] = 00, all IP set to operate with 5V signaling/bus termination.

      NOTE: The 3.3V reference is provided through a blocking diode which results in a small voltage drop.

VPX2IP LEDs

The VPX2IP has twenty LEDs. All the LEDs are located on the edge of the PCB near the bezel of the VPX2IP board. The LEDs are labeled in the silkscreen and on the bezel:

Figure 9: VPX2IP LED Layout



The ACK[1:0] LEDs assert when an ACK[1:0]* is asserted on that channels IP bus. Blinking logic is used to allow users to see when a single ACK* assertion occurs.

The 1.2V, 3.3V, 5.0V, 12.0V, and -12.0V LED's are associated with their own independent voltage monitoring circuits which accurately detect if any of the board's voltages are out of range. One LED for each power monitoring circuit. If the LED is on the voltage is in range, if the LED is off the voltage is out of range.

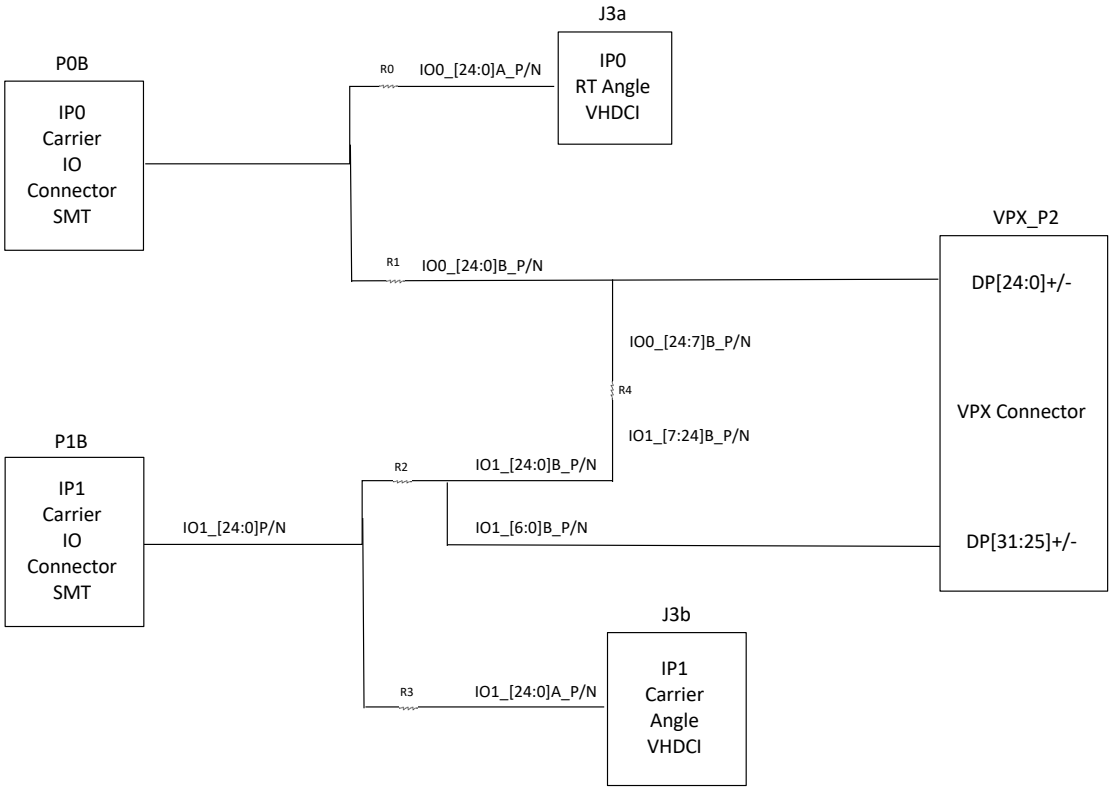
The eight (8) user controllable LEDs are user programmable with one of sixteen possible sources to provide a variety of status. The default setting provides link status at power up. When link-up occurs properly after power up LED3 through LED0 will light. These LEDs are on the rear of the PCB and show near the VHDCI connectors.

In addition, three LEDs are provided for switch status. Port 0 is the upstream port [toward host]. Port 1 is the FPGA side. These LEDs will blink in response to the current PCIe Gen in use. For example, Gen 2 on the upstream port and Gen 1 on the downstream. The rate of blinking indicates the Gen in use. Steady indicates Gen 3. The third LED should not go on – the FATAL LED indicates issues during link up. Check the selection for clock type in this case.

VPX2IP IP0/IP1 Connectivity Options

To provide local control and enable maximum designer flexibility, Dynamic Engineering allows engineers to specify resistor stuffing options to reconfigure the connectivity between IP0 and IP1 carrier I/O and the two VHDCI connectors or VPX connector, see table 11 for an illustration. Please contact Dynamic Engineering if you desire a non-standard trace/stuffing option/configuration.

Figure 10: VPX2IP IP0/IP1 Connectivity Options



Notes:

1) VPX_P0 (Power, JTAG, Reset, Global Address) and VPX_P1 (PCI Express) not shown are always stuffed.
2) Option 1 – VHDCI Header with bezel, no rear IO (VPX_P2) – R0 & R3 are 0 ohm and R1, R2 & R4 are open.
3) Option 2 – Rear IO, no VHDCI Header with blank bezel – R1 & R2 are 0 ohm and R0, R3, & R4 are open.
4) Differential bus traces are length/impedance matched – standard stuffing options leave virtually no stub.
5) R0, R1, R2, R3, and R4 are 0402, 1/16W pads – contact Dynamic Engineering for custom stuffing options.

J3 is a VHDCI connector.

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

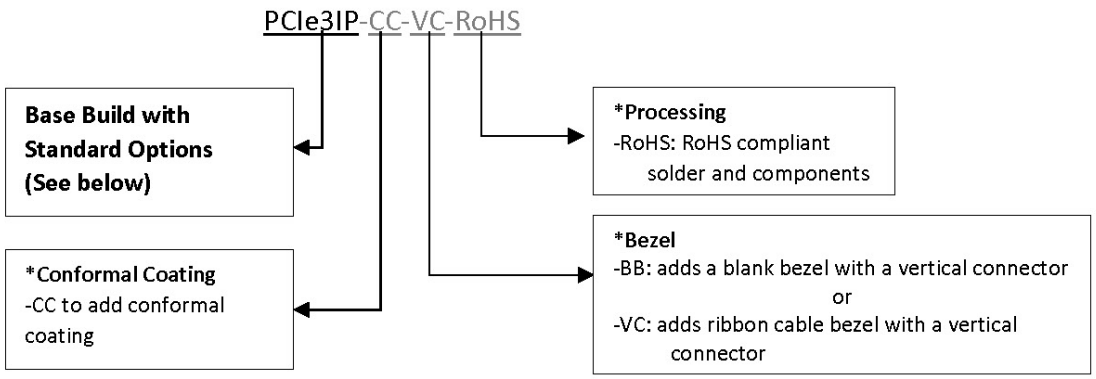
Ordering Information

Standard Temperature Range-Rated Components: -40 - 85°C

Table 44: Ordering Information

Board	Description
PCIe3IP	Half-length PCIe card with three IndustryPack Module positions, independent access, INT and MSI interrupt support, fused filtered power to the IP's, local power supplies, selectable 8 or 32 MHz operation per IP. www.dyneng.com/PCIe3IP
PCIe5IP	Full-length PCIe card with five IndustryPack Module positions, independent access, INT and MSI interrupt support, fused filtered power to the IP's, local power supplies, selectable 8 or 32 MHz operation per IP. www.dyneng.com/PCIe5IP
VPX2IP	3U 4HP with bezel or VPX connector/rear IO with blank bezel with two IndustryPack Module positions, independent access, INT and MSI interrupt support, fused filtered power to the IP's, local power supplies, selectable 8 or 32 MHz operation per IP. www.dyneng.com/VPX2IP
	Options: -RoHS: Use RoHS processing. Standard processing is "leaded." -CC: Option to add conformal coating -BB: Blank bezel with vertical connector **(PCIe3IP and PCIe5IP only)** -VC: Ribbon cable bezel with vertical connector **(PCIe3IP and PCIe5IP only)** -EJ: Add ejector style header connectors for the 2 non-bezel positions **(PCIe3IP and PCIe5IP only)** -RIO: I/O through rear VPX connector with blank bezel installed **(VPX2IP only)**
IP-Debug-Bus	Provides test points on IP control signals, power switching to allow hot swapping of IP's. Recommended for first time users www.dyneng.com/ipdbgbus
IP-Debug-IO	When using IP-Debug-Bus, IP-Debug-IO provides the connection to a 50-pin header for system IO connection or loop-back testing. www.dyneng.com/ipdbgio
Drivers and Reference Software	Clients of PCIeIP have access to existing drivers and reference software packages. The drivers detect the carrier and installed IP's, auto load known drivers for the IP's or IP-Generic [included with driver]. Support for Win7, and Linux with VxWorks in development.

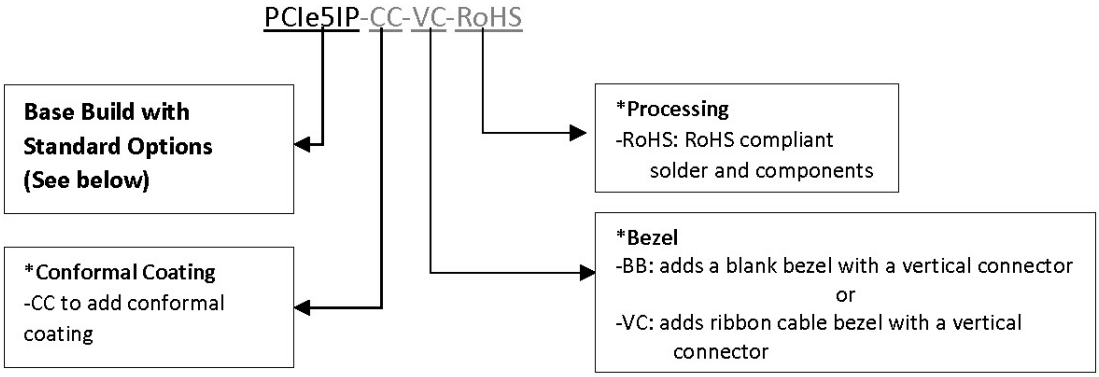
Figure 11: Ordering Options PCIe3IP



PCIe3IP comes standards as: Half-length PCIe card with three IndustryPack Module positions, independent access, INT and MSI interrupt support, fused filtered power to the IP's, local power supplies, selectable 8 or 32 MHz operation per IP. Standard Processing (non-RoHS)
*Only include the build option(s) desired
Ex. PCIe3IP-RoHS

Dynamic Engineering PCIe3IP Ordering Options Revision 01, August 13, 2020

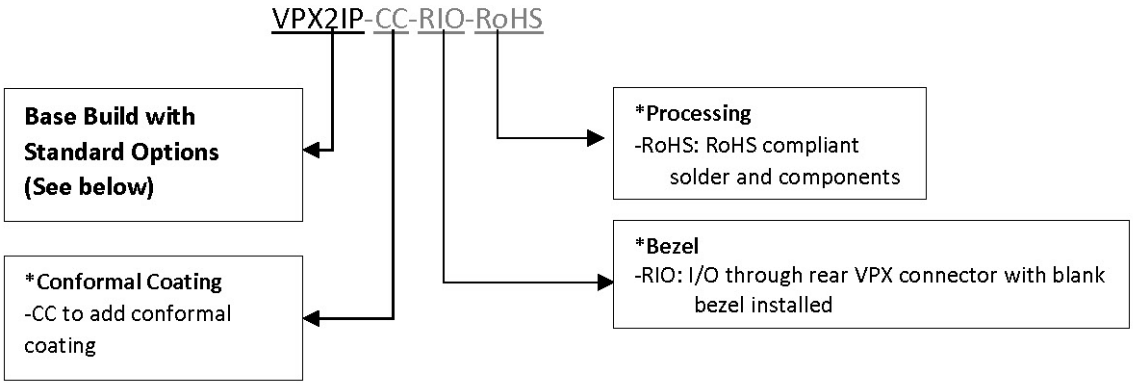
Figure 12: Ordering Options PCIe5IP



PCIe5IP comes standards as: Full-length PCIe card with five IndustryPack Module positions, independent access, INT and MSI interrupt support, fused filtered power to the IP's, local power supplies, selectable 8 or 32 MHz operation per IP.
*Only include the build option(s) desired
Ex. PCIe5IP-BB

Dynamic Engineering PCIe5IP Ordering Options Revision 01, August 13, 2020

Figure 13: Ordering Options VPX2IP



VPX2IP comes standards as: 3U 4HP with bezel or VPX connector/rear IO with blank bezel with two IndustryPack Module positions, independent access, INT and MSI interrupt support, fused filtered power to the IP's, local power supplies, selectable 8 or 32 MHz operation per IP.

*Only include the build option(s) desired
Ex. PCIe5IP-BB

Dynamic Engineering VPX2IP Ordering Options Revision 01, August 13, 2020

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
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