PEX 85XX
EEPROM – PEX 8518/8517/8512/8508
Design Note

July 2007, Version 1.1

Purpose and Scope

Default register values in the PEX 8518/8517/8512/8508 switches may not be appropriate for all designs. Software can program most device registers, however some registers need to be programmed prior to linkup or BIOS/OS enumeration and therefore the optional EEPROM or I²C interface can be used to initialize the PEX 8518/8517/8512/8508 registers. This document provides information for engineers to implement the serial EEPROM hardware and software for PEX 8518, PEX 8517, PEX 8512, and PEX 8508 PCI Express™ switches. This document also lists PEX 8518/8517/8512/8508 RDK EEPROM values that differ from default register values, in Table 26, beginning on page 33.

This document pertains only to the PEX 8518, PEX 8517, PEX 8512, and PEX 8508 devices. These three devices will be referred to as "PEX 85xx" for the remainder of this document.

1. EEPROM Usage

The PEX 85xx serial EEPROM controller downloads EEPROM contents to initialize registers when the EEPROM is present (EE_PR# input strapped low) and one of the following occurs:

- PEX_PERST# (Reset) input is returned high, following a Fundamental Reset (Cold or Warm Reset)
- Hot Reset is received at the Upstream port or the Upstream port exits a DL-Down condition, and if neither of the following Debug Control register bits are set:
  - Hot Reset EEPROM Load Disable (Port 0 offset 1DCh bit 17)
  - Upstream Hot Reset Severity Control (Port 0 offset 1DCh bit 16)

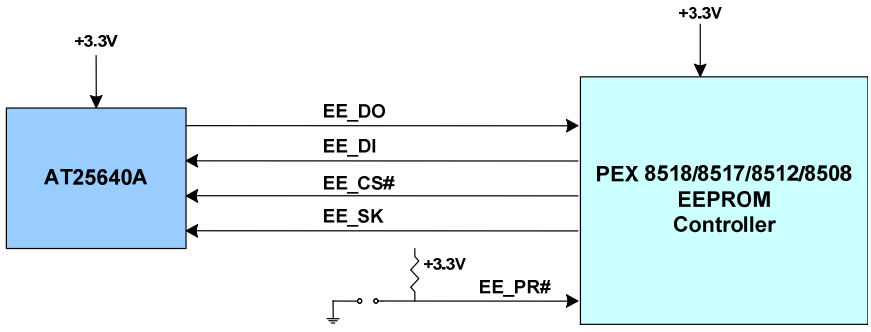

Figure 1. PEX 8518/8517/8512/8508 8KB SPI EEPROM Interface

PLX Technology, Inc, 870 Maude Avenue, Sunnyvale, CA 94085, Phone 408-774-9060, Fax 408-774-2169

2. EEPROM Basics

2.1. Compatible EEPROMs

If the optional EEPROM is implemented, the PEX 85xx switch requires an SPI-compatible, 64-Kbit (8-Kbit x 8-bit) to 512-Kbit EEPROM with 16-bit addressing that can operate at a frequency of 7.8MHz with 3.3V signaling. Please refer to the EEPROM manufacturer datasheets for operational specifications.

When the PEX 85xx is operating in either transparent mode or non-transparent mode without the need for Expansion ROM, 8KB of EEPROM storage is sufficient for the 7744 bytes of configuration data. Direct connection between the PEX 8518 and the Atmel AT25640A (8Kbit x 8-bit), 2.7V-5.5V EEPROM is shown in Figure 1 above.

Most SPI EEPROMs are limited to 5MHz operation, or can support higher frequency only at supply voltages greater than 3.3V. EEPROMs that are limited to 5 MHz operation are not compatible. The maximum voltage that the PEX 85xx switches will tolerate on the I/O pins is 4.6V. If the EEPROM is powered at 5V in order to operate at the required 7.8MHz frequency, external voltage level shifting circuitry is required between the EEPROM and the PEX 85xx EEPROM interface.

2.2. SPI Interface Pin Description

The PEX 85xx integrated EEPROM controller uses the Serial Peripheral Interface (SPI) bus, which is a 4-wire, synchronous, full duplex, master/slave serial communications interface. The four wires are:

1. Chip select EE_CS#, which the PEX 85xx asserts low to enable the EEPROM. PEX 85xx does not assert EE_CS# output if EE_PR# input is high (signaling that no EEPROM is present).
2. Serial clock EE_SK, which provides the interface timing and is driven only by a SPI bus master. PEX 85xx as master drives and toggles the EE_SK clock to shift serial data between the PEX 85xx and the EEPROM in blocks of 8 bits. PEX 85xx does not drive EE_SK clock output if EE_PR# input is high. PEX 85xx drives and toggles the EE_SK clock output only when it is accessing the EEPROM.
3. EE_DI (PEX 85xx output) carries the master-out slave-in serial data, which the PEX 85xx shifts out on the falling edge of the serial clock EE_SK.
4. EE_DO (PEX 85xx input) carries the master-in slave-out serial data, which the PEX 85xx latches on the rising edge of the serial clock EE_SK.

The PEX 85xx switches have weak internal pull-up resistors (~50 kΩ) on the EE_DI, EECS# and EE_DO signals to keep the SPI bus from floating when it is not being actively driven. The system designer may choose to add stronger pull-ups (5 kΩ-10 kΩ) on these lines.

2.3. Sequential Read

When the PEX 85xx reads the serial EEPROM, it asserts the EE_CS# chip select, and serially transmits the SPI Read command (0000 0011), followed by 16 address bits, on the EE_DI output (EEPROM DI input). In the 8KB EEPROM, the A15-A13 address bit values are Don't Care. The PEX 85xx then clocks in the serial data on the EE_DO input (EEPROM DO output). The PEX 85xx reads sequential data from the EEPROM by continuing to drive EE_CS# low, which causes the EEPROM to automatically increment its internal address register, and shift out the 8 data bits at each new address.

2.4. Data Format

All instructions, addresses and input data bytes are shifted with the most significant bit first. Dword data is stored in the serial EEPROM in Little Endian format. For example, the Device/Vendor ID value 851810B5h is stored in the EEPROM with the byte B5h at the lower EEPROM byte address.

The Atmel AT25640A can be programmed directly through the PEX 85xx switch, by using PLX software (PEX GUI and/or APIs) included in the PEX SDK.

3. PEX 85xx EEPROM Presence Detection and Error Detection

3.1. EEPROM Presence Detection

The EE_PR# input has a weak internal pull-up resistor (~50 kΩ), to hold the input in the inactive state when an EEPROM is not present and no trace is connected to this signal.

When the PEX_PERST# input (Fundamental Reset) is de-asserted, the PEX 85xx latches the EE_PR# (EEPROM Present) input to determine whether an EEPROM is present. If EE_PR# input state is detected as low, the PEX 85xx normally downloads data from the serial EEPROM and copies it into the PEX 85xx registers. If the EE_PR# input is high (EEPROM not present), the PEX 85xx EEPROM interface is disabled. If EE_PR# at reset is high and is then toggled low at runtime, the PEX 85xx will not detect the state change, and the EEPROM interface will remain disabled.

A corrupted EEPROM can be re-programmed if EE_PR# input is always low, and the EEPROM is either removed or PEX 85xx is forced to read all 1's from the EEPROM to cause a CRC error. To cause the PEX 85xx to read all 1's, the EE_DO input could be pulled high, or the EEPROM CS# could be switched off. At runtime, the EEPROM can then be carefully hot-plugged into its socket or otherwise re-enabled to allow the EEPROM to be re-programmed by PEX 85xx software.

3.2. CRC Error Detection

The PEX 85xx supports CRC (Cyclic Redundancy Code) error detection of the EEPROM contents. The EEPROM is programmed such that the last Dword of data at EEPROM byte address 1E3Ch is the CRC value, which is calculated as a function of all preceding data in the EEPROM image. When the PEX 85xx downloads initialization data from the EEPROM, it performs the same calculation used to generate the EEPROM CRC value, and compares its result to the CRC value downloaded from the EEPROM. If the calculated CRC value matches the CRC value read from the EEPROM, the probability is high that the EEPROM contents were downloaded correctly.

CRC error detection is enabled as default, by the initial value (0) of the CRC DISABLE bit, at Port 0 register 260h bit 21. If CRC error detection is enabled (default), and the CRC value that the PEX 85xx calculates after downloading the EEPROM contents matches the CRC value downloaded from the serial EEPROM, the PEX 85xx loads its registers with the EEPROM values. If the CRC value that the PEX 85xx calculates after downloading the EEPROM contents does not match the CRC value downloaded from the serial EEPROM, the PEX 85xx discards the downloaded EEPROM values, and instead sets its registers to default values.

CRC error detection can be disabled by setting the CRC DISABLE bit (Port 0 register 260h bit 21) to 1, in the serial EEPROM location corresponding to the CRC DISABLE bit. If the CRC DISABLE bit is set to disable CRC error detection, the PEX 85xx loads its registers with the EEPROM values. Software can also set this bit to cause the PEX 85xx to disable CRC error detection for subsequent runtime EEPROM downloads. However, disabling EEPROM CRC error detection is not recommended.

C/C++ code to compute the CRC value is included in the PLX PEX SDK. The CRC polynomial value is DB710641h. Example code to access and compute the CRC value for the PEX 8518 EEPROM is included in the file (default installation path):

C:\Plx\PlxSdk\Windows\PlxApi\PlxApi.c

3.3. EEPROM Presence and CRC Status Reporting

The PEX85xx reports the status of the EE_PR# (EEPROM Present) input, whether registers were loaded from EEPROM or whether an EEPROM CRC error was detected, in the EEPROM PRESENT bits in the EEPROM Status and Control register, as follows:

Serial EEPROM Present Status (Port 0 register 260h bits [17:16])

00b = Not present

01b = EEPROM present, no CRC error

10b = *Reserved*

11b = EEPROM present with CRC error – default reset values used

4. Expansion ROM

If the application requires Expansion ROM space, a 64KB EEPROM can be used with additional circuitry requirements. The lowest 8KB will be used for PEX 85xx register storage, while the uppermost 32KB can be Expansion ROM storage.

Applications requiring Expansion ROM may use the AT25HP512 (64Kbit x 8-bit) device. This interface requires some external buffering between the PEX 85xx EEPROM controller and the AT25HP512, as shown in Figure 2 below. This EEPROM can operate at 7.8MHz only when it is powered at 5V. Level translation buffers should be used to convert the PEX 85xx 3.3V output levels to 5V input levels compatible with the EEPROM. Similarly, the EEPROM output should be level translated from 5V to 3.3V in order to prevent electrical damage to the PEX 85xx EE_DO input. On Semiconductor's MC74VHC1GT125 tri-state buffer provides the necessary level translation for both directions. Four of these devices are required (one per pin).

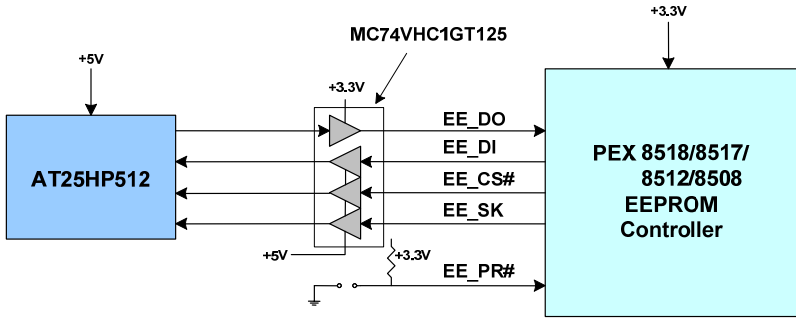

Figure 2. PEX 8518/8517/8512/8508 and 64KB SPI EEPROM Interface

The AT25HP512 EEPROM can be written only by using 128-byte Page Mode writes, which the PEX 85xx cannot perform. Therefore this EEPROM must be programmed by an external EEPROM programmer.

5. PEX 85xx EEPROM Software Interface

5.1. PEX 85xx EEPROM Control, Status and Data Registers

Software can access the EEPROM using two memory-mapped registers in the PEX 85xx upstream port. Address BAR0 + 260h is the EEPROM Status and Control register, in which bits [31:24] are a copy of the Status register within the EEPROM. Offset 264h is the EEPROM Buffer, which contains 4 bytes of data to either write to, or have been read from, the EEPROM. The register descriptions (Register 1 and Register 2) are shown below.

Register 1. 260h EEPROM Status and Control Register (only Port 0)

Bits	Name	Description	Type	EEPROM	Initial
EEPROM Control					
12:0	EEPROM BLOCK ADDRESS	EEPROM Block Address for 32 KB	R/W	Yes	0000h
15:13	EEPROM COMMAND	Commands to serial EEPROM Controller: 001b = Write EEPROM STATUS [31:24] bits to the EEPROM internal status register 010b = Write four bytes of data from the EEPROM BUFFER into memory location pointed to by the EEPROM Block Address 011b = Read four bytes of data from memory location pointed to by the EEPROM Block Address into EEPROM BUFFER 100b = Reset Write Enable latch 101b = Read EEPROM internal status register and write data to EEPROM STATUS [31:24] bits 110b = Set Write Enable latch All other encodings are *Reserved*	R/W	Yes	000b
EEPROM Status					
17:16	EEPROM PRESENT	Serial EEPROM Present status 00b = Not present 01b = EEPROM Present – no CRC error 10b = *Reserved* 11b = EEPROM Present, but with CRC error – default reset value used	RO	Yes	00b
19:18	EEPROM COMMAND STATUS	Serial EEPROM Command status 00b = EEPROM Command complete 01b = EEPROM Command **not** complete 10b = EEPROM Command complete with CRC error 11b = *Reserved*	RO	Yes	00b
20	EEPROM BLOCK ADDRESS UPPER BIT	EEPROM BLOCK ADDRESS upper bit [13] Extends EEPROM to 64K Bytes	R/W	Yes	0
21	CRC DISABLE	0b = EEPROM input data uses CRC 1b = EEPROM input data CRC disabled	R/W	Yes	0
23:22	-	***Reserved***			00b
Status Data from EEPROM					
24	EEPROM_RDY#	0b = EEPROM is ready to send data 1b = a write cycle is in progress	R/W	Yes	0
25	EEPROM_WEN	0b = EEPROM Write is disabled 1b = EEPROM Write is enabled	R/W	Yes	0
27:26	EEPROM_BP[1:0]	EEPROM Block-Write Protect bits (see Table 1 below)	R/W	Yes	00b
30:28	EEPROM WRITE STATUS	These bits = 000b when EEPROM is not in an internal Write cycle	R/W	Yes	000b
31	EEPROM_WPEN	EEPROM Write Protect Enable When this bit: = 0b and EEPROM_WEN = 1b, the EEPROM Status register is writable = 1b, the EEPROM Status register is protected	R/W	Yes	0

Table 1. Serial EEPROM Block-Write Protect Bits

BP[1:0]	Level	Array Addresses Protected			
		8 KB Device	16 KB Device	32 KB Device	64 KB Device
00b	0	None	None	None	None
01b	1 (top 1/4)	1800 – 1FFF	3000 – 3FFF	6000 – 7FFF	
10b	2 (top 1/2)	1000 – 1FFF	2000 – 3FFF	4000 – 7FFF	
11b	3 (All)	0000 – 1FFF	0000 – 3FFF	0000 – 7FFF	

Register 2. 264h EEPROM Buffer (only Port 0)

Bits	Name	Description	Type	EEPROM	Initial
31:0	EEPROM BUFFER	EEPROM Buffer	R/W	Yes	0h

In the EEPROM Status and Control register:

- The lower 13 bits contain the Dword EEPROM address (Byte address divided by 4) to be accessed
- The next 3 bits contain the EEPROM command code (operation to be performed)
- Bits [19:16] are Read-Only, so when writing this register the value is Don't Care
- Bits [23:20] must be 0 to read or write PEX 85xx register data to the EEPROM
- Bits [31:24] are a copy of the Status register within the EEPROM, so when writing or reading PEX 85xx register data to/from the EEPROM (using the EEPROM Write Data (Command 010b) or Read Data (Command 011b) instructions in bits [15:13]), bits [31:24] are not written to the Status register within the EEPROM.

The EEPROM will power up in the Write Disable state. Therefore, all programming instructions must be preceded by a Write Enable instruction (Command 110b).

To protect the device against inadvertent writes, the Write Disable instruction (Command 100b) disables all programming modes.

5.2. Software Read/Write Software Access to the EEPROM through the PEX 85xx

5.2.1. Reading and Writing Data

Using the PEX 85xx EEPROM Status and Control and EEPROM Buffer registers, software reads or writes 4 bytes (Dword-aligned) in every access to the EEPROM. Dword addresses for each PEX 85xx register are listed in the *PEX 85xx Data Book*, Appendix A, Tables A-1 and A-2. To determine the EEPROM Dword address for a particular PEX 85xx register, select the PEX 85xx register's byte address in the leftmost column, and then in that row select the column for the particular PEX 85xx port. The 13-bit Dword EEPROM address can then be entered directly into EEPROM Block Address field in the EEPROM Control register (Port 0 register 260h). The EEPROM command code, which designates the operation to be performed, is entered as the upper 3 bits [15:13] of the 16-bit EEPROM Control register.

5.2.2. To Write 4 Bytes to the Serial EEPROM

a. Write the 32-bit data into the EEPROM Buffer register at address Port 0 BAR0 + 264h.

b. Issue a Write Enable instruction (Command = 110b, Set Write Enable Latch), by writing the value 0000C000h into the EEPROM Status and Control register at address Port 0 BAR0 + 260h.

c. Calculate the combined Address and Command value to write into the EEPROM Control register (lower 16 bits of Port 0 register 260h), by combining the EEPROM 3-bit Write Data instruction, value 010b, as the upper 3 bits of the Word, with the 13-bit EEPROM Dword address obtained from Table A-1 or A-2. In other words, OR the value 4000h with the EEPROM Dword address value obtained from Table A-1 or A-2. The upper Word of the result should be clear. Then write the 32-bit value to register 260h to write register 264h data to the EEPROM.

d. Issue a Write Disable instruction (Command = 100b, Reset Write Enable Latch), by writing the value 00008000h to register 260h.

e. **WARNING!** Changing a value in the EEPROM will cause the CRC to be incorrect. Unless the CRC is recalculated and programmed into EEPROM byte address 1E3Ch (Dword address 78Fh), when the PEX 85xx reads the EEPROM to initialize its registers, it will detect the bad CRC and discard the EEPROM values, and instead load the registers with default values (unless Port 0 register 260h bit 21 is set in the EEPROM, which is strongly discouraged).

5.2.3. To Read 4 Bytes from the Serial EEPROM

a. Calculate the value to write into the EEPROM Control register (lower 16 bits of Port 0 register 260h), by combining the EEPROM 3-bit Read Data instruction, value 011b, as the upper 3 bits of the Word, with the 13-bit EEPROM Dword address obtained from Table A-1 or A-2. In other words, OR the value 6000h with the EEPROM Dword address value obtained from Table A-1 or A-2. The upper Word of the result should be clear. Then, write the 32-bit value to register 260h at Port 0 BAR0 + 260h.

b. Read the four bytes of EEPROM data from register 264h, at Port 0 BAR0 + 264h.

5.2.4. Reading and Writing EEPROM Status

Bits within the EEPROM Status register can enable write protection for data and/or the EEPROM Status register itself.

For hardware write protection, the serial EEPROM pinout includes the **WP#** (Write Protect) input; when WP# is held low, write operations to the EEPROM Status register are inhibited.

- **Bit 31, WPEN**, overrides WP# and enables/disables writes to the Status register.
  When WP# is high or WPEN = 0, and WEN = 1, the Status register is writable.
  When WP# is low and WPEN = 1, or WEN = 0, the Status register is protected.
- **Bit 25, WEN**, enables/disables writes to Unprotected Blocks of the memory array.
  When WEN = 0, writes to Unprotected Blocks in memory are disabled.
  When WEN = 1, writes to Unprotected Blocks in memory are enabled.
- **Bits 27:26, BP[1:0],** provide write protection for blocks of the memory array.
  Block Protection options protect the upper ¼, upper ½, or all of the EEPROM. PEX 85xx configuration data is stored in the lower addresses, so when using Block Protection, the entire EEPROM should be protected with BP[1:0] = 11b.

For board manufacturing with a blank EEPROM, the WP# input signal can be always strapped low and EEPROM can still be later initially programmed. Hardware write protection of the data is not enabled until software sets the Block Protection bits BP[1:0], and write protection of the Status register (protection of the BP[1:0] bits) is not enabled until software sets the WPEN bit. If WP# is low, once software sets the WPEN bit to write-protect the EEPROM status register, the WPEN value cannot be changed back to 0, nor can the BP[1:0] bits can be cleared to disable Block Protection. Write protection can be subsequently disabled after WP# is brought high.

Table 2 below summarizes write protection in the serial EEPROM.

Table 2. WPEN, WP# and WEN Operation

WPEN	WP#	WEN	Protected Blocks	Unprotected Blocks	Status Register
0	x	0	Protected	Protected	Protected
0	x	1	Protected	Writable	Writable
1	low	0	Protected	Protected	Protected
1	low	1	Protected	Writable	Protected
x	high	0	Protected	Protected	Protected
x	high	1	Protected	Writable	Writable

WPEN, WEN, and BP[1:0] status can be determined by reading the EEPROM Status register in PEX 85xx Port 0 register 260h bits [31:24]. To copy the EEPROM Status register contents into register 260h bits [31:24], issue a Read Status Register instruction (RDSR, Command = 101b), by writing the value 0000A000h to Port 0 register 260h.

The only writable bits in the EEPROM Status register are WPEN (bit 31) and BP[1:0] (bits [27:26]).

To enable Block Protection, write the following values to Port 0 register 260h:

00006000h (Write Enable)
0C002000h (Write BP[1:0] = 11b)
00008000h (Write Disable)

To disable Block Protection, write the following values to Port 0 register 260h:

00006000h (Write Enable)
00002000h (Write BP[1:0] = 00b)
00008000h (Write Disable)

To enable Block Protection and write protection of the EEPROM status register, write the following values to Port 0 register 260h:

00006000h (Write Enable)
8C002000h (Write BP[1:0] = 11b, WPEN = 1)
00008000h (Write Disable)

The EEPROM does not respond to Command Codes 000b or 111b.

5.3. EEPROM Programming Software

The PEX SDK includes software that can read and program the serial EEPROM that is connected to the switch. For information and instructions on using this application, please refer to the PEX SDK User's Manual.

6. EEPROM Content

It is convenient to use the EEPROM image for the PEX 8518RDK, PEX 8517RDK, PEX 8512RDK or PEX 8508RDK, included in the PEX SDK, as the basis for EEPROM content. Then use the PEX SDK GUI included in the PEX SDK to modify the image for the specific application. The following subsections list EEPROM settings that Differ from default registers, as well as other registers and bits that generally need to be evaluated for each specific application.

6.1. Silicon Revision

In Revision AB silicon, Revision ID (offset 08h) bit 0 is hardwired to 1.

- If the EEPROM is programmed with the value AAh, reading the register will return the value ABh.

In Revision AC silicon, Revision ID (offset 08h) bits [2:1] are reversed.

- If the EEPROM is programmed with the value AAh, reading the register will return the value ACh.
- If the EEPROM is programmed with the value ABh, reading the register will return the value ADh.
- If the EEPROM is programmed with the value ACh, reading the register will return the value AAh.

6.2. Switch Configuration

Modification of the RDK EEPROM image is most likely required in order to set the port configuration to match that defined by the PEX 85xx strapping pins, since these pins may be configured differently from that used in the PEX 85xx RDK. Registers that must be modified are:

1. Port 0 Port Configuration, offset 224h, to set number and width of ports as defined by strapping signals listed in Table 3 below. The EEPROM values overwrite the strapping signal values.

Table 3. Port Configuration Register and Corresponding Strapping Signals

Port Configuration Register (224h)	Default Value	Strapping Signals
Port 0	0h	STRAP_STN0_PORTCFG[4:0]

Valid Port Configuration Codes are shown in Table 4, Table 5and Table 6 below.

Table 4. PEX 8508 Configuration Codes

Code	Port 0	Port 1	Port 2	Port 3	Port 4
0,1,7,10-31	x2	x2	x2	x2	
2	x4	x4			
3	x4	x2	x2		
4	x4	x2	x1	x1	
5	x4	x1	x1	x2	
6	x4	x1	x2	x1	
8	x4	x1	x1	x1	x1
9	x2	x2	x2	x1	x1

Table 5. PEX 8512 Configuration Codes

Code	Port 0	Port 1	Port 2	Port 3	Port 4
3	x4	x4	x4		
8	x4	x2	x2	x2	x2

Table 6. PEX 8517 and PEX 8518 Configuration Codes

Code	Port 0	Port 1	Port 2	Port 3	Port 4
0,1,7,10-31	x4	x4	x4	x4	
2	x8	x8			
3	x8	x4	x4		
4	x8	x4	x2	x2	
5	x8	x2	x2	x4	
6	x8	x2	x4	x2	
8	x8	x2	x2	x2	x2
9	x4	x4	x4	x2	x2

2. Port 0 can be forced to train to x1 (rather than the width configured by register 224h), by setting Port 0 offset 228h bit 2, as shown in Table 7 below.

Table 7. Physical Layer Test Register (Offset 228h in Port 0)

Bit	Bit Field Name	Default Value	Value
2	Port_0_X1	0	1 = Port 0 is configured as x1 only 0 = Port 0 is not forced to train as x1

Setting this bit does not result in any additional power savings on unused SerDes.

3. Port 0 Debug Control, offset 1DCh, to select Transparent or Non-Transparent mode, one upstream port, and a single NT port (if Intelligent Adapter or Dual Host NT modes are enabled). The EEPROM values for the bit fields in register 1DCh overwrite the corresponding strapping signals, as listed in Table 8 below.

Table 8. Debug Control Register Bits and Corresponding Strapping Signals

Debug Control (1DCh) bits	Strapping Signals
Upstream Port Number [10:8]	STRAP_UPSTRM_PORT_SEL[2:0]
Mode Select [19:18]	STRAP_MODE_SEL[1:0]
NT-Port Number [27:24]	STRAP_NT_UPSTRM_PORT_SEL[3:0]

6.2.1. Changing the Switch Configuration

Changing the switch configuration (Ports 0 register 224h, number and/or width of ports) generally requires changes to additional register values in the EEPROM:

1. Port offset 160h, VC0 Resource Status register bits [17, 1]. Refer to Section 6.6.
2. Port offset 1F8h, ACK Transmission Latency Limit register bits [7:0]. Refer to Section 6.7.
3. Port offset 1F8h, Replay Timer Limit register bits [23:16]. Refer to Section 6.8.
4. Port 0 offset 668h, TIC Port Enable register bits [12:8, 3:0]. Refer to Section 6.9.
5. Port 0 offsets C00h[15:0] and C04h[31:16], ITCH VCnT Threshold registers. Refer to Section 6.10.

6.2.2. Changing the Upstream Port

Changing the designated upstream port (Port 0 register 1DCh bits [10:8]) generally requires changes to additional registers in the EEPROM:

1. Port offset 68h, PCI Express Capabilities register. Refer to Section 6.3 below.

a. Device / Port Type, bits [23:20].

b. Slot Implemented, bit 24.

2. Port offset 7Ch, Slot Capabilities register. Refer to Section 6.5.2 below.

6.3. Port Configuration

Each configured port's PCI Express Capabilities register, at offset 68h, may need to be modified depending upon whether a port other than 0 is defined as the upstream port, and whether downstream ports are connected to a device rather than a slot. Refer to Table 9 below.

Table 9. PCI Express Capabilities Register (Offset 68h in All Ports)

Bits	Bit Field Name	Default Value	Value
23:20	Device / Port Type	Upstream 5	Upstream port = 5h Downstream transparent port = 6h Non-Transparent port (NT-Virtual & NT-Link): Legacy PCI Express Endpoint (I/O BAR1 required) = 1h PCI Express Endpoint (I/O BAR1 not required, OS can close I/O space) = 0h ***Note:*** *BAR1 is enabled as default, in registers:* *NT-Virtual: BAR1 Setup D0h* *BAR1 Setup Shadow D80h* *NT-Link: BAR Setup E4h*
		Downstream 6	
		NT 0	
24	Slot Implemented	Upstream 1	Upstream and NT ports (not programmable) = 0 Downstream transparent port: 1 if Slot 0 if Device, or configured port is disabled
		Downstream 0	

6.4. RefClk Source

Link Status and Control register (78h) includes the Slot Clock Configuration bit that defines whether the upstream port, each downstream port, and the NT-Link port share a common clock source with the device at the other end of each link. Refer to Table 10 below.

Table 10. Link Status and Control Register (Offset 78h in All Ports)

Bit	Bit Field Name	Default Value	Value
28	Slot Clock Configuration	Upstream & NT 0 Downstream 1	1 = RefClk to PEX 85xx and RefClk at connector are from a common source. 0 = RefClk to PEX 85xx and RefClk at connector are asynchronous.

Since PEX 85xx Active State Power Management (ASPM) should remain disabled (78h[1:0] = 00b) due to errata, the L0s Exit Latency and N_FTS values, which can normally be reduced if the RefClk to all devices is synchronous, should remain the default values.

6.5. Hot Plug Signals

6.5.1. Upstream and NT-Link Ports

The PEX 85xx Upstream and NT ports are Hot Plug clients; Hot Plug signals associated with these ports are not used.

6.5.2. Downstream Transparent Ports

The PEX 85xx provides 9 Hot Plug signals for each downstream transparent port. The Hot Plug Controller in a downstream transparent port is enabled if the Power Controller Present bit in the Slot Capabilities register for that port is set (7Ch bit 1 = 1). This register also defines whether Hot Plug is supported by a PCIe slot, which Hot Plug signals are used by that port, Power Limits for slots connected to downstream transparent ports, and the Physical Slot Number assigned for physical slots that connect downstream transparent ports. Refer to Table 11 and Table 12 below.

Table 11. Downstream Transparent Port Hot Plug Signals

Inputs	Slot Capabilities (7Ch) Register Enable Bit	Functionality
HP_BUTTON#	0	Attention Button
HP_MRL#	2	Manually operated Retention Latch Sensor
HP_PRSNT#		Presence Detect Register 1E0h bit 5 controls whether Presence is detected by HP_PRSNT# or SerDes Receiver Detect
HP_PWRFLT#		Power Fault
Outputs		
HP_ATNLED#	3	Attention Indicator
HP_CLKEN#	1	Power Controller Present
HP_PERST#		Reset output
HP_PWREN#	1	Power Controller Present
HP_PWRLED#	4	Power Indicator Present

Table 12. Slot Capabilities Register (7Ch, Exists Only in Transparent Ports), Valid Only for Downstream Transparent Ports)

Bits	Bit Field Name	Default Value	Value
5	Hot Plug Surprise	Upstream 0 Downstream 1	Downstream Transparent port value: 0 if Slot does not support removal without prior notification 1 if Slot supports removal without prior notification
6	Hot Plug Capable	Upstream 0 Downstream 1	Downstream Transparent port value: 0 if Device, or Slot does not support Hot Plug 1 if Slot supports Hot Plug
14:7	Slot Power Limit Value	Upstream 0h Downstream 19h	If Slot Implemented bit is set, Downstream transparent port value is Max power (25W = 19h)
16:15	Slot Power Limit Scale	00b	If Slot Implemented bit is set, Downstream transparent port value is 00=1.0, 01=0.1, 10=0.01, 11=.001
31:19	Physical Slot Number	0h	Downstream Transparent port value: System unique non-zero value

If the Hot Plug Controller is enabled in a downstream transparent port, output signal timing must also be defined in that port's Power Management Hot Plug User Configuration register (1E0h). Refer to Table 13 below.

Table 13. Hot Plug Controller

Slot Capabilities Register (7Ch) Power Controller Present Bit (1)	Default Value	Hot Plug Controller
0	1	Disabled
1		Enabled 1. Set delay from power-applied to power-valid (TPEPV, 1E0h bits [4:3]) - delay from when HP_PWREN# and HP_PWRLED# outputs assert after PERST# input assertion, until HP_CLKEN# output is asserted. 2. Set delay from power-valid (HP_CLKEN# asserted) to HP_PERST# output de-assertion (TPVPERL), 1E0h bit 6)

6.5.3. Downstream Transparent Port HP_PERST# (Reset) Output

Whether HP_PERST# output is de-asserted is controlled by the Power Controller Control bit (Slot Control Register (80h) bit 10).

1. If the Hot Plug Controller is enabled (Power Controller Present, 7Ch bit 1 = 1 (default without EEPROM, or programmable only by EEPROM)), and if the Power Controller Control bit is clear (80h bit 10 = 0):

   HP_PERST# output de-asserts after TPVPERL delay (1E0h bit 6, 20 ms or 100 ms default) after power is valid at the slot (HP_CLKEN# asserted). Power is valid at the slot (HP_PWREN# and HP_PWRLED# (if enabled) outputs are asserted), after TPEPV delay (1E0h bits [4:3]) following completion of the EEPROM load. If HP_MRL# input is enabled (7Ch bit 2 = 1 (default without EEPROM, or programmable only by EEPROM)), the Hot Plug power up sequence ending with HP_PERST# de-assertion does not start until HP_MRL# is low (following 10 ms de-bounce).

2. If the Hot Plug Controller is enabled (Power Controller Present, 7Ch bit 1 = 1), and the Power Controller Control bit is set (80h bit 10 = 1), HP_PERST# output is not de-asserted.

3. If the Hot Plug Controller is disabled (Power Controller Present, 7Ch bit 1 = 0 (programmable only by EEPROM)), and the following conditions are true:

   - The Power Controller Control bit is clear (80h bit 10 = 0)
   - HP_MRL# is disabled (7Ch bit 2 = 0 (programmable only by EEPROM)) or if HP_MRL# is enabled (7Ch bit 2 = 1 (default without EEPROM, or programmable only by EEPROM)) with HP_MRL# input asserted, then HP_PERST# output de-asserts immediately after the EEPROM load.

     Otherwise, HP_PERST# output remains de-asserted.

HP_PERST# can also be toggled at runtime by toggling the Power Controller Control bit (80h bit 10), provided HP_PERST# initially de-asserts under one of the conditions listed above. Value of 1 asserts HP_PERST# (low) and value of 0 de-asserts HP_PERST# (high).

6.6. VC0 Negotiation Pending bit (All Ports, Offset 160h Bit [17])

The VC0 Negotiation Pending bit in each enabled port (as defined by Port 0 register 224h) must be set in the programmed EEPROM.

Software must first check that the VC0 Negotiation Pending bit in the Virtual Channel Extended Capability registers of each successive device in the link path is cleared (indicating link flow control initialization has completed), prior to attempting to access the device at the other end of the link.

6.7. ACK Transmission Latency Limit (All Ports, Offset 1F8h Bits [7:0])

The ACK Transmission Latency Limit register places a minimum amount of time (in clocks) that the switch port waits before prioritizing an ACK transmission.

Set the ACK Transmission Latency Limit in each transparent port, and in the NT-Virtual port. Recommended values for the ACK Transmission Latency Limit are based upon port width and Maximum Payload Size (128 or 256 bytes, as determined offset 70h bits [7:5]), as shown in Table 14 below.

Table 14. ACK Transmission Latency Limit, 1F8h [7:0]

Maximum Payload Size	Port Width			
	x1	x2	x4	x8
128B	FAh	80h	49h	43h
256B	FFh	D9h	76h	6Bh

If the recommended value is increased, there is greater chance of ACK collapsing, which can increase egress TLP throughput since ACK DLLP traffic is reduced. However, increasing the register value could slow ingress TLP throughput if the Replay buffer in the external device is not sufficiently deep to prevent it from throttling TLP transmission while waiting for an ACK.

For the NT-Virtual port width, use the port width of the NT-Link port. For example, if the NT-Link port width is x4 and Maximum Payload Size (70h[7:5]) is 128 bytes, set the NT-Virtual value to the x4 value 49h. The ACK Latency Timer value in the NT-Link port can remain at its default value 0h.

6.8. Replay Timer Limit, Upper 8 Bits (All Ports, Offset 1F8h Bits [23:16])

If a port's Transmit Retry Buffer contains TLPs for which no ACK or NAK has been received for a time period exceeding the programmed value of the Replay timer, the switch port's transmitter initiates a replay.

Set the Replay Timer Limit in each transparent port, and in the NT-Virtual port if Non-Transparent mode is enabled (in Port 0 register 1DCh bits [19:18]). Recommended values for the Replay Timer Limit are based upon link width and are calculated for Maximum Payload Size of 256 bytes; the same value should be used for Maximum Payload Size of 128 bytes (due to timer tolerance). Refer to Table 15 below.

Table 15. Replay Timer Limit, 1F8h [23:16]

Link Width			
x1	x2	x4	x8
05h	03h	02h	02h

For the NT port width, use the link width of the NT-Link port. For example, if the NT-Link port width is x4, set the NT-Virtual value to the x4 value, 02h. The Retry Timer Limit value in the NT-Link port can remain at its default value FFh.

6.9. TIC Control Register (Port 0 Offset 668h)

When a programmed EEPROM is present, the TIC Control register is loaded from EEPROM. The value indicates which ports are enabled (even if unused), as defined by Port 0 register 224h. To determine the value, start with FFFF_0000h, and then set the bit positions that correspond to enabled ports. For example, if Ports 0, 1 and 2 are enabled, the correct value is FFFF_0007h. Refer to Table 16 below.

Table 16. TIC Control Register (Offset 668h in Port 0)

Port Configuration	Number of Enabled Ports	TIC Control
Port 0 Register 224h		Port 0 Register 668h
0,1,4-7,10-31	4	FFFF000Fh
2	2	FFFF0003h
3	3	FFFF0007h
8,9	5	FFFF001Fh

The number of enabled ports is normally the same as the value of the Number of Ports Enumerated register value (Port 0 register 220h bits [22:20]), which is automatically updated by the switch after reset.

6.10. INCH Threshold Port Virtual Channel Registers (Port 0 Offsets A00h–A68h)

6.10.1. Flow Control Initialization Credits

With respect to system performance, of all register values, the Ingress Credits that are advertised to other devices can have the greatest impact.

The Station contains an Ingress Packet RAM of 1024 rows x 5 DW columns, which can be allocated among up to 5 enabled ports in the Station. Each packet requires 1 Header Credit, which allocates up to 5 DW of which 3 or 4 DW are for the packet header, and 1 DW is for the ECRC (if enabled).

Port 0 can be allocated a maximum of 32 Header Credits, for Posted, Non-Posted, and Completion transactions, for both Virtual Channels (VC0 and VC1). Credits for Posted, Non-Posted, and Completion transactions can be assigned different values for Virtual Channel 0 (VC0) and Virtual Channel 1 (VC1); however, the sum of all Header Credits for VC0 and VC1 together must not exceed the maximum value of 32 Header Credits.

Ports 1 and 2 can each be allocated a maximum of 24 Header Credits, for Posted, Non-Posted, and Completion transactions, for both Virtual Channels (VC0 and VC1). Credits for Posted, Non-Posted, and Completion transactions can be assigned different values for Virtual Channel 0 (VC0) and Virtual Channel 1 (VC1); however, the sum of all Header Credits for VC0 and VC1 together must not exceed the maximum value of 24 Header Credits per port.

Ports 3 and 4 can each be allocated a maximum of 16 Header Credits, for Posted, Non-Posted, and Completion transactions, for VC0 only (VC1 is not supported on Ports 3 and 4). The sum of all VC0 Header Credits for Ports 3 and 4 must not exceed the maximum value of 16 Header Credits per port.

Each Payload Credit corresponds to 16 bytes of data. Enough Payload Credits should be assigned for each header to accommodate the maximum packet size for each transaction type. Each Non-Posted transaction carries a maximum payload of 4 bytes. Posted and Completion Header Credit should be assigned at least the number of Payload Credits required to accommodate the Maximum Payload Size used in the system.

> [The Maximum Payload Size (Device Control register, offset 70h bits [7:5],) is programmed by software to the lowest value of Maximum Payload Size Supported (Device Capability register, offset 6Ch bits [2:0],) among all devices in the system. The value 000b indicates 128 bytes, and the value 001b indicates 256 bytes.]

Headers can only occupy the first column of the Packet RAM. The Payload data immediately follows the 3- or 4-Dword header, on the same row of RAM and on successive row(s) if the row containing the header fills up. If ECRC is enabled, the 1-Dword ECRC immediately follows the Payload data. Any remaining empty space in a row following the Payload and/or ECRC is unused. The number of rows of RAM that a packet will occupy is calculated by dividing the sum of the number of bytes in the header, payload, and ECRC (if enabled), by 20; if the result includes any non-integral remainder, increment the quotient.

For example, each Memory Read (Non-Posted, without data) packet occupies one row of the Packet RAM. A Posted or Completion packet having a 64-byte payload and 3 DW header (or a 4 DW header and no ECRC) occupies four rows of the Packet RAM. A Posted or Completion packet having a 3 DW header, 128-byte payload and no ECRC will occupy seven rows of RAM (8 rows if the header is 4 DW and/or if ECRC is enabled). A packet with a 256-byte payload will occupy fourteen rows of RAM.

Payload Credits can be calculated to include only the Payload Data, or alternatively to include any unused RAM in the remainder of a row. The number of Payload Credits is programmed into the lowest 9 bits ([8:0]) of each Ingress Credit (INCH) register; the number of Payload Credits for Posted and Completion packets is always a multiple of 8, since the lowest three bits ([2:0]) of the Posted and Completion registers are *Reserved*.

Table 17 below lists the maximum effective value for Payload Credits, for combinations of selected Header Credits and Maximum Payload Size.

Table 17. Effective Maximum Payload Credits for Header Credits and Payload Sizes

Headers		Maximum Payload Size					
		4		128		256	
dec	hex << 9	dec	hex	dec	hex	dec	hex
1	0200	1	01	8	08	16	10
2	0400	2	02	16	10	32	20
3	0600	3	03	24	18	48	30
4	0800	4	04	32	20	64	40
5	0A00	5	05	40	28	80	50
6	0C00	6	06	48	30	96	60
7	0E00	7	07	56	38	112	70
8	1000	8	08	64	40	128	80
9	1200	9	09	72	48	144	90
10	1400	10	0A	80	50	160	A0
11	1600	11	0B	88	58	176	B0
12	1800	12	0C	96	60	192	C0
13	1A00	13	0D	104	68	208	D0
14	1C00	14	0E	112	70	224	E0
15	1E00	15	0F	120	78	240	F0
16	2000	16	10	128	80	256	100
17	2200	17	11	136	88	272	110
18	2400	18	12	144	90	288	120
19	2600	19	13	152	98	304	130
20	2800	20	14	160	A0	320	140
21	2A00	21	15	168	A8	336	150
22	2C00	22	16	176	B0	352	160
23	2E00	23	17	184	B8	368	170
24	3000	24	18	192	C0	384	180
25	3200	25	19	200	C8	400	190
26	3400	26	1A	208	D0	416	1A0
27	3600	27	1B	216	D8	432	1B0
28	3800	28	1C	224	E0	448	1C0
29	3A00	29	1D	232	E8	464	1D0
30	3C00	30	1E	240	F0	480	1E0

INCH Threshold register values can be calculated by ORing the “hex << 9” column with the value in the appropriate “hex” column (the “4” column for Non-Posted, and the “128” or “256” hex column (as appropriate) for Posted and Completion transaction types.

Ingress Credits for the PEX 85xx RDK EEPROMs are programmed to the default values listed in the PEX 85xx Data Books. The Ingress Credit registers for Port 0 contain one set of default values, the Ingress Credit registers for Ports 1 and 2 share a different set of default values, and the Ingress Credit registers for Ports 3 and 4 share yet another different set of default values. Ingress Credits for the PEX 85xx RDK EEPROMs are programmed to the default register values shown in Table 18,
Table 19 and Table 20 below.

Table 18. PEX 85xx Default Ingress Credit Values for Port 0

	Headers		Payload		Register Value
	dec	hex	dec	hex	hex
VC0 Posted	12	C	120	78	1878
VC0 Non-Posted	7	7	7	7	0E07
VC0 Completion	10	A	104	68	1468
VC1 Posted	1	1	16	10	0210
VC1 Non-Posted	1	1	1	1	0201
VC1 Completion	1	1	16	10	0210
Total	32		263		

Table 19. PEX 85xx Default Ingress Credit Values for Ports 1 and 2

	Headers		Payload		Register Value
	dec	hex	dec	hex	hex
VC0 Posted	9	9	88	58	1258
VC0 Non-Posted	5	5	5	5	0A05
VC0 Completion	7	7	72	48	0E48
VC1 Posted	1	1	16	10	0210
VC1 Non-Posted	1	1	1	1	0201
VC1 Completion	1	1	16	10	0210
Total	24		198		

Table 20. PEX 85xx Default Ingress Credit Values for Ports 3 and 4

	Headers		Payload		Register Value
	dec	hex	dec	hex	hex
VC0 Posted	6	6	64	40	0C40
VC0 Non-Posted	4	4	4	4	0804
VC0 Completion	6	6	56	38	0C38
Total	16		124		

These values are selected for a mixture of packet types moving across the link, with all credits allocated to VC0 (since the concept of multiple Virtual Channels requires a system (hardware and software) solution and current systems have not implemented multiple VCs).

Credit values can be modified for optimization of the types of traffic in a system. For example, if all transactions to Port 0 are Posted, then credits could be changed (in the EEPROM) to 30 Posted header credits, 1 Non-Posted header credit, and 1 Completion header credit. A minimum of 1 Header Credit must always be allocated for each of VC0 Posted, VC0 Non-Posted, and VC0 Completion packets, along with the minimum number of Payload Credits to accommodate at least 1 packet carrying the Maximum Payload Size for each of the three packet types.

While the Ingress Packet RAM is large, it is possible to allocate more credits than the RAM can hold. In a working system, even though packets are continually being emptied from the Ingress Packet RAM to the Egress Packet RAM, the number of programmed credits for all ports should not be allowed to exceed the amount of Packet RAM, otherwise there is a potential for the RAM to overflow. The RAM will not overflow if all five ports are enabled and default Ingress Credit values are used.

6.11. Internal Credit Handler VCnT Threshold Registers (Port 0 Offsets C00h–C08h)

The ITCH VCnT Threshold registers define internal credits between the egress queue of one port and the ingress queues of ingress ports. The ITCH VCnT Threshold registers are programmed in the EEPROM with values to prevent filling of the Egress RAM with Posted packets, which might prevent Completion packets from making forward progress from the ingress queue to the egress queue inside the switch. The programmed values apply to all ports in the Station. While separate registers exist for Posted, Non-Posted and Completion packet counts, programming only the registers for Posted counts will prevent clogging of the Egress RAM for any combination of packets.

Recommended register values are based upon the number of ports enabled in the Station (as configured in Port 0 register 224h). A port is considered as enabled even if it is unused. The number of enabled ports is normally the same as the value of the Number of Ports Enumerated register value (Port 0 register 220h bits [22:20]), which is automatically updated by the switch after reset.

Recommended values for the VC0 Posted Lower and Upper Packet Count registers, based upon the number of enabled ports and only VC0 enabled, are listed in Table 21 below.

Table 21. ITCH VCnT VC0 Posted Packet Count, for One VC Enabled (VC0 only)

Ports	224h Value	Destination RAM Entries	VC0 Posted Lower Packet Count C00h[15:8]	VC0 Posted Upper Packet Count C00h[7:0]
2	2	256-256	8Ch	DCh
3	3	250-131-131	46h	62h
4	0,4,5,6	152-120-120-120	46h	62h
5	8,9	120–100–100–96–96	2Ah	46h

Recommended values for the VC0 and VC1 Posted Lower and Upper Packet Count registers, based upon the number of enabled ports, with both VC0 and VC1 enabled, are listed in Table 22 and Table 23 below.

Note: It is not recommended to use two VCs with 5 ports enabled, due to inefficient RAM utilization in this configuration.

Table 22. ITCH VCnT VC0 Posted Packet Count, for Two VCs Enabled

Ports	224h Value	Destination RAM Entries	VC0 Posted Lower Packet Count C00h[15:8]	VC0 Posted Upper Packet Count C00h[7:0]
2	2	256-256	54h	8Ch
3	3	250-131-131	1Ch	3Fh
4	0,4,5,6	152-120-120-120	1Ch	38h
5	8,9	120–100–100–96–96	1Ch	2Ah

Table 23. ITCH VCnT VC1 Posted Packet Count, for Two VCs Enabled

Ports	224h Value	Destination RAM Entries	VC1 Posted Lower Packet Count C04h[31:24]	VC1 Posted Upper Packet Count C04h[23:16]
2	2	256-256	2Ah	46h
3	3	250-131-131	14h	2Ah
4	0,4,5,6	152-120-120-120	14h	2Ah
5	8,9	120–100–100–96–96	12h	22h

6.12. SerDes Electrical Characteristics

SerDes Electrical Characteristics are programmed in Port 0 register addresses 248h – 258h of the PEX 85xx RDKs. The values are chip-specific and tuned to meet PCI-SIG amplitude and de-emphasis specifications with VTT = 1.5V. Other values can be programmed based upon oscilloscopic eye patterns measured for the particular design.

6.13. Non-Transparent Mode Registers

6.13.1. Device/Vendor ID, Subsystem ID and Subsystem Vendor ID

Some operating systems might require that the non-transparent bridges (NT-Virtual and NT-Link), which utilize device drivers, have different Vendor/Device IDs from that of the Transparent mode bridges (upstream and downstream transparent ports), which normally use a system driver included in the operating system.

Besides the Vendor/Device ID, the NT ports include a Subsystem ID and Subsystem Vendor ID. Normally the Vendor/Device ID identify the interface chip (the PLX switch), and the Subsystem ID and Subsystem Vendor ID identify the board. Therefore, normally the Vendor/Device ID in the NT ports should contain the default register values. This scheme allows a generic driver associated only with the Device/Vendor ID to be used with the board, or a specific driver associated with the Device/Vendor ID as well as the Subsystem and Subsystem Vendor ID to be used for the specific functionality and software developed for the board design. The Subsystem Vendor ID value must be a valid Vendor ID assigned to PCI-SIG members. If the board vendor is a member of PCI-SIG and uses it own Vendor ID, the company can freely assign its own Subsystem ID(s) to identify the board.

However, in the case of the NT-Link port, if the NT-Link interface receives a Hot Reset or its link goes down, the NT-Link registers are reset, and consequently the NT-Link registers containing the Device ID, Vendor ID, Subsystem ID and Subsystem Vendor ID are reset to PLX default values, and the EEPROM is not reloaded. Therefore, it is recommended that the NT Link Device ID, Vendor ID, Subsystem ID and Subsystem Vendor ID values remain PLX default values. In contrast, if the upstream port receives a Hot Reset or its link goes down, the transparent ports and NT-Virtual port registers are reset (unless Port 0 offset 1DCh[20] is set to disable such resets), and the EEPROM is reloaded (unless Port 0 offset 1DCh[17 or 16] is set).

Consequently, if the OS requires different Device/Vendor ID values for Transparent and Non-Transparent ports, it is recommended to change the value in the Transparent ports rather than the Non-Transparent ports.

The Subsystem IDs for the NT-Virtual and NT-Link ports can be the same value, or different values. Different Subsystem IDs might be desirable because the register sets for the NT-Virtual and NT-Link ports are different, and different software drivers for the two ports might be utilized.

If a company is not a member of PCI-SIG, PLX can assign, at no cost, Device ID(s) or Subsystem ID(s) for the product. In this case, the Vendor ID and Subsystem Vendor ID remain the default PLX value, 10B5h.

6.13.2. NT-Virtual BAR Setup and BAR Setup Shadow Registers

The BAR Setup (D4h-E0h) and corresponding BAR Setup Shadow (D84h-D90h) register pairs (for BARs 2-5) both need to be programmed prior to enabling NT-Virtual BIOS/OS configuration access to the NT-Virtual BAR registers, and therefore typically the BAR Setup and BAR Setup Shadow registers are both programmed by EEPROM (to the same value). Address space size must be a power of 2, and the corresponding size value to program into the BAR Setup and Shadow registers and enable the corresponding NT-Virtual BAR registers must be the two's complement of the address space size.

NT-Virtual BARs:

- BAR1 is an I/O BAR and is enabled as default in BAR1 Setup (D0h) and BAR1 Setup Shadow (D80h) registers. BAR1 should be disabled to conserve limited I/O resources, as recommended by the *PCI Express Base Specification*.
- BAR2 is a 32-bit memory BAR; minimum BAR size is 4 KB.
- BAR3 is a 32-bit memory BAR that uses a Look-Up Table (LUT) with 60 entries for individual address translation for each page. Page Size is programmable from 4 KB to 32 MB, corresponding to BAR3 sizes of 256 KB to 2 GB, respectively. LUT entries 30, 31, 32, and 33 are not available for use when a programmed EEPROM is used, and must be programmed to the same values as the NT-Virtual BAR Setup registers 2 through 5 (D4h through E0h), respectively.
- The BAR 4/5 group provides a 64-bit memory BAR; minimum BAR size is 4 KB. BAR4 can be used alone as a 32-bit BAR, in which case BAR5 is *reserved*.

While BAR Setup (and Shadow) registers are typically programmed by EEPROM, in certain applications it is possible that NT-Link host software can program the NT-Virtual BAR Setup and Shadow registers (the NT-Virtual register addresses are offset from NT-Link BAR0). If software, rather than the EEPROM, programs the BAR Setup and Shadow registers, the Virtual Interface Access Enable bit (Port 0 register 1DCh bit 28) should be initially disabled (0), to prevent enumeration of the NT-Virtual BAR registers from occurring until after the BAR Setup registers are programmed. After the BAR Setup registers are programmed, the Virtual Interface Access Enable bit should be set to allow NT-Virtual domain software to perform Type 0 accesses to configure the NT-Virtual BAR registers. Until this bit is set, Type 0 accesses to the NT-Virtual registers are retried (Completion with Completion Status Code set to Configuration Request Retry Status (CRS)). The Virtual Interface Access Enable bit default value is 1 if the EEPROM is not present, and otherwise 0.

6.13.3. NT-Link BAR Setup Registers

The BAR Setup registers (E8h-F4h) for BARs 2-5 need to be programmed prior to BIOS/OS configuration of the NT-Link BAR registers, and therefore typically the BARS are programmed by EEPROM. Address space size must be a power of 2, and the corresponding size value to program into the BAR Setup registers and enable the corresponding NT-Link BAR registers must be the two's complement of the address space size.

NT-Link BARs:

- BAR1 is an I/O BAR and is enabled as default in the BAR0/BAR1 Setup register (E4h). BAR1 can be disabled to conserve limited I/O resources.
- The BAR 2/3 group provides a 64-bit memory BAR; minimum BAR size is 4 KB. BAR2 can be used alone as a 32-bit BAR, in which case BAR3 is *reserved*.
- The BAR 4/5 group provides a 64-bit memory BAR; minimum BAR size is 4 KB. BAR4 can be used alone as a 32-bit BAR, in which case BAR5 is *reserved*.

While BAR Setup registers are typically programmed by EEPROM, in certain applications it is possible that NT-Virtual host software can program the NT-Link BAR Setup registers (the NT-Link register addresses are offset from Upstream Port BAR0); if software rather than the EEPROM programs the BAR Setup registers, the Link Interface Access Enable bit (Port 0 register 1DCh bit 29) should be initially disabled, to prevent enumeration of the NT-Link BAR registers from occurring until after the BAR Setup registers are programmed. After the BAR Setup registers are programmed, the Link Interface Access Enable bit should be set to allow NT-Link domain software to perform Type 0 accesses to configure the NT-Link BAR registers. Until this bit is set, Type 0 accesses in Intelligent Adapter mode are retried, and in Dual-Host mode, aborted. The Link Interface Access Enable bit default value in NT mode is 1.

6.13.4. BAR Limit Registers

BAR Limit registers, if used (for efficient use of memory resources) are typically programmed at runtime (because BARs are typically programmed at runtime, such as in a Plug 'n' Play system). BAR Limit registers contain an address; if the value is outside the range of the associated BAR register, the BAR Limit is ignored.

6.13.5. Memory BAR Address Translation Registers

Memory BAR Address Translation registers for NT-Virtual BARs 2, 4 and 5 contain the NT-Link domain base address. BAR2 and BAR4 Address Translation registers provide the lower 32-bit translation address. The BAR5 Address Translation register provides the upper 32-bit translation address when BARs 4/5 are configured as a 64-bit BAR. Typically the NT-Link domain host programs the Address Translation register values (such as for an NT-Link system memory page-locked buffer address) at runtime, using address NT-Link BAR0 + [10C3Ch-10C48h]. The Translation Address must be a multiple of the address space size.

NT-Virtual BAR3 provides a Look-Up Table (LUT) Address Translation set of registers, for 64 different NT-Link domain base addresses, in offsets C5Ch-D58h (NT-Link BAR0 + [10C5Ch-10D58h]). The NT-Virtual BAR3 Setup register (D8h) and BAR3 Setup Shadow register (D88h) provide programmable Page Sizes from 4 KB to 32 MB (i.e., BAR3 size is programmable from 256 KB to 2 GB, according to the selected Page Size).

Memory BAR Address Translation registers for NT-Link BARs contain the NT-Virtual domain base address. BAR2 and BAR4 Address Translation registers provide the lower 32-bit translation address. BAR3 and BAR5 Address Translation registers provide the upper 32-bit translation address when BARs 2/3 and 4/5 respectively are configured as 64-bit BARs. Typically, the NT-Virtual domain host programs this value (such as a system memory page-locked buffer address) at runtime, using address Upstream Port BAR0 + [11C3Ch-11C48h]. The Translation Address must be a multiple of the address space size.

Both sets of NT-Virtual and NT-Link Memory BAR Address Translation registers have the same offsets (C3Ch-C48h).

6.14. Shadowed Registers

In order for the PEX 85xx switch to properly route memory and I/O requests and completions, Port 0 contains Content Addressable Memory (CAM) registers that hold mirror copies of certain registers that exist in each port.

- AMCAM (Address Mapping CAM) registers determine the route of a memory request, and contain mirror copies of the Memory Base and Limit registers in each of the virtual PCI-PCI bridges in each transparent port.
- IOAMCAM (I/O Address Mapping CAM) registers determine the route of an I/O request, and contain mirror copies of the I/O Base and Limit registers in each of the virtual PCI-PCI bridges in each transparent port.
- BusNoCAM (Bus Number Mapping CAM) registers determine the route of a completion, and contain mirror copies of the Secondary Bus Number and Subordinate Bus Number registers in each of the virtual PCI-PCI bridges in each transparent port.

6.14.1. Shadowed Transparent Port Registers

Table 24 below lists the twelve registers in each Transparent port that are shadowed.

Table 24. Transparent Port Shadowed Registers

CSR Address	Description	Default Value	Shadow Registers in Port 0
10h	BAR0 (Upstream Port only)	00000000h	6C0h for Port 0, 6C8h for Port 1, 6D0h for Port 2, 6D8h for Port 3, 6E0h for Port 4
14h	BAR1 (Upstream Port only)	00000000h	6C4h for Port 0, 6CCh for Port 1, 6D4h for Port 2, 6DCh for Port 3, 6E4h for Port 4
18h	Bus Number	00000000h	2C8h for Port 0, 2CCh for Port 1, 2D0h for Port 2, 2D4h for Port 3, 2D8h for Port 4
1Ch [15:0]	I/O Base & Limit	01F1h	308h[15:0] for Port 0, 308h[31:16] for Port 1 30Ch[15:0] for Port 2, 30Ch[31:16] for Port 3, 310h[15:0] for Port 4
20h	Memory Base & Limit	0000FFF0h	348h for Port 0, 358h for Port 1, 368h for Port 2, 378h for Port 3, 388h for Port 4
24h	Prefetchable Memory Base & Limit	0001FFF1h	34Ch for Port 0, 35Ch for Port 1, 36Ch for Port 2, 37Ch for Port 3, 38Ch for Port 4
28h	Prefetchable Memory Upper Base (rev. AA default = FFFFFFFFh)	00000000h	350h for Port 0, 360h for Port 1, 370h for Port 2, 380h for Port 3, 390h for Port 4
2Ch	Prefetchable Memory Upper Limit	00000000h	354h for Port 0, 364h for Port 1, 374h for Port 2, 384h for Port 3, 394h for Port 4
30h	I/O Base Upper 16-Bits & I/O Limit Upper 16-Bits	0000FFFFh	680h for Port 0, 684h for Port 1, 688h for Port 2, 68Ch for Port 3, 690h for Port 4
14Ch	Port VC Capability 1	00000001h	840h for Port 0, 844h for Port 1, 848h for Port 2, 84Ch for Port 3, 850h for Port 4
15Ch	VC0 Resource Control	800000FFh	740h for Port 0, 748h for Port 1, 750h for Port 2, 758h for Port 3, 760h for Port 4
168h	VC1 Resource Control	01000000h	744h for Port 0, 74Ch for Port 1, 754h for Port 2, 75Ch for Port 3, 764h for Port 4

The nine PCI Type 1 Header registers (offsets 10h through 30h) are defined by the *PCI-PCI Bridge Architecture Specification*. The three Virtual Channel registers that are shadowed are part of the PCI Express Virtual Channel Capability that is defined by the *PCI Express Base Specification*.

The Type 1 Header registers are typically programmed at boot time by BIOS/OS. When a shadowed register is written, its corresponding CAM registers are automatically updated by the PEX 85xx switch. If standard PCI BIOS/OS enumerates the Type 1 Header registers, the corresponding shadow register locations in the serial EEPROM should generally be programmed to the corresponding Type 1 registers' default values, and the shadow registers will be automatically updated when the Type 1 registers are enumerated.

If a shadowed register is programmed to a non-default value in the EEPROM, then the corresponding shadow registers should be programmed to the value of the shadowed register.

One exception to the default value rule is the Bus Number (18h) register. The default register value is 0h, and in the corresponding BusNoCAM registers, the Secondary Bus Number default value is FFh (the register value is 0000FF00h). Until BIOS/OS enumerates the Bus Number register after the discovery phase, the shadow register value must be higher than the Type 1 register value in order to route Configuration Request completions to the upstream port.

The other exception to the default value rule is the Port VC Capability 1 (14Ch) register. The default value is 1h, and in the corresponding shadow registers, the default value is 0h since only the Low Priority Extended VC Count (bit 4) is shadowed.

6.14.2. Shadowed Non-Transparent Port Registers

Table 25 below lists the Non-Transparent port registers that are shadowed. Shadow registers in Port 0 are not loaded from EEPROM.

Table 25. Non-Transparent Port Shadowed Registers

Port	CSR Address	Description	Default Value	Shadow Registers	Comments
NT-Virtual	14h	Base Address 1 (BAR1, I/O)	00000001h	D6Ch in NT-Virtual, *0*	
NT-Virtual	18h	Base Address 2 (BAR2)	00000000h	D70h in NT-Virtual, *0*	
NT-Virtual	1Ch	Base Address 3 (BAR3)	00000000h	D74h in NT-Virtual, *0*	
NT-Virtual	20h	Base Address 4 (BAR4)	00000000h	D78h in NT-Virtual, *0*	
NT-Virtual	24h	Base Address 5 (BAR5)	00000000h	D7Ch in NT-Virtual, *0*	
NT-Virtual	D0h	BAR1 Setup	00000003h	D80h in NT-Virtual, *0*	
NT-Virtual	D4h	BAR2 Setup	00000000h	D84h in NT-Virtual, *0*	
NT-Virtual	D8h	BAR3 Setup	00000000h	D88h in NT-Virtual, *0*	
NT-Virtual	DCh	BAR4 Setup	00000000h	D8Ch in NT-Virtual, *0*	
NT-Virtual	E0h	BAR5 Setup	00000000h	D90h in NT-Virtual, *0*	
NT-Virtual	C3Ch	BAR2 Address Translation	00000000h	*Port 0*	
NT-Virtual	C44h	BAR4 Address Translation	00000000h	*Port 0*	
NT-Virtual	C48h	BAR5 Address Translation [63:32]	00000000h	*Port 0*	
NT-Virtual	C4Ch	BAR2 Limit	00000000h	*Port 0*	
NT-Virtual	C54h	BAR4 Limit	00000000h	*Port 0*	
NT-Virtual	C58h	BAR5 Limit [63:32}	00000000h	*Port 0*	
NT-Virtual	C5Ch – D58h	BAR3 Address Translation LUT	00000000h	*C5Ch-D58h in Port 0*	
NT-Link					
NT-Link	14Ch	Port VC Capability 1 (only bit 4 is shadowed)	00000001h	D64h in NT-Virtual,*0*	
NT-Link	15Ch	VC0 Resource Control (bits [31, 24, 7:0] are shadowed)	800000FFh	D5Ch in NT-Virtual, *0*	
NT-Link	168h	VC1 Resource Control (bits [31, 24, 7:0] are shadowed)	01000000h	D60h in NT-Virtual, *0*	

The primary purpose of the Non-Transparent shadow registers is to provide address and routing information in constant locations, regardless of which port is the NT port. The five NT-Virtual port Type 0 Base Address registers (BARs 1 – 5) are shadowed. Additionally the BAR Setup registers, which specify the ranges for the Base Address registers 2 – 5, are shadowed. The BAR1 Setup register enables/disables BAR1, which is an I/O BAR for I/O command access to registers (however since I/O mapping is strongly discouraged, BAR1 should be disabled unless required for legacy I/O access).

The NT-Virtual registers not only have the shadow registers, but the shadow registers themselves have their own shadow registers that exist in Port 0, at the same offsets as used in the NT-Virtual port. These secondary shadow registers (in Port 0) are not loaded directly from EEPROM; they are updated automatically whenever the NT-Virtual shadow registers are written by EEPROM or by software. For example, when software enumerates an NT-Virtual BAR register, the value is automatically copied to the corresponding shadow register in the NT-Virtual port, and that write automatically causes the value to be written to the same offset in Port 0.

The NT-Link port contains three shadow registers, which are the same PCI Express Virtual Channel Capability registers that are shadowed in the transparent ports. Only certain bits in these three registers need to be shadowed; shadow register bits that are described as *Reserved* should be programmed to 0.

Software Requirements for Setup of Non-Transparent Mode

Programming of registers for Non-Transparent mode operation registers requires some software setup, since the EEPROM cannot load all registers. The following operations are required:

1. Program the NT-Virtual BAR Setup Shadow registers (D80h to D90h) to the same value as the NT-Virtual BAR Setup registers (D0 to E0h); these registers can be programmed by EEPROM, and all of these registers must be programmed prior to programming the corresponding NT-Virtual BAR registers.
2. Software must write the NT-Virtual Send Lookup Table entries (NT-Virtual D94h to DB0h, cannot be programmed by EEPROM).
3. Software must write the NT-Link Receive Lookup Table entries (NT-Link DB4h to DF0h, cannot be programmed by EEPROM).
4. Software must program the NT-Link Command register to set the Master Enable (bit 2) and Memory Enable (bit 1) bits (NT-Link 04h cannot be programmed by EEPROM).
5. The NT-Virtual CPU must perform a Configuration Write to the NT-Virtual port; this requirement is satisfied by NT-Virtual host enumeration.
6. The NT-Link CPU must perform a Configuration Write to the NT-Link port; this requirement is satisfied by NT-Link host enumeration.

7. Table of RDK EEPROM Values that Differ from PEX 8518/8517/8512/8508 Default Values

Table 26 below lists register values that have been programmed into the PEX 8518RDK, PEX 8517RDK, PEX 8512RDK and PEX 8508RDK EEPROM images, that are not the default register values listed in the Data Book. These tables do not include all registers that are loaded from EEPROM (as listed in the Data Book), since most registers do not need to be changed from default register values (which are already programmed into the RDK EEPROM images).

Table 26. RDK EEPROM Values that Differ from PEX 8518/8517/8512/8508 Default Values

Ports	Offset	Register Name	Bits	Bit Field Name	Default Value	Program Value	Comments
0,1,2,3,4, NT-Virtual NT-Link	08h	Class Code and Revision ID	31:0		**Rev. AA** 060400AA	**Rev. AA** 060400AA	Silicon Revision ID In Rev. AB, bit 0 is hardwired to 1. In Rev. AC, bits [2:1] are reversed - EEPROM value of AAh read from register as ACh; - EEPROM value of ACh read from register as AAh. Class Code 060400h = transparent PCI-PCI Bridge 068000h = Other Bridge
					Rev. AB 060400AB	**Rev. AB** 060400AB	
					Rev. AC 060400AC	**Rev. AC** 060400AA	
8512 1,3,4	**7Ch**	**Slot Capabilities**			**00000CDF**	**00000C80**	
8512: 1,3,4	7Ch	Slot Capabilities	0	Attention Button Present	1	0	Upstream & NT port value = 0 Downstream transparent port: 1 if Slot has Attention Button; 0 if Device, or Slot not Hot Plug.
8512: 1,3,4	7Ch	Slot Capabilities	1	Power Controller Present	1	0	Upstream & NT port value = 0 Downstream transparent port: 0 to disable Hot Plug Controller 1 to enable Hot Plug Controller (HP_PWREN#, HP_PWRLED#, HP_CLKEN#, HP_PWRFLT#) If HPC is enabled (bit 1 = 1), set delays, for HP_CLKEN# output assertion, and subsequent HP_PERST# output de-assertion, in port offset 1E0h bits [6,4:3].
8512: 1,3,4	7Ch	Slot Capabilities	2	MRL Sensor Present	1	0	Upstream & NT port value = 0 Downstream transparent port: 1 if Slot has MRL Sensor; 0 if Device, or Slot is not Hot Plug.
8512: 1,3,4	7Ch	Slot Capabilities	3	Attention Indicator Present	1	0	Upstream & NT port value = 0 Downstream transparent port:: 1 if Slot has Attention Indicator; 0 if Device, or Slot is not Hot Plug.
8512: 1,3,4	7Ch	Slot Capabilities	4	Power Indicator Present	1	0	Upstream & NT port value = 0 Downstream transparent port: 1 if Slot has Power Indicator; 0 if Device, or Slot is not Hot Plug.
8512: 1,3,4	7Ch	Slot Capabilities	6	Hot Plug Capable	1	0	Upstream & NT port value = 0 Downstream transparent port: 0 if Device, or Slot not Hot Plug; 1 if Slot supports Hot Plug.
	80h	**Slot Status and Control**	**31:0**		**000007C0**	**00000000**	
1,2,3,4	80h	Slot Control	7:6	Attention Indicator Control	11b	00b	
1,2,3,4	80h	Slot Control	9:8	Power Indicator Control	11b	00b	
1,2,3,4	80h	Slot Control	10	Power Controller Control	1	0	Downstream transparent port: 0 = de-assert HP_PERST# (high); 1 = assert HP_PERST# (low).

Ports	Offset	Register Name	Bits	Bit Field Name	Default Value	Program Value	Comments
0	**1DCh**	**Debug Control**	**31:0**			**8508: Trans-parent Rev. AA, AB: 310C4000 Rev. AC: 312C4000** **8512: Trans-parent 312C4000** **8517: Trans-parent Rev. AA, AB: 330C4000 Rev. AC: 332C4000** **8518: Trans-parent Rev. AA, BA: 310C4000 Rev. AC: 312C4000**	**If in NT mode the EEPROM programs Port 0 to be the NT Port, then some Port 0 registers are loaded from the corresponding NT-Virtual register offsets in the EEPROM. If Port 0 is the NT port:** **1. The EEPROM location for NT-Virtual offset 1DCh must be programmed to the same value as Port 0 offset 1DCh.** **2. If 1DCh[30] = 1 to enable software or EEPROM control of PEX_LANE_GOOD# outputs (by writing to Port 0 register 1F4h[31:0]), then the EEPROM location for NT-Virtual offset 1F4h must be programmed to the same value as Port 0 offset 1F4h.**
0	1DCh	Debug Control	6:4	Cut-Thru Port Number	Strap Levels	000b	STRAP_UPSTRM_PORT_SEL[3:0]
0	1DCh	Debug Control	10:8	Upstream Port ID	Strap Levels	000b	STRAP_UPSTRM_PORT_SEL[3:0]
0	1DCh	Debug Control	19:18	Mode Select	Strap Levels	11b	STRAP_MODE_SEL[1:0] 11 = Transparent Mode 10 = NT Dual Host 01 = NT Intelligent Adapter
0	1DCh	Debug Control	22	Cut-through Enable	0	**Rev. AA, AB:** 0 **Rev. AC:** 1	
0	1DCh	Debug Control	27:24	NT Port Number	Strap Levels	**8508**: 1h **8512**: 1h **8517**: 3h **8518**: 1h	STRAP_NT_UPSTRM_PORT_SEL [3:0] In Transparent mode, can be set to Fh to match the signal strapping.
0	1DCh	Debug Control	29	Link Interface Access Enable	**Trans-parent** or **Intelligent Adapter** 0 **Dual Host** 1	1	Link Activation

Ports	Offset	Register Name	Bits	Bit Field Name	Default Value	Program Value	Comments
8508 0,1,2, NT-Virtual	**1F8h**	**ACK Transmission Latency Limit**	**31:0**		**00FF00FF**	**00030080**	
3,4						**000500FA**	
8512 0,1,2,3,4, NT-Virtual						**00020049**	
						00030080	
8517 0,1,2,3,4, NT-Virtual						**00020049**	
8518 0,1,2, NT-Virtual						**00020049**	
3,4						**00030080**	
8508 0,1,2, NT-Virtual	1F8h	ACK Transmission Latency Limit	7:0	ACK Transmission Latency Limit	FFh	80h	Value based upon link width and Maximum Payload Size (70h[7:5]). 128 byte Maximum Payload Size: x8 = 43h, x4 = 49h, x2 = 80h, x1 = FAh 256 byte Maximum Payload Size: x8 = 6Bh, x4 = 76h, x2 = D9h, x1 = FFh NT-Virtual value: must not be 0; set NT-Virtual value >= NT-Link width NT-Link value = 00h
3,4						FAh	
8512 0,1,2, NT-Virtual						49h	
3,4						80h	
8717 0,1,2,3,4, NT-Virtual						49h	
8518 0,1,2, NT-Virtual						49h	
3,4						80h	
8508 0,1,2, NT-Virtual	1F8h	ACK Transmission Latency Limit	23:16	Upper 8 Bits of Replay Timer Limit	FFh	03h	Value based upon link width. x8 = 02h, x4 = 02h, x2 = 03h, x1 = 05h Set NT-Virtual value according to the NT-Link width. NT-Link value = FFh
3,4						05h	
8512 0,1,2, NT-Virtual						02h	
3,4						03h	
8717 0,1,2,3,4, NT-Virtual						02h	
8518 0,1,2, NT-Virtual						02h	
3,4						03h	

Ports	Offset	Register Name	Bits	Bit Field Name	Default Value	Program Value	Comments
	224h	**Port Configuration**					**STRAP_STN0_PORTCFG[3:0]**
0	224h	Port Configuration	4:0	Port Configuration	Strap Levels	**8508:** 9h **8512:** 3h **8517:** 0h **8518:** 9h	x2, x2, x2, x1, x1 x4, x4, x4 x4, x4, x4, x4 x4, x4, x4, x2, x2
		Electrical Characteristics					
0	254h	SerDes Driver Equalization Level Select 1	31:0		88888888	99999999	
0	258h	SerDes Driver Equalization Level Select 2	31:0		88888888	99999999	
0	668h	TIC Port Enable	31:0	TIC_ UNP_STATUS	FFFFFFFF	**8508** FFFF001F **8512** FFFF0007 **8517** FFFF000F **8518** FFFF001F	Set upper 16-bits = FFFFh. For lower 16-bits, set each bit position that corresponds to an enabled port.

<table>
<tr><th>Ports</th><th>Offset</th><th>Register Name</th><th>Bits</th><th>Bit Field Name</th><th>Default Value</th><th>Program Value</th><th>Comments</th></tr>
<tr><td>8508
0</td><td rowspan="4">C00h</td><td rowspan="4">ITCH VC&T Threshold_1</td><td rowspan="4">31:0</td><td rowspan="4"></td><td rowspan="4">7FFF7FFF</td><td>7FFF2A46</td><td rowspan="4"></td></tr>
<tr><td>8512
0</td><td>7FFF2A46</td></tr>
<tr><td>8517
0</td><td>7FFF4662</td></tr>
<tr><td>8518
0</td><td>7FFF2A46</td></tr>
<tr><td>8508
0</td><td rowspan="4">C00h</td><td rowspan="4">ITCH VC&T Threshold_1</td><td rowspan="4">7:0</td><td rowspan="4">VC0 Posted Upper Packet Count</td><td rowspan="4">FF</td><td>46h</td><td rowspan="4">Value determined by number of ports in the Station.<table><tr><th>Ports</th><th>224h value</th><th>Value</th></tr><tr><td>2</td><td>2</td><td>DCh</td></tr><tr><td>3</td><td>3</td><td>62h</td></tr><tr><td>4</td><td>0,4,5,6</td><td>62h</td></tr><tr><td>5</td><td>8,9</td><td>46h</td></tr></table></td></tr>
<tr><td>8512
0</td><td>2Ah</td></tr>
<tr><td>8517
0</td><td>62h</td></tr>
<tr><td>8518
0</td><td>46h</td></tr>
<tr><td>8508
0</td><td rowspan="4">C00h</td><td rowspan="4">ITCH VC&T Threshold_1</td><td rowspan="4">15:8</td><td rowspan="4">VC0 Posted Lower Packet Count</td><td rowspan="4">7F</td><td>2Ah</td><td rowspan="4">Must be less than VC0 Posted Upper Packet Count.
Value determined by number of ports in the Station.<table><tr><th>Ports</th><th>224h value</th><th>Value</th></tr><tr><td>2</td><td>1</td><td>8Ch</td></tr><tr><td>3</td><td>2</td><td>46h</td></tr><tr><td>4</td><td>0,4,5,6</td><td>46h</td></tr><tr><td>5</td><td>8,9</td><td>2Ah</td></tr></table></td></tr>
<tr><td>8512
0</td><td>2Ah</td></tr>
<tr><td>8517
0</td><td>46h</td></tr>
<tr><td>8518
0</td><td>2Ah</td></tr>
</table>

Ports	Offset	Register Name	Bits	Bit Field Name	Default Value	Program Value	Comments
NT-Virtual							
NT-Virtual	D0h	BAR1 Setup	31:0		3h	0h	NT-Virtual Config register access 3h = enable BAR1 I/O, 256 bytes; All other codes disable BAR1.
0, NT-Virtual	D80h	BAR1 Setup Shadow	31:0		3h	0h	Shadow copy of D0h. EEPROM load of NT-Virtual offset D80h automatically shadows to NT Station (Port 0 or Port 8) only, and not to 8532/8524 non-NT Station (Port 0 or Port 8). EEPROM cannot load Ports 0 & 8.
NT-Link							
NT-Link	30h	Expansion ROM Base Address	31:0		1h	0h	32K = FFFF8001h
NT-Link	E4h	Configuration BAR Setup	31:0		3h	0h	NT-Link Config register access 3h = enable BAR1 I/O, 256 bytes; 0h = BAR0 only
NT-Virtual NT-Link							
NT-Virtual NT-Link	FCh	Configuration Data	31:0		0h	851810B5 or 851710B5 or 851210B5 or 850810B5	EEPROM indirect load bug

8. References

1. *PEX 85xx Data Books*:
   http://www.plxtech.com/products/expresslane/
2. *PCI Express™ Base Specification Revision 1.1*:
   http://www.pcisig.com/specifications/pciexpress/base
3. *Atmel AT25640A Data Sheet*:
   http://www.atmel.com/dyn/resources/prod_documents/doc3347.pdf
4. *Atmel AT25128A Data Sheet*:
   http://www.atmel.com/dyn/resources/prod_documents/doc3368.pdf
5. *Atmel AT25HP512 Data Sheet*:
   http://www.atmel.com/dyn/resources/prod_documents/doc1113.pdf