Carter Buck, Sr. Applications Engineer, PLX Technology

1 Introduction

This document provides the basic information for setup and demonstration of Non-Transparent Mode operation. This document applies to all PLX Switches that support Non-Transparent Mode. Section 2.3 (EEPROM Content) applies to PLX's Gen 2 Switches only (PLX's Gen 1 Switches used a different EEPROM which required the entire EEPROM to be loaded).

2 Content

Tables

- **NT-Virtual Port Registers**

  Table 1 lists the NT-Virtual Port registers that need to be programmed for packet addressing and routing. Since the BAR Setup registers need to be programmed prior to BIOS enumeration of the BAR registers, generally an EEPROM (or I^2C) is needed to program the NT registers.

- **Shadowed NT-Virtual Port Registers**

  Table 2 lists the NT-Virtual Port registers that are copied to each Station. These copies are termed shadow registers, which are also programmed by EEPROM (or I^2C).

- **NT-Link Port Registers**

  Table 3 lists the NT-Link Port registers that need to be programmed for addressing and routing. The NT-Link registers are not shadowed.

- **Sample EEPROM Values**

  Table 4 provides a sample serial EEPROM image for Non-Transparent mode, using Port 0 as the upstream port and Port 1 as the NT port.

2.1 Listings of NT Port Registers used for addressing and routing

- **NT-Virtual Port Registers**

Table 1. NT-Virtual Port BAR, BAR Setup, Address Translation, and Requester ID Registers

Ports	Offset	Register Name	Bits	Bit Field Name	Default Value	Program Value	Comments
NT-Virtual							
NT-Virtual	10h	BAR0	31:0		0h	0h	32-bit or lower 32-bit of 64-bit BAR
NT-Virtual	14h	BAR1	31:0		0h	0h	upper 32-bit of 64-bit BAR
NT-Virtual	D0h	Configuration BAR Setup	31:0		2h	2h	NT-Virtual Config register access 00 = Disable BAR0 and BAR1 01 = ***Reserved*** 10 = Enables BAR0 and Disables BAR1 (default) 11 = Enables BAR0 and BAR1
NT-Virtual	D80h	Configuration BAR Setup Shadow	31:0		2h	2h	Shadow copy of D0h
NT-Virtual	18h	BAR2	31:0		0h	0h	
NT-Virtual	D4h	BAR2 Setup	31:0		0	0h	32-bit or lower 32-bits of 64-bit BAR
NT-Virtual	D84h	BAR2 Setup Shadow	31:0		0h	0h	Shadow copy of D4h
NT-Virtual	C3Ch	BAR2 Address Translation	31:0		0h	0h	Link-side address. Must be a multiple of address space size.
NT-Virtual	1Ch	BAR3	31:0		0h	0h	
NT-Virtual	D8h	BAR3 Setup	31:0		0h	0h	32-bit or upper 32-bit of 64-bit BAR
NT-Virtual	D88h	BAR3 Setup Shadow	31:0		0h	0h	Shadow copy of D8h
NT-Virtual	C40h	BAR3 Address Translation	31:0		0h	0h	Link-side address. Must be a multiple of address space size.
NT-Virtual	20h	BAR4	31:0		0h	0h	
NT-Virtual	DCh	BAR4 Setup	31:0		0h	0h	32-bit or lower 32-bits of 64-bit BAR
NT-Virtual	D8Ch	BAR4 Setup Shadow	31:0		0h	0h	Shadow copy of DCh
NT-Virtual	C44h	BAR4 Address Translation	31:0		0h	0h	Link-side address. Must be a multiple of address space size.
NT-Virtual	24h	BAR5	31:0		0h	0h	
NT-Virtual	E0h	BAR5 Setup	31:0		0h	0h	32-bit or upper 32-bit of 64-bit BAR
NT-Virtual	D90h	BAR5 Setup Shadow	31:0		0h	0h	Shadow copy of E0h
NT-Virtual	C48h	BAR5 Address Translation	31:0		0h	0h	Link-side address. Must be a multiple of address space size.
NT-Virtual		**Requester ID Translation Lookup Table**					**Requester ID (B/D/F number)** **[2:0] Function** **[7:3] Device** **[15:8] Bus** **[30] No Snoop** **[31] Enable bit**
NT-Virtual	D94h	LUT Entry 0	31:0		0h	0h	
NT-Virtual	D98h	LUT Entry 1	31:0		0h	0h	
NT-Virtual	D9Ch	LUT Entry 2	31:0		0h	0h	
NT-Virtual	DA0h	LUT Entry 3	31:0		0h	0h	
NT-Virtual	DA4h	LUT Entry 4	31:0		0h	0h	
NT-Virtual	DA8h	LUT Entry 5	31:0		0h	0h	
NT-Virtual	DACh	LUT Entry 6	31:0		0h	0h	
NT-Virtual	DB0h	LUT Entry 7	31:0		0h	0h	

- **Shadowed NT-Virtual Port Registers**

  Table 2 below lists the NT-Virtual Port registers that are shadowed into the lowest number port of each Station (Ports 0, 4 and 8). The NT-Virtual BAR and BAR Setup registers (all located at offsets below 100h) also have corresponding shadow registers located at offsets above 100h in the NT-Virtual port.

  The registers located above offset 100h are additionally shadowed to the same offsets in each Station. If software writes to the listed NT-Virtual register (located above 100h), the value is automatically copied to the same offset in each Station. If EEPROM or I^2C writes to the NT-Virtual register, the value is not copied to each Station, and therefore EEPROM or I^2C must additionally write the value to each Station.

Table 2. Shadowed NT-Virtual Port Registers

Port	CSR Address	Description	Default Value	Shadow Registers	Comments
NT-Virtual	10h	Base Address 0 (BAR0)	00000000h	D68h in NT-Virtual, *0,4,8*	
NT-Virtual	14h	Base Address 1 (BAR1)	00000000h	D6Ch in NT-Virtual, *0,4,8*	
NT-Virtual	18h	Base Address 2 (BAR2)	00000000h	D70h in NT-Virtual, *0,4,8*	
NT-Virtual	1Ch	Base Address 3 (BAR3)	00000000h	D74h in NT-Virtual, *0,4,8*	
NT-Virtual	20h	Base Address 4 (BAR4)	00000000h	D78h in NT-Virtual, *0,4,8*	
NT-Virtual	24h	Base Address 5 (BAR5)	00000000h	D7Ch in NT-Virtual, *0,4,8*	
NT-Virtual	D0h	BAR0/1 Setup	00000002h	D80h in NT-Virtual, *0,4,8*	
NT-Virtual	D4h	BAR2 Setup	00000000h	D84h in NT-Virtual, *0,4,8*	
NT-Virtual	D8h	BAR3 Setup	00000000h	D88h in NT-Virtual, *0,4,8*	
NT-Virtual	DCh	BAR4 Setup	00000000h	D8Ch in NT-Virtual, *0,4,8*	
NT-Virtual	E0h	BAR5 Setup	00000000h	D90h in NT-Virtual, *0,4,8*	
NT-Virtual	C3Ch	BAR2 Address Translation	00000000h	*Ports 0,4,8*	
NT-Virtual	C40h	BAR3 Address Translation	00000000h	*Ports 0,4,8*	
NT-Virtual	C44h	BAR4 Address Translation	00000000h	*Ports 0,4,8*	
NT-Virtual	C48h	BAR5 Address Translation	00000000h	*Ports 0,4,8*	
NT-Virtual	D94h	Requester ID Translation 0	00000000h	*Ports 0,4,8*	
NT-Virtual	D98h	Requester ID Translation 1	00000000h	*Ports 0,4,8*	
NT-Virtual	D9Ch	Requester ID Translation 2	00000000h	*Ports 0,4,8*	
NT-Virtual	DA0h	Requester ID Translation 3	00000000h	*Ports 0,4,8*	
NT-Virtual	DA4h	Requester ID Translation 4	00000000h	*Ports 0,4,8*	
NT-Virtual	DA8h	Requester ID Translation 5	00000000h	*Ports 0,4,8*	
NT-Virtual	DACh	Requester ID Translation 6	00000000h	*Ports 0,4,8*	
NT-Virtual	DB0h	Requester ID Translation 7	00000000h	*Ports 0,4,8*	

For example, if BAR0/1 is to be a 64-bit BAR, the BAR0/1 Setup register (NT-Virtual register D0h) and BAR0/1 Setup Shadow register (NT-Virtual register D80h) must both be programmed to value 3h.

- If software programs the NT-Virtual BAR0/1 Setup Shadow register (D80h), the value (3h) is automatically copied to the same offset (D80h) in Ports 0, 4 and 8.
- If EEPROM or I^2C programs the NT-Virtual BAR0/1 Setup Shadow register (D80h), then EEPROM or I^2C must also write the value (3h) to register D80h in Ports 0, 4 and 8.

- **NT-Link Port Registers**

Table 3. NT-Link Port BAR, BAR Setup, Address Translation, and Requester ID Registers

Ports	Offset	Register Name	Bits	Bit Field Name	Default Value	Program Value	Comments
NT-Link							
NT-Link	10h	BAR0	31:0		0h	0h	32-bit or lower 32-bit of 64-bit BAR
NT-Link	14h	BAR1	31:0		0h	0h	upper 32-bit of 64-bit BAR
NT-Link	E4h	Configuration BAR Setup	31:0		2h	2h	NT-Link Config register access 00 = Disable BAR0 and BAR1 01 = ***Reserved*** 10 = Enables BAR0 and Disables BAR1 (default) 11 = Enables BAR0 and BAR1
NT-Link	18h	BAR2	31:0		0h	0h	
NT-Link	E8h	BAR2 Setup	31:0		0h	0h	32-bit or lower 32-bits of 64-bit BAR
NT-Link	C3Ch	BAR2 Address Translation	31:0		0h	0h	Virtual-side address. Must be a multiple of address space size.
NT-Link	1Ch	BAR3	31:0		0h	0h	
NT-Link	ECh	BAR3 Setup	31:0		0h	0h	32-bit or upper 32-bit of 64-bit BAR
NT-Link	C40h	BAR3 Address Translation	31:0		0h	0h	Virtual-side address. Must be a multiple of address space size.
NT-Link	20h	BAR4	31:0		0h	0h	
NT-Link	F0h	BAR4 Setup	31:0		0h	0h	32-bit or lower 32-bits of 64-bit BAR
NT-Link	C44h	BAR4 Address Translation	31:0		0h	0h	Virtual-side address. Must be a multiple of address space size.
NT-Link	24h	BAR5	31:0		0h	0h	
NT-Link	F4h	BAR5 Setup	31:0		0h	0h	32-bit or upper 32-bit of 64-bit BAR
NT-Link	C48h	BAR5 Address Translation	31:0		0h	0h	Virtual-side address. Must be a multiple of address space size.
NT-Link		**Requester ID Translation Lookup Table Entries_*n*_*m***					**Requester ID** ***n*** | ***m*** [0] Enable | [16] Enable [1] No Snoop | [17] No Snoop [7:3] Device | [23:19] Device [15:8] Bus | [31:24] Bus **(Bus << 8) || (Dev << 3) || (No Snoop Enable << 1) || Enable**
NT-Link	DB4h	Entry_0_1	31:0		0h	0h	
NT-Link	DB8h	Entry_2_3	31:0		0h	0h	
NT-Link	DBCh	Entry_4_5	31:0		0h	0h	
NT-Link	DC0h	Entry_6_7	31:0		0h	0h	
NT-Link	DC4h	Entry_8_9	31:0		0h	0h	
NT-Link	DC8h	Entry_10_11	31:0		0h	0h	
NT-Link	DCCh	Entry_12_13	31:0		0h	0h	
NT-Link	DD0h	Entry_14_15	31:0		0h	0h	
NT-Link	DD4h	Entry_16_17	31:0		0h	0h	
NT-Link	DD8h	Entry_18_19	31:0		0h	0h	
NT-Link	DDCh	Entry_20_21	31:0		0h	0h	
NT-Link	DE0h	Entry_22_23	31:0		0h	0h	
NT-Link	DE4h	Entry_24_25	31:0		0h	0h	
NT-Link	DE8h	Entry_26_27	31:0		0h	0h	
NT-Link	DECh	Entry_28_29	31:0		0h	0h	
NT-Link	DF0h	Entry_30_31	31:0		0h	0h	

2.2 Determining register values

- **Base Address Register (BAR)**

  Address space size must be 1 MB or greater, and must be a power of 2.

  BARs are typically programmed by system software, and the corresponding BAR Setup registers must be programmed prior to enumeration.

- **BAR Setup**

  The BAR Setup register value is calculated as the two's complement of the address space size. Then bits [3:0] of the lower 32-bit BAR must be configured if the address space is Prefetchable, or 64-bit.

  The two's complement can be calculated by using the **+/-** key on the calculator. For example, as 1 MB is 0x100000h, if 8 MB is required, enter 800000 into the hexadecimal calculator, then click the **+/-** key and then the **=** key; the two's complement of 8 MB is 0xFF800000.

- **Address Translation**

  Address Translation register values must be a multiple of the address space size (which must be a power of 2). Therefore, the programmed value must be the logical AND of the target address and the BAR Setup register value (with bits [3:0] cleared).

  If the system allocates less address space than requested in the BAR Setup register, the Translation Address must be calculated using the two's complement of the BAR size, instead of the BAR Setup register value.

  For example, if the BAR size is 8 MB and the target address is 0x80F70000, the Address Translation register value is the logical AND of the BAR Setup register value 0xFF800000, and the target address 0x80F70000, so the Address Translation register value must be 0x80800000. The difference between the two addresses (target address 0x80F70000 minus the Address Translation Register value 0x80800000), is 0x77000, which is an offset into the BAR. This offset plus the BAR value maps to the target address.

  Typically the destination address is not known until runtime, and that address may be dynamic, and so generally the Address Translation register values are not programmed into the serial EEPROM. However since these registers are shadowed into each Station, they are included in the sample EEPROM image below, as placeholders in case the Translation Address is programmed by EEPROM or I^2C.

- **Requester ID**

  The Requester ID value can be determined from the packet header. Often the Requester ID of the Host is 0/0/0, however each PCIe slot could have a different Requester ID value.

  A future software release will include a function for determining Requester ID values.

  In the meantime, one method of capturing the Requester ID value is to cause an Uncorrectable Error, which will cause the switch to log the packet header into the Header Log registers (offsets 0xFD0 – 0xFDC). The error status bits in the Device Status register (offset 0x70), the Uncorrectable Error Status register (offset 0xFB8) and the Correctable Error Status register (offset 0xFC4) must be cleared, to enable subsequent update of the Header Log registers.

  The following procedure, using the PLXMon (or PLXcm) application included in the SDK, can be used to capture the Requester ID of the upstream host:

  1. Launch PLXMon, then click Command / Select A Device, and select the switch upstream port. By default, PLXMon selects the first PLX device it finds following a bus scan.

  2. Clear the error registers (0x70, 0xFB8, 0xFC4) in the switch upstream port, by reading each register and writing the value back. Type the following commands:

     `PCI 70 <enter>` (returns the value in register 0x70)

```
PCI 70 <value> <enter>      (writes the value back)
PCI FB8 <enter>             (returns the value in register 0xFB8)
PCI FB8 <value> <enter>     (writes the value back)
PCI FC4 <enter>             (returns the value in register 0xFC4)
PCI FC4 <value> <enter>     (writes the value back)
```

3. Clear the Command register (offset 0x04) Master Enable and Memory Enable bits [2:1], as follows:

```
PCI 4 <enter>               (returns the value in register 0x04)
PCI 4 <value> <enter>       (clear bits [2:1] before writing the value back)
```

4. Perform a Memory Write, using the following command:

```
el s0 0 <enter>             (writes 0 to the read-only Device/Vendor ID register.
                             Since the Command register bits [2:1] are cleared, this
                             write causes an error.)
```

5. Restore the Command register (0x04) to its original value:

```
PCI 4 <value> <enter>
```

6. Read the Header Log register to determine the Requester ID:

```
PCI FD4 <enter>             (bits [31:24] contain Bus Number,
                             bits [23:16] contain Device / Function Numbers)
```

2.3 EEPROM Content

Table 4 below shows a sample PLX Gen 2 Switch EEPROM image, which configures Port 0 as the upstream port and Port 1 as the NT port. Therefore, NT-Virtual registers exist in Port 1. Note that three NT-Virtual registers, the BAR2 Address Translation register (0xC3C), the BAR2 Setup Shadow register (0xD84), and the Requester ID table entry (0xD94), are all copied to Ports 0, 4 and 8.

Table 4. Sample EEPROM Values

Port	Register	Value	Comments
0	1DC	31340080	**Debug Control** Register 0x1DC must be written twice, first with bit 7 set, next with bit 7 cleared. Bit 29 (Link Enable, disabled by default) is set.
0	1DC	31340000	
0	C3C	00100000	NT-Virtual BAR2 Address Translation (Port 0 shadow register)
0	D84	FFF00008	NT Virtual BAR2 Setup Shadow (Port 0 shadow register)
0	D94	80000000	NT-Virtual Requester ID (Port 0 shadow register). Host B/D/F = 0/0/0
1	D4	FFF00008	NT Virtual BAR2 Setup
1	C3C	00100000	NT-Virtual BAR2 Address Translation
1	D84	FFF00008	NT Virtual BAR2 Setup Shadow
1	D94	80000000	NT-Virtual Requester ID. Host B/D/F = 0/0/0
4	C3C	00100000	NT-Virtual BAR2 Address Translation (Port 4 shadow register)
4	D84	FFF00008	NT Virtual BAR2 Setup Shadow (Port 4 shadow register)
4	D94	80000000	NT-Virtual Requester ID (Port 4 shadow register). Host B/D/F = 0/0/0
8	C3C	00100000	NT-Virtual BAR2 Address Translation (Port 8 shadow register)
8	D84	FFF00008	NT Virtual BAR2 Setup Shadow (Port 8 shadow register)
8	D94	80000000	NT-Virtual Requester ID (Port 8 shadow register). Host B/D/F = 0/0/0
NT-Link	E8	FFF00008	NT-Link BAR2 Setup
NT-Link	C3C	00100000	NT-Link BAR2 Address Translation
NT-Link	DB4	00000001	NT-Link Requester ID. Host B/D/F = 0/0/0

In applications where two NT-Link ports are back-to-back, there is no host to enumerate NT-Link registers. In such case, the serial EEPROM or I^2C can enumerate the NT-Link ports. Table 5 below lists additional sample EEPROM entries for back-to-back NT applications. BAR0 for register access is enumerated, and BAR2 for memory transfers is also enumerated. Both NT-Link BARs can be targeted from the other device's NT-Virtual BAR2 in this example, by simply changing its NT-Virtual BAR2 Address Translation register.

Table 5. Additional Sample EEPROM Values for Back-to-Back NT

Port	Register	Value	Comments
NT-Link	04	00100006	NT-Link Command
NT-Link	10	00200000	NT-Link BAR0
NT-Link	18	00100008	NT-Link BAR2

2.4 Demonstrating NT Operation with PLXMon

Non-Transparent mode operation can be demonstrated between two host systems. For simple concept demonstrations and software development, a single PC can be used, in which both NT-Virtual and NT-Link ports connect to different slots in the same system. The PLX driver sets up a host buffer in system memory. The host can generate traffic through either the NT-Virtual port (accessed through the switch upstream port) or the NT-Link port. A cable provides the physical link to another slot in the system, through which traffic routes to the host buffer in system memory. This circuitous route demonstrates the addressing flexibility needed for a multi-host Non-Transparent application.

Once the hardware is set up, the EEPROM should be programmed with the sample values in Table 4, along with the appropriate Requester ID values as explained above. This EEPROM image requests 1 MB BARs, to simplify calculation of Translation Address register values (which will be determined at runtime). Since both NT ports will target the same host buffer in system memory, the Translation Addresses for both NT ports will be identical for this demonstration.

After the system in rebooted with the programmed EEPROM, launch PLXMon, and select either of the NT ports. Since in this example Port 0 is the upstream port and Port 1 is the NT port, the NT-Virtual port will be listed immediately following the switch upstream port. The NT-Link port will have a different Bus Number assigned.

To determine and program the Translation Address registers, type the following commands on the PLXMon command line screen.

1. Type “vars”, to display the PLXMon variables. The Host Buffer is called “hbuf”. Write down the physical address that is displayed for hbuf. Then select the other NT port and type “vars” again, to display the variables associated to that NT port; the physical address for hbuf should be the same for both NT ports. If the switch upstream port is selected and “vars” is executed again, the physical address of the hbuf associated with the upstream will likely have a different address. This demonstration will use the hbuf associated to the NT ports, to minimize having to select the upstream port to check its associated hbuf. PLXMon variables lose context when a different device or port is selected.

2. Since 1 MB BARs are used, the upper 12 address bits of the 32-bit BAR contain the base address, and therefore the Translation Address will also conveniently be the upper 3 nybbles of the hbuf physical address. The lower 5 nybbles of the hbuf physical address will be the offset from the BAR2 address value. So for example if the hbuf physical address of both NT ports is 0x05748000, the Translation Address will be 0x05700000, and the offset (remainder) will be 0x48000.

   To program the Translation Address into the NT-Virtual port as well as into the shadow registers in each Station, type:

   ```
   el s0 + 10c3c 05700000 <enter>
   ```

   To program the Translation Address into the NT-Link port, type:

   ```
   el s0 + 11c3c 05700000 <enter>
   ```

   The switch registers are now programmed for NT mode.

3. To test NT writes in one direction, make sure that one of the one of the NT ports is selected, and type:

   ```
   el s2+48000 12345678 <enter>
   ```

   Then check that the value (12345678) is now written into the hbuf, by typing:

   ```
   dl hbuf l 4 <enter>
   ```
   (the “l 4” tells PLXMon to return a length of 4 bytes)

4. Now test NT reads, using the following command:

   ```
   dl s2+48000 l 4 <enter>
   ```

5. To test NT writes in the other direction, use essentially the same commands as used in step 3 above. First, select the other NT port. Then write a different value to the adjacent location in the hbuf, by typing:

   ```
   el s2+48004 feedface <enter>
   ```

Then check that the value (feedface) is now written into the hbuf, by typing:

`dl hbuf+4 l 4 <enter>` (the “l 4” tells PLXMon to return a length of 4 bytes)

6. Now test NT reads in this direction, using the following command:

`dl s2+48000 l 8 <enter>` (the “l 8” tells PLXMon to return a length of 8 bytes)

PLXMon should return all 8 bytes (12345678 and feedface). This demonstrates that the hbuf can be partitioned into multiple sections for use by both NT ports.