Tsi108™/Tsi109™ Software Initialization Application Note

80B5000_AN001_07

October 2009

6024 Silver Creek Valley Road, San Jose, California 95138
Telephone: (800) 345-7015 • (408) 284-8200 • FAX: (408) 284-2775
Printed in U.S.A.

GENERAL DISCLAIMER

Integrated Device Technology, Inc. reserves the right to make changes to its products or specifications at any time, without notice, in order to improve design or performance and to supply the best possible product. IDT does not assume any responsibility for use of any circuitry described other than the circuitry embodied in an IDT product. The Company makes no representations that circuitry described herein is free from patent infringement or other rights of third parties which may result from its use. No license is granted by implication or otherwise under any patent, patent rights or other rights, of Integrated Device Technology, Inc.

CODE DISCLAIMER

Code examples provided by IDT are for illustrative purposes only and should not be relied upon for developing applications. Any use of the code examples below is completely at your own risk. IDT MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND CONCERNING THE NONINFRINGEMENT, QUALITY, SAFETY OR SUITABILITY OF THE CODE, EITHER EXPRESS OR IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. FURTHER, IDT MAKES NO REPRESENTATIONS OR WARRANTIES AS TO THE TRUTH, ACCURACY OR COMPLETENESS OF ANY STATEMENTS, INFORMATION OR MATERIALS CONCERNING CODE EXAMPLES CONTAINED IN ANY IDT PUBLICATION OR PUBLIC DISCLOSURE OR THAT IS CONTAINED ON ANY IDT INTERNET SITE. IN NO EVENT WILL IDT BE LIABLE FOR ANY DIRECT, CONSEQUENTIAL, INCIDENTAL, INDIRECT, PUNITIVE OR SPECIAL DAMAGES, HOWEVER THEY MAY ARISE, AND EVEN IF IDT HAS BEEN PREVIOUSLY ADVISED ABOUT THE POSSIBILITY OF SUCH DAMAGES. The code examples also may be subject to United States export control laws and may be subject to the export or import laws of other countries and it is your responsibility to comply with any applicable laws or regulations.

LIFE SUPPORT POLICY

Integrated Device Technology's products are not authorized for use as critical components in life support devices or systems unless a specific written agreement pertaining to such intended use is executed between the manufacturer and an officer of IDT.

1. Life support devices or systems are devices or systems which (a) are intended for surgical implant into the body or (b) support or sustain life and whose failure to perform, when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury to the user.
2. A critical component is any components of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

Contents

About this Document

This section discusses the following topics:

Scope

This document explains how to write initialization software for use with the Tsi108 and Tsi109, such as how to access each device's registers from the PCI/X and Processor Interfaces. It also describes the Tsi108/Tsi109's main data paths, and how to program their registers for efficient port-to-port access.

Revision History

80B5000_AN001_07, Formal, October 2009

This document was rebranded as IDT. It does not include any technical changes.

80B5000_AN001_06, Formal, March 2007

Changed item 3 in Figures 9 and 10 to indicate "The Switch Fabric address forwarded to the HLP Interface." This item previously incorrectly indicated the address was forwarded to the PCI/X Interface.

80B5000_AN001_05, Formal, November 2005

This version was revised to support the following product name changes: Tsi108A to Tsi109, Tsi108x to Tsi108/Tsi108, and Tsi108 Family to Tsi108/Tsi109. No technical changes were implemented.

80B5000_AN001_04, Formal, June 2005

This version was revised to indicate support for the Tsi108 and Tsi109 devices. No technical changes were implemented.

80B5000_AN001_03, Formal, February 2005

This version was revised to incorporate several minor changes.

80B5000_AN001_02, Formal, February 2005

This version was revised to incorporate several minor changes.

80B5000_AN001_01, Formal, January 2005

This is the first version of this document. It is intended for internal use only.

1. Internal Register Access

This section describes how to access Tsi108/Tsi109's registers in order to place the device in an operational state. It is assumed that the correct hardware configuration has been chosen for the given design. For information on the complete Tsi108/Tsi109 programming model, see the Register section of the *Tsi108/Tsi109 User Manual*.

Using software, the Tsi108/Tsi109 register space can be configured from the PCI/X, Processor, or I^2C Interface depending on the system architecture requirements. The I^2C Interface, however, can only initialize a sub-set of Tsi108/Tsi109's registers. For information on initializing Tsi108/Tsi109's registers through the I^2C Interface, see the I^2C Interface section of the *Tsi108/Tsi109 User Manual*.

In order to initialize the Tsi108 or Tsi109, the user must have a detailed understanding of the *Tsi108/Tsi109 User Manual*, relevant bus specifications, and overall system configuration.

1.1 Device Register Map

The Tsi108/Tsi109 contains 64 Kbytes of internal register space for initialization and configuration. Each of the Tsi108/Tsi109's interfaces has a separate register block, as indicated in Table 1.

Table 1: Device Register Map

Address[15:0][a]	Register Block
0x0000 ─ 0x0FFF	HLP Interface
0x1000 – 0x1FFF	PCI/X Interface
0x2000 – 0x2FFF	Clock Generator
0x3000 – 0x3FFF	Processor Interface
0x4000 – 0x4FFF	Memory Controller
0x5000 – 0x5FFF	DMA Controller
0x6000 – 0x6FFF	Ethernet Controller
0x7000 – 0x73FF	I^2C Controller
0x7400 – 0x77FF	Interrupt Controller
0x7800 – 0x79FF	UART 0 Interface
0x7A00 – 0x7BFF	GPIO Interface
0x7C00 – 0x7FFF	UART 1 Interface

a. The Address bus in the table defines the 16 least significant bits of the Address bus. For example, when accessing Tsi108/Tsi109's registers from the PCI/X bus this would be AD[15:0]. When accessing the Tsi108/Tsi109 from the Processor bus it would be processor address bits A[20:35] for a 36-bit processor, and processor address bits A[16:31] for a 32-bit processor.

1.2. Required Initialization from PCI/X and Processor Interface

This section describes the necessary register accesses that the Tsi108/Tsi109 must receive to enable its registers to be visible from the PCI/X Interface and the Processor Interface.

1.2.1 How to Initialize from the Process Bus Interface

The PB_REG_BAR register (offset 0x410) must be programmed in order to get access to the 64 Kbytes of internal register space (see Table 2). The PB_REG_BAR window differs from other address windows because it is enabled upon power-up. The reset value for PB_REG_BAR is 0xC000_0001.

PB_REG_BAR[BA] must be initialized in order for the Tsi108/Tsi109 to know what address range to decode for access to the Tsi108/Tsi109 registers. If the Processor is operating in 36-bit mode the base address upper field, BA_UPPER, is required as well. Once these fields are programmed, Tsi108/Tsi109's register space is accessible from the Processor bus.

Table 2: Processor Bus Register Access

Register	Field	Description
PB_REG_BAR	BA[4:19]	Base Address: These bits are compared with processor address bits A[4:19] when connected to a 36-bit processor bus or A[0:15] when connected to a 32-bit processor bus to determine if the transaction falls within the PB_REG_BAR.
PB_REG_BAR	BA_UPPER[0:3]	Base Address Upper: When the processor is operating in 36-bit mode these bits are compared with processor address bus bits A[0:3] to determine if the transaction falls within the PB_REG_BAR. These bits are ignored in 32-bit address mode.
PB_REG_BAR	END_MODE[1:0]	Endian Mode: These bits allow for swapping between Big-endian and Little-endian mode. These bits should be programmed based on whether the processor is operating in Big-endian or Little-endian mode.
PB_REG_BAR	EN	Enable: This bit is written to enable/disable the PB_REG_BAR.

1.2.2 How to Initialize from the PCI/X Bus

The following Tsi108/Tsi109 registers must be accessed through PCI/X configuration cycles in order to get access to the 64 Kbytes of internal register space:

1. PE_CSR[MS] (offset 0x004) must be enabled to allow the Tsi108/Tsi109 to accept memory cycles as a PCI/X target (see Table 3).
2. MISC_CSR[BAR0_EN] (offset 0x040) must be enabled to allow write access to PCI Base Address Register 0 and PCI Base Address Upper Register 0.
3. The base address field (BA[31:16]) within P2O_BAR0 (offset 0x010) and the upper base address field (BA[63:32]) within P2O_BAR0_UPPER (offset 0x014) must be initialized for the Tsi108/Tsi109 to know what address range to decode for access to its registers. Once these fields are programmed, Tsi108/Tsi109's 64 Kbytes of register space is accessible from the PCI/X bus through memory cycles.

Table 3: PCI/X Bus Register Access

Register	Field	Description
PE_CSR	MS	Memory Space Enable: This bit is written to allow the Tsi108/Tsi109 to accept memory cycles as a PCI/X target.
MISC_CSR	BAR0_EN	Base Address Register 0 Enable: This register must be enabled so that the P2O_BAR0 and P2O_BAR0_UPPER registers are visible in PCI/X configuration space. If this bit is cleared reads to the P2O_BAR0 and P2O_BAR0_UPPER registers will return zero and writes have no effect.
P20_BAR0	BA[31:16]	Base Address: These bits are compared with PCI/X address bits A[31:16] to determine if the transaction falls within the 64-bit register bar.
P2O_BAR0_UPPER	BA[63:32]	Base Address Upper: These bits are compared with PCI/X address bits A[63:32] to determine if the transaction falls within the 64-bit register bar.

2. Port-to-Port Access

In order for port-to-port access to occur through the Tsi108/Tsi109, certain registers must be initialized with software. This section describes the necessary accesses to Tsi108/Tsi109's registers to allow port-to-port communication. It explains the various paths through the device and the register settings required to access these paths.

The main Tsi108/Tsi109 data paths are indicated in the following list. The remainder of this section describes the required Tsi108/Tsi109 register settings to enable access through these data paths.

1. Processor Interface to Memory Controller (Direct path)
2. Processor Interface to Memory Controller (Non-direct path)
3. Processor Interface to PCI/X Interface
4. Processor Interface to HLP Interface
5. PCI/X Interface to Memory Controller (Non-Snoop path)
6. PCI/X Interface to Memory Controller (Snoop path)
7. PCI/X Interface to HLP Interface
8. Ethernet Controller to Memory Controller (Transmit and Receive paths)

For all accesses through the Memory Controller to SDRAM it is assumed that the specific memory configuration has already been configured. The configuration of the Memory Controller registers depends on the type of system memory used and where it is located.

When the Tsi108/Tsi109 is connected to a 36-bit processor, Tsi108/Tsi109 bits PB_A[0:35] connect to Processor Bus bits A[0:35]. When the Tsi108/Tsi109 is connected to a 32-bit processor, Tsi108/Tsi109 bits PB_A[4:35] are connected to Processor Bus bits A[0:31] and PB_A[0:3] should be pulled low on the board.

2.1. Processor to Memory Controller (Direct Path)

In order to communicate through the Tsi108/Tsi109 from the Processor bus directly to memory (Direct path) through the Memory Controller, a Processor bus to SDRAM base address register must be initialized with software. This base address register defines an address range (in Processor bus address space) where the Tsi108/Tsi109 responds to a Processor's request. If the Processor's request falls within one of the Tsi108/Tsi109 Processor bus to SDRAM base address ranges, the transaction is forwarded through the Memory Controller to memory (SDRAM). The Tsi108/Tsi109 supports two Processor Bus to SDRAM base address registers: PB_SDRAM_BAR1 (offset 0x41C) and PB_SDRAM_BAR2 (offset 0x420) (see Tables 4, 5, and 6).

2.1.1 How to Initialize the Processor Bus to SDRAM BARs

The Processor bus to SDRAM BARs can be initialized using PCI/X memory write cycles from the PCI/X bus through the P2O_BAR0 and P2O_BAR0_UPPER registers, or from the Processor bus through the PB_REG_BAR. Table 4 describes the internal register settings associated with mapping Processor bus transactions directly to memory (SDRAM) through the Memory Controller.

Table 4: PB SDRAM Base Address Register Initialization

Register[a]	Field	Description
PB_SDRAM_BAR[1:2]	BA[4:7]	Base Address: These bits are compared with processor address bits A[4:7] when connected to a 36-bit processor bus or A[0:3] when connected to a 32-bit processor bus to determine if the transaction falls within the PB_SDRAM_BAR.
PB_SDRAM_BAR[1:2]	TA[4:7]	Translation Address: When address translation is enabled (ATE set to 1), these bits replace processor address bits A[4:7] when connected to a 36-bit processor bus or A[0:3] when connected to a 32-bit processor bus to address memory (SDRAM).
PB_SDRAM_BAR[1:2]	ATE	Address Translation Enable: When enabled the TA and TA_UPPER fields replace the appropriate processor bus address when forwarding to memory (SDRAM).
PB_SDRAM_BAR[1:2]	BA_UPPER[0:3]	Base Address Upper: When the processor is operating in 36-bit mode these bits are compared with processor address bus bits A[0:3] to determine if the transaction falls within the PB_SDRAM_BAR. These bits are ignored in 32-bit address mode.
PB_SDRAM_BAR[1:2]	TA_UPPER[0:3]	Translation Address Upper: When address translation is enabled (ATE set to 1), these bits replace processor address bits A[0:3] when the processor bus is operating in 36-bit mode to address memory (SDRAM).
PB_SDRAM_BAR[1:2]	SIZE[3:0]	Size: These bits define the PB_SDRAM_BAR window size, see Tables 5 and 6.
PB_SDRAM_BAR[1:2]	WR_PRTC	Write Protect: This bit indicates whether the address window is write-protected. If enabled the address window is write-protected and any writes to it will cause an error.
PB_SDRAM_BAR[1:2]	EN	Enable: This bit enables the PB_SDRAM_BAR address window.

a. The Tsi108/Tsi109 supports two Processor to SDRAM base address windows.

Table 5: PB_SDRAM_BAR Address Window Size for 36-bit Processors

SIZE[3:0]	Window Size (Bytes)	Active BAR (32-bit mode, 36-bit mode)	Offset (32-bit mode, 36-bit mode)
0000	256M	A[4:7], A[0:7]	A[8:35], A[8:35]
0001	512M	A[4:6], A[0:6]	A[7:35], A[7:35]
0010	1G	A[4:5], A[0:5]	A[6:35], A[6:35]
0011	2G	A[4], A[0:4]	A[5:35], A[5:35]
0100[a]	4G	NA, A[0:3]	A[4:35}, A[4:35]
0101	8G	NA, A[0:2]	NA, A[3:35]
0110	16G	NA, A[0:1]	NA, A[2:35]
0111	32G	NA, A[0]	NA, A[1:35]
1000[b]	64G	NA, NA	NA, A[0:35]

a. If extended addressing (36-bit mode) is disabled in the processor and the SIZE is 4'b0100, then any address on the processor bus is claimed by the Tsi108/Tsi109 and forwarded to memory.

b. If extended addressing is enabled in the processor and the SIZE is 4'b1000, then any address on the processor bus is claimed by the Tsi108/Tsi109 and forwarded to memory.

Table 6: PB_SDRAM_BAR Address Window Size for 32-bit Processors

SIZE[3:0]	Window Size (Bytes)	Active BAR	Offset
0000	256M	A[0:3]	A[4:31]
0001	512M	A[0:2]	A[3:31]
0010	1G	A[0:1]	A[2:31]
0011	2G	A[0]	A[1:31]
0100[a]	4G	NA	A[0:31]

a. If the SIZE is 4'b0100 then any address on the processor bus is claimed by the Tsi108/Tsi109 and forwarded to memory.

2.1.2 Processor to Memory Controller (Direct Path) Examples

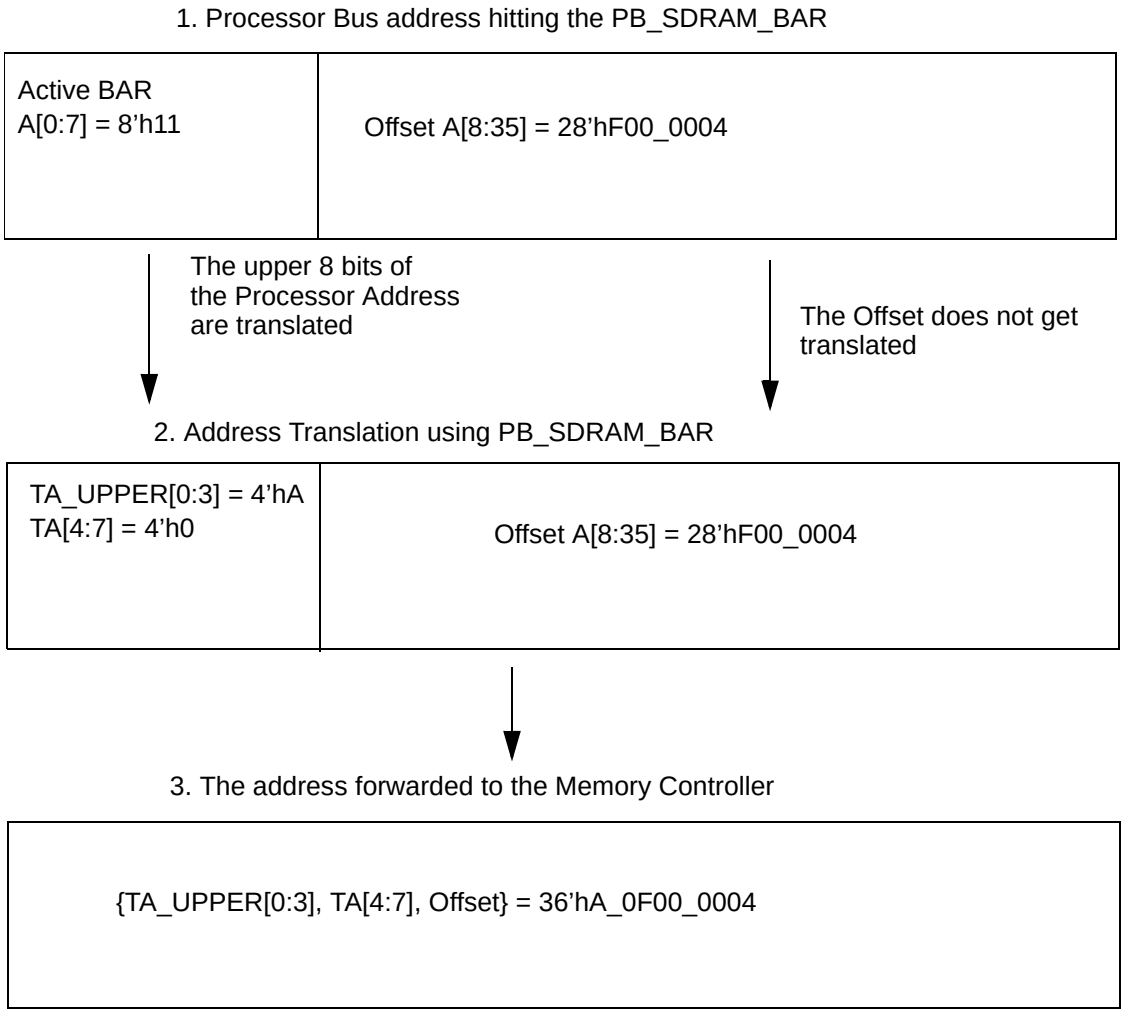
Example 1: 36-bit Processor with extended addressing enabled, access to memory through the Direct path from the processor bus.

In this example a 256-Mbyte window is configured using the PB_SDRAM_BAR2 register. All processor bus accesses that fall within that window are forwarded directly to the Memory Controller and onto memory. The values programmed in the PB_SDRAM_BAR2 for this example will forward all processor bus addresses that fall within 36'h1_1000_0000 – 36'h1_1FFF_FFFF directly to the Memory Controller as 36'hA_0000_0000 – 36'hA_0FFF_FFFF. Using the PB_SDRAM_BAR2 register the following fields are programmed as described below. Figure 1 shows the address translation process resulting from a Processor bus address (36'h1_1F00_0004) hitting the PB_SDRAM_BAR2.

Register settings:

- **PB_SDRAM_BAR2:** BA = 4'h1; TA = 4'h0; ATE = 1; BA_UPPER = 4'h1; TA_UPPER = 4'hA; SIZE = 4'h0 (256M); WR_PRTC = 0; EN = 1

Figure 1: Direct Path Address Translation Process, 36-bit Processor



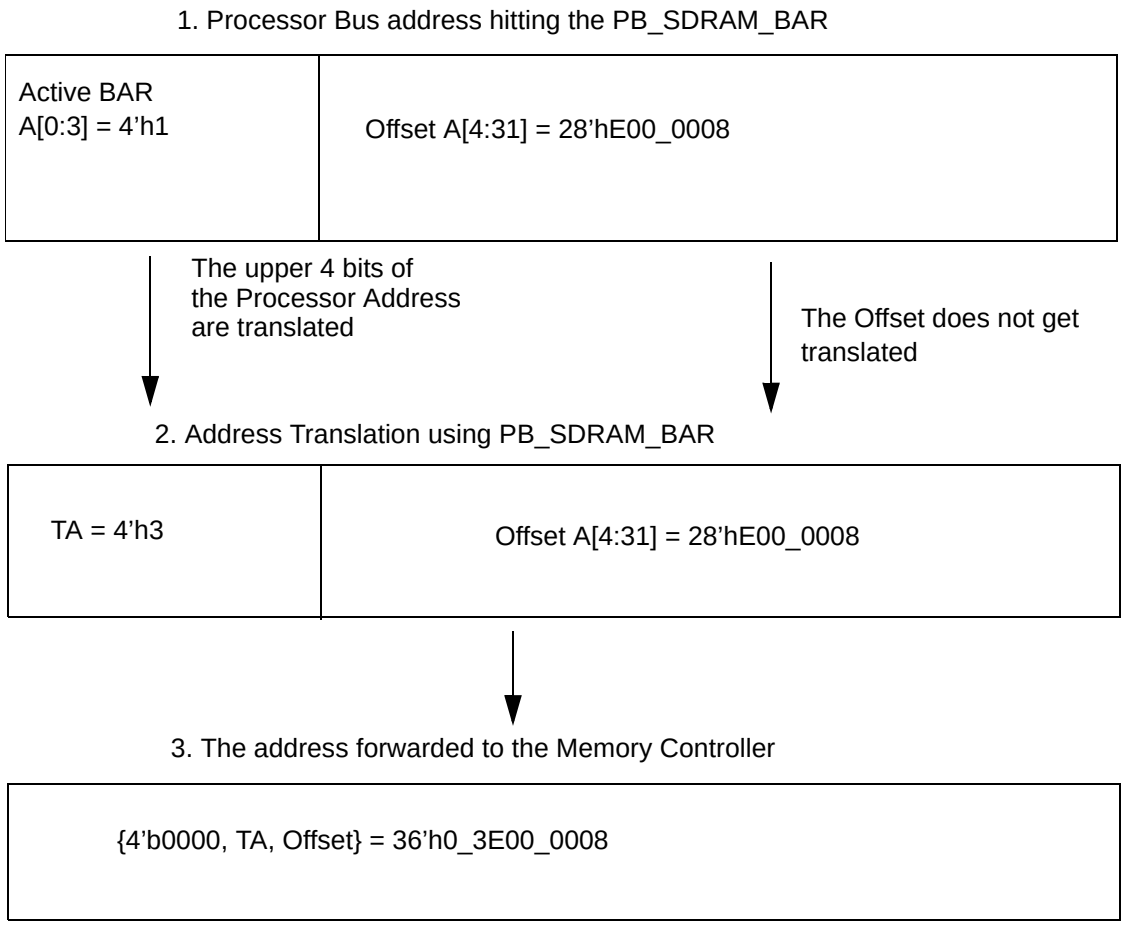
Example 2: 32-bit Processor, access to memory through the Direct path from the processor bus.

In this example a 256-Mbyte window is configured using the Tsi108/Tsi109 PB_SDRAM_BAR2 register. All processor bus accesses that fall within that window are forwarded directly to the Memory Controller and onto memory. The values programmed in the PB_SDRAM_BAR2 for this example will forward all processor bus addresses that fall within 32'h1000_0000 – 32'h1FFF_FFFF directly to the Memory Controller as 32'h3000_0000 – 32'h3FFF_FFFF. Using the PB_SDRAM_BAR2 register the following fields are programmed as described below. Figure 2 shows the address translation process resulting from a Processor bus address (32'h1E00_0008) hitting the PB_SDRAM_BAR2.

Register settings:

- **PB_SDRAM_BAR2:** BA = 4'h1; TA = 4'h3; ATE = 1; SIZE = 4'h0 (256M); WR_PRTC = 0; EN = 1

Figure 2: Direct Path Address Translation Process, 32-bit Processor



Note: The upper 4 address bits are pulled down on the board and passed to the Memory Controller.

2.2. Processor to Memory Controller (Non-direct Path)

In order to communicate from the Processor bus through the Switch Fabric to memory (Non-direct path) through the Memory Controller, the following registers must be initialized: one of the PB to Switch Fabric Base Address Registers (PB_OCN_BARx), as well as the associated PB Switch Fabric BAR Lower (PB_BARx_LOWER_LUT_ADDRy) and Upper (PB_BARx_UPPER_LUT_ADDRy) Lookup Table Address Registers. A PB to Switch Fabric base address register defines an address range (in Processor bus address space) where the Tsi108/Tsi109 responds to a Processor's request.

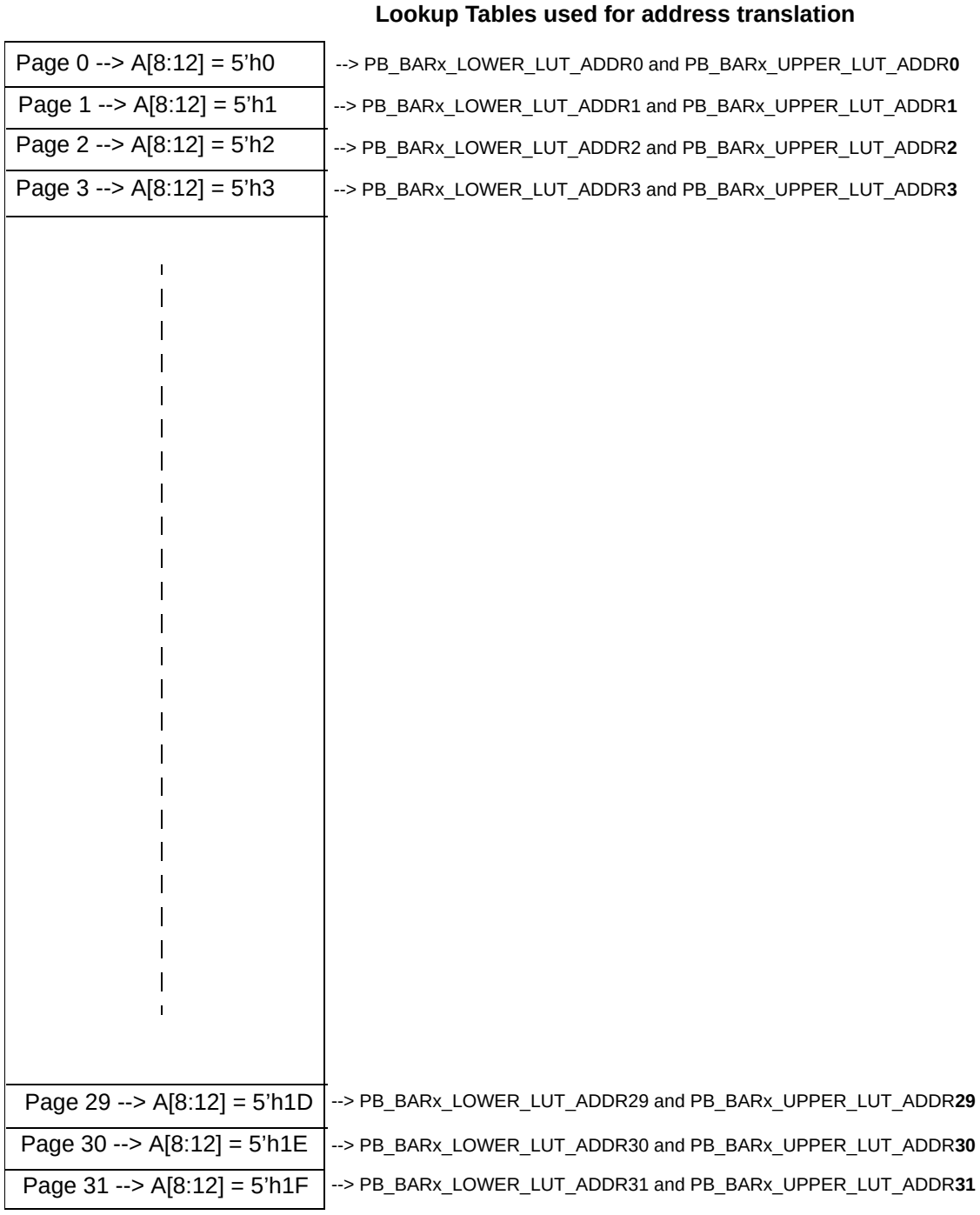
If the processor's request falls within one of the Tsi108/Tsi109's PB to Switch Fabric base address ranges, the transaction is forwarded to the Tsi108/Tsi109 Switch Fabric. The Tsi108/Tsi109 supports two unique PB to Switch Fabric BARs: PB_OCN_BAR1 (offset 0x414) and PB_OCN_BAR2 (offset 0x418) (see Tables 7 to 10).

In order to direct the request from the Switch Fabric to another Tsi108/Tsi109 interface the address must be translated. The PB Switch Fabric BAR Upper and Lower Lookup Table Address Registers must be programmed to provide the proper address translation to target the Memory Controller to memory. Each of the PB to Switch Fabric BARs (PB_OCN_BAR1 and PB_OCN_BAR2) can have up to 32 pages of memory associated with them. The PB to Switch Fabric BAR Upper and Lower Lookup Tables make up those 32 pages, each individual page of memory requires an initialization of an Upper and Lower BAR Lookup (see Figure 3 along with Tables 11 and 12).

The PB to Switch Fabric BAR Upper and Lower Lookup Table Address Registers (PB_BAR[1:2]_LOWER_LUT_ADDR{0:31} and PB_BAR[1:2]_UPPER_LUT_ADDR{0:31}) are located in the Processor bus register block at offsets 0x800 – 0x9FC.

Figure 3: PB to Switch Fabric BAR Lookup Table Addressing

This example describes the Page addressing and address translation process for a 256M window, processor operating in 36-bit mode.



2.2.1 How to Initialize the Processor Bus to SDRAM Non-direct Path

The PB to Switch Fabric Base Address, PB Switch Fabric BAR Upper and Lower Lookup Table Address Registers can be initialized using PCI/X memory write cycles from the PCI/X bus through the P2O_BAR0 and P2O_BAR0_UPPER Registers, or from the Processor bus through the PB_REG_BAR. Tables 7 to 13 describe the internal register settings associated with mapping Processor Bus transactions through the Switch Fabric to the Memory Controller and onto memory.

Table 7: PB Switch Fabric Base Address Register 1 Initialization

Register	Field	Description
PB_OCN_BAR1[a]	BA[4:7]	Base Address: These bits are compared with processor address bits A[4:7] when connected to a 36-bit processor bus or A[0:3] when connected to a 32-bit processor bus to determine if the transaction falls within PB_OCN_BAR1.
PB_OCN_BAR1	BA_UPPER[0:3]	Base Address Upper: When the processor is operating in 36-bit mode these bits are compared with processor address bus bits A[0:3] to determine if the transaction falls within PB_OCN_BAR1. These bits are ignored in 32-bit address mode.
PB_OCN_BAR1	SIZE[3:0]	Size: These bits define the PB_OCN_BAR1 window size, see Tables 9 and 10.
PB_OCN_BAR1	BOOT	Boot from HLP Interface: Used for booting the processor from the Host Local Port (HLP) Interface. When enabled all processor bus accesses to PB_OCN_BAR1 are forwarded to the HLP Interface as well as: • All memory regions are write-protected • Tsi108/Tsi109 lookup tables are not used • Endian mode is set as “packet swap” • Address bits, indicated by signals PB_A[16:35], are used as the offset; the remainder of the address is zero. **Note**: Software must clear this bit after the processor finishes booting from the HLP and the PB_LOWER_LUT_ADDRx/PB_UPPER_LUT_ADDRx registers are programmed. **Caution:** When BOOT is 0, PB_OCN_BAR1 must not be accessed until the corresponding lookup table registers, PB_UPPER_LUT_ADDRx and PB_LOWER_LUT_ADDRx, are programmed. This sequence must be followed or Tsi108/Tsi109 behavior is undefined.
PB_OCN_BAR1	EN	Enable: This bit enables the PB_OCN_BAR1.

a. The PB_OCN_BAR1 register differs from PB_OCN_BAR2 in that it comes up enabled and at a pre-defined base address to the HLP Interface for accessing boot-up flash. The PB_OCN_BAR1’s reset value is 0xF000_0003.

Table 8: PB Switch Fabric Base Address Register 2 Initialization

Register	Field	Description
PB_OCN_BAR2[a]	BA[4:7]	Base Address: These bits are compared with processor address bits A[4:7] when connected to a 36-bit processor bus or A[0:3] when connected to a 32-bit processor to determine if the transaction falls within PB_OCN_BAR2.
PB_OCN_BAR2	BA_UPPER[0:3]	Base Address Upper: When the processor is operating in 36-bit mode these bits are compared with processor address bus bits A[0:3] to determine if the transaction falls within PB_OCN_BAR2. These bits are ignored in 32-bit address mode.
PB_OCN_BAR2	SIZE[3:0]	Size: These bits define the PB_OCN_BAR2 window size, see Tables 9 and 10.
PB_OCN_BAR2	EN	Enable: This bit enables the PB_OCN_BAR2.

a. The PB_OCN_BAR2 register's reset value is 0x0000_0000.

Table 9: PB_OCN_BAR[1:2] Window Size for 36-bit Processors

Size[3:0]	Window Size (Bytes)	Page Size (Bytes)	Active BAR (32-bit, 36-bit mode)	Lookup Index[a] (32-bit, 36-bit mode)	Offset (32-bit, 36-bit mode)
0000	256M	8M	A[4:7], A[0:7]	A[8:12], A[8:12]	A[13:35],A[13:35]
0001	512M	16M	A[4:6], A[0:6]	A[7:11], A[7:11]	A[12:35],A[12:35]
0010	1G	32M	A[4:5], A[0:5]	A[6:10], A[6:10]	A[11:35],A[11:35]
0011	2G	64M	A[4], A[0:4]	A[5:9], A[5:9]	A[10:35],A[10:35]
0100[b]	4G	128M	NA, A[0:3]	A[4:8], A[4:8]	A[9:35],A[9:35]
0101	8G	256M	NA, A[0:2]	NA, A[3:7]	NA, A[8:35]
0110	16G	512M	NA, A[0:1]	NA, A[2:6]	NA, A[7:35]
0111	32G	1G	NA, A[0]	NA, A[1:5]	NA, A[6:35]
1000[c]	64G	2G	NA, NA	NA, A[0:4]	NA, A[5:35]

a. The address bits that comprise the Lookup Index define which page within the window the access is destined for. Each PB_OCN_BAR window is comprised of 32 distinct memory pages.

b. If extended addressing (36-bit mode) is disabled in the processor and the SIZE is 4'b0100, then any address on the processor bus is claimed by the Tsi108/Tsi109 and forwarded to the Switch Fabric.

c. If extended addressing is enabled in the processor and the SIZE is 4'b1000, then any address on the processor bus is claimed by the Tsi108/Tsi109 and forwarded to the Switch Fabric.

Table 10: PB_OCN_BAR[1:2] Window Size for 32-bit Processors

Size[3:0]	Window Size (Bytes)	Page Size (Bytes)	Active BAR	Lookup Index	Offset
0000	256M	8M	A[0:3]	A[4:8]	A[9:31]
0001	512M	16M	A[0:2]	A[3:7]	A[8:31]
0010	1G	32M	A[0:1]	A[2:6]	A[7:31]
0011	2G	64M	A[0]	A[1:5]	A[6:31]
0100[a]	4G	128M	NA	A[0:4]	A[5:31]

a. If the SIZE is 4'b0100 then any address on the processor bus is claimed by the Tsi108/Tsi109 and forwarded to the Switch Fabric.

Table 11: PB Switch Fabric BAR Lower Lookup Table Address Register Initialization

Register	Field	Description
PB_BAR[1:2]_LOWER_LUT_ADDR{0..31}	TA[31:23]	Translation Address [31:23]: When the ATE bit is enabled these bits replace the processor bus address.
PB_BAR[1:2]_LOWER_LUT_ADDR{0..31}	END_MODE[1:0]	Endian Mode of destination Port: 00 = No Swap (data invariant) 01 = Endian Swap (swapping based on address and size of the transaction) used when processor is in Little-endian mode. 10 = Packet swap (byte swap) 11 = Word swap (position of upper and lower 4 bytes reversed) **Note:** When swapping is enabled, some combinations of transaction alignments and sizes are not supported because they lead to non-contiguous bytes being accessed.
PB_BAR[1:2]_LOWER_LUT_ADDR{0..31}	WR_PRTC	Write Protect: When enabled this bit indicates the window is write-protected. **Note:** Writing to a write-protected window will cause an error.
PB_BAR[1:2]_LOWER_LUT_ADDR{0..31}	ATE	Address Translation Enable: This bit when enabled enables the TA field TA[63:10] to overwrite the processor bus address.
PB_BAR[1:2]_LOWER_LUT_ADDR{0..31}	DST_PORT[3:0]	Switch Fabric Destination Port Number: Each Tsi108/Tsi109 external interface is connected to a Switch Fabric port (see Table 13).

Table 12: PB Switch Fabric BAR Upper Lookup Table Address Register Initialization

Register	Field	Description
PB_BAR[1:2]_UPPER_LUT_ADDR{0..31}	TA[63:32]	Translation Address [63:32]: When the ATE bit is enabled these bits replace the processor bus address.

Table 13: Switch Fabric Port Encoding

DST_PORT[3:0]	Device Block
0000	HLP Interface
0001	PCI/X Interface
0010	Processor Interface (Master)
0011	Processor Interface (Slave)
0100	Memory Controller
0101	DMA Controller
0110	Ethernet Controller
0111	Reserved

2.2.2 Processor to Memory Controller (Non-direct) Examples

Example 1: 36-bit Processor with extended addressing enabled, access to memory through the Non-direct path from the processor bus.

In this example the PB_OCN_BAR2 window size has been programmed to 256M, all processor bus addresses that fall within 36'h1_1000_0000 – 36'h1_1FFF_FFFF will be accepted and forwarded through the Switch Fabric as indicated by the values programmed in the PB_OCN_BAR2 register below. Once the transaction has been accepted through the PB_OCN_BAR2 register it is up to the associated lookup tables to determine both the destination port and proper address translation. Since the window size programmed in the PB_OCN_BAR2 register is 256 Mbytes, that means there are 32, 8-Mbyte pages associated with that address window.

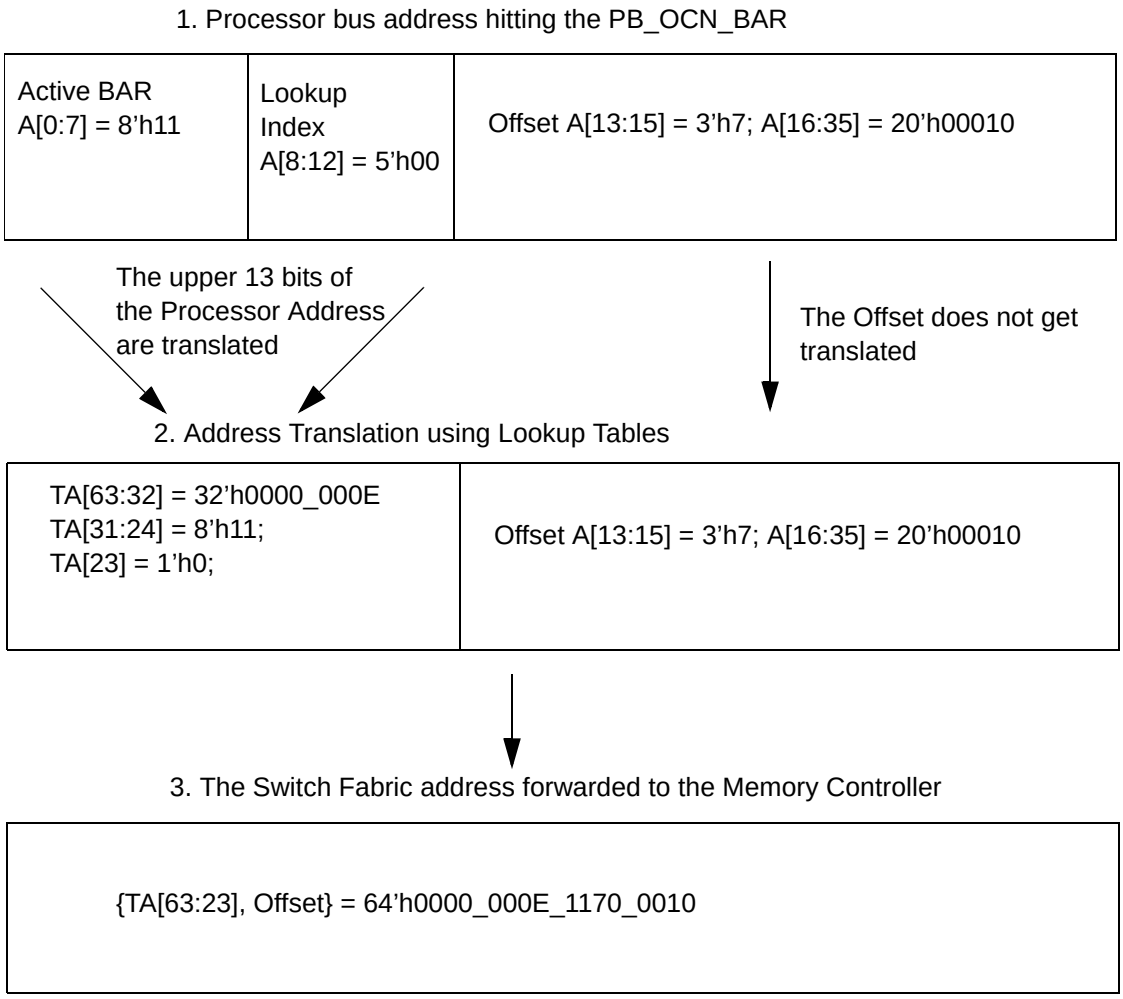
The PB_BARx_UPPER_LUT_ADDR{0..31} and PB_BARx_LOWER_LUT_ADDR{0..31} registers must be programmed to determine both the destination port and address translation associated with the 32 pages. In this example, only one 8-Mbyte page (page 0) of the 256-Mbyte window is configured to access memory, by programming the PB_BAR2_UPPER_LUT_ADDR0 and PB_BAR2_LOWER_LUT_ADDR0 registers, page 0 addresses memory as indicated by the DST_PORT field (see Table 13). Address translation has been enabled and all Switch Fabric address accesses that address page 0 will be forwarded to the Memory Controller as 64'h0000_000E_1100_0000 – 64'h0000_000E_117F_FFFF.

Address hits to page 0 are determined as follows using the information provided in Table 9. Processor bits A[0:7] define the active BAR which in this example are defined as 8'h11. Processor bits A[8:12] define the lookup index (that is, what page the access is mapped to), in order to hit page 0, these bits need to be 5'h00. The remaining Processor bits A[13:35] define the address offset which in this case address the entire 8M page of memory. Figure 4 shows the address translation process resulting from a Processor bus address (36'h1_1070_0010) hitting the PB_OCN_BAR2.

Register settings:

- **PB_OCN_BAR2:** BA = 4’h1; BA_UPPER = 4’h1; SIZE = 4’h0 (256M Window); EN = 1
- **PB_BAR2_UPPER_LUT_ADDR0:** TA[63:32] = 32’h0000_000E
- **PB_BAR2_LOWER_LUT_ADDR0:** TA[31:24] = 8’h11, TA[23] = 1’h0; END_MODE = 2’b00; WR_PRTC = 0; ATE = 1; DST_PORT = 4’b0100 (Memory Controller)

Figure 4: Non-direct Path Address Translation Process, 36-bit Processor



Example 2: 32-bit Processor, access to memory through the Non-direct path from the processor bus.

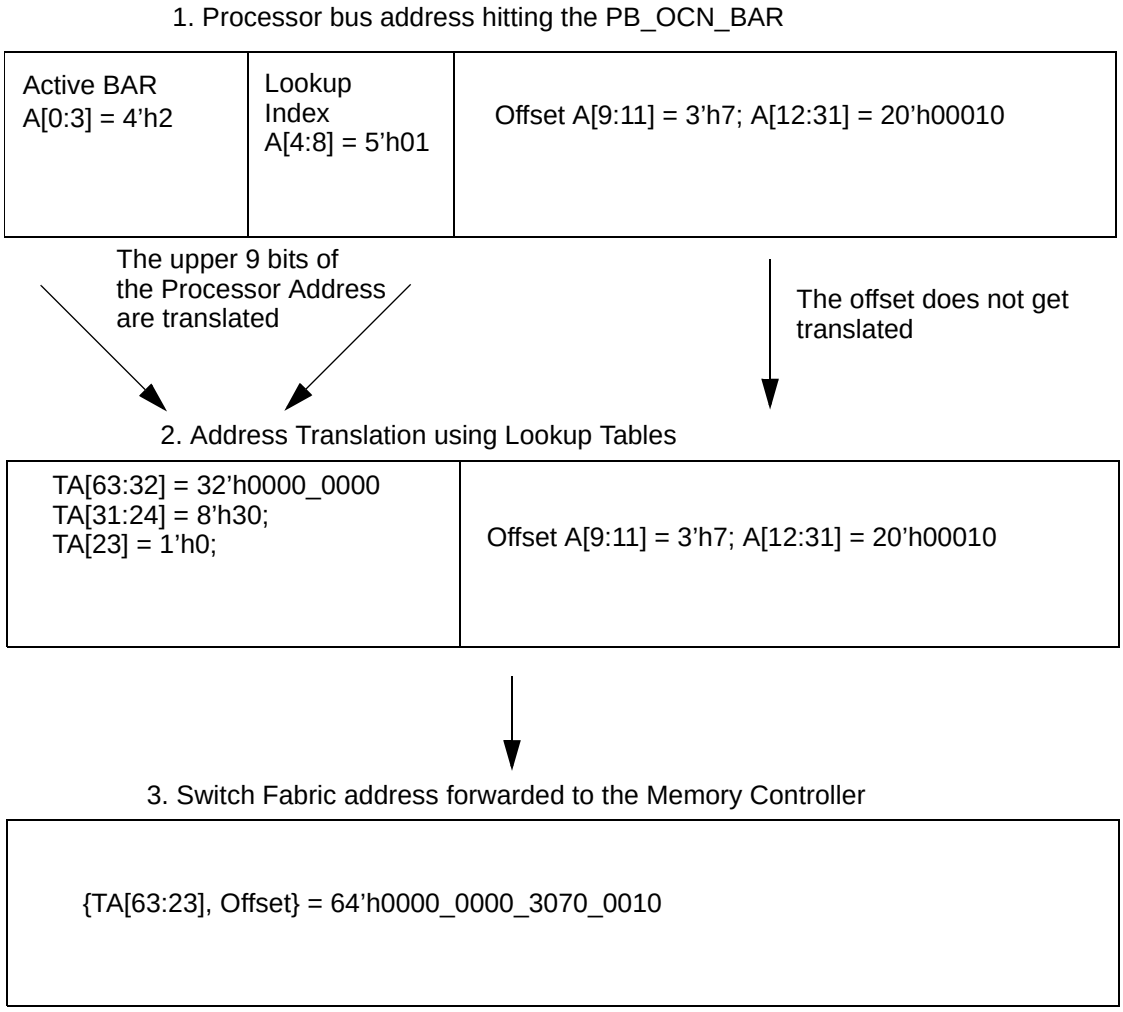
In this example the PB_OCN_BAR2 window size has been programmed to 256 Mbytes, all processor bus addresses that fall within 36’h2000_0000 – 36’h2FFF_FFFF will be accepted and forwarded through the Switch Fabric as indicated by the values programmed in the PB_OCN_BAR2 register below. Once the transaction has been accepted through the PB_OCN_BAR2 register it is up to the associated lookup tables to determine both the destination port and proper address translation. Since the window size programmed in the PB_OCN_BAR2 register is 256 Mbytes, that means there are 32, 8-Mbyte pages associated with that address window. The PB_BARx_UPPER_LUT_ADDR{0..31} and PB_BARx_LOWER_LUT_ADDR{0..31} registers must be programmed to determine both the destination port and address translation associated with those 32 pages.

In this example only one 8-Mbyte page (page 1) of the 256-Mbyte window is configured to access memory (SDRAM), by programming the PB_BAR2_UPPER_LUT_ADDR1 and PB_BAR2_LOWER_LUT_ADDR1 registers, page 1 addresses memory as indicated by the DST_PORT field (see Table 13). Address translation has been enabled and all Switch Fabric address accesses that address page 1 will be forwarded to the Memory Controller as 64’h0000_0000_3000_0000 – 64’h0000_0000_307F_FFFF. Address hits to page 1 are determined as follows using the information provided in Table 10. Processor bits A[0:3] define the active BAR which in this example are defined as 4’h2. Processor bits A[4:8] define the lookup index (that is, what page the access is mapped to), in order to hit page 1, these bits need to be 5’h01. The remaining Processor bits A[9:31] define the address offset which in this case address the entire 8-Mbyte page of memory. Figure 5 shows the address translation process resulting from a Processor bus address (32’h20F0_0010) hitting the PB_OCN_BAR2.

Register settings:

- **PB_OCN_BAR2:** BA = 4’h2; SIZE = 4’h0 (256M Window); EN = 1
- **PB_BAR2_UPPER_LUT_ADDR1:** TA[63:32] = 32’h0000_0000
- **PB_BAR2_LOWER_LUT_ADDR1:** TA[31:24] = 8’h30, TA[23] = 1’h0; END_MODE = 2’b00; WR_PRTC = 0; ATE = 1; DST_PORT = 4’b0100 (Memory Controller)

Figure 5: Non-direct Path Address Translation Process, 32-bit Processor



2.3 Processor to PCI/X

In order to communicate through the Tsi108/Tsi109 from the Processor bus through the Switch Fabric to the PCI/X Interface, the following registers must be initialized: a PB to Switch Fabric Base Address Register (PB_OCN_BARx), as well as the associated PB Switch Fabric BAR Lower (PB_BARx_LOWER_LUT_ADDRy) and Upper (PB_BARx_UPPER_LUT_ADDRy) Lookup Table Address Registers.

A PB to Switch Fabric BAR defines an address range (in Processor bus address space) where the Tsi108/Tsi109 responds to a Processor's request. If the Processor's request falls within one of the Tsi108/Tsi109's PB to Switch Fabric base address ranges, the transaction is forwarded to the Switch Fabric. The Tsi108/Tsi109 supports two unique PB to Switch Fabric BARs, PB_OCN_BAR1 (offset 0x414) and PB_OCN_BAR2 (offset 0x418).

In order to direct the request from the Switch Fabric to another Tsi108/Tsi109 interface the address must be translated. The PB Switch Fabric BAR Upper and Lower Lookup Table Address Registers must be programmed to provide the proper address translation to target the PCI/X Interface. Each of the PB to Switch Fabric BARs (PB_OCN_BAR1 and PB_OCN_BAR2) can have up to 32 pages of memory associated with them. The PB to Switch Fabric BAR Upper and Lower Lookup Tables make up those 32 pages, each individual page of memory requires an initialization of an Upper and Lower BAR Lookup. The PB to Switch Fabric BAR Upper and Lower Lookup Table Address Registers (PB_BAR[1:2]_LOWER_LUT_ADDR{0:31} and PB_BAR[1:2]_UPPER_LUT_ADDR{0:31}) are located in the Processor bus register block at offsets 0x800 – 0x9FC.

2.3.1 How to Initialize the Processor Bus to PCI/X Path

The PB to Switch Fabric Base Address, PB Switch Fabric BAR Upper and Lower Lookup Table Address Registers can be initialized using PCI/X memory write cycles from the PCI/X bus through the P2O_BAR0 and P2O_BAR0_UPPER registers, or from the Processor bus through the PB_REG_BAR. Tables 7 through 13 describe the internal register settings associated with mapping Processor bus transactions through the Switch Fabric to the PCI/X interface.

2.3.2 Processor to PCI/X Examples

Example 1: 36-bit Processor with extended addressing enabled, access to PCI/X from the processor bus.

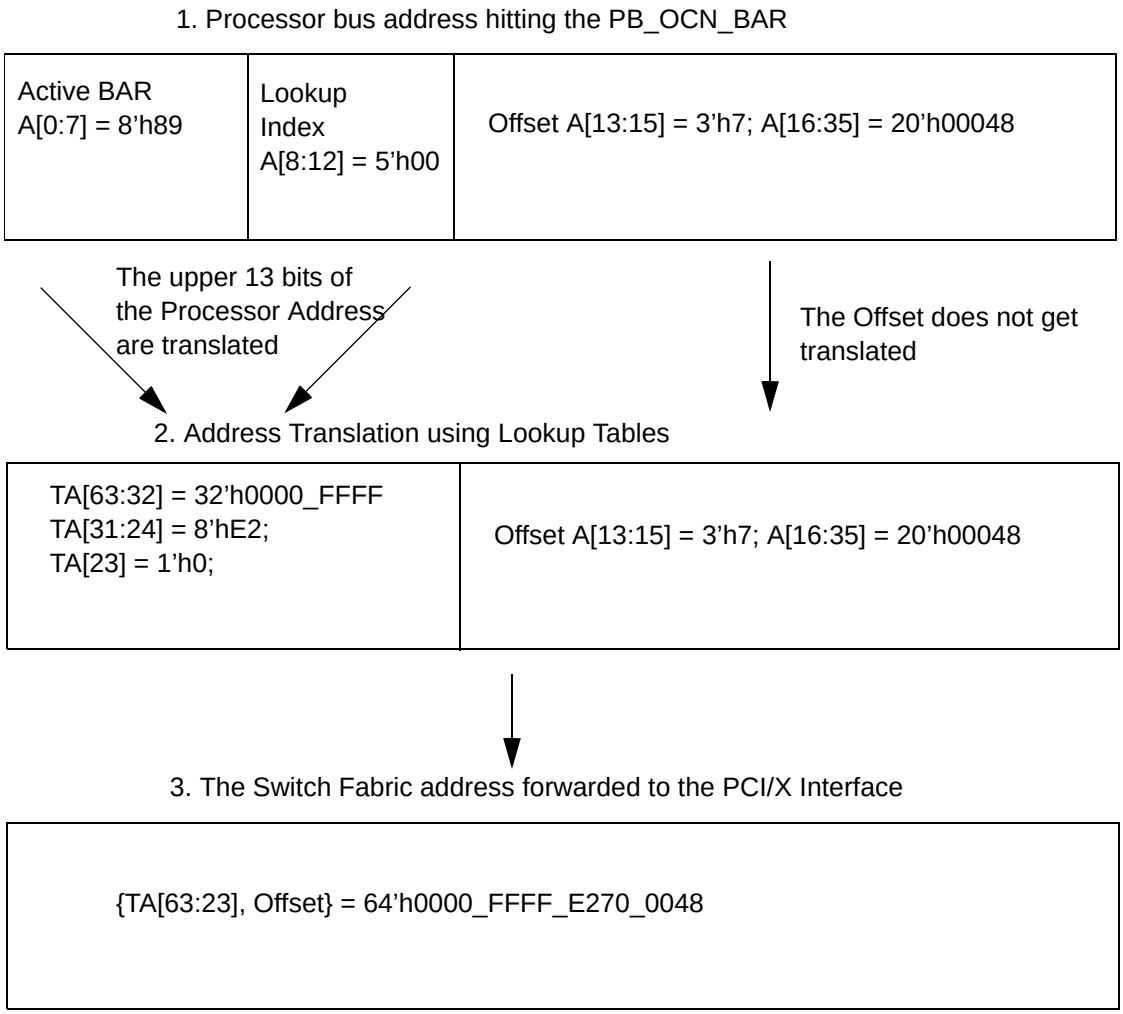
In this example the PB_OCN_BAR2 window size has been programmed to 256 Mbytes, all processor bus addresses that fall within 36’h8_9000_0000 – 36’h8_9FFF_FFFF will be accepted and forwarded through the Switch Fabric as indicated by the values programmed in the PB_OCN_BAR2 register below. Once the transaction has been accepted through the PB_OCN_BAR2 register it is up to the associated lookup tables to determine both the destination port and proper address translation. Since the window size programmed in the PB_OCN_BAR2 register is 256 Mbytes, that means there are 32, 8-Mbyte pages associated with that address window. The PB_BARx_UPPER_LUT_ADDR{0..31} and PB_BARx_LOWER_LUT_ADDR{0..31} registers must be programmed to determine both the destination port and address translation associated with those 32 pages.

In this example only one 8-Mbyte page (page 0) of the 256-Mbyte window is configured to access the PCI/X Interface, by programming the PB_BAR2_UPPER_LUT_ADDR0 and PB_BAR2_LOWER_LUT_ADDR0 registers, page 0 addresses the PCI/X port as indicated by the DST_PORT field (see Table 13). Address translation has been enabled and all Switch Fabric address accesses that address page 0 will be forwarded to the PCI/X Interface as 64’h0000_FFFF_E200_0000 – 64’h0000_FFFF_E27F_FFFF. Address hits to page 0 are determined as follows using the information provided in Table 9. Processor bits A[0:7] define the active BAR which in this example are defined as 8’h89. Processor bits A[8:12] define the lookup index (that is, what page the access is mapped to), in order to hit page 0, these bits need to be 5’h00. The remaining Processor bits A[13:35] define the address offset which in this case address the entire 8-Mbyte page of memory. Figure 6 shows the address translation process resulting from a Processor bus address (36’h8_9070_0048) hitting the PB_OCN_BAR2.

Register settings:

- **PB_OCN_BAR2:** BA = 4’h9; BA_UPPER = 4’h8; SIZE = 4’h0 (256M Window); EN = 1
- **PB_BAR2_UPPER_LUT_ADDR0:** TA[63:32] = 32’h0000_FFFF
- **PB_BAR2_LOWER_LUT_ADDR0:** TA[31:24] = 8’hE2, TA[23] = 1’h0; END_MODE = 2’b00; WR_PRTC = 0; ATE = 1; DST_PORT = 4’b0001 (PCI/X Interface)

Figure 6: PCI/X Path Address Translation Process, 36-bit Processor



Example 2: 32-bit Processor, access to PCI/X from the processor bus.

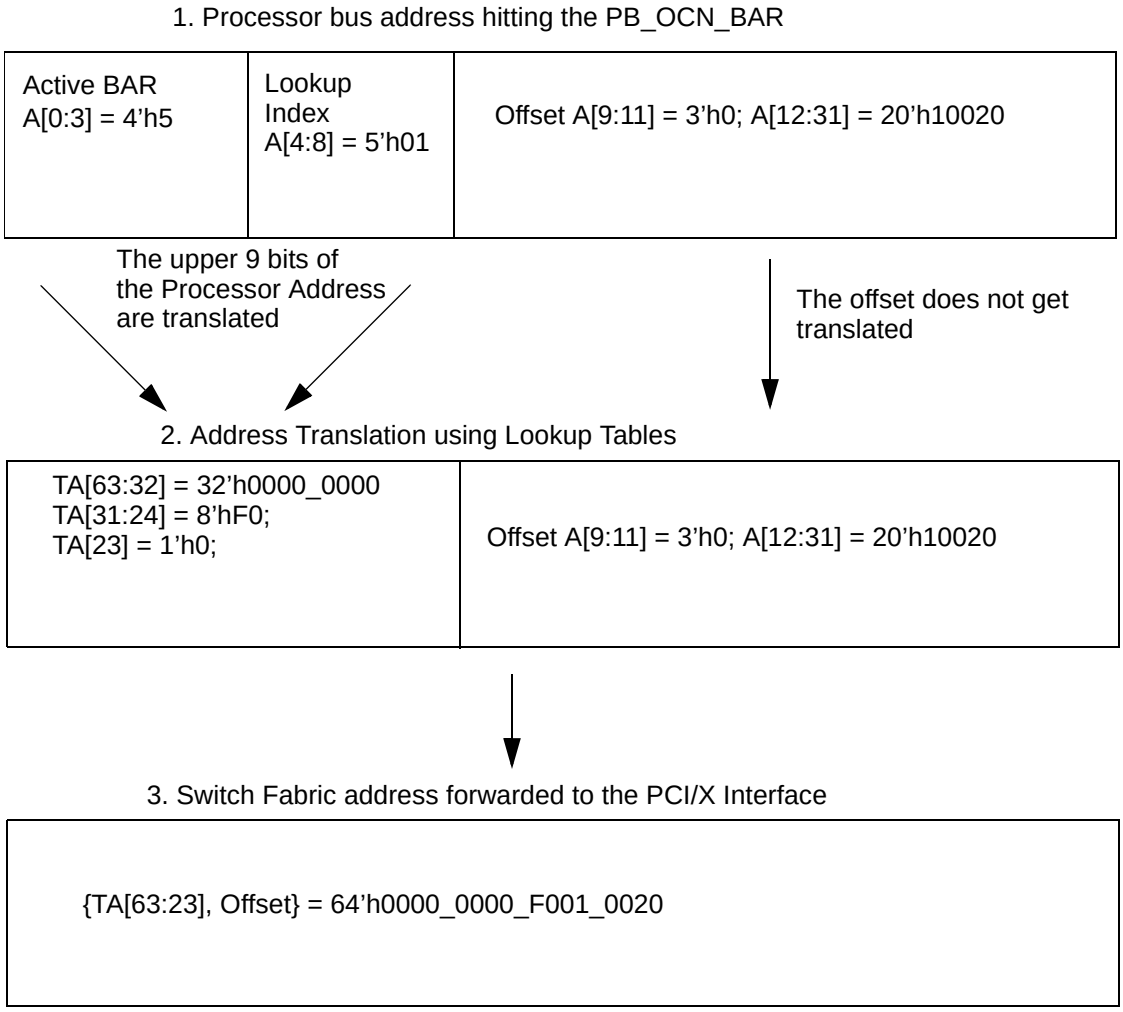
In this example the PB_OCN_BAR2 window size has been programmed to 256 Mbytes, all processor bus addresses that fall within 36’h5000_0000 – 36’h5FFF_FFFF will be accepted and forwarded through the Switch Fabric as indicated by the values programmed in the PB_OCN_BAR2 register below. Once the transaction has been accepted through the PB_OCN_BAR2 register it is up to the associated lookup tables to determine both the destination port and proper address translation. Since the window size programmed in the PB_OCN_BAR2 register is 256 Mbytes, that means there are 32, 8-Mbyte pages associated with that address window. The PB_BARx_UPPER_LUT_ADDR{0..31} and PB_BARx_LOWER_LUT_ADDR{0..31} registers must be programmed to determine both the destination port and address translation associated with those 32 pages.

In this example only one 8-Mbyte page (page 1) of the 256-Mbyte window is configured to access the PCI/X Interface, by programming the PB_BAR2_UPPER_LUT_ADDR1 and PB_BAR2_LOWER_LUT_ADDR1 registers, page 1 addresses the PCI/X port as indicated by the DST_PORT field (see Table 13). Address translation has been enabled and all Switch Fabric address accesses that address page 1 will be forwarded to the PCI/X Interface as 64’h0000_0000_F000_0000 – 64’h0000_0000_F07F_FFFF. Address hits to page 1 are determined as follows using the information provided in Table 10. Processor bits A[0:3] define the active BAR which in this example are defined as 4’h5. Processor bits A[4:8] define the lookup index (that is, what page the access is mapped to), in order to hit page 1, these bits need to be 5’h01. The remaining Processor bits A[9:31] define the address offset which in this case address the entire 8-Mbyte page of memory. Figure 7 shows the address translation process resulting from a Processor bus address (32’h5081_0020) hitting the PB_OCN_BAR2.

Register settings:

- **PB_OCN_BAR2:** BA = 4’h5; SIZE = 4’h0 (256M Window); EN = 1
- **PB_BAR2_UPPER_LUT_ADDR1:** TA[63:32] = 32’h0000_0000
- **PB_BAR2_LOWER_LUT_ADDR1:** TA[31:24] = 8’hF0, TA[23] = 1’h0; END_MODE = 2’b00; WR_PRTC = 0; ATE = 1; DST_PORT = 4’b0001 (PCI/X Interface)

Figure 7: PCI/X Path Address Translation Process, 32-bit Processor



2.3.3 Additional Switch Fabric to PCI/X Transaction Mapping Capabilities

The previous sections describe how to map Processor bus accesses to the PCI/X bus through the Switch Fabric. Based on that configuration, the Tsi108/Tsi109 will only initiate memory cycles on the PCI/X bus. As a result, all read and write commands from the Processor bus or any other Tsi108/Tsi109 port that get mapped to the PCI/X bus will result in the Tsi108/Tsi109 initiating a memory type transaction on the PCI/X bus (that is, Memory Read, Memory Read Line, Memory Read Multiple, Memory Read Block, Memory Read DWORD, Memory Write, Memory Write Block and Dual Address Cycle).

The Switch Fabric to PCI/X path offers additional functionality to convert Switch Fabric addresses received because of a Processor bus access (for example, into a Configuration or an I/O access on the PCI/X bus by implementing a 64-bit Configuration BAR and a 64-bit I/O BAR). It also supports an additional layer of address translation by implementing three additional translation BARs: two 64-bit Prefetchable Memory BARs, and one 32-bit Memory BAR.

The configuration BAR registers (PFAB_BAR0 and PFAB_BAR0_UPPER), when enabled define a 16-Mbyte fixed address window that is used to generate configuration transactions on the PCI/X bus. The I/O Switch Fabric BAR registers (PFAB_IO and PFAB_IO_UPPER), when enabled define a 64-Kbyte fixed address window that is used to generate I/O transactions on the PCI/X bus. The MEM32 Switch Fabric BAR register (PFAB_MEM32), when enabled defines an address window that is used for address translation from the Switch Fabric to 32-bit Memory transactions on the PCI/X bus. The PFAB_MEM32 BAR supports a 32-bit address decode (that is, the upper Switch Fabric address bits [63:32] must be zeros), and can be a minimum of 512M to 1G in size. The PFAB_MEM32 address window supports address translation through the use of a remap register (PFAB_MEM32_REMAP) and a mask register (PFAB_MEM32_MASK). The PFM Switch Fabric BAR registers (PFAB_PFM3 and PFAB_PFM4), when enabled define an address window used for address translation from the Switch Fabric to 64-bit Memory transactions on the PCI/X bus. The PFAB_PFM3 and PFAB_PFM4 BARs support a 64-bit address decode, and can be a minimum of 1G to 2G in size. The PFAB_PFM3 and PFAB_PFM4 address windows support address translation through the use of remap registers (PFAB_PFM[3:4]_REMAP_UPPER and PFAB_PFM[3:4]_REMAP_LOWER) and mask registers (PFAB_PFM[3:4]_MASK).

When any of the above mentioned Switch Fabric to PCI/X BARs are enabled and the Switch Fabric address falls within the defined address window, the PCI/X address or command are altered as described within the individual BAR. These registers are located in the PCI/X bus register block at offsets 0x204 through 0x23C.

2.3.4 How to Initialize the Switch Fabric to PCI/X Transaction Mapping

The Switch Fabric to PCI/X BARs can be initialized using PCI/X memory write cycles from the PCI/X bus through the P2O_BAR0 and P2O_BAR0_UPPER registers, or from the Processor bus through the PB_REG_BAR. Tables 14 to 24 describe the internal register settings associated with setting up the optional Switch Fabric to PCI/X bus transaction mapping.

Table 14: PFAB_BAR0 Configuration Address Register Initialization

Register	Field	Description
PFAB_BAR0	PFAB_BAR[31:24]	PCI/X Configuration Generation Base Address bits [31:24]: These bits are compared with the Switch Fabric address bits [31:24].
PFAB_BAR0	BAR0_EN	PFAB BAR0 Window Enable: Writing a 1 to this location enables the configuration address window. When enabled all Switch Fabric transactions with addresses that fall within the window are get initiated as configuration cycles on the PCI/X bus.

Table 15: PFAB_BAR0_UPPER Configuration Address Register Initialization

Register	Field	Description
PFAB_BAR0_UPPER	PFAB_BAR0[63:32]	PCI/X Configuration Generation Base Address bits [63:32]: These bits are compared with the Switch Fabric address bits [63:32].

Table 16: PFAB_IO Base Address Register Initialization

Register	Field	Description
PFAB_IO	BAR[31:16]	PCI/X IO Generation Base Address bits [31:16]: These bits are compared with the Switch Fabric address bits [31:16].
PFAB_IO	EN	PFAB_IO Window Enable: Writing a 1 to this location enables the IO address window. When enabled all Switch Fabric transactions with addresses that fall within the window get initiated as I/O cycles on the PCI/X bus.

Table 17: PFAB_IO_UPPER Base Address Register Initialization

Register	Field	Description
PFAB_IO_UPPER	BAR[63:32]	PCI/X IO Generation Base Address bits [63:32]: These bits are compared with the Switch Fabric address bits [63:32].

Table 18: PFAB_MEM32 Base Address Register Initialization

Register	Field	Description
PFAB_MEM32	BA[31:29]	32-bit Memory Base Address bits [31:29]: These bits are compared with the Switch Fabric address bits [31:29].
PFAB_MEM32	SIZE	MEM_32 Window Size: 0 = 512 Mbytes 1= 1 Gbyte
PFAB_MEM32	EN	MEM_32 Window Enable: Writing a 1 to this location enables the MEM_32 address window. When enabled all Switch Fabric transactions with addresses that fall within the window get translated as defined by the appropriate remap and mask registers defined in Tables 19 and 20 onto the PCI/X bus.

Table 19: PFAB_MEM32_REMAP Register Initialization

Register	Field	Description
PFAB_MEM32_ REMAP	Remap[31:12]	MEM_32 Remap Value: These bits are substituted with the Switch Fabric address bits [31:12] provided the corresponding Mask value is enabled.

Table 20: PFAB_MEM32_MASK Register Initialization

Register	Field	Description
PFAB_MEM32_MASK	Mask[31:12]	MEM_32 Mask Value: These bits when enabled, enable the substitution of the Switch Fabric address bits [31:12] with the MEM_32 Remap Value Remap[31:12].

Table 21: PFAB_PFM[3:4] Base Address Register Initialization

Register	Field[a]	Description
PFAB_PFM[3:4]	BA[31:30]	PFM[3:4] Base Address bits [31:30]: These bits are compared with the Switch Fabric address bits [31:30].
PFAB_PFM[3:4]	BA[59:48]	PFM[3:4] Base Address bits [59:48]: These bits are compared with the Switch Fabric address bits [59:48].
PFAB_PFM[3:4]	SIZE	PFM[3:4] Window Size: 0 = 1 Gbytes 1= 2 Gbyte
PFAB_PFM[3:4]	EN	PFM[3:4] Window Enable: Writing a 1 to this location enables the PFM[3:4] address window. When enabled all Switch Fabric transactions with addresses that fall within the window get translated as defined by the appropriate remap and mask registers defined in Tables 22 and 24 onto the PCI/X bus.
PFAB_PFM[3:4]	BA[47:32]	PFM[3:4] Base Address bits [47:32]: These bits are compared with the Switch Fabric address bits [47:32].

a. The BA[63:60] value is hardwired to 4’h0.

Table 22: PFAB_PFM[3:4]_REMAP_UPPER Register Initialization

Register	Field	Description
PFAB_PFM[3:4]_ REMAP_UPPER	Remap[63:44]	PFM[3:4] Upper Remap Value: These bits are substituted with the Switch Fabric address bits [63:44] provided the corresponding Mask value is enabled.

Table 23: PFAB_PFM[3:4]_REMAP_LOWER Register Initialization

Register	Field	Description
PFAB_PFM[3:4]_ REMAP_LOWER	Remap[31:12]	PFM[3:4] Lower Remap Value: These bits are substituted with the Switch Fabric address bits [31:12] provided the corresponding Mask value is enabled.
PFAB_PFM[3:4]_ REMAP_LOWER	Remap[43:32]	PFM[3:4] Lower Remap Value: These bits are substituted with the Switch Fabric address bits [43:32] provided the corresponding Mask value is enabled.

Table 24: PFAB_PFM[3:4]_MASK Register Initialization

Register	Field[a]	Description
PFAB_PFM[3:4]_MASK	Mask[31:12]	PFM[3:4] Mask Value: These bits when enabled, enable the substitution of the Switch Fabric address bits [31:12] with the PFM[3:4] Remap Value Remap[31:12].
PFAB_PFM[3:4]_MASK	Mask[43:32]	PFM[3:4] Mask Value: These bits when enabled, enable the substitution of the Switch Fabric address bits [43:32] with the PFM[3:4] Remap Value Remap[43:32].

a. Remap[63:44] are always substituted for Switch Fabric address bits[63:44] as the Mask bits only provide selectable replacement for bits [43:12] of the original Switch Fabric address.

2.4. Switch Fabric to PCI/X Configuration Cycle Example

Example 1: The following example describes the most common use of the Switch Fabric to PCI/X transaction mapping capabilities. The example will describe the use of the Configuration BAR to transform the Switch Fabric address received into a Type 0 configuration cycle on the PCI/X bus (the example assumes the PCI/X Interface is connected to PCI/X Bus #1).

In this example the PFAB_BAR0 and PFAB_BAR0_UPPER registers are programmed such that all Switch Fabric addresses destined for the PCI/X Interface that fall within 64’h0000_0000_F000_0000 – 64’h0000_0000_F0FF_FFFF will be accepted and forwarded to the PCI/X bus as a configuration cycle. Figure 8 uses the Switch Fabric address as a result of a Processor bus access destined for the PCI/X Interface, as defined in the previous example (see Figure 7). The Switch Fabric address (64’h0000_0000_F001_0020) falls within the window defined by the PFAB_BAR0 and PFAB_BAR0_UPPER registers.

Register settings:

- **PFAB_BAR0:** PFAB_BAR0[31:24] = 8’hF0; BAR0_EN = 1
- **PFAB_BAR0_UPPER:** PFAB_BAR0[63:32]= 32’h0000_0000

Figure 8: Switch Fabric to PCI/X Configuration Cycle Generation

1. Switch Fabric address hitting the PFAB_BAR0 and PFAB_BAR0_UPPER

BAR Decode Addr[63:24] = 40'h0000_0000_F0	Outgoing Bus # Addr[23:16] = 8'h01	Outgoing Device # Addr[15:11] = 5'b00000	Outgoing Func. # Addr[10:8] = 3'b000	Outgoing Register # Addr[7:2] = 6'b001000	Addr[1:0] = Undefined

The transformation from Switch Fabric address to Configuration Address is defined in the PCI/X Interface section of the *Tsi108/Tsi109 User Manual* as well as Table 25 below. The fields describe above determine what type of configuration cycle (Type 0 or Type 1) as well as which address line is selected for the IDSEL.

3. Configuration Type 0 Transaction Address forwarded to the PCI/X Interface

Configuration Type 0 at Address = 32'h0001_0020

Note: Bit AD[16] of the above configuration address is asserted due to the Outgoing Device # above being 5'b00000. The device number determines which PCI Address line will be asserted when the Tsi108/Tsi109 in the Type0 Configuration cycle. The device the Tsi108/Tsi109 is addressing the Type 0 configuration cycle to should have its IDSEL input connected to AD[16]. This is PCI spec. and is described in Table 26.

Table 25: Switch Fabric Address and its conversion to PCI/X Configuration Cycles

Incoming Switch Fabric Address Bits	Used For....
Addr[63:24]	PFAB_BAR0 decode.
Addr[23:16]	Outgoing PCI/X bus number. The configuration type is based on the value in this field. For example, if the value matches our destination bus number (the BUS_NUM field in the PE_PCI/X_S, PCI/X offset 0x0F4) then the Tsi108/Tsi109 will generate a Type 0 configuration cycle on the PCI/X bus, at all other times the Tsi108/Tsi109 will generate a configuration Type 1.
Addr[15:11]	Outgoing device number. For Type 0 configuration cycles this number determines which Address line is asserted for the IDSEL connection. This comes directly out of the PCI specification as well as Table 26 below.
Addr[10:8]	Outgoing function number.
Addr[7:2]	Outgoing register number.
Addr[1:0]	These bits have no effect and are undefined.

Table 26: IDSEL Generation

Device Number	Address AD[31:16]
0 0000b	0000 0000 0000 0001b
0 0001b	0000 0000 0000 0010b
0 0010b	0000 0000 0000 0100b
0 0011b	0000 0000 0000 1000b
0 0100b	0000 0000 0001 0000b
0 0101b	0000 0000 0010 0000b
0 0110b	0000 0000 0100 0000b
0 0111b	0000 0000 1000 0000b
0 1000b	0000 0001 0000 0000b
0 1001b	0000 0010 0000 0000b
0 1010b	0000 0100 0000 0000b
0 1011b	0000 1000 0000 0000b
0 1100b	0001 0000 0000 0000b
0 1101b	0010 0000 0000 0000b

Table 26: IDSEL Generation *(Continued)*

Device Number	Address AD[31:16]
0 1110b	0100 0000 0000 0000b
0 1111b	1000 0000 0000 0000b
1 XXXXb	0000 0000 0000 0000b

2.5 Processor to HLP

In order to communicate through the Tsi108/Tsi109 from the Processor bus through the Switch Fabric to the Host Local Port (HLP) Interface the following registers must be initialized: a PB to Switch Fabric Base Address Register (PB_OCN_BARx), as well as the associated PB Switch Fabric BAR Lower (PB_BARx_LOWER_LUT_ADDRy) and Upper (PB_BARx_UPPER_LUT_ADDRy) Lookup Table Address Registers. A PB to Switch Fabric BAR defines an address range (in Processor bus address space) where the Tsi108/Tsi109 responds to a Processor's request. If the Processor's request falls within one of the Tsi108/Tsi109's PB to Switch Fabric base address ranges, the transaction is forwarded through to the Switch Fabric. The Tsi108/Tsi109 supports two unique PB to Switch Fabric BARs: PB_OCN_BAR1 (offset 0x414) and PB_OCN_BAR2 (offset 0x418).

In order to direct the request from the Switch Fabric to another Tsi108/Tsi109 interface the address must be translated. The PB Switch Fabric BAR Upper and Lower Lookup Table Address Registers must be programmed to provide the proper address translation to target the HLP Interface. Each of the PB to Switch Fabric BARs (PB_OCN_BAR1 and PB_OCN_BAR2) can have up to 32 pages of memory associated with them. The PB to Switch Fabric BAR Upper and Lower Lookup Tables make up those 32 pages, and each individual page of memory requires an initialization of an Upper and Lower BAR Lookup. The PB to Switch Fabric BAR Upper and Lower Lookup Table Address Registers (PB_BAR[1:2]_LOWER_LUT_ADDR{0:31} and PB_BAR[1:2]_UPPER_LUT_ADDR{0:31}) are located in the Processor bus register block at offsets 0x800 – 0x9FC.

2.5.1 How to Initialize the Processor Bus to HLP Path

The HLP Interface is unique in that the Tsi108/Tsi109 automatically comes up ready to boot from HLP. The Tsi108/Tsi109 PB to Switch Fabric Base Address Register (PB_OCN_BAR1) comes up enabled to access HLP at the HLP pre-defined address 32'h0xfff0_0000. All processors whether 32-bit or 36-bit, boot up in 32-bit mode. At power-up all Switch Fabric BAR1 accesses are forwarded to HLP. The Processor bus to HLP path can also be configured manually. The PB to Switch Fabric Base Address, PB Switch Fabric BAR Upper and Lower Lookup Table Address Registers can be initialized using PCI/X memory write cycles from the PCI/X bus through the P2O_BAR0 and P2O_BAR0_UPPER Registers or from the Processor bus through the PB_REG_BAR. Tables 7 through 13 describe the internal register settings associated with mapping Processor bus transactions through the Switch Fabric and onto the HLP Interface.

2.5.2 Processor to HLP Examples (After Initial BOOT)

Example 1: 36-bit Processor with extended addressing enabled, access to HLP from the processor bus.

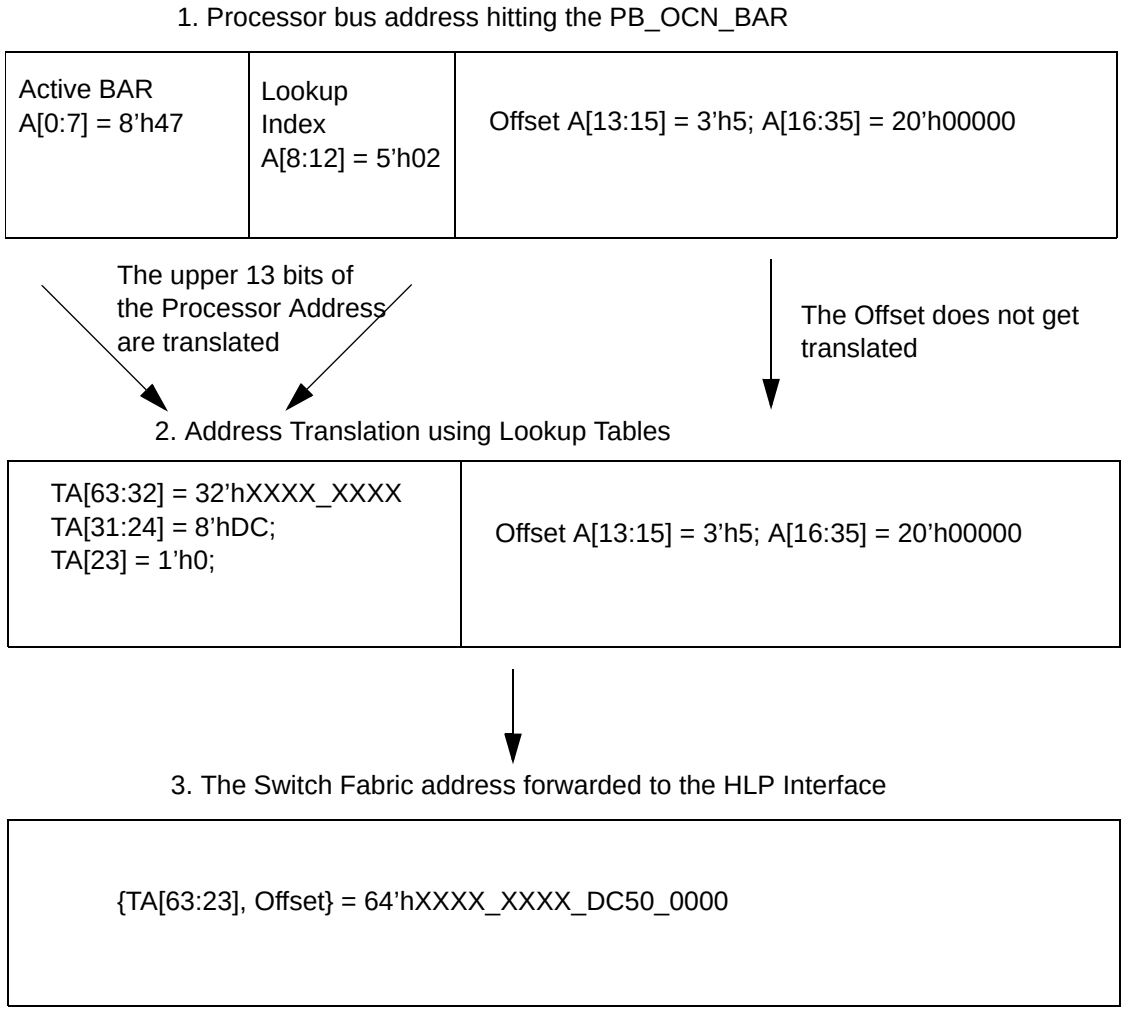
In this example the PB_OCN_BAR2 window size has been programmed to 256 Mbytes, all processor bus addresses that fall within 36'h4_7000_0000 – 36'h4_7FFF_FFFF will be accepted and forwarded through the Switch Fabric as indicated by the values programmed in the PB_OCN_BAR2 register below. Once the transaction has been accepted through the PB_OCN_BAR2 register it is up to the associated lookup tables to determine both the destination port and proper address translation. Since the window size programmed in the PB_OCN_BAR2 register is 256 Mbytes, that means there are 32, 8-Mbyte pages associated with that address window.

The PB_BARx_UPPER_LUT_ADDR{0..31} and PB_BARx_LOWER_LUT_ADDR{0..31} registers must be programmed to determine both the destination port and address translation associated with those 32 pages. In this example only one 8-Mbyte page (page 2) of the 256-Mbyte window is configured to access the HLP Interface, by programming the PB_BAR2_UPPER_LUT_ADDR2 and PB_BAR2_LOWER_LUT_ADDR2 registers, page 2 addresses the HLP port as indicated by the DST_PORT field (see Table 13). Address translation has been enabled and all Switch Fabric address accesses that address page 2 will be forwarded to the HLP interface as 64'hXXXX_XXXX_DC00_0000 – 64'hXXXX_XXXX_DC7F_FFFF. Address hits to page 2 are determined as follows using the information provided in Table 9. Processor bits A[0:7] define the active BAR which in this example are defined as 8'h47. Processor bits A[8:12] define the lookup index (that is, what page the access is mapped to), in order to hit page 2, these bits need to be 5'h02. The remaining Processor bits A[13:35] define the address offset which in this case address the entire 8-Mbyte page of memory. Figure 9 shows the address translation process resulting from a Processor bus address (36'h4_7150_0000) hitting the PB_OCN_BAR2.

Register settings:

- **PB_OCN_BAR2:** BA = 4'h7; BA_UPPER = 4'h4; SIZE = 4'h0 (256M Window); EN = 1
- **PB_BAR2_UPPER_LUT_ADDR2:** TA[63:32] = 32'hXXXX_XXXX (The 64-bit Switch Fabric address is truncated to the 32-bit address presented on the HLP AD pins, therefore the upper translation bits are don't cares.)
- **PB_BAR2_LOWER_LUT_ADDR2:** TA[31:24] = 8'hDC, TA[23] = 1'h0; END_MODE = 2'b00; WR_PRTC = 0; ATE = 1; DST_PORT = 4'b0000 (HLP Interface)

Figure 9: HLP Path Address Translation Process, 36-bit Processor



Example 2: 32-bit Processor, access to HLP from the processor bus.

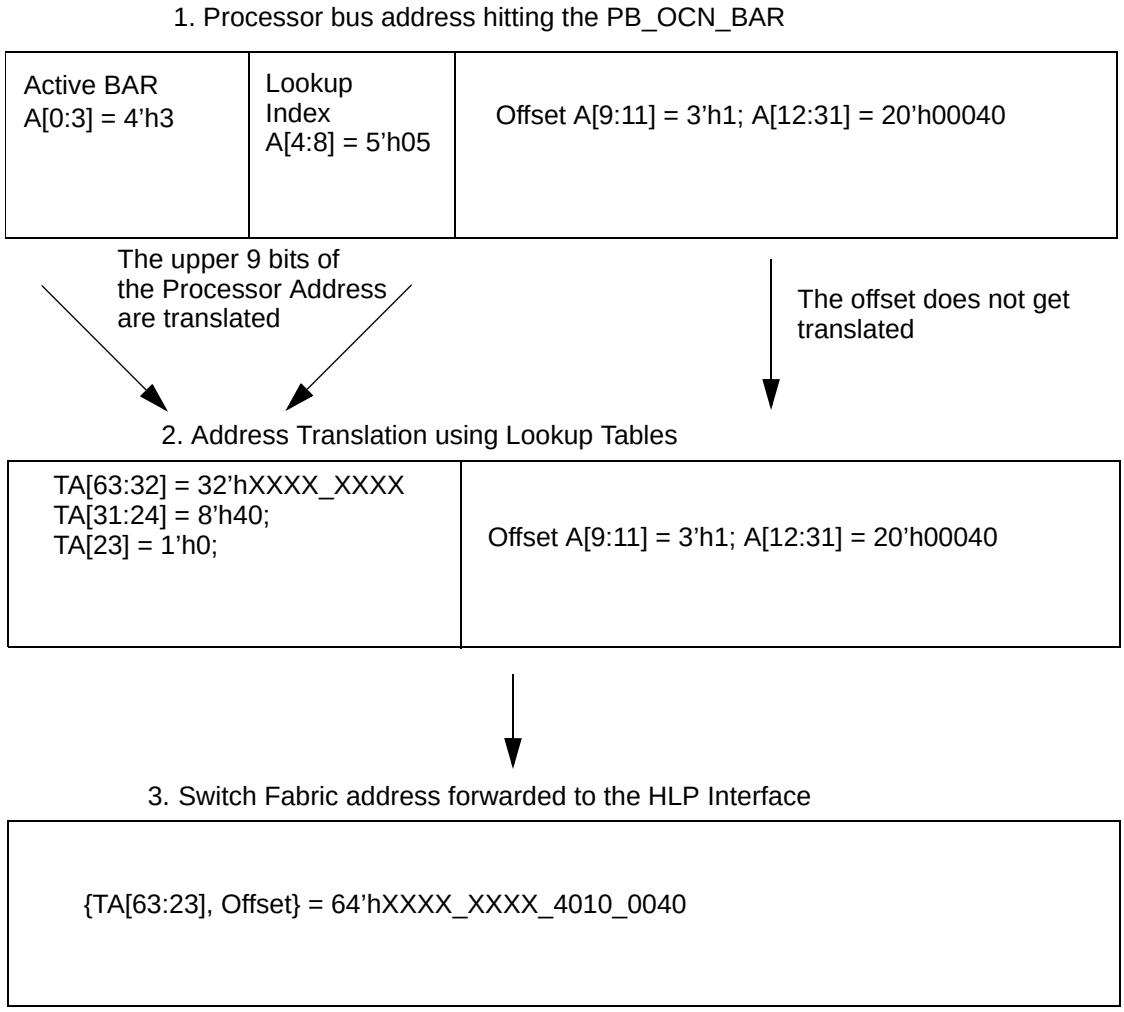
In this example the PB_OCN_BAR2 window size has been programmed to 256 Mbytes, all processor bus addresses that fall within 36'h3000_0000 – 36'h3FFF_FFFF will be accepted and forwarded through the Switch Fabric as indicated by the values programmed in the PB_OCN_BAR2 register below. Once the transaction has been accepted through the PB_OCN_BAR2 register it is up to the associated lookup tables to determine both the destination port and proper address translation. Since the window size programmed in the PB_OCN_BAR2 register is 256 Mbytes, that means there are 32, 8-Mbyte pages associated with that address window.

The PB_BARx_UPPER_LUT_ADDR{0..31} and PB_BARx_LOWER_LUT_ADDR{0..31} registers must be programmed to determine both the destination port and address translation associated with those 32 pages. In this example only one 8-Mbyte page (page 5) of the 256-Mbyte window is configured to access the HLP Interface, by programming the PB_BAR2_UPPER_LUT_ADDR5 and PB_BAR2_LOWER_LUT_ADDR5 registers, page 5 addresses the HLP port as indicated by the DST_PORT field (see Table 13). Address translation has been enabled and all Switch Fabric address accesses that address page 5 will be forwarded to the HLP Interface as 64'hXXXX_XXXX_4000_0000 – 64'hXXXX_XXXX_407F_FFFF. Address hits to page 5 are determined as follows using the information provided in Table 10. Processor bits A[0:3] define the active BAR which in this example are defined as 4'h3. Processor bits A[4:8] define the lookup index (that is, what page the access is mapped to), in order to hit page 5, these bits need to be 5'h05. The remaining Processor bits A[9:31] define the address offset which in this case address the entire 8-Mbyte page of memory. Figure 10 shows the address translation process resulting from a Processor bus address (32'h3290_0040) hitting the PB_OCN_BAR2.

Register settings:

- **PB_OCN_BAR2:** BA = 4'h3; SIZE = 4'h0 (256M Window); EN = 1
- **PB_BAR2_UPPER_LUT_ADDR5:** TA[63:32] = 32'hXXXX_XXXX (The 64-bit Switch Fabric address is truncated to the 32-bit address presented on the HLP AD pins, therefore the upper translation bits are don't cares.)
- **PB_BAR2_LOWER_LUT_ADDR5:** TA[31:24] = 8'h40, TA[23] = 1'h0; END_MODE = 2'b00; WR_PRTC = 0; ATE = 1; DST_PORT = 4'b0000 (HLP Interface)

Figure 10: HLP Path Address Translation Process, 32-bit Processor



2.6. PCI/X to Memory Controller (Non-snoop Path)

In order to communicate from the PCI/X bus through the Switch Fabric to memory (Non-snoop path) using the Memory Controller the following registers must be initialized: a PCI Base Address Register (P2O_BARx and P2O_BARx_UPPER), the PCI Page Size Register (P2O_PAGE_SIZES), as well as the associated PCI BAR Lower (P2O_BARx_LUTy) and Upper (P2O_BARx_LUT_UPPERy) Lookup Table Address Registers.

A PCI base address register defines an address range (in PCI/X bus address space) where the Tsi108/Tsi109 responds to a PCI/X bus initiator's request. If the PCI/X bus initiator's request falls within one of the Tsi108/Tsi109 PCI base address ranges, the transaction is forwarded through to the Switch Fabric. The Tsi108/Tsi109 supports two unique PCI BARs: P2O_BAR2 (offset 0x018), P2O_BAR2_UPPER (offset 0x01C); and P2O_BAR3 (offset 0x020), P2O_BAR3_UPPER (offset 0x024) (see Tables 27 and 28). The PCI BARs are enabled through writes to the PCI Page Size Register. The size of the individual address window is also configurable within this register (see Tables 29 and 30). The PCI Page Size Register (P2O_PAGE_SIZES) is located in the PCI/X bus register block at offset 0x04C.

In order to direct the request from the Switch Fabric to another Tsi108/Tsi109 interface the address must be translated. The PCI BAR Upper and Lower Lookup Table Address Registers must be programmed to provide the proper address translation to target the Memory Controller to memory (SDRAM). Each of the PCI Base Address Registers (P2O_BAR2, P2O_BAR2_UPPER and P2O_BAR3, P2O_BAR3_UPPER) can have up to 32 pages of memory associated with them. The PCI BAR Upper and Lower Lookup Tables make up those 32 pages, and each individual page of memory requires an initialization of a Lower BAR Lookup and possibly an Upper BAR Lookup depending on whether address translation is enabled or disabled in the PCI Page Size Register (see Figure 11 along with Tables 31 and 32). The PCI BAR Upper and Lower Lookup Table Address Registers (P2O_BAR[2:3]_LUT{0:31} and P2O_BAR[2:3]_LUT_UPPER{0:31}) are located in the PCI/X bus register block at offsets 0x500 – 0x6FC.

Figure 11: PCI/X BAR Lookup Table Addressing

This example describes the Page addressing and address translation process for a 256K window when address translation is enabled in the P2O_PAGE_SIZES register.

	Lookup Tables used for address translation
Page 0 --> AD[13:17] = 5'h0	--> P2O_BARx_LUT0 and P2O_BARx_LUT_UPPER0
Page 1 --> AD[13:17] = 5'h1	--> P2O_BARx_LUT1 and P2O_BARx_LUT_UPPER1
Page 2 --> AD[13:17] = 5'h2	--> P2O_BARx_LUT2 and P2O_BARx_LUT_UPPER2
Page 3 --> AD[13:17] = 5'h3	--> P2O_BARx_LUT3 and P2O_BARx_LUT_UPPER3
Page 29 --> AD[13:17] = 5'h1D	--> P2O_BARx_LUT29 and P2O_BARx_LUT_UPPER29
Page 30 --> AD[13:17] = 5'h1E	--> P2O_BARx_LUT30 and P2O_BARx_LUT_UPPER30
Page 31 --> AD[13:17] = 5'h1F	--> P2O_BARx_LUT31 and P2O_BARx_LUT_UPPER31

2.6.1 How to Initialize the PCI/X Bus to SDRAM Non-snoop Path

The PCI Base Address Registers, PCI Page Size Register, PCI BAR Upper and Lower Lookup Table Address Registers can be initialized using PCI/X memory write cycles from the PCI/X bus through the P2O_BAR0 and P2O_BAR0_UPPER Registers, or from the Processor bus through the PB_REG_BAR. Table 13 and Tables 27 to 32 describe the internal register settings associated with mapping PCI/X bus transactions through the Switch Fabric to the Memory Controller and onto memory.

Table 27: PCI Base Address Register Initialization

Register	Field	Description
P2O_BAR[2:3]	BA[31:15]	Base Address: These bits are compared with PCI/X address bits AD[31:15] to determine if the transaction falls within the P2O_BAR[2:3]. If the address is a 64-bit address these bits are combined with the values in the P2O_BAR[2:3]_UPPER register.
P2O_BAR[2:3]	PRFTCH	Prefetchable: This window is a prefetchable window only.
P2O_BAR[2:3]	TYPE	Address Type: The PCI/X BAR is a 64-bit BAR located anywhere in 64-bit address space.
P2O_BAR[2:3]	IO_MODE	PCI Bus Address Space: The PCI/X BAR is a memory only BAR

Table 28: PCI Base Address Upper Register Initialization

Register	Field	Description
P2O_BAR[2:3]_ UPPER	BA[63:32]	Base Address Upper: This field defines the upper address portion of the 64-bit PCI/X BAR. If the window is based on a 32-bit address bus these bits remain zero.

Table 29: PCI Page Size Register Initialization

Register	Field	Description
P2O_PAGE_SIZES	BAR2_SIZE[15:11]	BAR2 Page Size: When the P2O_BAR2 is enabled these bits define the size of the address window. The address window consists of 32 memory pages (see Table 30).
P2O_PAGE_SIZES	BAR2_NOTRAN	BAR2 Address Translation Control: This bit controls whether address translation is enabled or disabled. Unlike other BARs the PCI BAR can either use the translation address placed in the appropriate lookup tables or the PCI/X bus address can be passed on to the Switch Fabric directly. When address translation is disabled, the lookup tables are only used to determine the destination port. The reset value of this register enables address translation.
P2O_PAGE_SIZES	BAR2_EN	Fabric BAR2 Enable: This bit is used to enable P2O_BAR2 and P2O_BAR2_UPPER. Writes to these registers are disabled until BAR2_EN is written.
P2O_PAGE_SIZES	BAR3_SIZE[7:3]	BAR3 Page Size: When the P2O_BAR3 is enabled these bits define the size of the address window. The address window consists of 32 memory pages (see Table 30).
P2O_PAGE_SIZES	BAR3_NO_TRAN	BAR3 Address Translation Control: This bit controls whether address translation is enabled or disabled. Unlike other BARs the PCI BAR can either use the translation address placed in the appropriate lookup tables or the PCI/X bus address can be passed on to the Switch Fabric directly. When address translation is disabled, the lookup tables are only used to determine the destination port. The reset value of this register enables address translation.
P2O_PAGE_SIZES	BAR3_EN	Fabric BAR3 Enable: This bit is used to enable P2O_BAR3 and P2O_BAR3_UPPER. Writes to these registers are disabled until BAR3_EN is written.

Table 30: P2O_BAR[2:3]_SIZE Initialization

BAR_SIZE	Window Size (Bytes)	Page Size (Bytes)	Active BAR	Lookup Index	Offset
5'h00	32K	1K	AD[63:15]	AD[14:10]	AD[9:0]
5'h01	64K	2K	AD[63:16]	AD[15:11]	AD[10:0]
5'h02	128K	4K	AD[63:17]	AD[16:12]	AD[11:0]
5'h03	256K	8K	AD[63:18]	AD[17:13]	AD[12:0]
5'h04	512K	16K	AD[63:19]	AD[18:14]	AD[13:0]
5'h05	1M	32K	AD[63:20]	AD[19:15]	AD[14:0]
5'h06	2M	64K	AD[63:21]	AD[20:16]	AD[15:0]
5'h07	4M	128K	AD[63:22]	AD[21:17]	AD[16:0]
5'h08	8M	256K	AD[63:23]	AD[22:18]	AD[17:0]
5'h09	16M	512K	AD[63:24]	AD[23:19]	AD[18:0]
5'h0A	32M	1M	AD[63:25]	AD[24:20]	AD[19:0]
5'h0B	64M	2M	AD[63:26]	AD[25:21]	AD[20:0]
5'h0C	128M	4M	AD[63:27]	AD[26:22]	AD[21:0]
5'h0D	256M	8M	AD[63:28]	AD[27:23]	AD[22:0]
5'h0E	512M	16M	AD[63:29]	AD[28:24]	AD[23:0]
5'h0F	1G	32M	AD[63:30]	AD[29:25]	AD[24:0]
5'h10	2G	64M	AD[63:31]	AD[30:26]	AD[25:0]
5'h11	4G	128M	AD[63:32]	AD[31:27]	AD[26:0]
5'h12	8G	256M	AD[63:33]	AD[32:28]	AD[27:0]
5'h13	16G	512M	AD[63:34]	AD[33:29]	AD[28:0]
5'h14	32G	1G	AD[63:35]	AD[34:30]	AD[29:0]
5'h15	64G	2G	AD[63:36]	AD[35:31]	AD[30:0]
5'h16	128G	4G	AD[63:37]	AD[36:32]	AD[31:0]
5'h17	256G	8G	AD[63:38]	AD[37:33]	AD[32:0]
5'h18	512G	16G	AD[63:39]	AD[38:34]	AD[33:0]
5'h19	1T	32G	AD[63:40]	AD[39:35]	AD[34:0]
5'h1A	2T	64G	AD[63:41]	AD[40:36]	AD[35:0]

Table 30: P2O_BAR[2:3]_SIZE Initialization *(Continued)*

BAR_SIZE	Window Size (Bytes)	Page Size (Bytes)	Active BAR	Lookup Index	Offset
5'h1B	4T	128G	AD[63:42]	AD[41:37]	AD[36:0]
5'h1C	8T	256G	AD[63:43]	AD[42:38]	AD[37:0]
5'h1D	16T	512G	AD[63:44]	AD[43:39]	AD[38:0]
5'h1E	32T	1T	AD[63:45]	AD[44:40]	AD[39:0]
5'h1F	64T	2T	AD[63:46]	AD[45:41]	AD[40:0]

Table 31: PCI BAR Lower Lookup Table Address Register Initialization

Register	Field	Description
P2O_BAR[2:3]_ LUT{0..31}	BAR[2:3]_PAGE_ ADDR	PCI/X BAR[2:3] Translation Address: When the BAR[2:3]_NOTRAN bit in the P20_PAGE_SIZES register is a zero, these bits replace the lower PCI/X bus address bits. The bits replaced depend on the Size of the Address Window.
P2O_BAR[2:3]_ LUT{0..31}	BAR[2:3]_DESTID	Switch Fabric Destination Port Number: Each Tsi108/Tsi109 external interface is connected to a Switch Fabric port (see Table 13).

Table 32: PCI BAR Upper Lookup Table Address Register Initialization

Register	Field	Description
P2O_BAR[2:3]_LUT_UP PER{0..31}	BAR[2:3]_PAGE_ ADDR	PCI/X BAR[2:3] Translation Address: When the BAR[2:3]_NOTRAN bit in the P20_PAGE_SIZES register is a zero, these bits replace the upper PCI/X bus address bits. The bits replaced depend on the Size of the Address Window.

2.6.2 PCI/X to Memory Controller (Non-snoop) Example

Example: Since the address translation process involves the same steps for a 32-bit or a 64-bit PCI/X address, only one example applies. The example below shows the translation process for a 32-bit PCI/X address, access to memory through the Non-snoop path.

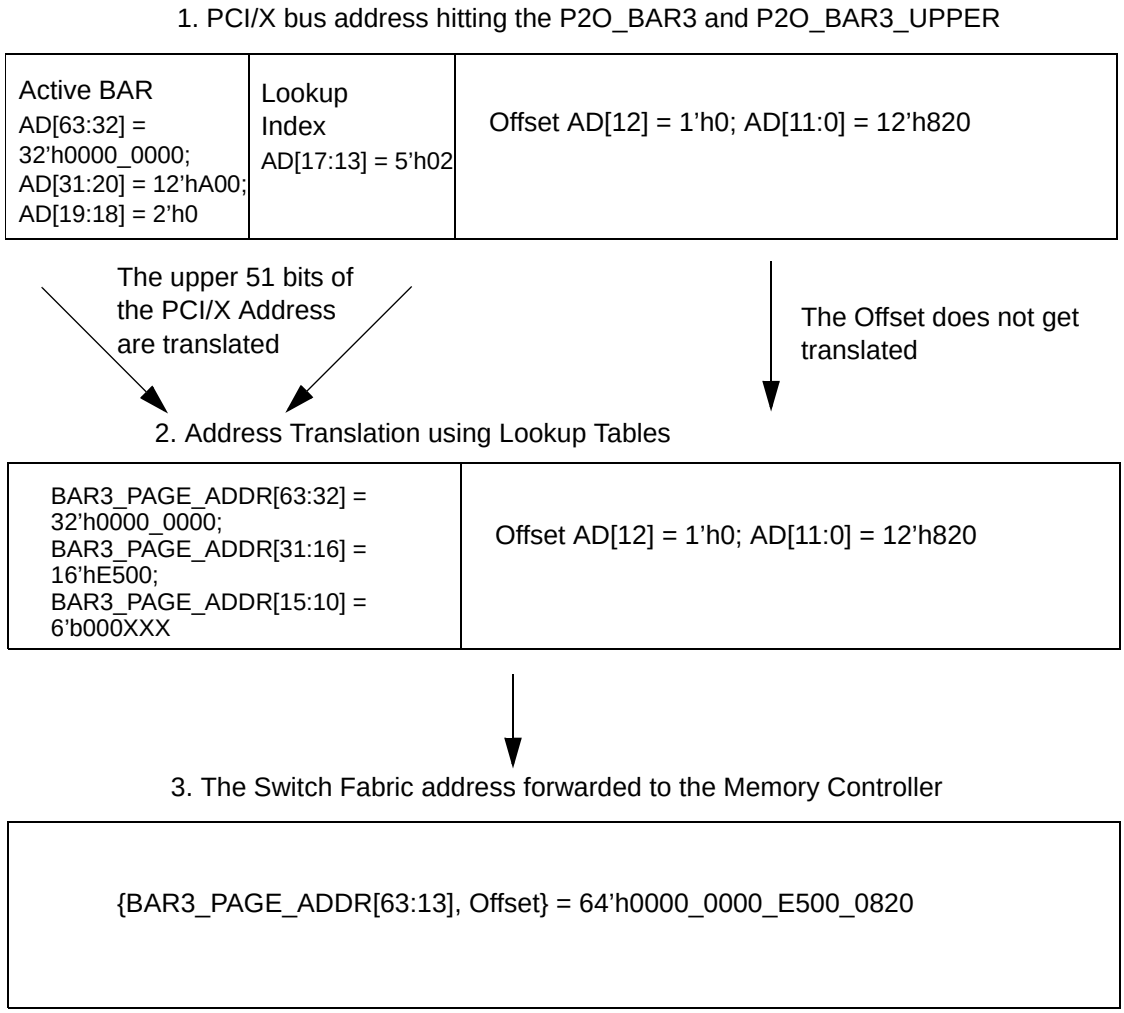
In this example the P2O_PAGE_SIZES window size (BAR3_SIZE) has been programmed to 256 Kbytes, all PCI/X bus addresses that fall within 64’h0000_0000_A000_0000 – 64’h0000_0000_A003_FFFF will be accepted and forwarded through the Switch Fabric as indicated by the values programmed in the P2O_BAR3 and P2O_BAR3_UPPER registers below. Once the transaction has been accepted through the P2O_BAR3 and P2O_BAR3_UPPER registers it is up to the associated lookup tables to determine both the destination port and proper address translation, provided translation is enabled. Since the window size programmed in the P2O_PAGE_SIZES register is 256 Kbytes, that means there are 32, 8-Kbyte pages associated with that address window.

The P2O_BARx_LUT{0..31} and P2O_BARx_LUT_UPPER{0..31} registers must be programmed to determine both the destination port and address translation associated with those 32 pages. In this example only one 8-Kbyte page (page 2) of the 256-Kbyte window is configured to access memory, by programming the P2O_BAR3_LUT2 and P2O_BAR3_LUT_UPPER2 registers, page 2 addresses memory as indicated by the BAR3_DESTID field (see Table 13). Address translation is enabled as indicated by the value in the P2O_PAGE_SIZES[BAR3_NOTRAN] field and all Switch Fabric address accesses that address page 2 will be forwarded to the Memory Controller as 64’h0000_0000_E500_0000 – 64’h0000_0000_E500_1FFF. Address hits to page 2 are determined as follows using the information provided in Table 30. PCI/X bits AD[63:18] define the active BAR which in this example are defined as {44’h0000_0000_A00, 2’h0}. PCI/X bits AD[17:13] define the lookup index (that is, what page the access is mapped to), in order to hit page 2, these bits need to be 5’h02. The remaining PCI/X bits AD[12:0] define the address offset that in this case address the entire 8-Kbyte page of memory. Figure 12 shows the address translation process resulting from a PCI/X bus address (32’hA000_4820) hitting the P20_BAR3 and P2O_BAR3_UPPER registers.

Register settings:

- **P2O_PAGE_SIZES:** BAR3_EN = 1; BAR3_NOTRAN = 0; BAR3_SIZE = 5’h03 (256K Window)
- **P2O_BAR3:** BA[31:20] = 16’hA00, BA[19:18] = 2’h0, BA[17:15] = 3’bXXX; PRFTCH = 1; TYPE = 2’h2; IO_MODE = 0
- **P2O_BAR3_UPPER:** BA[63:32] = 32’h0000_0000
- **P2O_BAR3_LUT2:** BAR3_PAGE_ADDR[31:16] = 16’hE500, BAR3_PAGE_ADDR[15:10] = 6’b000XXX; BAR3_DESTID = 4’b0100 (Memory Controller)
- **PB_BAR3_LUT_UPPER2:** BAR3_PAGE_ADDR[31:0] = 32’h0000_0000

Figure 12: Non-snoop Path Address Translation Process, 32-bit PCI/X Address



2.7. PCI/X to Memory Controller (Snoop Path)

In order to communicate through the Tsi108/Tsi109 from the PCI/X bus through the Switch Fabric to Processor Bus Master Interface (Snoop path) and onto memory the following registers must be initialized: a PCI Base Address Register (P2O_BARx and P2O_BARx_UPPER), the PCI Page Size Register (P2O_PAGE_SIZES), the associated PCI BAR Lower (P2O_BARx_LUTy) and Upper (P2O_BARx_LUT_UPPERy) Lookup Table Address Registers, as well as a Processor bus to SDRAM Base Address Register (PB_SDRAM_BARx).

A PCI BAR defines an address range (in PCI/X bus address space) where the Tsi108/Tsi109 responds to a PCI/X bus initiator's request. If the PCI/X bus initiator's request falls within one of the Tsi108/Tsi109 PCI base address ranges, the transaction is forwarded to the Switch Fabric. The Tsi108/Tsi109 supports two unique PCI BARs: P2O_BAR2 (offset 0x018), P2O_BAR2_UPPER (offset 0x01C); and P2O_BAR3 (offset 0x020), P2O_BAR3_UPPER (offset 0x024) (see Tables 27 and 28). The PCI Base Address Registers are enabled through writes to the PCI Page Size Register, the size of the individual address window is also configurable within this register (see Tables 29 and 30). The PCI Page Size Register (P2O_PAGE_SIZES) is located in the PCI/X bus register block at offset 0x04C.

In order to direct the request from the Switch Fabric to another Tsi108/Tsi109 interface the address must be translated. The PCI BAR Upper and Lower Lookup Table Address Registers must be programmed to provide the proper address translation to target the Processor Master Interface. Each of the PCI BARs (P2O_BAR2, P2O_BAR2_UPPER and P2O_BAR3, P2O_BAR3_UPPER) can have up to 32 pages of memory associated with them. The PCI BAR Upper and Lower Lookup Tables make up those 32 pages, and each individual page of memory requires an initialization of a Lower BAR Lookup and possibly an Upper BAR Lookup depending on whether address translation is enabled or disabled in the PCI Page Size Register (see Figure 11 along with Tables 31 and 32).

The PCI BAR Upper and Lower Lookup Table Address Registers (P2O_BAR[2:3]_LUT{0:31} and P2O_BAR[2:3]_LUT_UPPER{0:31}) are located in the PCI/X bus register block at offsets 0x500 – 0x6FC. The registers programmed so far are used to route the PCI/X address to the Processor Master Interface and onto the Processor bus. An additional register needs to be programmed to allow access to memory; a Processor bus to SDRAM BAR is used for that purpose. The Processor bus to SDRAM Base Address Registers (PB_SDRAM_BAR[1:2]) are located in the Processor bus register block at offsets 0x41C and 0x420 (see Tables 4 and 5, and 6).

2.7.1 How to Initialize the PCI/X Bus to SDRAM Snoop Path

The PCI BARs, PCI Page Size Register, PCI BAR Upper Lower Lookup Table Address Registers and Processor bus to SDRAM Base Address Register, can be initialized using PCI/X memory write cycles from the PCI/X bus through the P2O_BAR0 and P2O_BAR0_UPPER Registers, or from the Processor bus through the PB_REG_BAR. Tables 4 to 6, 13, and 27 through 32 describe the internal register settings associated with mapping PCI/X bus transactions through the Switch Fabric to the Processor Master Interface and onto memory.

2.7.2 PCI/X to Memory (Snoop) Example

Example: Since the address translation process involves the same steps for a 32-bit or a 64-bit PCI/X address, only one example applies. The example below shows the translation process for a 32-bit PCI/X address, access to the Processor Master Interface through the Switch Fabric and onto memory through the Processor bus direct path. The Snoop path allows processors on the 60x/MPX bus to snoop the transaction in support of cache coherency.

In this example the P2O_PAGE_SIZES window size (BAR3_SIZE) has been programmed to 256 Kbytes, all PCI/X bus addresses that fall within 64’h0000_0000_C000_0000 – 64’h0000_0000_C003_FFFF will be accepted and forwarded through the Switch Fabric as indicated by the values programmed in the P2O_BAR3 and P2O_BAR3_UPPER registers below. Once the transaction has been accepted through the P2O_BAR3 and P2O_BAR3_UPPER registers it is up to the associated lookup tables to determine both the destination port and proper address translation, provided translation is enabled. Since the window size programmed in the P2O_PAGE_SIZES register is 256 Kbytes, that means there are 32, 8-Kbyte pages associated with that address window.

The P2O_BARx_LUT{0..31} and P2O_BARx_LUT_UPPER{0..31} registers must be programmed to determine both the destination port and address translation associated with the 32 pages. In this example only one 8-Kbyte page (page 6) of the 256-Kbyte window is configured to access the Processor Bus Master Interface, by programming the P2O_BAR3_LUT6 and P2O_BAR3_LUT_UPPER6 registers, page 6 addresses the Processor Bus Master Interface as indicated by the BAR3_DESTID field (see Table 13). Address translation is enabled as indicated by the value in the P2O_PAGE_SIZES[BAR3_NOTRAN] field and all Switch Fabric address accesses that address page 6 will be forwarded to the Processor Bus Master Interface as 64’h0000_0008_A000_0000 – 64’h0000_0008_A000_1FFF.

Address hits to page 6 are determined as follows using the information provided in Table 30 on page 44. PCI/X bits AD[63:18] define the active BAR which in this example are defined as {44’h0000_0000_C00, 2’h0}. PCI/X bits AD[17:13] define the lookup index (that is, what page the access is mapped to), in order to hit page 6, these bits need to be 5’h06. The remaining PCI/X bits AD[12:0] define the address offset, which in this case address the entire 8-Kbyte page of memory. The registers initialized thus far in this example allow the PCI/X transaction to be mapped to the Processor bus.

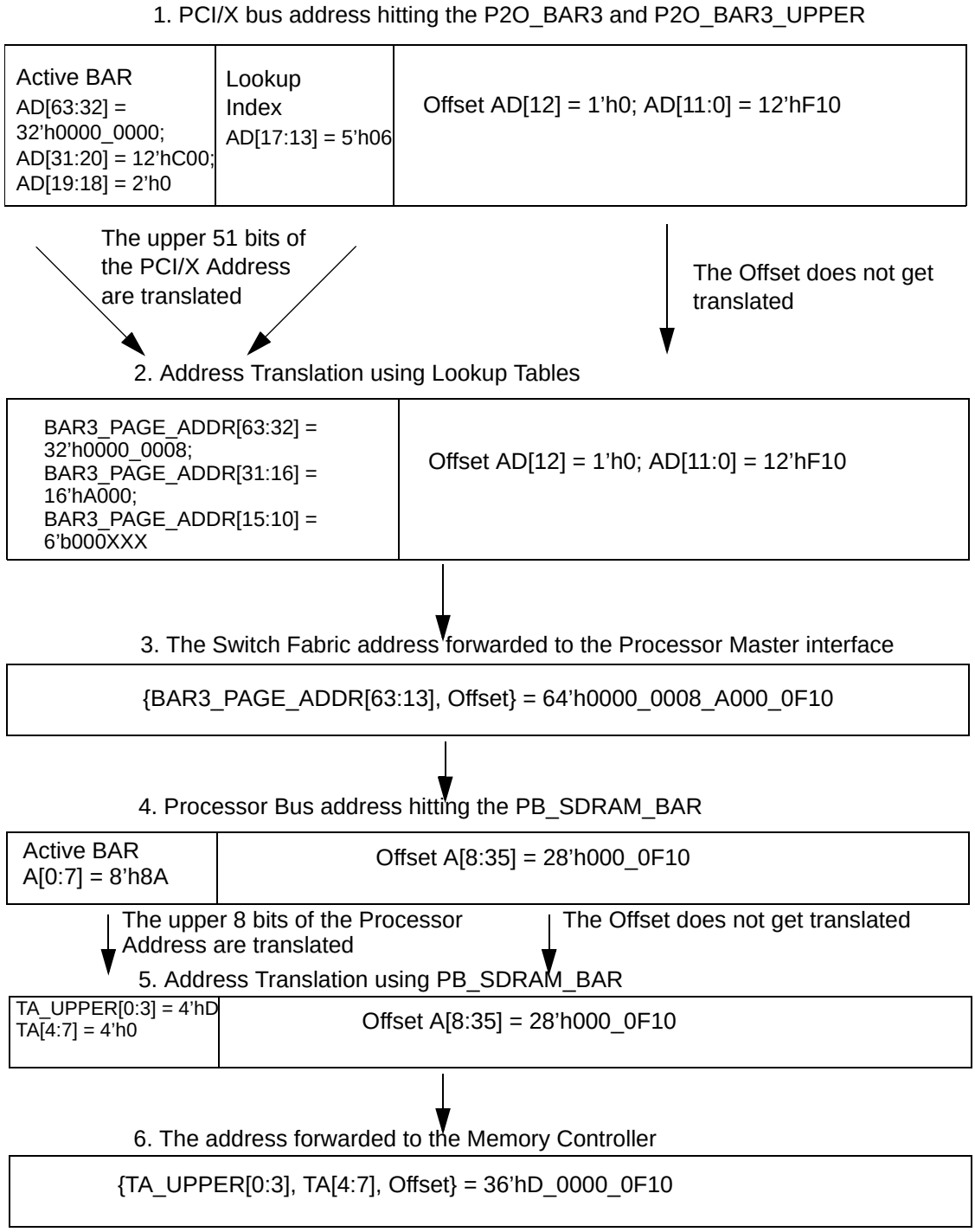
An additional register needs to be initialized to allow for the Processor bus address to be forwarded onto memory. In this example a 256-Mbyte window is configured using the Tsi108/Tsi109 PB_SDRAM_BAR2 register, all processor bus accesses that fall within that window are forwarded directly to the Memory Controller and onto memory. The values programmed in the PB_SDRAM_BAR2 for this specific example will forward all processor bus addresses that fall within 36'h8_A000_0000 – 36'h8_AFFF_FFFF directly to the Memory Controller as 36'hD_0000_0000 – 36'hD_0FFF_FFFF.

Figure 13 shows the address translation process resulting from a PCI/X bus address (32'hC000_CF10) hitting the P20_BAR3 and P2O_BAR3_UPPER registers.

Register settings:

- **P2O_PAGE_SIZES:** BAR3_EN = 1; BAR3_NOTRAN = 0; BAR3_SIZE = 5'h03 (256K Window)
- **P2O_BAR3:** BA[31:20] = 16'hC00, BA[19:18] = 2'h0, BA[17:15] = 3'bXXX; PRFTCH = 1; TYPE = 2'h2; IO_MODE = 0
- **P2O_BAR3_UPPER:** BA[63:32] = 32'h0000_0000
- **P2O_BAR3_LUT6:** BAR3_PAGE_ADDR[31:16] = 16'hA000, BAR3_PAGE_ADDR[15:10] = 6'b000XXX; BAR3_DESTID = 4'b0010 (Processor Bus Master Interface)
- **PB_BAR3_LUT_UPPER6:** BAR3_PAGE_ADDR[31:0] = 32'h0000_0008
- **PB_SDRAM_BAR2:** BA = 4'hA; TA = 4'h0; ATE = 1; BA_UPPER = 4'h8; TA_UPPER = 4'hD; SIZE = 4'h0 (256M); WR_PRTC = 0; EN = 1

Figure 13: Snoop Path Address Translation Process, 32-bit PCI/X Address



2.8. PCI/X to HLP

In order to communicate from the PCI/X bus through the Switch Fabric to external peripherals through the HLP Interface, the following registers must be initialized: a PCI Base Address Register (P2O_BARx and P2O_BARx_UPPER), the PCI Page Size Register (P2O_PAGE_SIZES), as well as the associated PCI BAR Lower (P2O_BARx_LUTy) and Upper (P2O_BARx_LUT_UPPERy) Lookup Table Address Registers. A PCI BAR defines an address range (in PCI/X bus address space) where the Tsi108/Tsi109 responds to a PCI/X bus initiator's request. If the PCI/X bus initiator's request falls within one of the Tsi108/Tsi109's PCI base address ranges, the transaction is forwarded through to the Switch Fabric. The Tsi108/Tsi109 supports two unique PCI BARs: P2O_BAR2 (offset 0x018), P2O_BAR2_UPPER (offset 0x01C); and P2O_BAR3 (offset 0x020), P2O_BAR3_UPPER (offset 0x024) (see Tables 27 and 28). The PCI BARs are enabled through writes to the PCI Page Size Register. The size of the individual address window is also configurable within this register (see Tables 29 and 30). The PCI Page Size Register (P2O_PAGE_SIZES) is located in the PCI/X bus register block at offset 0x04C.

In order to direct the request from the Switch Fabric to another Tsi108/Tsi109 interface the address must be translated. The PCI BAR Upper and Lower Lookup Table Address Registers must be programmed to provide the proper address translation to target the HLP Interface. Each of the PCI BARs (P2O_BAR2, P2O_BAR2_UPPER and P2O_BAR3, P2O_BAR3_UPPER) can have up to 32 pages of memory associated with them. The PCI BAR Upper and Lower Lookup Tables make up those 32 pages, and each individual page of memory requires an initialization of a Lower BAR Lookup and possibly an Upper BAR Lookup depending on whether address translation is enabled or disabled in the PCI Page Size Register (see Figure 11 along with Tables 31 and 32). The PCI BAR Upper and Lower Lookup Table Address Registers (P2O_BAR[2:3]_LUT{0:31} and P2O_BAR[2:3]_LUT_UPPER{0:31}) are located in the PCI/X bus register block at offsets 0x500 – 0x6FC.

2.8.1 How to Initialize the PCI/X Bus to HLP Path

The PCI BARs, PCI Page Size Register, PCI BAR Upper and Lower Lookup Table Address Registers can be initialized using PCI/X memory write cycles from the PCI/X bus through the P2O_BAR0 and P2O_BAR0_UPPER Registers, or from the Processor bus through the PB_REG_BAR. Table 13 and Tables 27 through 32 describe the internal register settings associated with mapping PCI/X bus transactions through the Switch Fabric to the HLP Interface and onto external peripherals such as ROM, EEPROM, FLASH, and SRAM.

2.8.2 PCI/X to HLP Example

Example: Since the address translation process involves the same steps for a 32-bit or a 64-bit PCI/X address, only one example applies. The following example shows the translation process for a 32-bit PCI/X address, access to the HLP Interface.

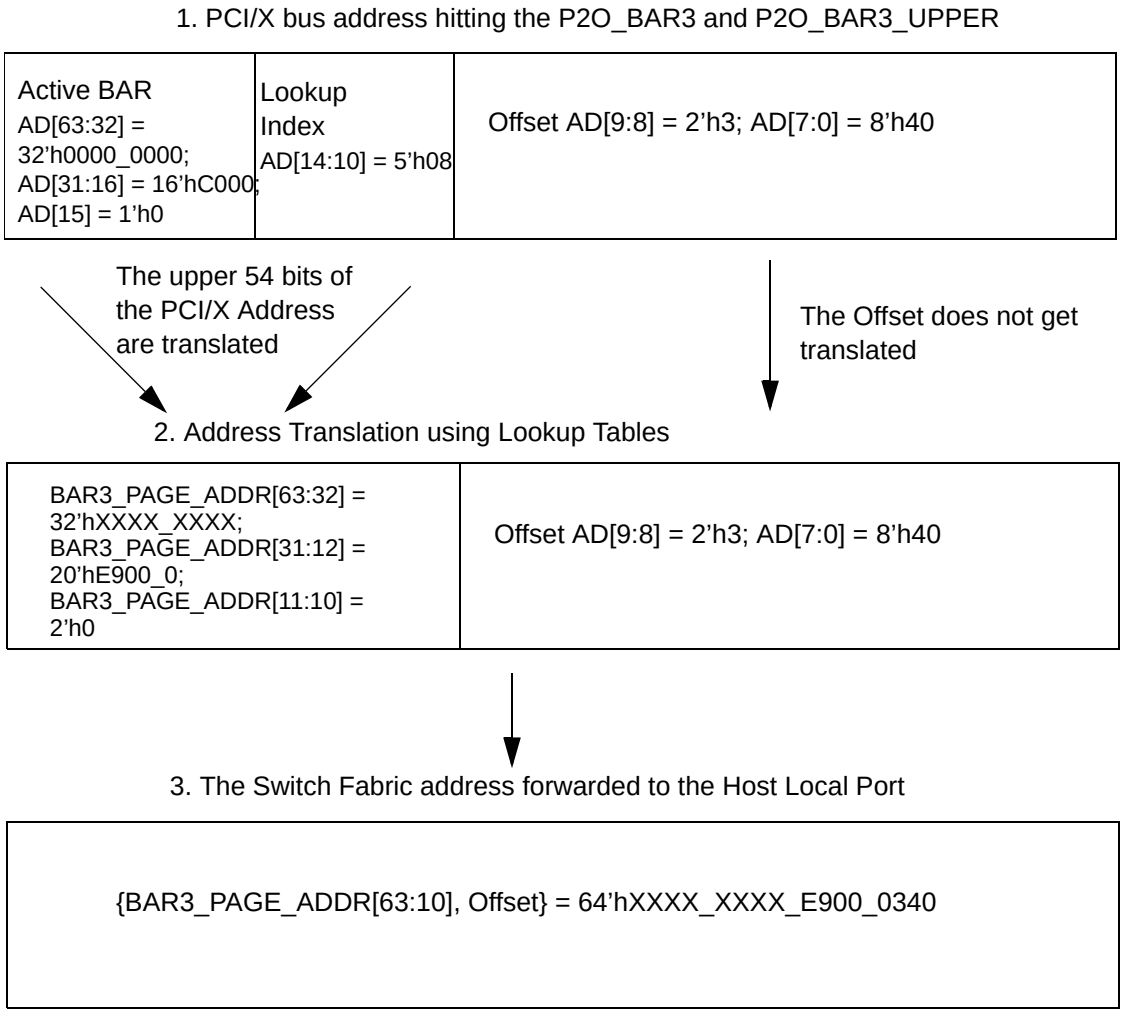
In this example the P2O_PAGE_SIZES window size (BAR3_SIZE) has been programmed to 32 Kbytes, all PCI/X bus addresses that fall within 64'h0000_0000_7000_0000 – 64'h0000_0000_7000_7FFF will be accepted and forwarded through the Switch Fabric as indicated by the values programmed in the P2O_BAR3 and P2O_BAR3_UPPER registers below. Once the transaction has been accepted through the P2O_BAR3 and P2O_BAR3_UPPER registers it is up to the associated lookup tables to determine both the destination port and proper address translation, provided translation is enabled. Since the window size programmed in the P2O_PAGE_SIZES register is 32 Kbytes, that means there are 32, 1-Kbyte pages associated with that address window.

The P2O_BARx_LUT{0..31} and P2O_BARx_LUT_UPPER{0..31} registers must be programmed to determine both the destination port and address translation associated with the 32 pages. In this example only one 1-Kbyte page (page 8) of the 32-Kbyte window is configured to access HLP peripherals, by programming the P2O_BAR3_LUT8 and P2O_BAR3_LUT_UPPER8 registers, page 8 addresses HLP as indicated by the BAR3_DESTID field (see Table 13). Address translation is enabled as indicated by the value in the P2O_PAGE_SIZES[BAR3_NOTRAN] field and all Switch Fabric address accesses that address page 8 will be forwarded to the HLP Interface as 64'hXXXX_XXXX_E900_0000 – 64'hXXXX_XXXX_E900_03FF. Address hits to page 8 are determined as follows using the information provided in Table 30. PCI/X bits AD[63:15] define the active BAR which in this example are defined as {48'h0000_0000_7000, 1'h0}. PCI/X bits AD[14:10] define the lookup index (that is, what page the access is mapped to), in order to hit page 8, these bits need to be 5'h08. The remaining PCI/X bits AD[9:0] define the address offset, which in this case address the entire 1K page of memory. Figure 14 shows the address translation process resulting from a PCI/X address (32'hC000_2340) hitting the P20_BAR3 and P2O_BAR3_UPPER registers.

Register settings:

- **P2O_PAGE_SIZES:** BAR3_EN = 1; BAR3_NOTRAN = 0; BAR3_SIZE = 5'h00 (32K Window)
- **P2O_BAR3:** BA[31:16] = 16'hC000, BA[15] = 0; PRFTCH = 1; TYPE = 2'h2; IO_MODE = 0
- **P2O_BAR3_UPPER:** BA[63:32] = 32'h0000_0000
- **P2O_BAR3_LUT2:** BAR3_PAGE_ADDR[31:12] = 20'hE9000, BAR3_PAGE_ADDR[11:10] = 2'h0; BAR3_DESTID = 4'b0000 (Host Local Port Interface)
- **PB_BAR3_LUT_UPPER2:** BAR3_PAGE_ADDR[31:0] = 32'hXXXX_XXXX (The 64-bit Switch Fabric address is truncated to the 32-bit address presented on the HLP AD pins, therefore the upper translation bits are don't cares.)

Figure 14: HLP Path Address Translation Process, 32-bit PCI/X Address



2.9. Ethernet Controller to Memory

The Ethernet Controller requires a lengthy initialization process and configuration process. For more information on how to program the Ethernet Controller, see the *Tsi108/Tsi109 User Manual*.

IMPORTANT NOTICE AND DISCLAIMER

RENESAS ELECTRONICS CORPORATION AND ITS SUBSIDIARIES (“RENESAS”) PROVIDES TECHNICAL SPECIFICATIONS AND RELIABILITY DATA (INCLUDING DATASHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT OF THIRD-PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for developers who are designing with Renesas products. You are solely responsible for (1) selecting the appropriate products for your application, (2) designing, validating, and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, or other requirements. These resources are subject to change without notice. Renesas grants you permission to use these resources only to develop an application that uses Renesas products. Other reproduction or use of these resources is strictly prohibited. No license is granted to any other Renesas intellectual property or to any third-party intellectual property. Renesas disclaims responsibility for, and you will fully indemnify Renesas and its representatives against, any claims, damages, costs, losses, or liabilities arising from your use of these resources. Renesas' products are provided only subject to Renesas' Terms and Conditions of Sale or other applicable terms agreed to in writing. No use of any Renesas resources expands or otherwise alters any applicable warranties or warranty disclaimers for these products.

(Disclaimer Rev.1.01 Jan 2024)

Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu,
Koto-ku, Tokyo 135-0061, Japan
www.renesas.com

Contact Information

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit www.renesas.com/contact-us/.

Trademarks

Renesas and the Renesas logo are trademarks of Renesas Electronics Corporation. All trademarks and registered trademarks are the property of their respective owners.