Tsi620 Address Translation Application Note

November 30, 2011

6024 Silver Creek Valley Road San Jose, California 95138
Telephone: (408) 284-8200 • FAX: (408) 284-3572
Printed in U.S.A.

GENERAL DISCLAIMER

Integrated Device Technology, Inc. ("IDT") reserves the right to make changes to its products or specifications at any time, without notice, in order to improve design or performance. IDT does not assume responsibility for use of any circuitry described herein other than the circuitry embodied in an IDT product. Disclosure of the information herein does not convey a license or any other right, by implication or otherwise, in any patent, trademark, or other intellectual property right of IDT. IDT products may contain errata which can affect product performance to a minor or immaterial degree. Current characterized errata will be made available upon request. Items identified herein as "reserved" or "undefined" are reserved for future definition. IDT does not assume responsibility for conflicts or incompatibilities arising from the future definition of such items. IDT products have not been designed, tested, or manufactured for use in, and thus are not warranted for, applications where the failure, malfunction, or any inaccuracy in the application carries a risk of death, serious bodily injury, or damage to tangible property. Code examples provided herein by IDT are for illustrative purposes only and should not be relied upon for developing applications. Any use of such code examples shall be at the user's sole risk.

1. Tsi620 Address Translation Application Note

Topics discussed include the following:

- “Tsi620 Address Translation Overview” on page 4
- “Concepts for Mapping Addresses” on page 5
- “Untranslatable Transactions” on page 9
- “Common Control Values” on page 10
- “The RapidIO-to-PCI Translation Path” on page 11
- “PCI to RapidIO Address Mapping” on page 19
- “RapidIO-to-RapidIO Address Mapping” on page 23

Revision History

November 30, 2011, Formal

This version of the document includes a new appendix about Tsi620 “Configuration Scripts”.

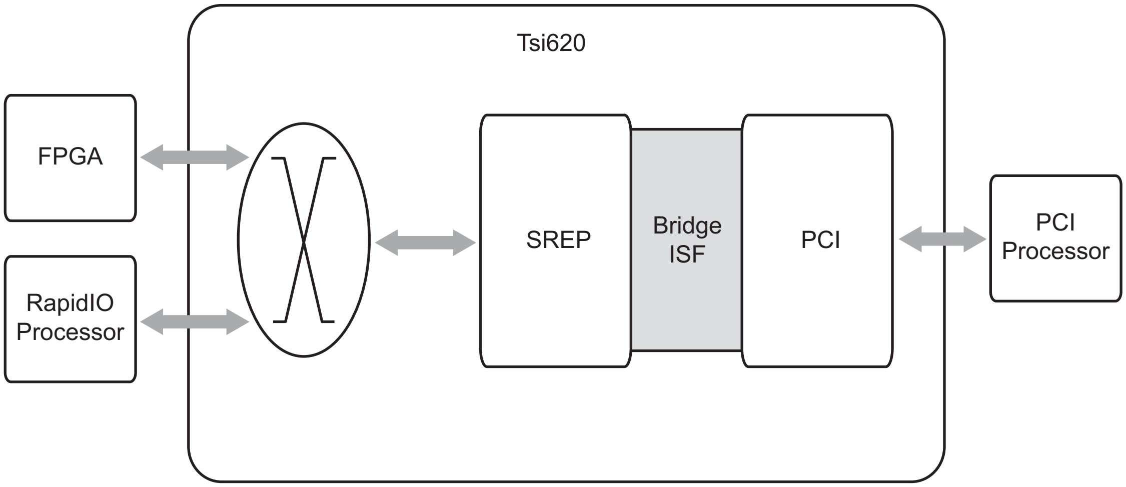
February 17, 2009, Formal

This is the first version of the *Tsi620 Address Translation Application Note*.

1.1 Tsi620 Address Translation Overview

This document discusses the application of Tsi620 address translation based on the system displayed in Figure 1. This system requires bridging from RapidIO to PCI, from PCI to RapidIO, and non-transparent bridging between RapidIO to RapidIO.

Figure 1: Example System



Tsi620 address translation determines how transactions are bridged between the RapidIO and PCI busses. Items that are considered include:

- Transaction type
- Read and write permissions
- Transaction priority
- Logical addresses accepted and used

The register initialization for this system is found in a number of scripts that can be run using the RapidFET tool, which is available from Fabric Embedded Tools. These scripts include:

- Tsi620_SREP_cmnctls.txt
- Tsi620_rio_2_pci_64luts_config.txt
- Tsi620_rio_2_pci_config.txt
- Tsi620_pci_2_rio_config.txt
- Tsi620_rio_2_rio_config.txt

Tip: To obtain a copy of the RapidFET tool, go to www.fetcorp.com.

1.2 Concepts for Mapping Addresses

The concepts used to discuss and design Tsi620 address translation are address translation paths, base address registers (BARs), lookup tables (LUTs), and address spaces. These concepts are explained in the following sections.

1.2.1 Address Translation Paths

An address translation path consists of the source of a transaction, and also the destination of the transaction when the source and destination use different protocols or are constrained differently from each other. The Tsi620 supports three address translation paths (see also Figure 2):

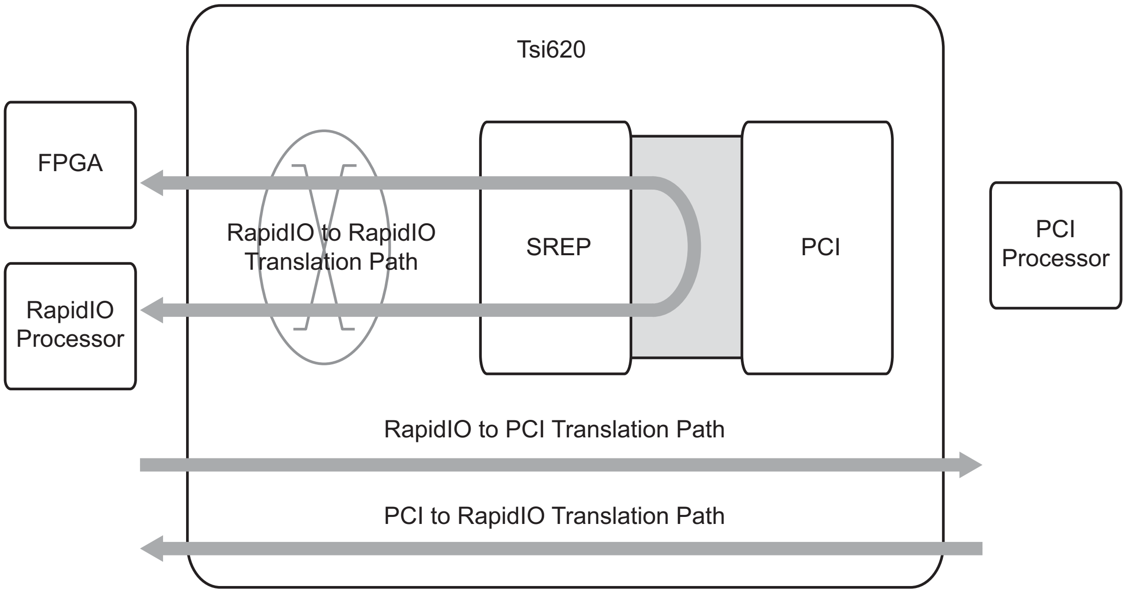
- PCI-to-RapidIO
- RapidIO-to-PCI
- RapidIO-to-RapidIO

The PCI-to-RapidIO translation path controls the translation of PCI memory transactions into RapidIO maintenance, doorbell, and logical I/O packets. PCI devices use this address translation path to configure and communicate with RapidIO devices.

The RapidIO-to-PCI translation path controls the translation of RapidIO memory transactions into PCI configuration, I/O, and memory space transactions. RapidIO devices use this address translation path to configure and communicate with PCI devices.

The RapidIO-to-RapidIO translation path controls the translation of one RapidIO memory transaction into another RapidIO transaction. This translation path restricts how one RapidIO device can communicate with another, or to bridge aspects of the RapidIO protocol that are otherwise incompatible between devices. For example, the RapidIO-to-RapidIO translation path can be used to restrict the memory regions and transaction types that an endpoint can access, or to translate between 8-bit and 16-bit destination IDs.

Figure 2: Address Translation Paths


Tsi620
FPGA
RapidIO Processor
RapidIO to RapidIO Translation Path
SREP
PCI
PCI Processor
RapidIO to PCI Translation Path
PCI to RapidIO Translation Path

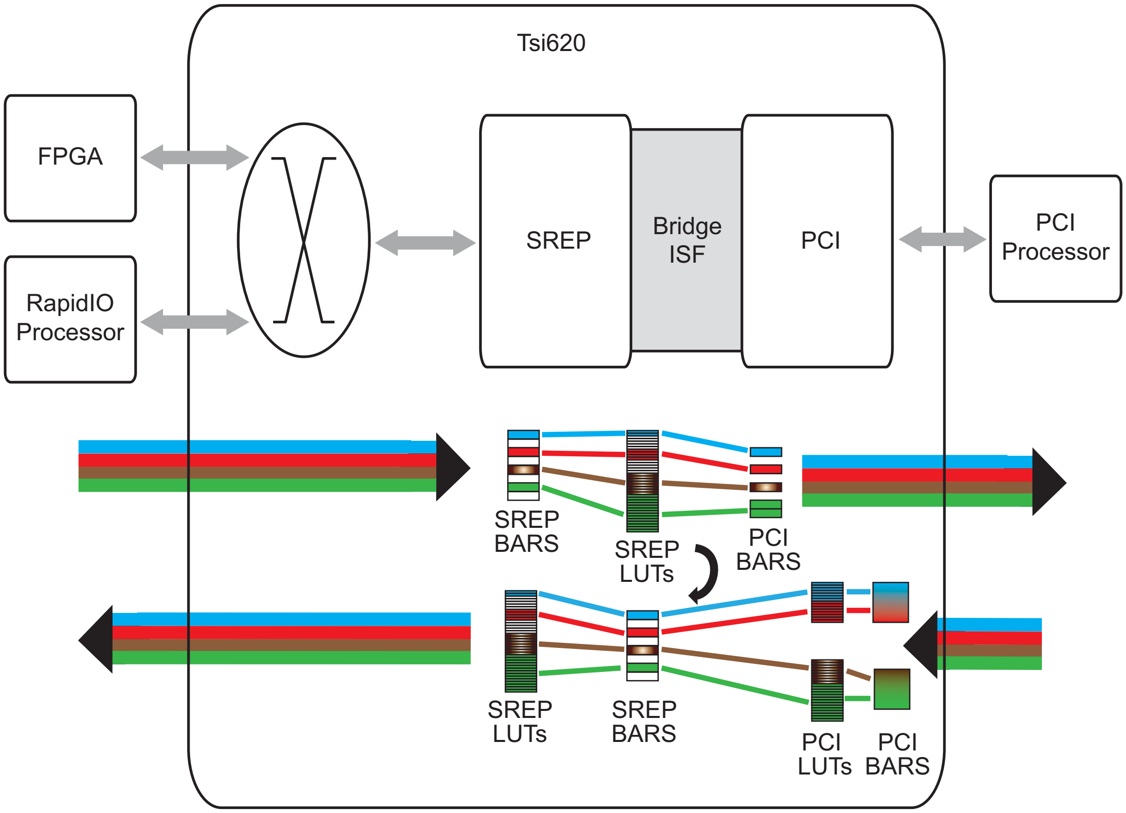
1.2.2 BARs and LUTs

Base address registers (BARs) and lookup tables (LUTs) are used to define the address ranges for translation, and define how the translation is performed (see Figure 3). The Tsi620's Serial RapidIO Endpoint (SREP) and PCI blocks both have two sets of BARs and LUTs: one for the RapidIO-to-PCI address translation path, and one for the PCI-to-RapidIO translation path. The RapidIO-to-RapidIO translation path uses both sets of RapidIO BARs and LUTs.

BARs determine an address range, and are defined as a starting (base) address and a size. There are alignment and size restrictions on most BARs. For example, the SREP BARs must be a power of two in size, permitting a range from 4 KB up to 16 GB.

LUTs control how equally-sized subsections of the address range defined by a BAR are translated. LUTs are associated with a particular BAR. In PCI, 32 LUTs are dedicated to each BAR. In RapidIO, each address path has 256 LUTs that can be distributed among the eight BARs.

Figure 3: BARs and LUTs Locations



The PCI block has fewer BARs and LUTs than the SREP block. In the RapidIO-to-PCI direction, PCI transaction types — for example, configuration transactions — are associated with specific PCI BARs and LUTs. The selection of BARs is therefore fairly easy for the PCI block since designers are constrained to use the BARs dedicated to producing the type of transactions they require. For example, the Configuration Cycle BAR is used to create PCI configuration transactions.

In contrast, while the SREP block has many more BARs and LUTs, only a few are dedicated to specific transaction types. Designers have more flexibility to decide how to allocate BARs and LUTs to address spaces. In the example system the BARs are selected linearly within each address translation path, while the LUTs were chosen more randomly.

1.2.3 Address Spaces

Address translation is performed through several different address spaces (see Figure 4). The address spaces which a transaction passes through are unique to the address translation path for the transaction.

The PCI-to-RapidIO translation path consists of three address spaces:

- Tsi620 PCI Target Address Space – This is the Tsi620 PCI BAR that a PCI device must address in order to communicate with a RapidIO endpoint.
- Tsi620 SREP Bridge ISF Address Space – Transactions that hit the Tsi620 PCI Target Address Space are translated to the Tsi620 SREP Bridge ISF space.
- RapidIO Endpoint Target Address Space – The SREP translates transactions in the SREP Bridge ISF Address Space to a RapidIO Endpoint's Target Address Space. The RapidIO Endpoint's Target Address Space includes the RapidIO DestID and memory address.

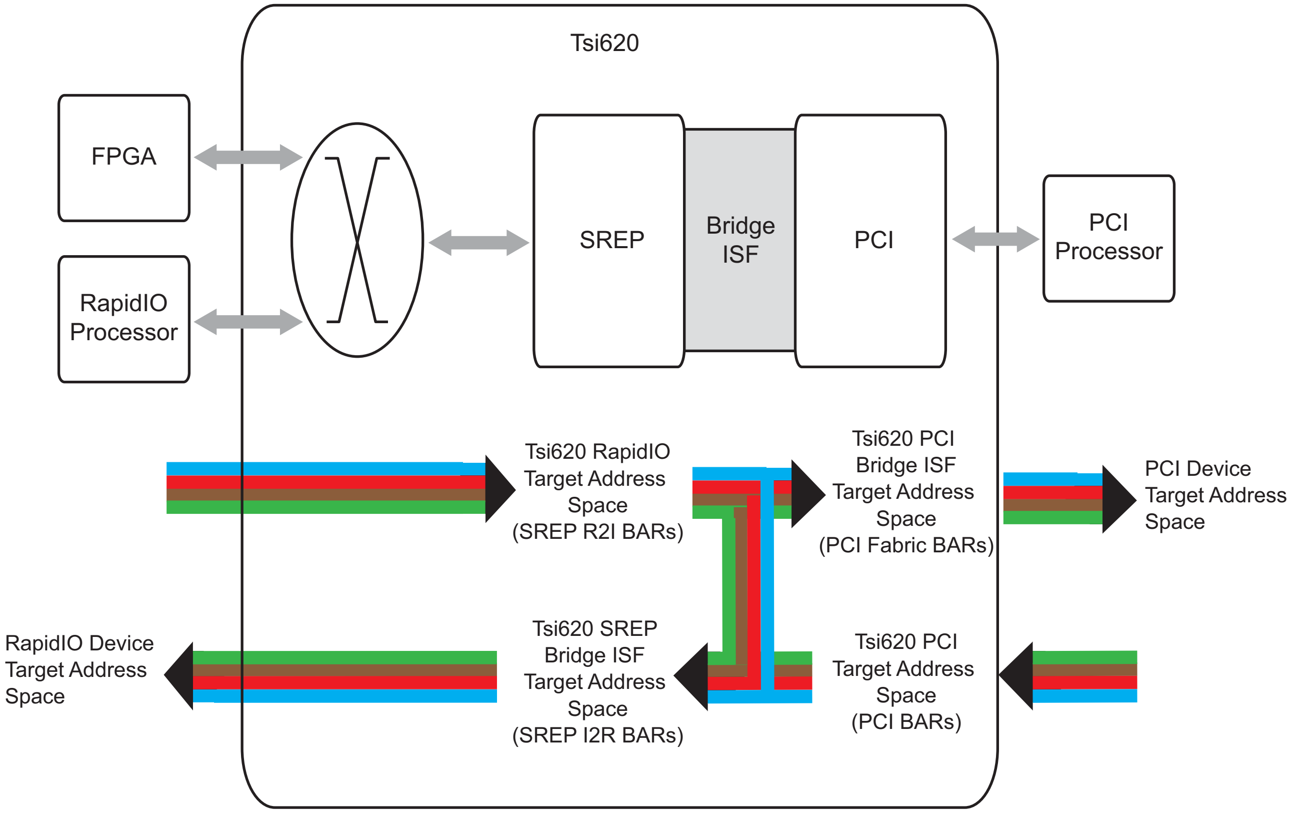
The RapidIO-to-PCI translation path consists of three address spaces:

- Tsi620 RapidIO Target Address Space – This is the SREP R2I BAR that a RapidIO device must address in order to communicate with a PCI device.
- Tsi620 PCI Bridge ISF Target Address Space – The Tsi620 RapidIO Target Address Space is translated to Tsi620 PCI Bridge ISF Target Address Space.
- PCI Device Target Address Space – The PCI block is configured to translate transactions in the PCI Bridge ISF Target Address Space to a PCI Device's Target Address Space. The PCI Device's Target Address Space is defined by the type of PCI transaction determined by this translation, along with the memory address permitted for the transaction.

The RapidIO-to-RapidIO translation path also consists of three address spaces. These address spaces are a part of the PCI-to-RapidIO and RapidIO-to-PCI translation paths:

- Tsi620 RapidIO Target Address Space – This is the SREP R2I BAR that a RapidIO device must address to communicate with a RapidIO device.
- Tsi620 SREP Bridge ISF address space – Transactions that hit the Tsi620 RapidIO Target Address Space are translated to the Tsi620 SREP Bridge ISF space.
- RapidIO Endpoint Target Address Space – The SREP translates transactions in the SREP Bridge ISF address space to a RapidIO Endpoint's Target Address space. The RapidIO Endpoint's Target Address Space includes a RapidIO destination ID and memory address.

Figure 4: Address Spaces for Address Translation


Tsi620
FPGA
RapidIO Processor
SREP
Bridge ISF
PCI
PCI Processor
Tsi620 RapidIO Target Address Space (SREP R2I BARs)
Tsi620 PCI Bridge ISF Target Address Space (PCI Fabric BARs)
PCI Device Target Address Space
RapidIO Device Target Address Space
Tsi620 SREP Bridge ISF Target Address Space (SREP I2R BARs)
Tsi620 PCI Target Address Space (PCI BARs)

The Bridge ISF Address Spaces is 64 bits in size, allowing system designers a large range of options for choosing address ranges when implementing translation. The PCI bus Memory address space is also 64 bits; however, for simplicity, in our example system we have constrained the PCI addresses to 32 bits. The RapidIO Address Spaces in the example are all 34 bits.

1.3 Untranslatable Transactions

Some PCI transactions cannot be bridged to RapidIO. Similarly, some RapidIO transactions cannot be bridged to PCI. These untranslatable transactions are described in the following sections.

1.3.1 Untranslatable PCI Transactions

PCI configuration transactions are terminated by the Tsi620 PCI interface. They cannot be translated to RapidIO transactions. PCI I/O transactions cannot be accepted by the Tsi620 PCI interface. They cannot be bridged to RapidIO.

The Tsi620 can generate PCI I/O read and write transactions but cannot accept these transactions.

1.3.2 Untranslatable RapidIO Transactions

RapidIO maintenance read and write transactions are terminated by the SREP. They cannot be translated to PCI transactions.

RapidIO port-write and doorbell packets are buffered separately in the SREP. Port-write and doorbell packet reception can cause a PCI interrupt, or trigger other event notification mechanisms. However, port-write and doorbell packets cannot be bridged to PCI transactions.

RapidIO message, data streaming, flow control, and atomic transactions are not supported by SREP. They too cannot be translated to PCI transactions.

1.4 Common Control Values

SREP contains control values that affect its behavior as a RapidIO target and a RapidIO master. These are discussed in the following sections.

The Tsi620_SREP_cmnctls.txt script is provided to configure the control values that are necessary for translation. For more information, see the "Configuration Scripts" appendix.

1.4.1 SREP RapidIO Target Controls

SREP has a variety of control registers that are applied to both the destination ID and source ID of request packets. Example control values are listed in Table 1. For more information on these control values, see the "SREP Transport Layer "chapter of the *Tsi620 User Manual*.

SREP also supports the standard SREP_RIO_LIO_REGACC register, which allows all Tsi620 registers to be accessed using NREAD and NWRITE/NWRITE_R transactions.

Operation of the SREP_RIO_LIO_REGACC register has not been incorporated into the example system because the register accesses are not translated/bridged.

The destination and source ID control values listed in the table also apply to NREAD and NWRITE/NWRITE_R packets whose address is in the SREP_RIO_LIO_REGACC range.

Table 1: RapidIO Destination and Source ID Control Registers

Tsi620 Register Address	Register Name	Example Register Value	Description
2_0060	SREP_RIO_DEST_ID	0x00020002	SourceID for all SREP RapidIO request packets. Destination ID to be used by all RapidIO transactions sent to the Tsi620. In this example, the 8- and 16-bit destination IDs are both set to '2'.
2_0400	SREP_DESTID_CHK_CTL	0x0000000A	Determines which packet destination IDs will be accepted by the Tsi620. In this example, it is set to accept only the 8- or 16-bit destination ID '2' programmed in the SREP_RIO_DEST_ID register.

Table 1: RapidIO Destination and Source ID Control Registers *(Continued)*

Tsi620 Register Address	Register Name	Example Register Value	Description
2_0404	SREP_DESTID_LG_CTL	0x00000000	Determines what range of 16-bit destination IDs can be accepted. In this example, it is disabled.
2_0408	SREP_REG_ACC_CTL	0x4C385336	Determines what source IDs can write to SREP registers. In this example, only the 16-bit source ID of the RapidIO host processor is allowed to write to the SREP registers.
2_040C	SREP_REG_ACC_SMCHK_CTL	0x00000000	Determines which 8-bit source IDs can write to SREP registers. The lower 16 bits are the sourceID of the last transaction to write to SREP registers from RapidIO. In this example, all 8-bit source IDs are rejected to prevent the FPGA from changing SREP register values.
2_0410	SREP_REG_ACC_LRGCHK_CTL	0xFFFF0100	Determines which 16-bit source IDs can write to SREP registers. This is restricted to the RapidIO processor in this example system. In this example, only the 16-bit source ID of the RapidIO processor (0x0100) is allowed to write to the SREP registers.

1.4.2 SREP RapidIO Master Controls

The SREP_RIO_PORT_GEN_CTL[MAST_EN] bit must be set to 1 before SREP can generate any RapidIO request packets.

1.5 The RapidIO-to-PCI Translation Path

RapidIO NREAD and NWRITE/NWRITE_R/SWRITE packets can follow the RapidIO-to-PCI translation path. RapidIO NREAD, NWRITE, and NWRITE_R packets can be bridged to PCI configuration or I/O transactions. RapidIO NREAD, NWRITE, NWRITE_R, and SWRITE packets can be bridged to PCI memory transactions.

TIP

The register accesses required to configure this example of the Tsi620 RapidIO-to-PCI translation path are found in the script named Tsi620_rio_2_pci_config.txt.

In the following example, it is assumed that the RapidIO processor needs to generate PCI configuration, I/O, and memory cycles. Table 2 lists the address spaces and address ranges that the RapidIO device must be able to access on the PCI bus for this example.

Table 2: Tsi620 PCI Device Target Address Space

PCI Device Target Address Space	Address Range	Description
Configuration	N/A	PCI configuration transactions are used to access PCI device registers. Configuration transactions do not operate in the same bus address space as PCI I/O and Memory transactions.
I/O	0x0000_0000 0x0000_FFFF	PCI I/O transactions occur in up to the first 64 KB of address space.
Non-Prefetchable Memory	0x0020_0000 0x003F_FFFF	In this example, 2 MB of non-prefetchable memory must be accessible by the RapidIO processor.
Prefetchable Memory	0x3000_0000 0x33FF_FFFF 0x4000_0000 0x43FF_FFFF	The RapidIO processor must be able to access two separate 64-MB chunks of memory on the PCI bus. Each 64-KB chunk of memory is divided into four 16-MB chunks. The first and third 16-MB chunks can only be written to, while the second and fourth 16-MB chunks can only be read from.

Every transaction type and/or memory range in the PCI Device Memory Space must have a corresponding memory range in the Tsi620 PCI Bridge ISF Target Address Space. The Tsi620 PCI Interface has separate BARs to support PCI configuration, I/O, non-prefetchable memory, and prefetchable memory transactions. The mapping for this example from the Tsi620 PCI Bridge ISF Target Address Space to the PCI Device Memory Map is displayed in Table 3.

Table 3: Tsi620 PCI Fabric Target Address Space Example Configuration

PCI Device Target Address Space	Tsi620 PCI Bridge ISF Target Address Space	Register Name	Tsi620 Register Address	Example Register Value
Configuration	0x0000_0000 0x0003_FFFF	PFAB_BAR0	0x2_2204	0x0000_0001
		PFAB_BAR0_UPPER	0x2_2208	0x0000_0000
I/O	0x0004_0000 0x0004_FFFF	PFAB_IO	0x2_220C	0x0004_0001
		PFAB_IO_UPPER	0x2_2210	0x0000_0000
Non-Prefetchable Memory 0x0020_0000 0x003F_FFFF	0x2000_0000 0x201F_FFFF	PFAB_MEM32	0x2_2214	0x2001_0000
		PFAB_MEM32_REMAP	0x2_2218	0x0020_0000
		PFAB_MEM32_MASK	0x2_221C	0xFFE0_0000

Table 3: Tsi620 PCI Fabric Target Address Space Example Configuration *(Continued)*

PCI Device Target Address Space	Tsi620 PCI Bridge ISF Target Address Space	Register Name	Tsi620 Register Address	Example Register Value
First Prefetchable Memory 0x3000_0000 0x33FF_FFFF	0x1_0000_0000 0x1_03FF_FFFF	PFAB_PFM3	0x2_2220	0x0001_0001
		PFAB_PFM3_REMAP_UPPER	0X2_2224	0x0000_0000
		PFAB_PFM3_REMAP_LOWER	0x2_2228	0x3000_0000
		PFAB_PFM3_MASK	0x2_222C	0xFC00_0FFF
Second Prefetchable Memory 0x4000_0000 0x43FF_FFFF	0x2_0000_0000 0x2_03FF_FFFF	PFAB_PFM4	0x2_2230	0x0001_0002
		PFAB_PFM4_REMAP_UPPER	0X2_2234	0x0000_0000
		PFAB_PFM4_REMAP_LOWER	0x2_2238	0x4000_0000
		PFAB_PFM4_MASK	0x2_223C	0xFC00_0FFF

Every address in the Tsi620 PCI Bridge ISF Target Address Space must have a corresponding address range in the Tsi620 RapidIO Target Address Space. To minimize the use of SREP BARs and LUTs, the four PCI Device Target Address Spaces can be consolidated into two Tsi620 RapidIO Target Address Spaces, one for prefetchable memory and the other for non-prefetchable memory:

- 128 MB of Prefetchable memory for memory transactions
- 4 MB of Non-Prefetchable memory, which includes 256 KB for Configuration space, 64 KB for I/O space, and 2 MB for Non-Prefetchable memory space

There are two ways to implement the Tsi620 RapidIO Target Address Space non-prefetchable memory space. The first example, found in the Tsi620_rio_2_pci_config.txt script and illustrated in Figure 4, uses three BARs and three LUT entries. By using three BARs, the Tsi620 RapidIO Target Address Space contains the exact amount of required address space.

The second example, found in the Tsi620_rio_2_pci_64luts_config.txt script and illustrated in Figure 5, uses only one BAR but 64 LUT entries. In this script, the Tsi620 RapidIO Target Address Space contains a single 4-MB block of memory, which is distributed among the Tsi620 PCI Bridge ISF Address Spaces using LUT entries. The 4-MB size is necessary because SREP BARs must be a power of 2 in size. The 4-MB space is therefore larger than what is required. In some applications, this extra memory space could be used as a second route to get to the Tsi620 PCI Bridge ISF Address Spaces. In this example, however, read and write permission is denied to the extra memory space.

The register settings required for the Tsi620 RapidIO Target Address Space using four BARs for the non-prefetchable memory space are displayed in Table 4.

⚠ The register accesses must be performed in the order in which they occur in the table. Otherwise, the LUT entries programmed may not correspond to the LUT entries used by the BAR.

Table 4: Tsi620 RapidIO Target Address Space Example Implemented With 4 BARs

PCI Device Target Address Space	Tsi620 Fabric Target Address Space	Tsi620 RapidIO Target Address Space	Tsi620 Register Name and Offset	Example Register Value
Configuration Memory	0x0000_0000 0x0003_FFFF	0x1_0020_0000 0x1_0023_FFFF	SREP_R2I_BAR0_LUT_CSR (0x2_0500)	0x0000_0600
			SREP_R2I_BAR0_LOWER (0x2_0508)	0x0020_0011
			SREP_R2I_LUT_PTY_CTL (0x2_0590)	0x0000_0001
			SREP_R2I_LUT_UPPER (0x20_05A0)	0x0000_0000
			SREP_R2I_LUT_LOWER (0x2_05AC)	0x0000_0013
I/O Memory	0x0004_0000 0x0004_FFFF	0x1_0024_0000 0x1_0024_FFFF	SREP_R2I_BAR1_LUT_CSR (0x2_0510)	0x0000_0400
			SREP_R2I_BAR1_LOWER (0x2_0518)	0x0024_0011
			SREP_R2I_LUT_PTY_CTL (0x2_0590)	0x0000_0002
			SREP_R2I_LUT_UPPER (0x20_05A0)	0x0000_0000
			SREP_R2I_LUT_LOWER (0x2_05AC)	0x0010_0013
Non-Prefetchable Memory 0x0020_0000 0x003F_FFFF	0x2000_0000 0x201F_FFFF	0x1_0000_0000 0x1_001F_FFFF	SREP_R2I_BAR2_LUT_CSR (0x2_0520)	0x0000_0900
			SREP_R2I_BAR2_LOWER (0x2_0528)	0x0000_0011
			SREP_R2I_LUT_PTY_CTL (0x2_0590)	0x0000_0000
			SREP_R2I_LUT_UPPER (0x20_05A0)	0x0000_0000
			SREP_R2I_LUT_LOWER (0x2_05AC)	0x0020_0013

Table 4: Tsi620 RapidIO Target Address Space Example Implemented With 4 BARs *(Continued)*

PCI Device Target Address Space	Tsi620 Fabric Target Address Space	Tsi620 RapidIO Target Address Space	Tsi620 Register Name and Offset	Example Register Value
First and Second Prefetchable Memory 0x3000_0000 0x33FF_FFFF 0x4000_0000 0x43FF_FFFF	0x1_0000_0000 0x1_03FF_FFFF 0x2_0000_0000 0x2_03FF_FFFF	0x2_0000_0000 0x2_07FF_FFFF	SREP_R2I_BAR3_LUT_CSR (0x2_0530)	0x0004_0F03
			SREP_R2I_BAR3_LOWER (0x2_0538)	0x0000_0012
			SREP_R2I_LUT_PTY_CTL (0x2_0590)	0x8000_0040
First Prefetchable Memory 0x3000_0000 0x33FF_FFFF	0x1_0000_0000 0x1_00FF_FFFF (Write Only)	0x2_0000_0000 0x2_00FF_FFFF (Write Only)	SREP_R2I_LUT_UPPER (0x20_05A0)	0x0000_0001
			SREP_R2I_LUT_LOWER (0x2_05AC)	0x0000_0012
	0x1_0100_0000 0x1_01FF_FFFF (Read Only)	0x2_0100_0000 0x2_01FF_FFFF (Read Only)	SREP_R2I_LUT_UPPER (0x20_05A0)	0x0000_0001
			SREP_R2I_LUT_LOWER (0x2_05AC)	0x0100_0011
	0x1_0200_0000 0x1_02FF_FFFF (Write Only)	0x2_0200_0000 0x2_02FF_FFFF (Write Only)	SREP_R2I_LUT_UPPER (0x20_05A0)	0x0000_0001
			SREP_R2I_LUT_LOWER (0x2_05AC)	0x0200_0012
	0x1_0300_0000 0x1_03FF_FFFF (Read Only)	0x2_0300_0000 0x2_03FF_FFFF (Read Only)	SREP_R2I_LUT_UPPER (0x20_05A0)	0x0000_0001
			SREP_R2I_LUT_LOWER (0x2_05AC)	0x0300_0011

Table 4: Tsi620 RapidIO Target Address Space Example Implemented With 4 BARs *(Continued)*

PCI Device Target Address Space	Tsi620 Fabric Target Address Space	Tsi620 RapidIO Target Address Space	Tsi620 Register Name and Offset	Example Register Value
Second Prefetchable Memory 0x4000_0000 0x43FF_FFFF	0x2_0000_0000 0x2_00FF_FFFF (Write Only)	0x2_0400_0000 0x2_04FF_FFFF (Write Only)	SREP_R2I_LUT_UPPER (0x20_05A0)	0x0000_0002
			SREP_R2I_LUT_LOWER (0x2_05AC)	0x0000_0012
	0x2_0100_0000 0x2_01FF_FFFF (Read Only)	0x2_0500_0000 0x2_05FF_FFFF (Read Only)	SREP_R2I_LUT_UPPER (0x20_05A0)	0x0000_0002
			SREP_R2I_LUT_LOWER (0x2_05AC)	0x0100_0011
	0x2_0200_0000 0x2_02FF_FFFF (Write Only)	0x2_0600_0000 0x2_06FF_FFFF (Write Only)	SREP_R2I_LUT_UPPER (0x20_05A0)	0x0000_0002
			SREP_R2I_LUT_LOWER (0x2_05AC)	0x0200_0012
	0x2_0300_0000 0x2_03FF_FFFF (Read Only)	0x2_0700_0000 0x2_07FF_FFFF (Read Only)	SREP_R2I_LUT_UPPER (0x20_05A0)	0x0000_0002
			SREP_R2I_LUT_LOWER (0x2_05AC)	0x0300_0012

The register settings required for the Tsi620 RapidIO Target Address Space example using one BAR for the non-prefetchable memory space are displayed in Table 5. There are many sequential accesses to the SREP_R2I_LUT_UPPER and SREP_R2I_LUT_LOWER registers required to complete the programming. These accesses demonstrate the SREP_R2I_LUT_PTY_CTL[AUTO_INC] feature, which allows a series of writes to the SREP_R2I_LUT_UPPER and SREP_R2I_LUT_LOWER registers to initialize sequential LUT entries without repetitively writing to the SREP_R2I_LUT_PTY_CTL[LUT_INDEX] field.

The register accesses must be performed in the order in which they occur in the table. Otherwise, the LUT entries programmed may not correspond to the LUT entries used by the BAR.

A complete list of register writes appears in the Tsi620_rio_2_pci_64luts_config.txt script.

Table 5: Tsi620 RapidIO Target Address Space Example for Non-Prefetchable Memory Space Implemented with 1 BAR

Tsi620 RapidIO Target Address Space	Tsi620 Register Name and Offset	Example Register Value	Tsi620 Fabric Target Address Spaces	PCI Device Target Address Space
4-MB window for PCI Configuration, I/O and Non-Prefetchable Memory 0x1_0000_0000 0x1_003F_FFFF	SREP_R2I_BAR0_LUT_CSR (0x2_0500)	0x0000_0A06	-	
	SREP_R2I_BAR0_LOWER (0x2_0508)	0x0000_0011		
	SREP_R2I_LUT_PTY_CTL (0x2_0590)	0x8000_0000		
Non-Prefetchable Memory LUT Entries for PCI Configuration Space 0x1_0000_0000 0x1_0003_FFFF	SREP_R2I_LUT_UPPER (0x20_05A0)	0x0000_0000	0x0000_0000 0x0003_FFFF	Configuration
	SREP_R2I_LUT_LOWER (0x2_05AC)	0x0000_0013		
	SREP_R2I_LUT_UPPER (0x20_05A0)	0x0000_0000		
	SREP_R2I_LUT_LOWER (0x2_05AC)	0x0001_0013		
	SREP_R2I_LUT_UPPER (0x20_05A0)	0x0000_0000		
	SREP_R2I_LUT_LOWER (0x2_05AC)	0x0002_0013		
	SREP_R2I_LUT_UPPER (0x20_05A0)	0x0000_0000		
	SREP_R2I_LUT_LOWER (0x2_05AC)	0x0003_0013		
Non-Prefetchable Memory LUT Entries for PCI Configuration Space 0x1_0004_0000 0x1_0004_FFFF	SREP_R2I_LUT_UPPER (0x20_05A0)	0x0000_0000	0x0004_0000 0x0004_FFFF	I/O
	SREP_R2I_LUT_LOWER (0x2_05AC)	0x0004_0013		

Table 5: Tsi620 RapidIO Target Address Space Example for Non-Prefetchable Memory Space Implemented with 1 BAR *(Continued)*

Tsi620 RapidIO Target Address Space	Tsi620 Register Name and Offset	Example Register Value	Tsi620 Fabric Target Address Spaces	PCI Device Target Address Space
Non-Prefetchable Memory LUT Entries for UNUSED Memory Space 0x1_0005_0000 0x1_001F_FFFF	SREP_R2I_LUT_UPPER (0x20_05A0)	0x0000_0000	Unused Space 0x0_0005_0000 0x0_001F_FFFF	N/A
	SREP_R2I_LUT_LOWER (0x2_05AC)	0x0005_0010		
	SREP_R2I_LUT_UPPER (0x20_05A0)	0x0000_0000		
	SREP_R2I_LUT_LOWER (0x2_05AC)	0x0005_0010		
	. . .			
	SREP_R2I_LUT_UPPER (0x20_05A0)	0x0000_0000		
	SREP_R2I_LUT_LOWER (0x2_05AC)	0x001F_0010		
Non-Prefetchable Memory LUT Entries for PCI Non-Prefetchable Memory Space 0x1_0020_0000 0x1_003F_FFFF	SREP_R2I_LUT_UPPER (0x20_05A0)	0x0000_0000	0x2000_0000 0x201F_FFFF	Non-Prefetchable Memory 0x0020_0000 0x003F_FFFF
	SREP_R2I_LUT_LOWER (0x2_05AC)	0x0020_0010		
	SREP_R2I_LUT_UPPER (0x20_05A0)	0x0000_0000		
	SREP_R2I_LUT_LOWER (0x2_05AC)	0x0021_0010		
	. . .			
	SREP_R2I_LUT_UPPER (0x20_05A0)	0x0000_0000		
	SREP_R2I_LUT_LOWER (0x2_05AC)	0x003F_0010		

1.6 PCI to RapidIO Address Mapping

In the following example, it is assumed that the PCI processor needs to be able to generate RapidIO Maintenance, Doorbell, and Logical I/O (NREAD/NWRITE/NWRITE_R/SWRITE) packets. Table 6 lists the address spaces and ranges that the PCI device must access on the RapidIO microprocessor.

Table 6: RapidIO Endpoint Target Address Space

RapidIO Endpoint Target Address Space	RapidIO Endpoint Target Address Range	Description
Maintenance	0x0000_0000 0x0003_FFFF	In this example, the RapidIO endpoint has 256 KB of register space which the PCI processor must be able to access. Note that RapidIO Maintenance address space is completely separate from the Logical I/O address space.
Doorbell	N/A	RapidIO Doorbell transactions do not have a memory address.
Logical I/O Memory	0x4000_0000 0x5FFF_FFFF 0x6000_0000 0x7FFF_FFFF	In this example, the 1 GB memory range is divided into two 512 MB chunks. The first chunk is accessed using NREADs and NWRITE/SWRITE packets. The second chunk is accessed using NREADs and NWRITE_Rs.

Each RapidIO Endpoint Target Address Space must have a corresponding Tsi620 SREP Bridge ISF Target Address Space associated with it. One Tsi620 SREP I2R BAR is necessary for Maintenance space, one for Doorbells, and one for the Logical I/O Memory space.

The register settings required to implement the Tsi620 SREP Bridge ISF address space are found in Table 7. Note that the BAR that implements the Logical I/O Memory space requires two LUT entries. The first LUT entry translates Bridge ISF writes to RapidIO SWRITEs/NWRITES, while the second translates Bridge ISF writes to RapidIO NWRITE_Rs. Also note that the Doorbell BAR does not have any LUTs associated with it.

TIP

As the NWRITE_R responses will not be propagated to the PCI bus, software can determine when all outstanding NWRITE_Rs have completed by polling the SREP_R2I_EVENT_STATUS.MNWR_NO_ACK bit.

Table 7: Tsi620 RapidIO Bridge ISF Target Address Space Example

RapidIO Device Transaction Target Address Space	Tsi620 SREP Bridge ISF Target Address Space	Tsi620 Register Name	Tsi620 Register Address	Example Register Value
Maintenance	0x4_FFFC_0000 0x4_FFFF_FFFF	SREP_I2R_BAR0_LUT_CSR	0x2_0600	0x0000_0600
		SREP_I2R_BAR0_UPPER	0x2_0604	0x0000_0004
		SREP_I2R_BAR0_LOWER	0x2_0608	0xFFFC_0001
		SREP_I2R_LUT_PTY_CTL	0x2_06A0	0x0000_0000
		SREP_I2R_LUT_TA_UPPER	0x2_06A4	0x01C1_0100
		SREP_I2R_LUT_TA_LOWER	0x2_06AC	0x0000_00A0
		SREP_I2R_LUT_TA_RIO_PARMS	0x2_06BC	0x0000_8081
Doorbell	0x4_FFFB_F000 0x4_FFFB_FFFF	SREP_I2R_DB_BAR_UPPER	0x2_0690	0x0000_0004
		SREP_I2R_DB_BAR_LOWER	0x2_0694	0xFFFB_F001
Logical I/O Memory 0x0_4000_0000 0x0_7FFF_FFFF	0x4_C000_0000 0x4_FFFF_FFFF	SREP_I2R_BAR1_LUT_CSR	0x2_0610	0x0001_1201
		SREP_I2R_BAR1_UPPER	0x2_0614	0x0000_0004
		SREP_I2R_BAR0_LOWER	0x2_0618	0xC000_0001
	Logical I/O Memory (NREAD, SWRITE) 0x4_C000_0000 0x4_DFFF_FFFF	SREP_I2R_LUT_PTY_CTL	0x2_06A0	0x8000_0001
		SREP_I2R_LUT_TA_UPPER	0x2_06A4	0x00D1_0100
		SREP_I2R_LUT_TA_LOWER	0x2_06AC	0x4000_0020
		SREP_I2R_LUT_TA_RIO_PARMS	0x2_06BC	0x0000_2460
	Logical I/O Memory (NREAD, NWRITE_R) 0x4_E000_0000 0x4_FFFF_FFFF	SREP_I2R_LUT_TA_UPPER	0x2_06A4	0x00D1_0100
		SREP_I2R_LUT_TA_LOWER	0x2_06AC	0x6000_0020
		SREP_I2R_LUT_TA_RIO_PARMS	0x2_06BC	0x0000_2455

Similar to the RapidIO to PCI translation path, the Tsi620 SREP Bridge ISF Target Address Spaces can be consolidated into just two Tsi620 PCI Target Address Spaces:

- 260 KB, non-prefetchable, for Doorbells and Maintenance Space
- 1 GB, prefetchable, for Logical I/O

The PCI BARs must be a power of two in size, so the non-prefetchable BAR size is rounded up to 512 KB. The register settings required to implement the Tsi620 PCI Target Address Space are displayed in Table 8. The script which executes the necessary configuration for the example is titled “Tsi620_pci_2_rio_config.txt”.

Table 8: Tsi620 PCI Target Address Space

Tsi620 PCI Target Address Space	Tsi620 Register Name and Offset	Register Value	Tsi620 SREP Bridge ISF Target Address Space	RapidIO Device Target Address Space
Non-prefetchable, for Doorbells and Maintenance 0x0040_0000 0x0047_FFFF	P2O_PAGE_SIZES (0x2_204C)	0x0000_0101	Memory space controlled by P2O_BAR2_LUT values	Memory space controlled by P2O_BAR2_LUT values
	P2O_PAGE_SIZES (0x2_204C)	0x0000_2179		
	P2O_BAR2 (0x2_2018)	0x0040_0004		
	P2O_BAR2_UPPER (0x2_201C)	0x0000_0000		
Unused BAR address range, 0x0040_0000 0x0043_BFFF	P2O_BAR2_LUT0 (0x2_2500)	0x0000_0000	Unused space, translate to address that will result in an error response	N/A
	P2O_BAR2_LUT0_UPPER (0x2_2504)	0xFFFF_FFFF		
	P2O_BAR2_LUT1 (0x2_2508)	0x0000_4000		
	P2O_BAR2_LUT1_UPPER (0x2_250C)	0xFFFF_FFFF		
	. . .			
	P2O_BAR2_LUT14 (0x2_2570)	0x0003_8000		
	P2O_BAR2_LUT14_UPPER (0x2_2574)	0xFFFF_FFFF		
For Doorbells, 0x0043_C000 0x0043_FFFF Note: This is 16K, which is larger than the DB BAR.	P2O_BAR2_LUT15 (0x2_2578)	0xFFFB_C000	0x4_FFFB_F000 0x4_FFFB_FFFF	Doorbell
	P2O_BAR2_LUT15_UPPER (0x2_257C)	0x0000_0004		

Table 8: Tsi620 PCI Target Address Space *(Continued)*

Tsi620 PCI Target Address Space	Tsi620 Register Name and Offset	Register Value	Tsi620 SREP Bridge ISF Target Address Space	RapidIO Device Target Address Space
For Maintenance, 0x0044_0000 0x0047_FFFF	P2O_BAR2_LUT16 (0x2_2580)	0xFFFC_0000	0x4_FFFC_0000 0x4_FFFF_FFFF	Maintenance
	P2O_BAR2_LUT16_UPPER (0x2_2584)	0x0000_0004		
	P2O_BAR2_LUT17 (0x2_2588)	0xFFFC_4000		
	P2O_BAR2_LUT17_UPPER (0x2_258C)	0x0000_0004		
	. . .			
	P2O_BAR2_LUT31 (0x2_25F8)	0xFFFF_C000		
	P2O_BAR2_LUT31_UPPER (0x2_25FC)	0x0000_0004		
Prefetchable Space 0x8000_0000 0xBFFF_FFFF	P2O_BAR3 (0x2_2020)	0x8000_000C	0x4_C000_0000 0x4_FFFF_FFFF	Logical I/O Memory 0x0_4000_0000 0x0_7FFF_FFFF
	P2O_BAR3_UPPER (0x2_2024)	0x0000_0000		
	P2O_BAR3_LUT0 (0x2_2600)	0xC000_0000		
	P2O_BAR3_LUT0_UPPER (0x2_2604)	0x0000_0004		
	P2O_BAR3_LUT1 (0x2_2608)	0xC200_0000		
	P2O_BAR3_LUT1_UPPER (0x2_260C)	0x0000_0004		
	. . .			
	P2O_BAR3_LUT31 (0x2_26F8)	0xFE00_0000		
	P2O_BAR2_LUT31_UPPER (0x2_26FC)	0x0000_0004		

1.7 RapidIO-to-RapidIO Address Mapping

In this example, the FPGA supports 8-bit destination IDs while the RapidIO processor uses 16-bit destination IDs. This has the effect of preventing accesses between the FPGA and the RapidIO processor. In the example system, however, there are two flows that must be supported between the RapidIO processor and the FPGA. The RapidIO processor must configure the FPGA using maintenance accesses, and the FPGA must be able to write to a small region of memory in the RapidIO processor.

Table 9 lists the RapidIO-to-RapidIO Device Target Address Spaces for the FPGA, the RapidIO processor, and the Tsi620. The next step is to map the Tsi620 RapidIO Target Address Spaces to the Tsi620 SREP Bridge ISF Target Address Spaces.

Table 9: RapidIO-to-RapidIO Device Target Address Spaces

RapidIO Target		
Device	**Address Space**	**Source**
FPGA	Maintenance Packets, 8-bit destination IDs	RapidIO Processor
RapidIO Processor	0xD000_0000 0xD003F_FFFF, 16-bit destination IDs	FPGA

The FPGA requires a Tsi620 RapidIO Target Address Space to write to a memory region in the RapidIO processor. The RapidIO processor requires a Tsi620 RapidIO Target Address Space to perform maintenance reads and writes to the FPGA. Each of the Tsi620 RapidIO Target Address Spaces must in turn be mapped to a Tsi620 SREP Bridge ISF address space, which in turn will be mapped to the appropriate RapidIO or FPGA RapidIO Target Address Space.

Table 10: RapidIO-to-RapidIO Tsi620 RapidIO Target Address Spaces

RapidIO Target			
Device	**Address Space**	**Source**	**Tsi620 SREP Bridge ISF Address Space**
FPGA	Maintenance Packets, 8-bit destination IDs	RapidIO Processor	0xD040_0000 0xD043_FFFF
RapidIO Processor	0xD000_0000 0xD003F_FFFF, 16-bit destination IDs	FPGA	0xD000_0000 0xD003F_FFFF

The register settings displayed in Table 11 are required to map the Tsi620 SREP Bridge ISF Address Space to the RapidIO Target Address Spaces. This is accomplished using two I2R BARs. Each BAR is associated with one LUT entry. The example script which implements the necessary register writes is titled “Tsi620_rio_2_rio_config.txt”.

Table 11: RapidIO-to-RapidIO Tsi620 RapidIO Bridge ISF Target Address Spaces

RapidIO Device Target Address Space	Tsi620 SREP Bridge ISF Target Address Space	Tsi620 Register Name	Tsi620 Register Address	Example Register Value
Maintenance Space for FPGA	0xD040_0000 0xD043_FFFF	SREP_I2R_BAR4_LUT_CSR	0x2_0640	0x0080_0600
		SREP_I2R_BAR4_UPPER	0x2_0644	0x0000_0000
		SREP_I2R_BAR4_LOWER	0x2_0648	0xD040_0001
		SREP_I2R_LUT_PTY_CTL	0x2_06A0	0x0000_0080
		SREP_I2R_LUT_TA_UPPER	0x2_06A4	0x01C0_0001
		SREP_I2R_LUT_TA_LOWER	0x2_06AC	0x0000_00A0
		SREP_I2R_LUT_TA_RIO_PARMS	0x2_06BC	0x0000_8081
Memory Space on RapidIO Processor	0xD000_0000 0xD03F_FFFF	SREP_I2R_BAR5_LUT_CSR	0x2_0650	0x0081_0A00
		SREP_I2R_BAR5_UPPER	0x2_0654	0x0000_0000
		SREP_I2R_BAR5_LOWER	0x2_0658	0xD000_0001
		SREP_I2R_LUT_PTY_CTL	0x2_06A0	0x0000_0081
		SREP_I2R_LUT_TA_UPPER	0x2_06A4	0x00D1_0100
		SREP_I2R_LUT_TA_LOWER	0x2_06AC	0xD000_0020
		SREP_I2R_LUT_TA_RIO_PARMS	0x2_06BC	0x0000_2460

The register settings displayed in Table 12 are required to map the Tsi620 RapidIO Target Address Spaces to the Tsi620 RapidIO Bridge ISF Target address spaces. Two R2I BARs are used, one for each Tsi620 RapidIO Bridge ISF Target address space. One LUT is associated with each R2I BAR.

Table 12: RapidIO-to-RapidIO Tsi620 SREP Bridge ISF Target Address Spaces

Tsi620 RapidIO Target Address Space	Tsi620 Register Name and Offset	Example Register Value	Tsi620 SREP Bridge ISF Target Address Space	RapidIO Device Target Address Space
0xD040_0000 0xD043_FFFF	SREP_R2I_BAR5_LUT_CSR (0x2_0550)	0x0050_0600	0xD040_0000 0xD043_FFFF	Maintenance Space for FPGA
	SREP_R2I_BAR5_LOWER (0x2_0558)	0xD040_0010		
	SREP_R2I_LUT_PTY_CTL (0x2_0590)	0x0000_0050		
	SREP_R2I_LUT_UPPER (0x20_05A0)	0x0000_0000		
	SREP_R2I_LUT_LOWER (0x2_05AC)	0xD040_0003		
0xD000_0000 0xD03F_FFFF	SREP_R2I_BAR6_LUT_CSR (0x2_0560)	0x0060_0A00	0xD000_0000 0xD03F_FFFF	Memory Space on RapidIO Processor
	SREP_R2I_BAR6_LOWER (0x2_0568)	0xD000_0010		
	SREP_R2I_LUT_PTY_CTL (0x2_0590)	0x0000_0060		
	SREP_R2I_LUT_UPPER (0x20_05A0)	0x0000_0000		
	SREP_R2I_LUT_LOWER (0x2_05AC)	0xD000_0003		

A. Configuration Scripts

```
// This script configures the Tsi620 Buffer Release Management
// functions in both PCI-to-RapidIO and RapidIO-to-PCI directions,
// using default settings.
//
// For an explanation of the functions of the various registers, see
// the Tsi620 Flow Control Application Note.
//
// PCI to RapidIO Buffer Release Management Configuration
//
// Assumes standard PCI-to-RapidIO priority mapping:
// SREP_I2R_LUT_TA_LOWER.RD_PRIO = 0,
// SREP_I2R_LUT_TA_LOWER.WR_PRIO = 2
//
// Sets timeouts to maximum, and re-engages BRM after timing out if
// the buffer fill level hits STOP. These settings assume that the
// probability of deadlocks is low, and periods of congestion are long.
//

w 206e4 0x80FF1C04 // SREP_I2R_BREL
w 206FC 0x80FF0702 // SREP_B2S_BREL
w 1180c 0x00010203 // SP8_RIO_WM

// RapidIO to PCI Buffer Release Management Configuration
//
// Assumes the following:
// One buffer is sufficient to deal with decomposed PCI transactions.
// - This means that PCI reads must be less than 256 bytes in size.
// Sets timeouts to maximum, and re-engages BRM after timing out if
// the buffer fill level hits STOP. These settings assume that the
// probability of deadlocks is low, and periods of congestion are
// long.
//

w 205b0 0x00000080 // SREP_R2I_ISF_REQ_PRIO_CSR
w 205b4 0x11220000 // SREP_R2I_ISF_RESP_PRIO_CSR
w 205c0 0x01010203 // SREP_R2I_WM
w 205C4 0x80FF1D02 // SREP_R2I_BREL
w 205C8 0x00010203 // SREP_R2I_ISF_WM
w 205CC 0x80FF0802 // SREP_R2I_ISF_BREL
w 206F0 0x00010203 // SREP_NWR_ERR_WM
w 206F4 0x80FF1D02 // SREP_NWR_ERR_BREL
```

```
// This script configures the Tsi620 PCI to RapidIO Address
// translation functions according to the system description
// in the Tsi620 Address Translation Application Note.
//
// For an explanation of the functions of the various registers,
// see the Tsi620 User Manual.
//
// This script contains two parts: Bridge ISF to RapidIO
// translation, and PCI to Bridge ISF translation.
// This is consistent with the Tsi620 Address Translation
// Application Note.
//
//  ----->>>>> Bridge ISF to RapidIO Translation START <<<<<-----
//

// RapidIO Maintenance Space - 256 KB
// Tsi620 Bridge ISF address range 0x4_FFFC_0000 to 0x4_FFFF_FFFF

w 20600 0x00000600  // SREP_I2R_BAR0_LUT_CSR
w 20604 0x00000004  // SREP_I2R_BAR0_UPPER
w 20608 0xFFFC0001  // SREP_I2R_BAR0_LOWER
w 206A0 0x00000000  // SREP_I2R_LUT_PTY_CTL - Use LUT 0
w 206A4 0x01C10100  // SREP_I2R_LUT_TA_UPPER
w 206AC 0x00000000  // SREP_I2R_LUT_TA_LOWER
w 206BC 0x00008081  // SREP_I2R_LUT_TA_RIO_PARMS

// Doorbell Space - 4 KB
// Tsi620 Bridge ISF address range 0x4_FFFB_F000 to 0x4_FFFB_FFFF

w 20690 0x00000004  // SREP_I2R_DB_BAR_UPPER
w 20694 0xFFFBF001  // SREP_I2R_DB_BAR_LOWER

// Memory - 1 GB
// RapidIO Device Target Address Range: 0x0_4000_0000 to 0x0_7FFF_FFFF
// Tsi620 SREP Bridge ISF address Range: 0x4_C000_0000 to 0x4_FFFF_FFFF
//
// Note that programming the LUT entries makes use of the AUTO_INC
// feature. Because of this, the 'W' (write without readback) command
// is used. Otherwise, the readback of the SREP_R2I_LUT_LOWER register
// would cause unintended incrementing of the LUT_INDEX.

w 20610 0x00021201  // SREP_I2R_BAR1_LUT_CSR
w 20614 0x00000004  // SREP_I2R_BAR1_UPPER
w 20618 0xC0000001  // SREP_I2R_BAR1_LOWER
w 206A0 0x80000002  // SREP_I2R_LUT_PTY_CTL - Use LUTs #2-3

W 206A4 0x00D10100  // SREP_I2R_LUT_TA_UPPER
W 206AC 0x40000020  // SREP_I2R_LUT_TA_LOWER
W 206BC 0x00002460  // SREP_I2R_LUT_TA_RIO_PARMS - NREAD/SWRITE
```

```
W 206A4 0x00D10100  // SREP_I2R_LUT_TA_UPPER
W 206AC 0x60000020  // SREP_I2R_LUT_TA_LOWER
W 206BC 0x00002455  // SREP_I2R_LUT_TA_RIO_PARMS - NREAD/NWRITE_R

//  ----->>>>> Bridge ISF to RapidIO Translation END <<<<<-----
//
//  ----->>>>> PCI to Bridge ISF Translation START <<<<<-----
//
// RapidIO Target Address Spaces
// RapidIO Maintenance Space - 256 KB
// Doorbell Space - 4 KB
// RapidIO Device Target Address Range: 0x0_4000_0000 to 0x0_7FFF_FFFF
//
// Tsi620 SREP Bridge ISF Address Spaces
// 0x4_FFFC_0000 to 0x4_FFFF_FFFF - RapidIO Maintenance
// 0x4_FFFB_F000 to 0x4_FFFB_FFFF - Doorbell
// 0x4_C000_0000 to 0x4_FFFF_FFFF - Memory
//
// Tsi620 PCI Target Address Spaces
// 0x0040_0000 to 0x0047_FFFF - Maintenance and Doorbells
// 0x8000_0000 to 0xBFFF_FFFF - Memory Space

// Create 512 KB Range for RapidIO Maintenance + Doorbell
// This is too large for the requirements, so there is an
// Unused space at the beginning of the BAR. The address
// Space is mapped as follows with respect to Tsi620 PCI
// Target Address Space:
// 0x0040_0000 to 0x5F_BFFF - Unused
// 0x005F_C000 to 0x5F_FFFF - Doorbells (too big!)
// 0x0060_0000 to 0x7F_FFFF - Maintenance

w 2204C 0x00000101 // P2O_PAGE_SIZES  // Enable writes to this register
w 2204C 0x00002179 // P2O_PAGE_SIZES  // Set up 512 KB Non-prefetchable BAR
w 22018 0x00400004 // P2O_BAR2
w 2201C 0x00000000 // P2O_BAR2_UPPER
w 22020 0x8000000C // P2O_BAR3 - Set up 1 GB prefetchable Memory BAR
w 22024 0x00000000 // P2O_BAR3_UPPER

// Write the P2O BAR2 LUT Entries
w 22500 0x00000000 // P2O_BAR2_LUT0
w 22504 0xFFFFFFFF // P2O_BAR2_LUT0_UPPER - UNUSED
w 22508 0x00004000 // P2O_BAR2_LUT1
w 2250C 0xFFFFFFFF // P2O_BAR2_LUT1_UPPER - UNUSED
w 22510 0x00008000 // P2O_BAR2_LUT2
w 22514 0xFFFFFFFF // P2O_BAR2_LUT2_UPPER - UNUSED
w 22518 0x0000C000 // P2O_BAR2_LUT3
w 2251C 0xFFFFFFFF // P2O_BAR2_LUT3_UPPER - UNUSED
w 22520 0x00010000 // P2O_BAR2_LUT4
```

```
w 22524 0xFFFFFFFF // P2O_BAR2_LUT4_UPPER - UNUSED
w 22528 0x00014000 // P2O_BAR2_LUT5
w 2252C 0xFFFFFFFF // P2O_BAR2_LUT5_UPPER - UNUSED
w 22530 0x00018000 // P2O_BAR2_LUT6
w 22534 0xFFFFFFFF // P2O_BAR2_LUT6_UPPER - UNUSED
w 22538 0x0001C000 // P2O_BAR2_LUT7
w 2253C 0xFFFFFFFF // P2O_BAR2_LUT7_UPPER - UNUSED
w 22540 0x00020000 // P2O_BAR2_LUT8
w 22544 0xFFFFFFFF // P2O_BAR2_LUT8_UPPER - UNUSED
w 22548 0x00024000 // P2O_BAR2_LUT9
w 2254C 0xFFFFFFFF // P2O_BAR2_LUT9_UPPER - UNUSED
w 22550 0x00028000 // P2O_BAR2_LUT10
w 22554 0xFFFFFFFF // P2O_BAR2_LUT10_UPPER - UNUSED
w 22558 0x0002C000 // P2O_BAR2_LUT11
w 2255C 0xFFFFFFFF // P2O_BAR2_LUT11_UPPER - UNUSED
w 22560 0x00030000 // P2O_BAR2_LUT12
w 22564 0xFFFFFFFF // P2O_BAR2_LUT12_UPPER - UNUSED
w 22568 0x00034000 // P2O_BAR2_LUT13
w 2256C 0xFFFFFFFF // P2O_BAR2_LUT13_UPPER - UNUSED
w 22570 0x00038000 // P2O_BAR2_LUT14
w 22574 0xFFFFFFFF // P2O_BAR2_LUT14_UPPER - UNUSED
w 22578 0xFFFBC000 // P2O_BAR2_LUT15
w 2257C 0x00000004 // P2O_BAR2_LUT15_UPPER - Doorbell
w 22580 0xFFFC0000 // P2O_BAR2_LUT16
w 22584 0x00000004 // P2O_BAR2_LUT16_UPPER - Maintenance
w 22588 0xFFFC4000 // P2O_BAR2_LUT17
w 2258C 0x00000004 // P2O_BAR2_LUT17_UPPER - Maintenance
w 22590 0xFFFC8000 // P2O_BAR2_LUT18
w 22594 0x00000004 // P2O_BAR2_LUT18_UPPER - Maintenance
w 22598 0xFFFCC000 // P2O_BAR2_LUT19
w 2259C 0x00000004 // P2O_BAR2_LUT19_UPPER - Maintenance
w 225A0 0xFFFD0000 // P2O_BAR2_LUT20
w 225A4 0x00000004 // P2O_BAR2_LUT20_UPPER - Maintenance
w 225A8 0xFFFD4000 // P2O_BAR2_LUT21
w 225AC 0x00000004 // P2O_BAR2_LUT21_UPPER - Maintenance
w 225B0 0xFFFD8000 // P2O_BAR2_LUT22
w 225B4 0x00000004 // P2O_BAR2_LUT22_UPPER - Maintenance
w 225B8 0xFFFDC000 // P2O_BAR2_LUT23
w 225BC 0x00000004 // P2O_BAR2_LUT23_UPPER - Maintenance
w 225C0 0xFFFE0000 // P2O_BAR2_LUT24
w 225C4 0x00000004 // P2O_BAR2_LUT24_UPPER - Maintenance
w 225C8 0xFFFE4000 // P2O_BAR2_LUT25
w 225CC 0x00000004 // P2O_BAR2_LUT25_UPPER - Maintenance
w 225D0 0xFFFE8000 // P2O_BAR2_LUT26
w 225D4 0x00000004 // P2O_BAR2_LUT26_UPPER - Maintenance
w 225D8 0xFFFEC000 // P2O_BAR2_LUT27
w 225DC 0x00000004 // P2O_BAR2_LUT27_UPPER - Maintenance
w 225E0 0xFFFF0000 // P2O_BAR2_LUT28
w 225E4 0x00000004 // P2O_BAR2_LUT28_UPPER - Maintenance
```

```
w 225E8 0xFFFF4000 // P2O_BAR2_LUT29
w 225EC 0x00000004 // P2O_BAR2_LUT29_UPPER - Maintenance
w 225F0 0xFFFF8000 // P2O_BAR2_LUT30
w 225F4 0x00000004 // P2O_BAR2_LUT30_UPPER - Maintenance
w 225F8 0xFFFFC000 // P2O_BAR2_LUT31
w 225FC 0x00000004 // P2O_BAR2_LUT31_UPPER - Maintenance


// Write the P2O BAR3 LUT Entries
w 22600 0xC0000000 // P2O_BAR3_LUT0
w 22604 0x00000004 // P2O_BAR3_LUT0_UPPER
w 22608 0xC2000000 // P2O_BAR3_LUT1
w 2260C 0x00000004 // P2O_BAR3_LUT1_UPPER
w 22610 0xC4000000 // P2O_BAR3_LUT2
w 22614 0x00000004 // P2O_BAR3_LUT2_UPPER
w 22618 0xC6000000 // P2O_BAR3_LUT3
w 2261C 0x00000004 // P2O_BAR3_LUT3_UPPER
w 22620 0xC8000000 // P2O_BAR3_LUT4
w 22624 0x00000004 // P2O_BAR3_LUT4_UPPER
w 22628 0xCA000000 // P2O_BAR3_LUT5
w 2262C 0x00000004 // P2O_BAR3_LUT5_UPPER
w 22630 0xCC000000 // P2O_BAR3_LUT6
w 22634 0x00000004 // P2O_BAR3_LUT6_UPPER
w 22638 0xCE000000 // P2O_BAR3_LUT7
w 2263C 0x00000004 // P2O_BAR3_LUT7_UPPER
w 22640 0xD0000000 // P2O_BAR3_LUT8
w 22644 0x00000004 // P2O_BAR3_LUT8_UPPER
w 22648 0xD2000000 // P2O_BAR3_LUT9
w 2264C 0x00000004 // P2O_BAR3_LUT9_UPPER
w 22650 0xD4000000 // P2O_BAR3_LUT10
w 22654 0x00000004 // P2O_BAR3_LUT10_UPPER
w 22658 0xD6000000 // P2O_BAR3_LUT11
w 2265C 0x00000004 // P2O_BAR3_LUT11_UPPER
w 22660 0xD8000000 // P2O_BAR3_LUT12
w 22664 0x00000004 // P2O_BAR3_LUT12_UPPER
w 22668 0xDA000000 // P2O_BAR3_LUT13
w 2266C 0x00000004 // P2O_BAR3_LUT13_UPPER
w 22670 0xDC000000 // P2O_BAR3_LUT14
w 22674 0x00000004 // P2O_BAR3_LUT14_UPPER
w 22678 0xDE000000 // P2O_BAR3_LUT15
w 2267C 0x00000004 // P2O_BAR3_LUT15_UPPER
w 22680 0xE0000000 // P2O_BAR3_LUT16
w 22684 0x00000004 // P2O_BAR3_LUT16_UPPER
w 22688 0xE2000000 // P2O_BAR3_LUT17
w 2268C 0x00000004 // P2O_BAR3_LUT17_UPPER
w 22690 0xE4000000 // P2O_BAR3_LUT18
w 22694 0x00000004 // P2O_BAR3_LUT18_UPPER
w 22698 0xE6000000 // P2O_BAR3_LUT19
w 2269C 0x00000004 // P2O_BAR3_LUT19_UPPER
```

```
w 226A0 0xE8000000 // P2O_BAR3_LUT20
w 226A4 0x00000004 // P2O_BAR3_LUT20_UPPER
w 226A8 0xEA000000 // P2O_BAR3_LUT21
w 226AC 0x00000004 // P2O_BAR3_LUT21_UPPER
w 226B0 0xEC000000 // P2O_BAR3_LUT22
w 226B4 0x00000004 // P2O_BAR3_LUT22_UPPER
w 226B8 0xEE000000 // P2O_BAR3_LUT23
w 226BC 0x00000004 // P2O_BAR3_LUT23_UPPER
w 226C0 0xF0000000 // P2O_BAR3_LUT24
w 226C4 0x00000004 // P2O_BAR3_LUT24_UPPER
w 226C8 0xF2000000 // P2O_BAR3_LUT25
w 226CC 0x00000004 // P2O_BAR3_LUT25_UPPER
w 226D0 0xF4000000 // P2O_BAR3_LUT26
w 226D4 0x00000004 // P2O_BAR3_LUT26_UPPER
w 226D8 0xF6000000 // P2O_BAR3_LUT27
w 226DC 0x00000004 // P2O_BAR3_LUT27_UPPER
w 226E0 0xF8000000 // P2O_BAR3_LUT28
w 226E4 0x00000004 // P2O_BAR3_LUT28_UPPER
w 226E8 0xFA000000 // P2O_BAR3_LUT29
w 226EC 0x00000004 // P2O_BAR3_LUT29_UPPER
w 226F0 0xFC000000 // P2O_BAR3_LUT30
w 226F4 0x00000004 // P2O_BAR3_LUT30_UPPER
w 226F8 0xFE000000 // P2O_BAR3_LUT31
w 226FC 0x00000004 // P2O_BAR3_LUT31_UPPER

//  ----->>>>> PCI to Bridge ISF Translation END <<<<<-----
```

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.