PowerPC™

Using Remote ROM With The 660 Bridge

Application Note AN-PPC-012

The IBM27-82660 PowerPC to PCI Bridge and Memory Controller allows the boot ROM to be located on an ISA or other tertiary bus.

MPRA12APU-01
Dale Elson
April, 1996

IBM®

Using Remote ROM With The 660 Bridge

Dale Elson

The IBM27-82660 PowerPC to PCI Bridge and Memory Controller allows the boot ROM to be located on an ISA or other tertiary bus.

This application note describes the remote ROM mode of the IBM27-82660 PowerPC to PCI Bridge and Memory Controller (660). Remote ROM mode was documented in initial versions (MPR660UMU-01 and earlier) of the 660 User's Manual. The remote ROM mode documentation was removed from the most recent (MPR660UMU-02) version of the user's manual because the function was not supported at the time that the user's manual was released. The remote ROM mode is now supported and will continue to be supported.

This application note serves as an addendum to MPR660UMU-02. The information in this application note will be included in subsequent revisions of the 660 User's Manual.

1 Pin Descriptions

Information about remote ROM mode operation is added to three of the signal descriptions in section 2 of the 660 User's Manual. These additions are indicated in bold in Table 1.

Table 1. 660 Bridge Signal Descriptions

Signal	663	664	Description
ROM_OE#	—	O 47	ROM output enable. ROM_OE# enables direct-attached ROM. **The signal is always high for remote ROM.**
ROM_WE#	—	O 60	ROM write enable. Write enable for flash ROM for direct-attach ROM. **The signal is always high for remote ROM.**
AOS_RR_MMRS	I 166	O 69	All Ones Select/ROM Remote/Mask MEM_RD_SMPL. This signal is used to force the data bus to 64 one-bits when the CPU reads memory of PCI space that is unoccupied. To force all ones, this signal must be asserted on the CPU_CLK that the CPU read latch samples data. When ROM_LOAD is asserted, this signal is used to determine the location of the ROM. When ROM_REMOTE is deasserted, it indicates that the ROM is locally on the PCI bus. In this case, ROM data always arrives on PCI_AD[31:24]. **When ROM_REMOTE is asserted, the ROM is assumed to be on a tertiary bus (such as the ISA bus). In this case, ROM data arrives like all other one-byte PCI targets—first byte on PCI_AD[7:0], second byte on PCI_AD[15:8], etc.** When the PCI is burst reading memory, MASK_MEM_RD_SMPL is asserted after the first MEM_RD_SMPL. It then stays asserted until the PCI–to–MEM read latch is empty. Note: On PCI burst reads of memory, the PCI–to–MEM read latch keeps getting refilled as long as data from the memory is available before the PCI uses it up.

2 ROM Operations

The following sections are added to section 7 of the 660 User's Manual.

The first ROM access method (remote ROM mode) attaches the ROM device to an external PCI agent which supports the PowerPC Reference Platform ROM space map and access protocol. CPU bus master transfers to ROM space are forwarded to the PCI bus and claimed by the PCI agent, which supplies the ROM device data. This PCI device is typically a PCI to ISA bridge. The ROM device attaches to the ISA bridge through the ISA bus lines, thereby saving a PCI bus load. The 660 supplies write-protect capability in this mode.

The ROM mode is indicated to the 660 on the strapping pin configuration bits during power-on-reset (POR).

2.1 Remote ROM Mode

In a system that uses the remote ROM mode, the ROM device attaches to a PCI agent. When a CPU bus master reads from memory addresses mapped to ROM space, the 660 arbitrates for the PCI bus and then masters a memory read transaction on the PCI bus. The PCI agent claims the transaction and supplies the ROM device data. CPU writes to the ROM and ROM write-protection operations are also forwarded to the PCI agent.

As shown in Figure 1, the ROM access flows from the CPU to the 660 over the CPU bus, from the 660 to the PCI agent over the PCI bus, and from the PCI agent to the ROM device. The ROM device attaches to the PCI agent, not to the PCI_AD lines, so a PCI bus load is saved by the remote ROM method.

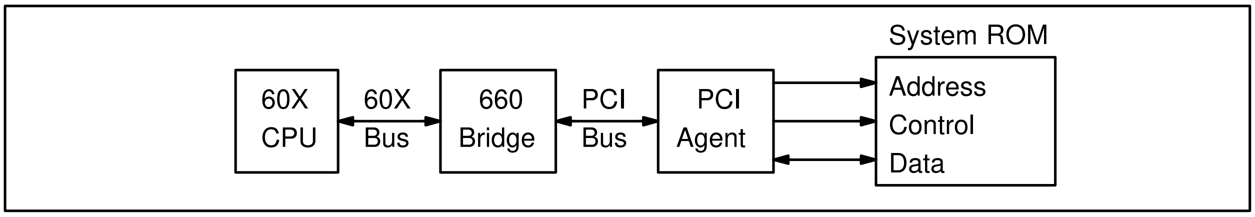


Figure 1. Remote ROM Connections

2.1.1 Remote ROM Reads

For remote ROM reads, the 660 arbitrates for the PCI bus, initiates eight single-byte PCI accesses, releases the PCI bus, and completes the CPU transfer. The eight single bytes of ROM data are assembled into a double-word in the 663 and passed to the CPU. Figure 2 shows the beginning of the operation, including the first two PCI transactions. Figure 3 shows the last part of the operation, including the last two PCI transactions.

During and following reset, compliant PCI agents are logically disconnected from the PCI bus except for the ability to respond to configuration transactions. These agents have not yet been configured with necessary operational parameters. PCI agents capable of the remote ROM access protocol reset with the ability to respond to remote ROM accesses before being fully configured. The CPU begins reading instructions at FFF0 0100h before it can configure the PCI devices.

The ROM read discussion assumes that the system is in big-endian mode. For the effects of little-endian mode operation on ROM reads, see Section 2.1.1.4.

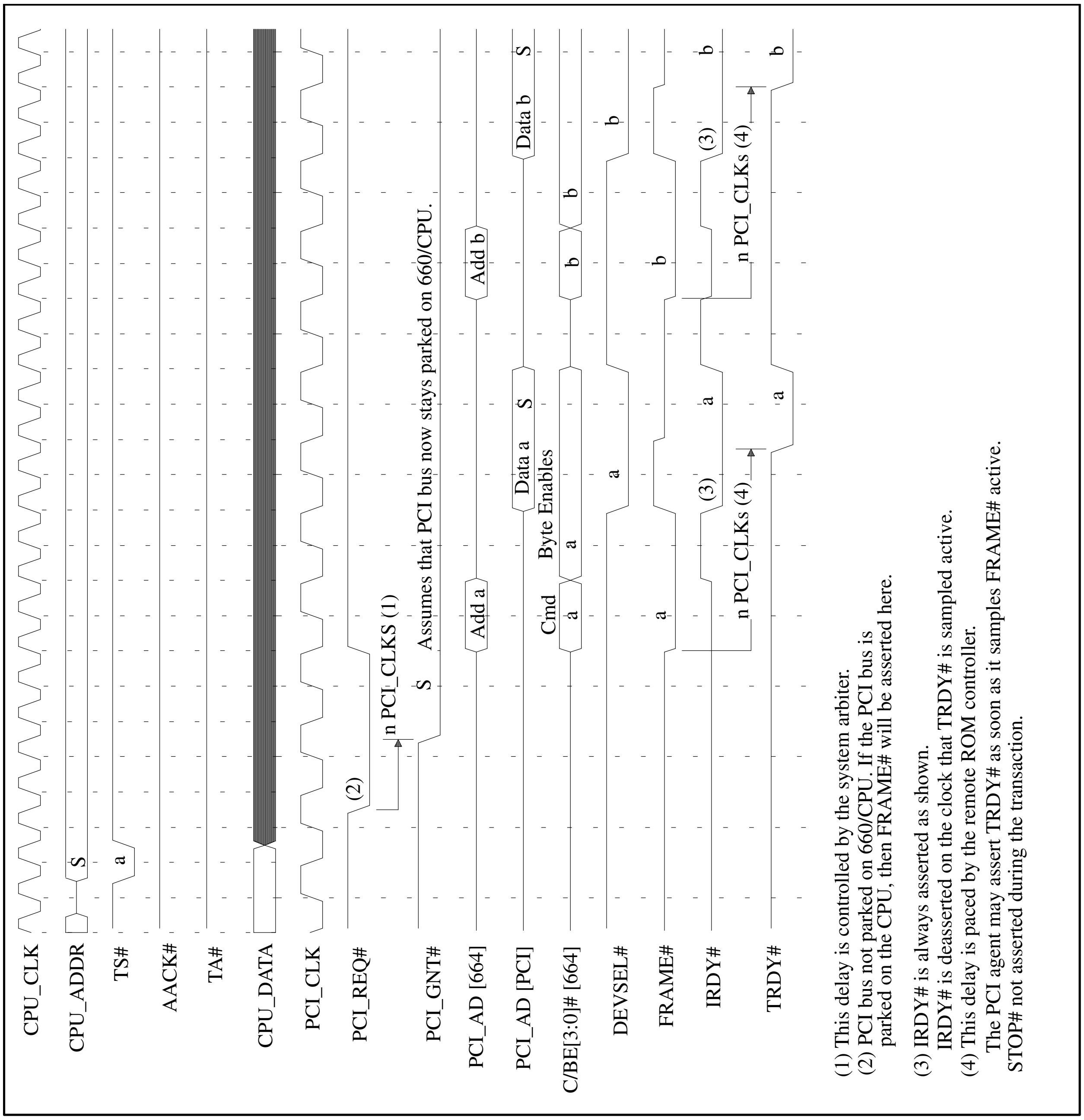


Figure 2. Remote ROM Read – Initial Transactions

2.1.1.1 Remote ROM Read Sequence

In response to a CPU bus read in the 4G – 2M to 4G address range, the 660 requests the PCI bus from the PCI arbiter. When the PCI bus is granted (or if the bus is already parked on the CPU), the 660 initiates a series of PCI memory-read transactions as shown in Table 2 for a CPU read from FFE0 0000h to FFFF FFFF. Note that the last column in Table 2 shows the effect of little-endian mode operation. See Section 2.1.1.4.

The address of the first transaction is the low-order byte of the double-word pointed to by the CPU address (see Section 2.1.1.2). The 660 expects the low-order byte of ROM data in the 8-byte double-word to be returned on PCI byte lane 0, PCI_AD[7:0]. As shown in The 660 then masters seven more PCI read transactions, each time receiving back one byte of ROM data and driving it onto the CPU data bus as shown in Table 2. Note that the byte enables are incrementing within each 4-byte word pointed to by the PCI address.

At the completion of the eighth PCI read, the 660 drives the assembled double-word onto the CPU data bus. The 660 then signals completion of the transfer to the CPU.

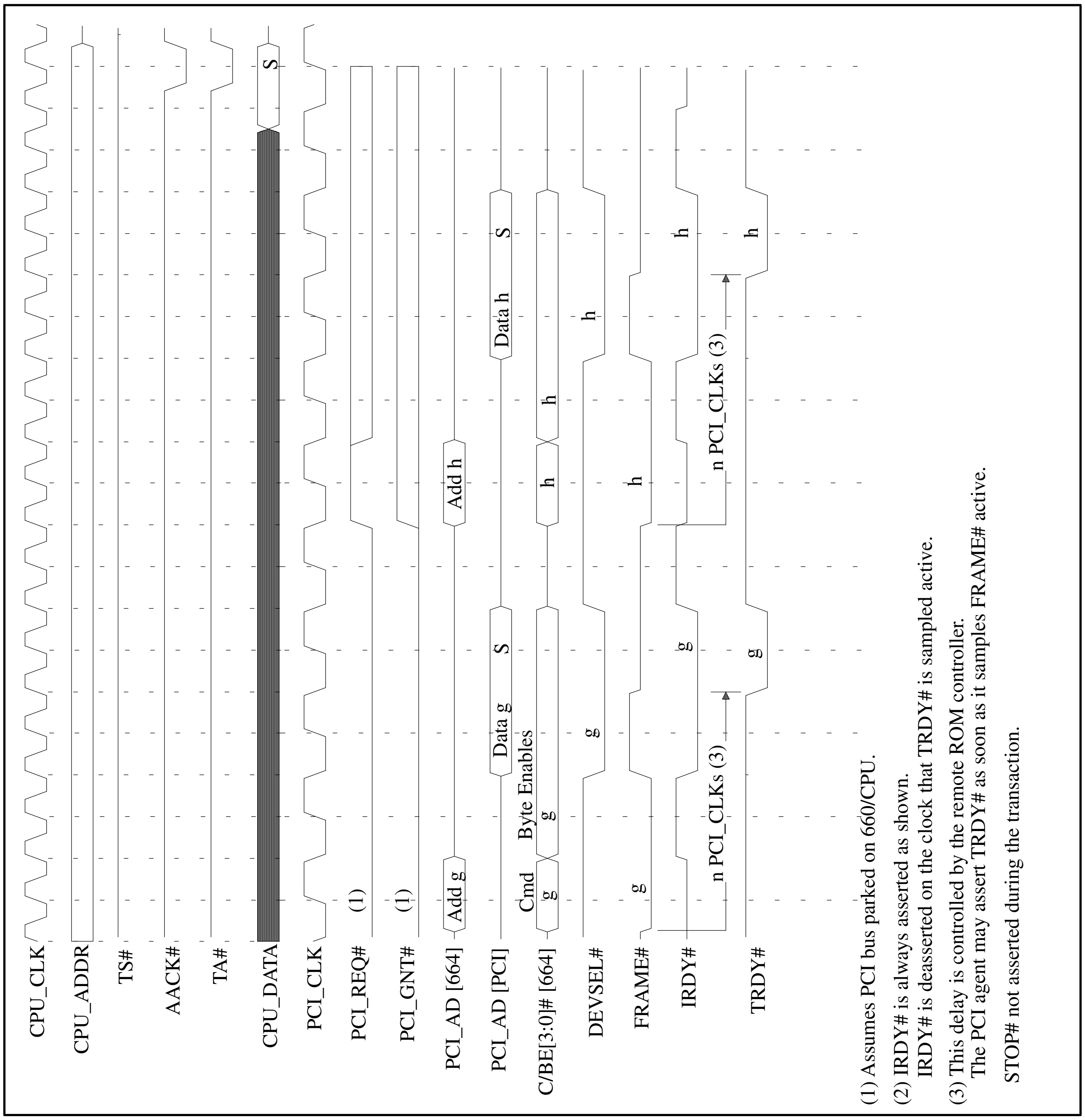


Figure 3. Remote ROM Read – Final Transactions

Remote ROM reads are not pipelined. The 660 does not assert AACK# to the CPU until the end of the remote ROM read sequence. The 660 asserts PCI_REQ# throughout the entire remote ROM read sequence.

Table 2. Remote ROM Read Sequence, CPU Address = FFFX XXX0

PCI Access #	PCI Bus Read Memory Address PCI_AD[31:0]	Byte Enables PCI_C/BE[3:0]#	ROM Addr	ROM Data	Big Endian CPU_DATA [0:63]	Little Endian CPU_DATA [0:63]
1	FFFX XXX0h	1110	0	a	—	—
2	FFFX XXX0h	1101	1	b	—	—
3	FFFX XXX0h	1011	2	c	—	—
4	FFFX XXX0h	0111	3	d	—	—
5	FFFX XXX4h	1110	4	e	—	—
6	FFFX XXX4h	1101	5	f	—	—
7	FFFX XXX4h	1011	6	g	—	—
8	FFFX XXX4h	0111	7	h	abcd efgh	hgfe dcba

2.1.1.2 Address, Transfer Size, and Alignment

The initial PCI address generated during the remote ROM read sequence is formed by copying the high-order 29 bits of the CPU address, and forcing the three low order bits PCI_AD[2:0] to 000b. This generates a base address that is aligned on an 8-byte boundary. While reading the lower 4 bytes, the 660 indicates which byte it is requesting using the PCI byte enables C/BE[3:0]#. After the first four bytes of ROM data are read, the 660 increments the address on the PCI_AD lines by 4 before executing the second four PCI reads.

The CPU read address need not be aligned on an 8-byte boundary. A CPU read from any address (in ROM space) of any length that does not cross an 8–byte boundary within a double-word returns all eight bytes of that double-word data from the ROM. For example, the operations shown in Table 2 could have been caused by a CPU memory read to FFF0 0100h, FFF0 0101h, or FFF0 0105h.

Errors occurring during remote ROM reads are handled as usual for the error type. No special rules are in effect.

2.1.1.3 Burst Reads

The 660 supports burst reads in remote ROM mode. The 660 supports a pseudo burst mode, which supplies the same eight bytes of data (from the ROM) to the CPU on each beat of a 4-beat CPU burst.

A burst ROM read begins with the 660 executing a single-beat ROM read operation, which assembles eight bytes of ROM data into a double-word on the CPU data bus. For a burst ROM read, the 660 asserts TA# for four CPU_CLK cycles, with AACK# asserted on the fourth cycle. The same data remains asserted on the CPU data bus for all four of the data cycles.

For a single-beat read, the 660 asserts TA# and AACK# for one CPU_CLK cycle, and the CPU completes the transfer.

2.1.1.4 Endian Mode Considerations

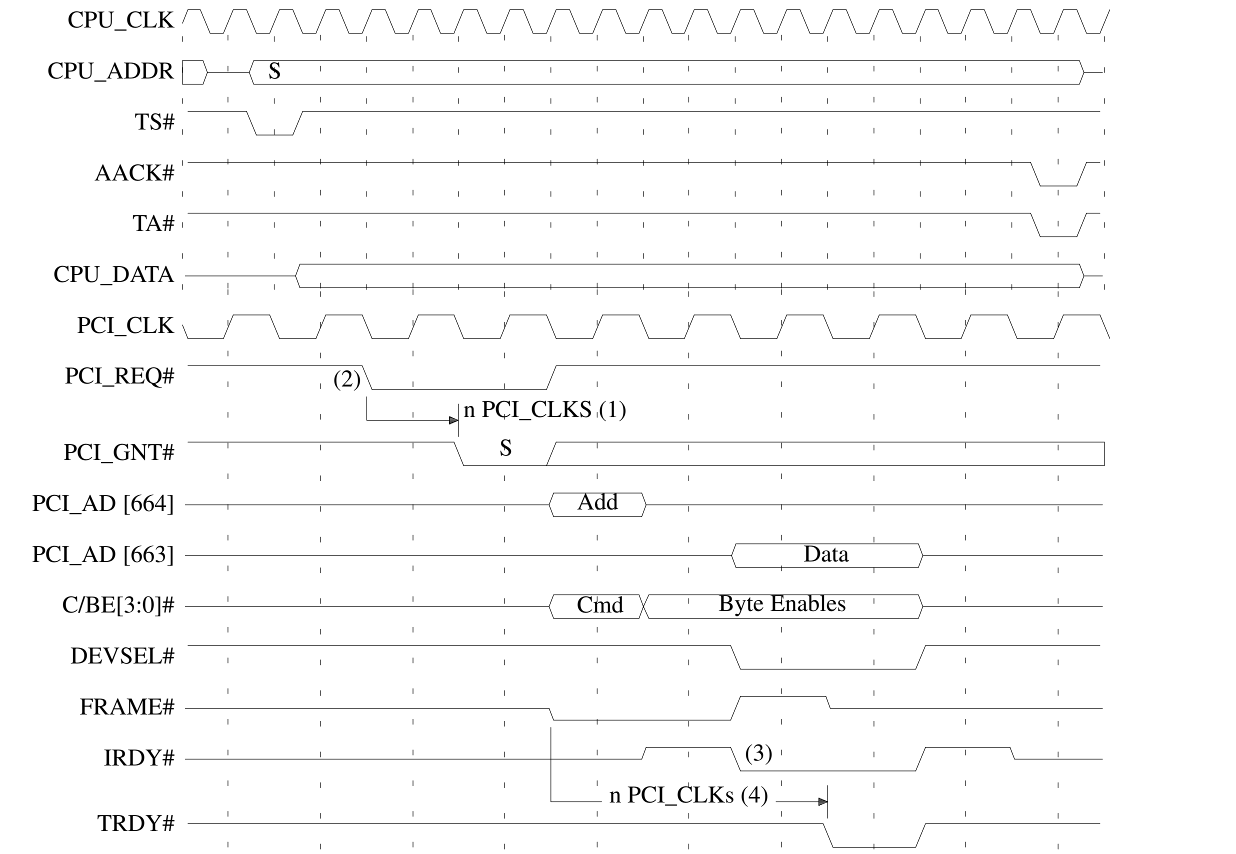
In little-endian mode, the address munging done by the CPU has no effect because PCI_AD[2:0] are forced to 000 during the address phase by the 660 at the beginning of the transaction. However, in little-endian mode the byte swapper is enabled, so the bytes of ROM data returned to the CPU are swapped as shown in the last column of Table 2.

2.1.1.5 4-Byte Reads

The 660 handles 4-byte ROM reads (and all ROM reads of less than 8 bytes) as if they were 8-byte reads. All 8 bytes are gathered by the 660, and all 8 bytes are driven onto the CPU data bus.

2.1.2 Remote ROM Writes

While the 660 is configured for remote ROM operation, the 660 forwards all CPU to ROM write transfers to the PCI bus as memory writes. The PCI agent that is controlling the remote ROM acts as the PCI target during CPU to ROM write transfers, executes the write cycle to the ROM, and may provide ROM write-protection.



(1) This delay is controlled by the system arbiter.

(2) PCI Bus not parked on CPU. If the PCI bus is parked on the CPU, FRAME# is asserted here.

(3) IRDY# is always asserted as FRAME# is deasserted.
IRDY# is deasserted on the clock that TRDY# is sampled active.

(4) This delay is paced by the remote ROM controller.
The PCI agent may assert TRDY# as soon as it samples FRAME# active.

STOP# not asserted during the transaction.

Figure 4. Remote ROM Write

2.1.2.1 Write Sequence

A CPU bus master begins a remote ROM write transaction by initiating a one-byte, single-beat memory write transfer to CPU bus address range 4G – 2M to 4G (FF80 0000h to FFFF FFFFh).

The 660 decodes the CPU transfer, arbitrates for the PCI bus, and initiates a memory write PCI transaction to the same address in the 4G – 2M to 4G address range.

The PCI agent that is controlling the remote ROM (such as the PCI to ISA Bridge), claims the transaction, manages the write cycle to the ROM device, and signals TRDY#.

The 660 then completes the PCI transaction, and signals AACK# and TA# to the CPU. Note that remote ROM writes are neither posted or pipelined.

2.1.2.2 Write Protection

Write protection can be provided by the PCI agent that controls the ROM. In addition, some flash ROM devices can have the means to permanently lock out sectors by writing control sequences. The 660 also has a write lockout in the Bridge Chipset Options 2 register (bit 0 of index BBh).

2.1.2.3 Address, Size, Alignment, and Endian Mode

In remote ROM mode, CPU memory writes from 4G – 2M to 4G cause the 660 to generate PCI bus memory write transactions to 4G – 2M to 4G. The 660 does not allow CPU masters to access the rest of the PCI memory space from 2G to 4G.

In remote ROM mode, PCI bus master memory write transactions from 4G – 2M to 4G are ignored by the 660. However, the PCI agent that controls the ROM responds to these transactions. In contrast, in direct-attach ROM mode, the 660 forwards PCI bus master memory transactions from 2G to 4G (to populated memory locations) to system memory from 0 to 2G.

Remote ROM writes must be one-byte, single-beat transfers.

The endian mode of the system has no net effect on a ROM write because the transfer size is one byte. The address is munged by the CPU and unmunged by the 660. The data comes out of the CPU on the byte lane associated with the munged address, and then is swapped by the 660 to the byte lane associated with the unmunged address. Thus a ROM write in little-endian mode puts the data byte in the same ROM location as does the same ROM write in big-endian mode.

2.2 ROM-Related Bridge Control Registers

2.2.1 ROM Lockout Bit for Remote-Attach ROM

The ROM write-protect bit for remote ROM is in the Bridge Chipset Options 2 register (index BBh). While enabled (bit 0 is 1), writes to the remote ROM are forwarded to the PCI memory space. While disabled (bit 0 is 0), writes to the remote ROM are treated as no-ops and an error is signalled. After the bit is set to 0 (disabled), it cannot be set to 1 (enabled).

USA and Canada:
IBM Microelectronics Division
1580 Route 52, Bldg. 504
Hopewell Junction, NY 12533-6531
Tel: (800) PowerPC
Fax: (800) PowerFax
http://www.chips.ibm.com
http://www.ibm.com
ftp://ftp.austin.ibm.com/pub/PPC_support

Document Number: MPRA12APU–01