PowerPC™ 60x/PCI Bus Bridge Design for PowerPC Reference Platform

Yongjae Rim
Power Personal System Division
IBM Corporation
Austin, Texas 78758
Office (512) 838-8508
FAX (512) 838-5989
e-mail: yrim@austin.ibm.com

Abstract

*PowerPC Reference Platform specification defines the open computer system specification which is based on the PowerPC microprocessor. The specification recommends a system to provide industry standard buses to provide a wide variety of expansion capability. The existing PowerPC microprocessors implement a proprietary bus called 60x bus. The strategic direction of the PowerPC Reference Platform is to implement the **PCI** (Peripheral Component Interconnect) local bus. The PCI local bus is a high performance, industry standard bus specification. This paper discusses design and implementation of the 60x/PCI bus bridge chip for PowerPC Reference Platform design.*

1 Introduction

It has been noticed in the computer industry that compute power of conventional microprocessors far exceed the capacities of most industry standard system buses. Graphics-oriented operating systems, such as OS/2™, Windows-NT™ and System7™, have created a performance bottleneck between the processor and I/O architecture. Local bus implementation has shown substantial performance improvement in such environments.

Although the PowerPC™ microprocessor buses deliver high throughput, their use isoften limited to the communication layer among processor, memory controller and high speed I/O devices, because of the compatibility with existing industry standards and the implementation cost. To provide a wide variety of expansion capability and low cost adaptors, the PowerPC Reference Platform recommends a system provide industry standard buses.

The current implementation of the PowerPC Reference Platform is to implement the **PCI** (Peripheral Component Interconnect) local bus. The PCI local bus is a high performance, industry standard bus specification used in a wide range of computer system applications. It is a 32-bit or 64-bit bus with multiplexed address and data.

The goal of this paper is to provide a guideline for implementing a 60x/PCI bridge chip which can be used in PowerPC Reference Platform system design. In section 2, brief descriptions of 60x and PCI buses are presented. The high level descriptions of bus protocols and the functional capabilities are to be presented. The discussion in chapter 3 is devoted to functional requirements for the PowerPC Reference Platform. This paper provides examples for implementing these functional requirements. For the discussion in this paper, the memory controller and the arbiter are assumed to be integrated into the bridge chip because of the improved performance and cost reduction.

2 60X bus and PCI bus

2.1 60X bus

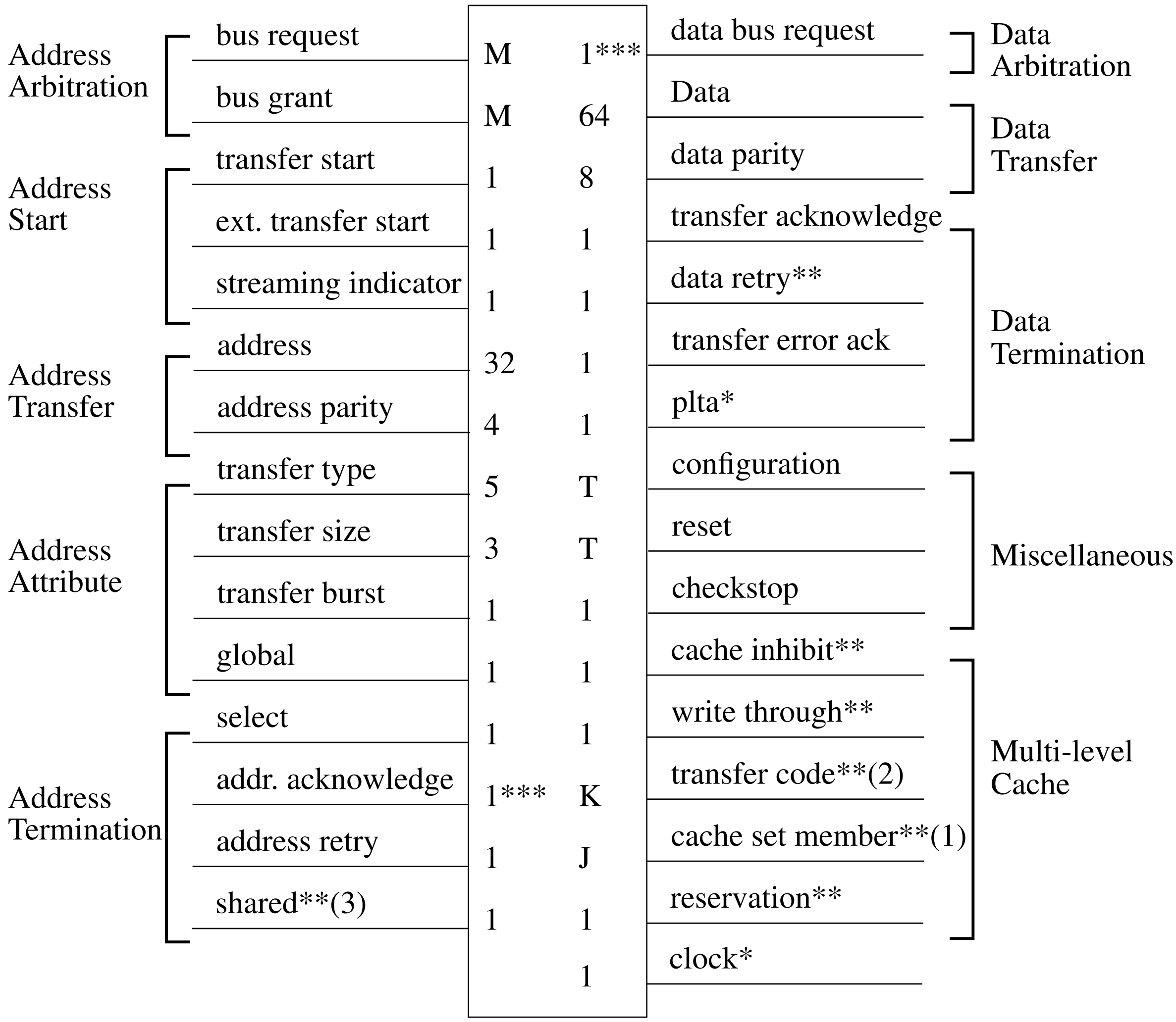
There are two PowerPC processor interfaces defined: 60x and 6xx. In this paper, the discussion is focused only on 60x bus architecture.

The 60x bus interface is designed to support the PowerPC memory model primarily for 32-bit implementations. The interface uses separate address and data busses and a variety control and status signals. The address is 32 bits wide and the data is 64 bits wide. These buses have byte wide parity bits. The bus supports multiple masters through arbitration provided by the system. Bus parking is allowed to eliminate the arbitration overhead.

The 60x bus supports maintenance of a coherent memory system in bus based multiprocessor system using '**MESI**' (Modified, Exclusive owned, Shared and Invalid) snoop protocol. Note that 603 supports '**MEI**' (Modified, Exclusive owned, and Invalid) protocol, instead. In PowerPC Architecture, the coherency unit is implementation dependent. In 601, the cache line has two sectors of 32 bytes and each sector maintain its coherency independently. The cache line sizes of 603 and 604 are 32 bytes.

The 60x bus supports two transfer protocols: **Basic Transfer Protocol** and **Extended Transfer Protocol**. The **Basic Transfer Protocol** supports transfers of any number of contiguous bytes within an aligned double word. The **Basic Transfer Protocol** also supports transfer of aligned 32-byte blocks via a multi-beat transfer. In the **Basic Transfer Protocol**, the separate address and data buses can be used in a system to implement single envelope, pipelined or split transaction operation.

Extended Transfer Protocol is used for accesses to Direct Store Segments. In the **Extended Transfer Protocol**, a positive reply for each transaction is provided. PowerPC Reference Platform specification recommends not to use the **Extended Transfer Protocol**



Note:
-M(Bus master number)
-J(cache set number): 3 for 601, 1 for 603, and 2 for 604
-K(Transfer code size): 2 for 601and 603, and 4 for 604
-603 has no SHARED pin (MEI protocol)
** Optional signals
*** Multiple copies of these signals to have point-to-point connection between arbiter and masters

Figure 1. 60X bus interface signals

because of its limited performance. The Extended Transfer Protocol is provided primarily for the compatibility with Power architecture.

Figure 1 on page 4 shows the signals for 60x buses. The detailed discussion for each signals on 60x bus is available in [2].

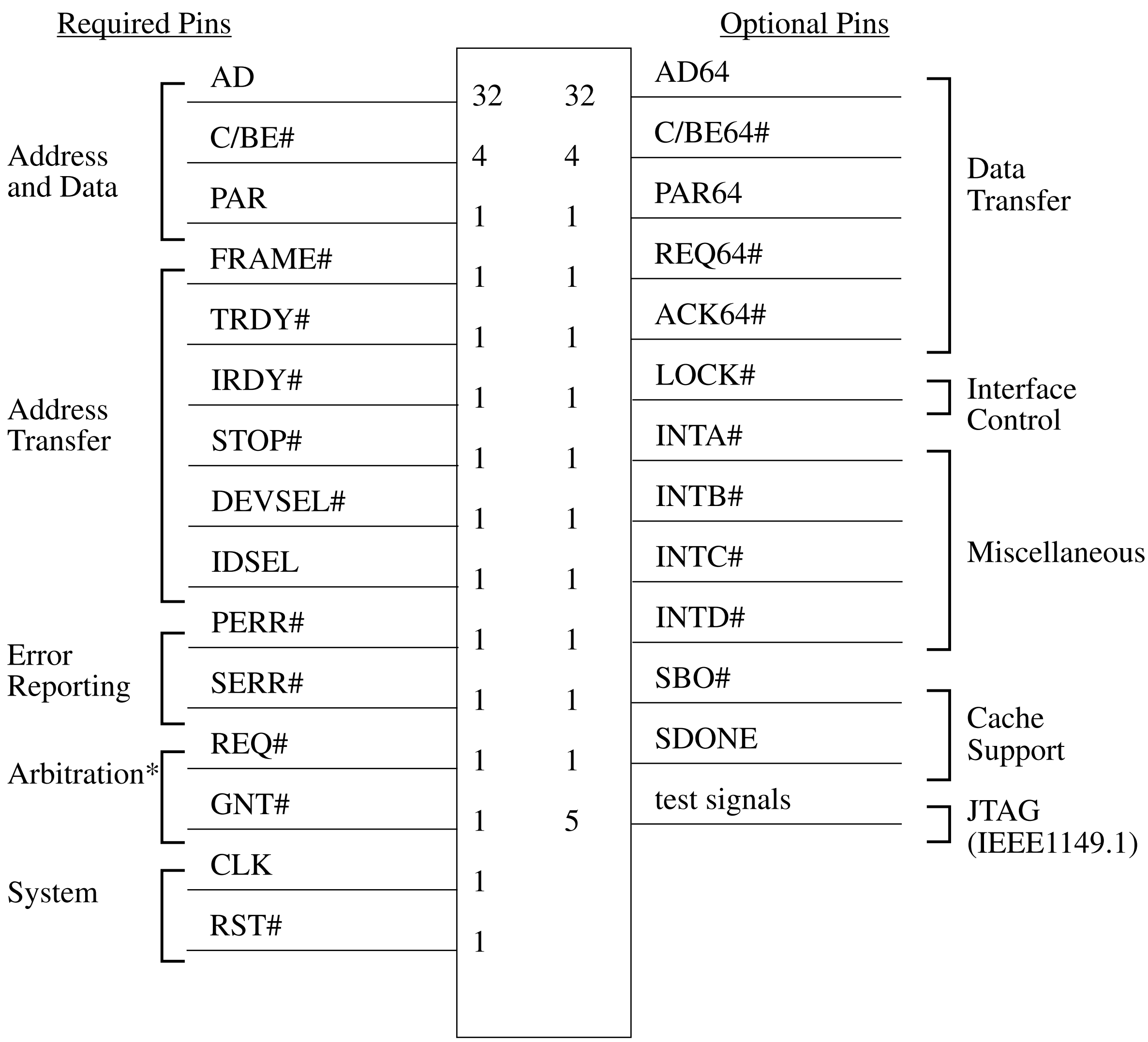


Figure 2. PCI bus interface signals

2.2 PCI bus

The PCI local bus is a 32-bit or 64 bit bus with multiplexed address and data lines. The multiplexed architecture reduces pin count and packaging size of PCI components. The bus protocol provides a transparent update capability from 32-bit to 64-bit data path.

The PCI is a synchronous bus with up to 33MHz clock speed, delivering up to 264 MByte/s peak (with 64 bit data path) performance. The bus allows variable length linear and toggle mode burst transfer for both read and write to improve write dependent graphics performance.

The bus allows multiple masters using a central arbitration scheme and simple request-grant handshaking. Bus parking is allowed to reduce the latency of random accesses.

PCI devices are fully auto configured using configuration cycle. This allows the device drivers for PCI components to be portable across various classes of platforms.

The PCI cache coherency mechanism assumes a flat address space and a single level bridge topology. The PCI bus provides maintenance of cache coherency using three state (Modified, Standby and Invalid) snoop protocol. The coherency unit is a cache line. When the system has a write-back cache policy resource locking has a potential deadlock. The deadlock is avoided by requiring targets that support cacheable write-back even when the bus is locked. The granularity of lock is defined to be 16 bytes aligned.

Figure 2 on page 5 shows the signals for PCI buses. The detailed discussion for each signals on PCI bus is available in [3].

3 Functional requirements on 60x/PCI bus bridge

This chapter discusses the functional requirements on the 60x/PCI bus bridge for PowerPC Reference Platform systems. To make the discussion easy to follow, reference is made to an example implementation. Although there are several designs in progress in the industry, examples are selected from the chip set (**IBM2782654A** and **IBM2782653B**) from IMD (IBM Microelectronics Division) due to its public availability. The chip set has a functional partition such that **IBM2782654A** generates the control signals and the **IBM2782653B** provides buffering and routing of data and addresses under the control of **IBM2782654A**.

The IBM chip set provides the memory control interface as well as the central bus arbitration function. These two functions are integrated into the bridge function because of performance and cost reasons. Although PowerPC Reference Platform does not require a bridge chip to include the memory controller and bus arbiter, it is assumed in the following discussion.

3.1 Address decoding

PowerPC Reference Platform defines four types of address spaces: system memory, I/O memory, system I/O and ROM. Due to the characteristics of PCI protocol, the 60x/PCI bridge must partition the system I/O space to include at least PCI I/O and PCI configuration spaces.

The **IBM2782654A** decodes the CPU addresses to map into five address spaces: system memory, PCI memory, PCI I/O, PCI configuration and system ROM spaces. The system I/O space is partitioned into PCI I/O, PCI configuration, and Planar register spaces. The Planar register space is allocated to map the system register and some ISA I/O registers. Some of the functions provided with the registers include physical memory map, error status, interrupt vectors and FLASH memory access ports.

The PowerPC Reference Platform allows a system designer to choose the memory map. **IBM2782654A** uses predetermined logic to decode memory addresses. In this case, the system designer must follow the memory map provided by the bridge chip. Another way of implementing the decoding is to use a programmable decoding table. With the programmable table, a system designer can choose or modify the memory map suitable for his requirements.

The bridge generates a PCI bus cycle when it recognizes an address within PCI memory, PCI I/O or PCI configuration spaces. If the ROM is located on the PCI bus or on the tertiary bus, the bridge may generate the PCI cycle for addresses in the ROM space.

When the tertiary bridge is implemented it is recommended that the addresses of devices on the tertiary bus be mapped in PCI I/O spaces. It helps simplify the address decoding on the bridge. In this case, the I/O address of the devices on the tertiary bus device must be decoded by the tertiary bus bridge.

3.2 System memory control

The PowerPC Reference Platform requires a minimum of 8MByte memory and 32 Mbyte of expansion capability. In **IBM2782654A**, the extension capability is supported by using eight SIMM address mapping registers. Since each mapping register is associated with a SIMM slot on a system the bridge allows a system to have up to eight SIMM slots. The content of the register indicates the starting real address of the corresponding SIMM.

SIMM mapping registers partition the real memory space into 8MByte segments. This implies that the capacity of the SIMM module must be equal to or multiple of 8 MBytes. **IBM2782654A** supports 8 and 32 MByte SIMMs. The mapping registers are located in the System I/O space.

It is recommended the bridge support SIMMs with various capacities. **IBM2782654A** uses the memory address mapping table to support this capability. The table has eight entries to allow up to eight SIMM slots in the system. The width of the entry determines the maximum memory capacity that the system is allowed. For example, if the entry is eight bits wide and the entry is represented in 4MByte segment, the system can have up to 1GByte memory. The size of the segment determines by the minimum SIMM capacity supported.

Another advantage of using the memory address mapping table is that the system can boot with faulty SIMM modules. That is, by properly tagging the table entry at system boot time, the table can hide the existence of the faulty module on the boot such that the physical memory map is built without the faulty SIMMs.

3.3 Support for the burst mode transfer

In general, there are two types of burst mode transfer from a bus bridge's point of view: 60x master initiated and PCI master initiated burst transfers. 60x master initiated burst transfers are for the cache line transactions. The burst transfer for cache line transactions is required by PowerPC Reference Platform specification. An I/O burst transfer is not available on PowerPC Reference Platform system because the specification prohibits a transaction which transfers more than four bytes into the system I/O space.

The IBM chip set supports 60x master initiated burst transfers whose addresses are aligned on a 4-byte boundary. During burst transfer, the **IBM2782653B** increments the memory address. For 60x master initiated burst transfer, the most immediate double words are fetched first. Then, the eight LSBs of the address are increased by eight with wrapping around to fetch the 32 bytes aligned data. Since the **IBM2782654A** allows one master on 60x, the only possible burst type initiate by a 60x master is a cache line transfer due to cache miss or snoop write back.

The IBM chip set allows the PCI initiated burst at word-aligned address. If eight byte access is allowed as for a 60x master initiated burst, it may create a situation that a word in the double word is in the memory and the other word may not be. To avoid this situation, PCI master initiated burst is based on a word transfer and the address is increased by 4. Note that a PCI burst may cause more than one page fault when the size of data is larger than the page size. On the other hand, there is at most one page fault in 60x master initiated burst transaction because the transaction is caused for one cache line transfer.

3.4 Bus arbitration

It is strongly recommended that a designer includes bus arbiters in the bridge design for both 60x and PCI buses. **IBM2782654A** provides arbitration capability of the PCI and 60x buses, managing the 60x and PCI buses as a single logical bus such that one master is granted ownership of both buses at any time.

The arbiter in **IBM2782654A** employs a priority and timer scheme to provide performance and fairness. For example, the refresh cycle has the highest priority and PCI devices have the lowest. At the same time, the timer is used to provide a fairness of accessing the bus for all masters. That is, once the timer expires, the current bus master must relinquish bus ownership if there is any other bus master requesting the bus. For performance reasons, the snoop cache hit causes the current PCI master to relinquish the PCI bus and to start the cache write back independent of its priority.

IBM2782654A has two pairs of BR (Bus Request) and BG (Bus Grant) on 60x bus for a CPU and a L2 cache. This places limits of using the bridge for multi-processor systems.

3.5 System ROM access

The PowerPC Reference Platform strongly recommends the use of FLASH memory for system ROM. FLASH allows for field upgrade using software, making system ROM update easy. If the bridge implements the ROM interface, write to the ROM space must be supported. Since most FLASHes are implemented such that they require different power requirement for a write operation, special power control must be provided.

One way to access ROM is to use conventional SRAM interface. The IBM chip set, however, implemented two state machines in **IBM2782654A** to generate a special ROM cycle. In every special cycle, the ROM is accessed on a byte basis. The **IBM2782654A** sends control signals to the **IBM2782653B** to properly align the address for ROM access. The **IBM2782653B** increments the address and packs the data byte into a proper format for the FLASH access.

For a read access, eight bytes are read from ROM and buffered in **IBM2782653B** independent of the actual size to be transferred between the controller and the CPU. After the eight bytes are accumulated in the **IBM2782653B**, the data is transferred to the CPU.

The FLASH write cycle is executed by writing a four byte word on the address 0x'FFFFFFF0'. The word written is composed of three bytes of FLASH address and a byte of data to be stored at that address. The FLASH write is protected by writing any data to the address 0x'FFFFFFF1'.

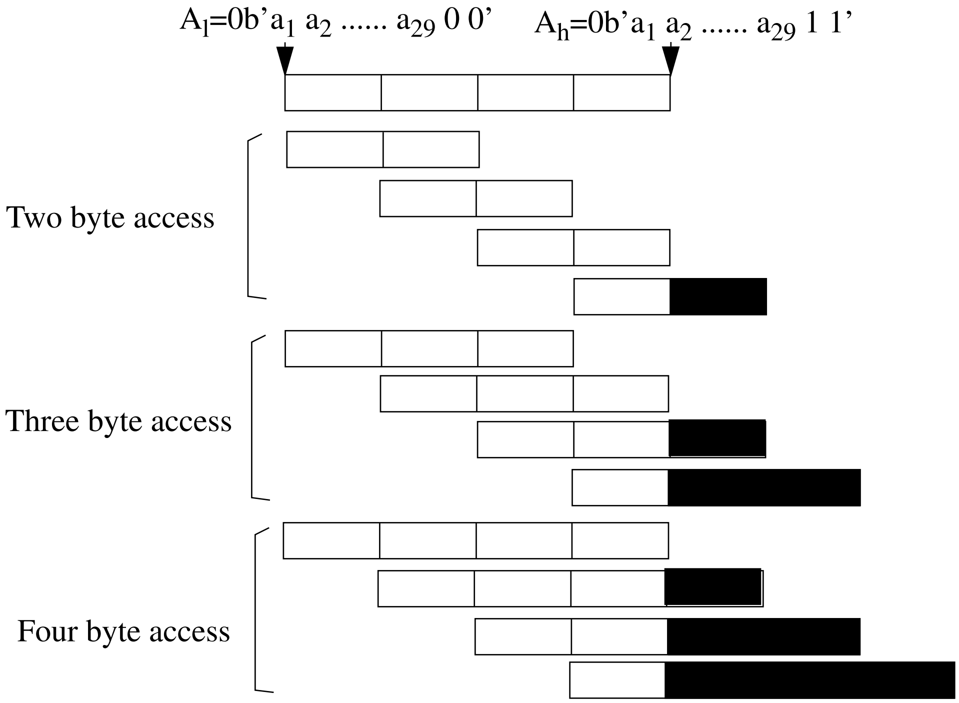
3.6 Bi-Endian support

Bi-Endian support is required by the PowerPC Reference Platform. Current PowerPC microprocessors are programmable to run in either big-Endian or little-Endian mode. There are two different implementations of little-Endian byte ordering. One is "address-consistent view", and the other is "byte-consistent view."[5] The current PowerPC microprocessors are implemented in byte-consistent view.

The little-Endian design on current PowerPC implementation may introduce data inconsistency when CPU tries to access I/O storage because most little-Endian I/O devices are implemented in address-consistent view. To avoid this data inconsistency, byte lane swapping and address unmunging[1][1] are required.

The example bridge design has implemented bi-Endian mode data path such that the data byte lanes are swapped and the address is unmunged in little-Endian mode to translate the byte-consistent ordering to the address-consistent ordering. The detailed discussion about Endian mode and its implication on PowerPC Reference Platform is provided in [1][4].

1. Address munging is a technique to implement the byte-consistent view. To implement byte-consistent view, the 3 LSBs in the 32 bit address is XOR with b'111', b'011' or b'001' depending on the data size.



For address A which points to the first data byte accessed,
$A_l <= A <= A_h$

Figure 3. I/O access address rule

3.7 Data access alignment

PowerPC Reference Platform strongly recommends the software programmers to use only aligned data accesses. Since the level of support for the unaligned data access varies on microprocessor implementations, the usage of the unaligned data access may cause unpredictable behavior.

PowerPC Reference Platform requires system I/O access to follow an addressing rule. The addressing rule states that the all access must be made within a word boundary. The examples of valid and invalid I/O accesses are shown in Figure 3 on page 12. The data bytes without black boxes are properly addressed accesses and others are the ones that are not properly addressed accesses. **IBM2782654A** generates TEA (Transfer Error Acknowledgment) when an improperly addressed data access is detected.

3.8 Support for ECC (Error Correction Code)

ECC is used to enable a system to continue working with memory faults. The most commonly used ECC scheme is SECDED (single-bit error correction and double-bit error detection). Using this ECC, a single bit error can be corrected by hardware in the run time without causing a system failure.

It is often assumed that ECC costs more memory space and I/O signals to implement than doing Parity. As the width of the memory data path increases, however, these costs for ECC increase logarithmically while the cost for Parity increases proportional to the size of data. For example, to implement the 64 bit ECC for 1-bit error correction and 2-bit error detection, the Hamming code with distance 3 is required. In other words, total (64 * 4 - 3) different Hamming code, or 8 bit syndromes, are required to implement the ECC code of the 64 bit data. Since these syndromes are stored in the memory with associated data, 64 bit data requires 72 bit memory space. Using the same analysis, it requires 7 bits to implement SECDED ECC for 32 bit data, or 9 bits for 128 bit data. On the other hand, the byte-wide parity requires 4, 8 and 16 bits of memory space and I/O signals for 32, 64 and 128 bit data.

ECC requires logic for syndrome generation, error correction and error detection. Comparing to Parity, ECC requires a significant amount of logic. However, since the bridge chip is usually an I/O-bound design[1] the cost for extra logic should be containable.

3.9 Support of multiple bus speed.

In the existing PC environment, the system bus speeds are equal to or slower than 33 MHz because of the limitations on driver technology and the board level design constraint. To help the system designer utilize the optimal system bus, most of the PowerPC microprocessors provide the capability of sampling the 60x bus at a slower speed than its internal clock. For example, the 601

1. The design is called I/O bound if the chip size is increased not by the amount of logic required but by the boundary space to provide the required I/O signal pads on the chip

microprocessor is able to sample the 60x bus at a reduced rate, such as 1/2 or 1/3 of the 601 internal clock speed.

Table 1: SetUP register configuration examples

60x speed (MHz)	PCI speed (MHz)	Bit 1 (Bus speed)	Bit 2 (Extra memory cycle)
25	25	0	x
33	33	0	x
40	20	1	0
50	25	1	1
66	33	1	1

To provide a flexibility on selecting the bus speed and memory speed, it is recommended to have at least 1/2 clock divisor in the bridge. This clock divisor allows the use of low speed RAM with the fast bus. Since the dual-PCI bridge has two independent PCI interfaces, it is desired to have separate speed control for each PCI bus to support the devices.

The **IBM2782654A** implements the SetUp register to allow the various combinations of the 60x and PCI bus speeds. The **IBM2782654A** uses two bits to figure out if 1) the speed of 60x and PCI buses are same or not, and 2) extra delay is necessary to propagate the memory data to the fast bus. Table 1, “SetUP register configuration examples,“ indicates the possible settings for the two bits for different clock speed.

3.10 L2 cache support

L2 cache is introduced to reduce the memory latency for a system which has L1 cache. In a system with a dual-PCI bridge[1] which tends to have more traffic than a single PCI bridge, L2 cache is also useful to reduce the amount of memory traffic by filtering out some of L1 cache miss transactions.

1. Dual-PCI bridge is a bus bridge which connects a host bus to two PCI buses.

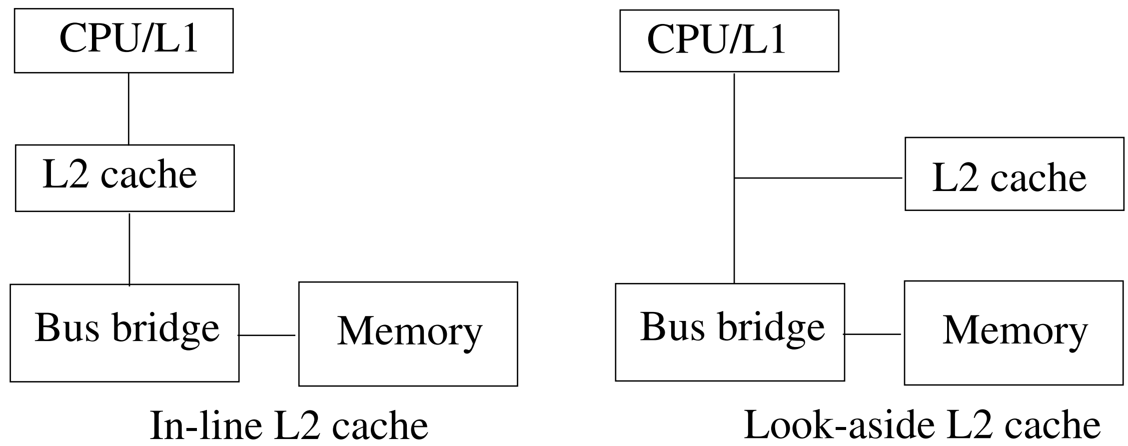


Figure 4. L2 cache implementations

An L2 cache can be implemented as either in-line or look-aside cache. The connectivity of the two caches is shown in Figure 4 on page 15. Since the look-aside cache sends the L2 cache hit signal after the address is sent to the bridge, the effective amount of 60x master traffic does not decrease.

The bridge chip requires an L2_Cache_Hit signal input to support the look-aside caches. For all L1 cache transactions, if L2 cache found a valid copy in its directory, the L2 controller must send the L2_Cache_Hit signal to the bridge. When the bridge chip is used with an in-line cache, the L2_Cache_Hit signal can be wired to false and the L2 cache miss reported by the controller is served properly.

3.11 Error handling

The hardware support for error handling helps for hardware debugging during system bring up and software debugging. The PowerPC Reference Platform strongly recommend to have at least Parity error and its address reporting capability. The memory controller and the bus bridge should be able to isolate the parity error by buses and by addresses. The IBM chip set reports the following errors:

- Parity error
- Transfer size error
- Extended Transfer Protocol PIO cycles

- ecowx and eciwx instructions
- PCI target abort
- PCI target system time-out
- PCI master abort

Once an error is detected, the bridge chip sends the TEA (Transfer Error Acknowledgment) signal to the CPU and sets the corresponding error status bit in a register located in System I/O space. Since all the errors are reported through TEA in the current PowerPC microprocessors the bridge chip requires registers to record the type of error to help a microprocessor identify the type of error occurred. If the error is a memory parity error, the address where the error occurred is saved in the Memory Parity Error Register.

3.12 Power Management

Power Management is recommended by the PowerPC Reference Platform. When implemented, the software must manage the device and the system power. The hardware component must provide the interface for its power state report and power state change to the software.

The PowerPC Reference Platform recommends devices implement four power states: On, Power Managed, Low Power, Off. The On state means the device is in full functional mode, and Off state means that the device is powered off. The Power Managed state differs from Low Power state such that in Power Managed state the device and its subsystem are operational with reduced performance while in the Low Power state they are not operation but preserve the state information.

The memory controller in the bridge must support CAS before RAS refresh in order to allow low power DRAM refresh. The refresh cycle must use Real Time Clock when the CPU is either in Power Managed mode. The bridge must be able to turn off the memory controller in self-refresh mode.

The bridge chip must be able to control the power of subsystems independently such that I/O subsystem can be turned off independent of the memory controller. In maximum power save mode, the bridge must tri-state the interfaces, including addresses, data and control signals.

4 Conclusion

This paper has discussed for 60x/PCI bus bridge design. The discussion has been provided from two aspects; requirements and recommendations. Requirements are the features those must be implemented for applications to be compliant with the PowerPC Reference Platform. Some functional features are recommended to improve system performance. Considering the compute power that a PowerPC microprocessor delivers, a designer should try to implement all the recommended features to deliver a bridge chip which helps reduce the performance bottleneck from I/O and memory systems. Some features, such as programmable address decode table and memory mapping registers, are recommend to improve the flexibility of the bridge chip for various applications.

The discussion is primarily focused on requirements and strong recommendations for PowerPC Reference Platform design criteria to be used in client systems (e.g. notebook and desktop. Other considerations may take precedence in the event that a design is intended for usage in a unique or special environment. With respect to higher-performance server applications, additional consideration should be given to expansion capability and to exploit the parallelism in transactions and/or memory interface.

Reference

[1]. PowerPC Architecture, First Edition, May, 1993.

[2]. IBM System 60X bus Architecture, Book I, version 1.3, Feb 25, 1993.

[3]. PCI Local Bus Specification, Revision 2.0, April 30, 1993.

[4]. Shin-Tai Phan, “Bi-Endian designs in PowerPC Reference Platform,” IBM PPS internal note, March 1994.

[5]. H. Kirrmann, “Data Format and Bus Compatibility in Mulitprocessors,” IEEE Micro, Vol. 3, No. 4, August 1983, pp. 32-47.

Trademarks

PowerPC and OS/2 are trademarks of the International Business Machines Corporation.

Windows-NT is a trademark of the Microsoft Corporation.

System 7 is a trademark of Apple Computer, Inc.