6.7.2.5 Write-Through L2 Timing Diagrams

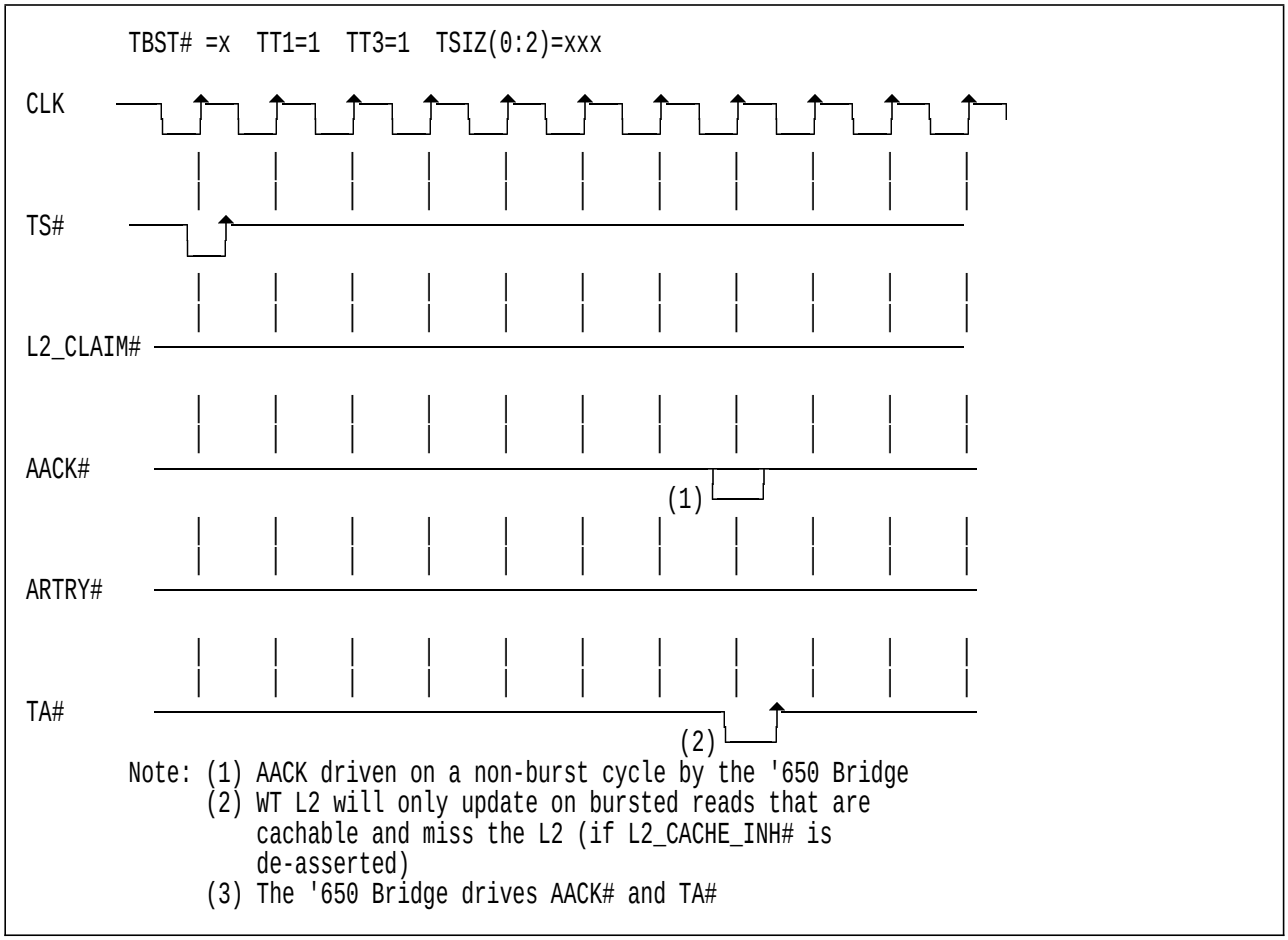


Figure 29. WT L2 Response to Read Misses

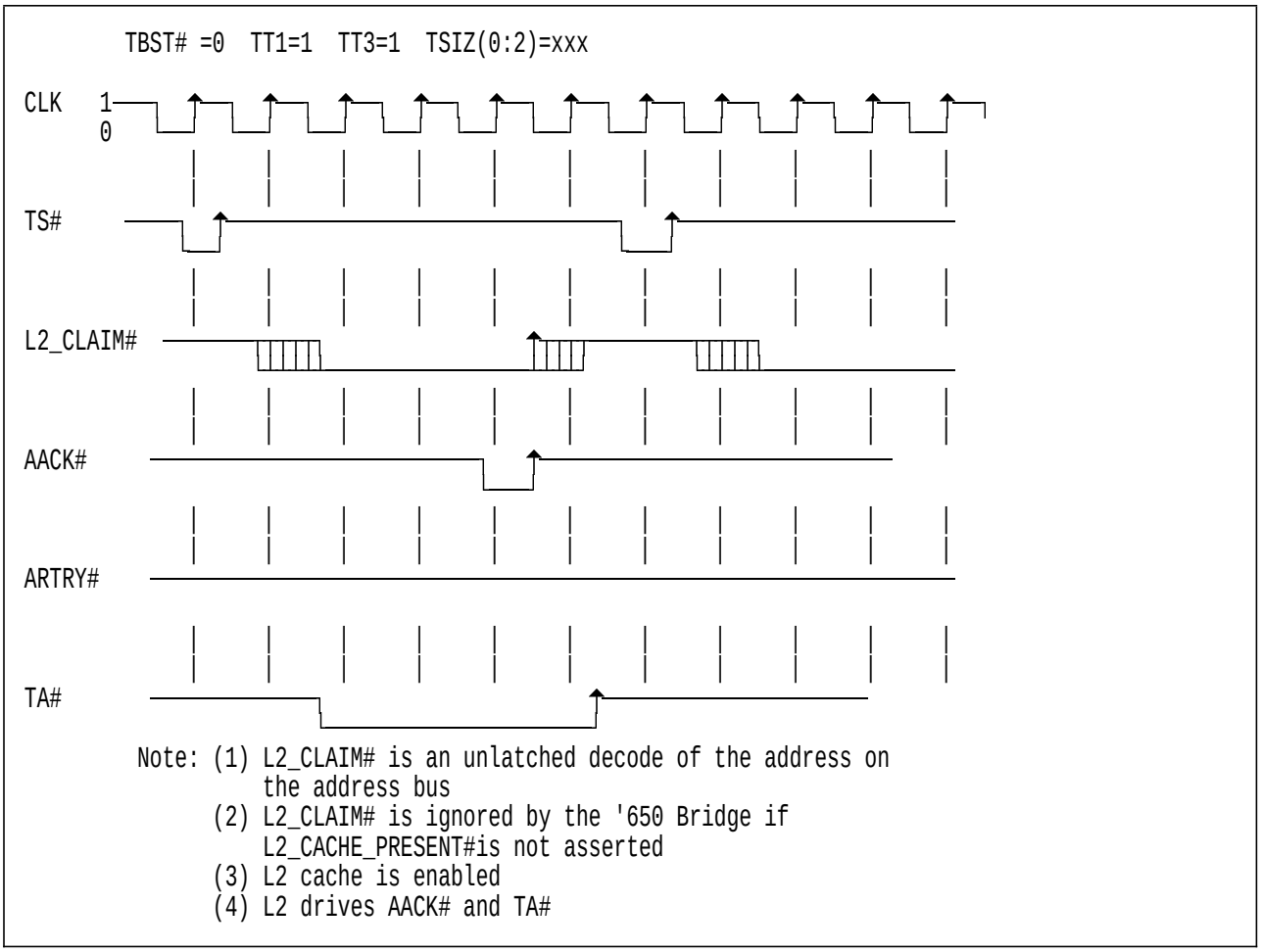


Figure 30. WT L2 Response to Burst READ Hits

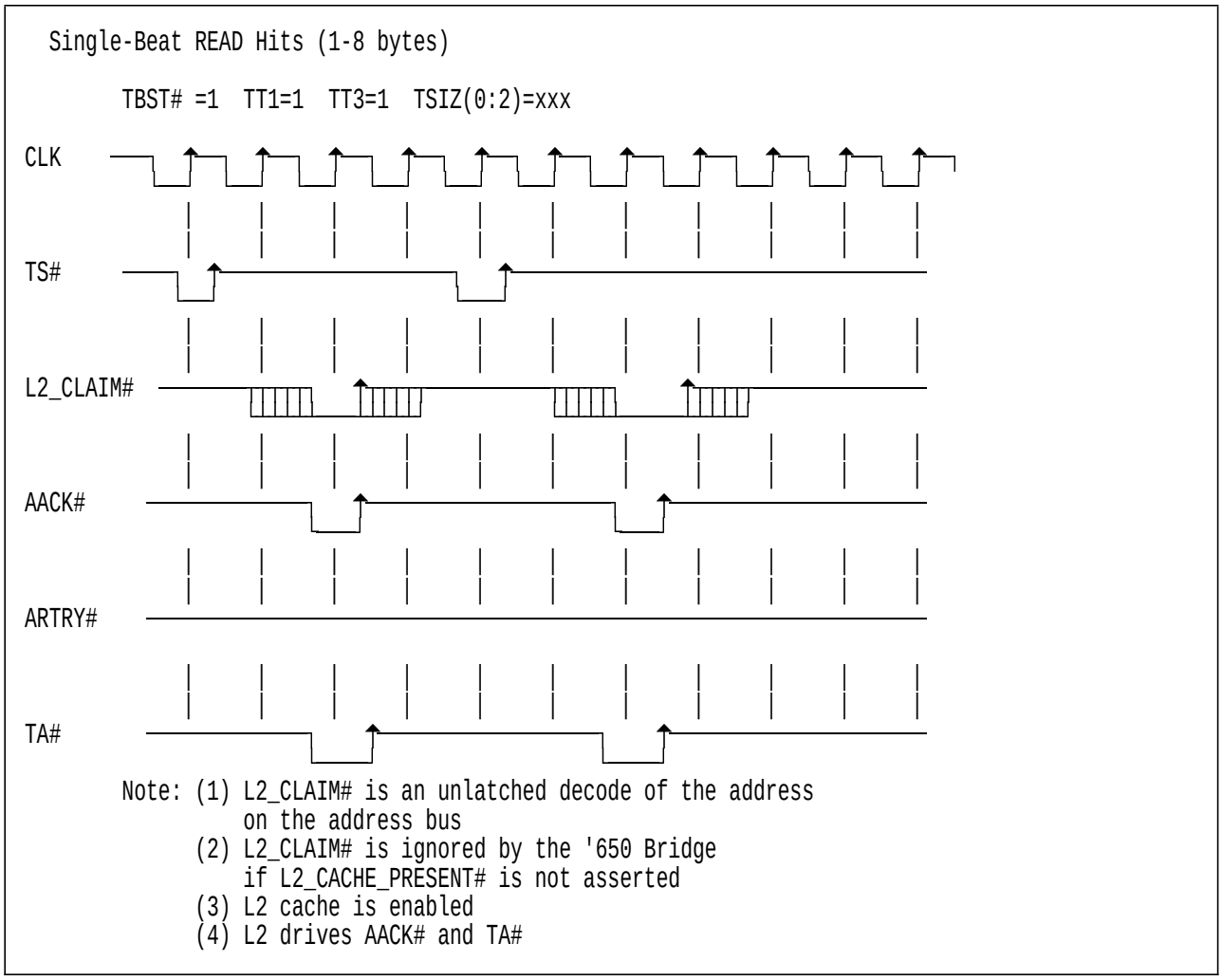


Figure 31. WT L2 Response to Single-Beat READ Hits (1-8 bytes)

Figure 32 shows the WT L2 response for a write cycle. If the cycle is a burst write by the PowerPC processor, the L2 will cache these accesses, overwriting any data that may already be in the cache. Since I/O (PCI) snoop write cycles are never more than 4 bytes, the L2 will invalidate data on write hits, or ignore cycles that are misses.

Since the L2 is a write-through cache design, the L2 never contains “dirty” data. Therefore, the WT L2 will never assert ARTRY# or BR_L2#.

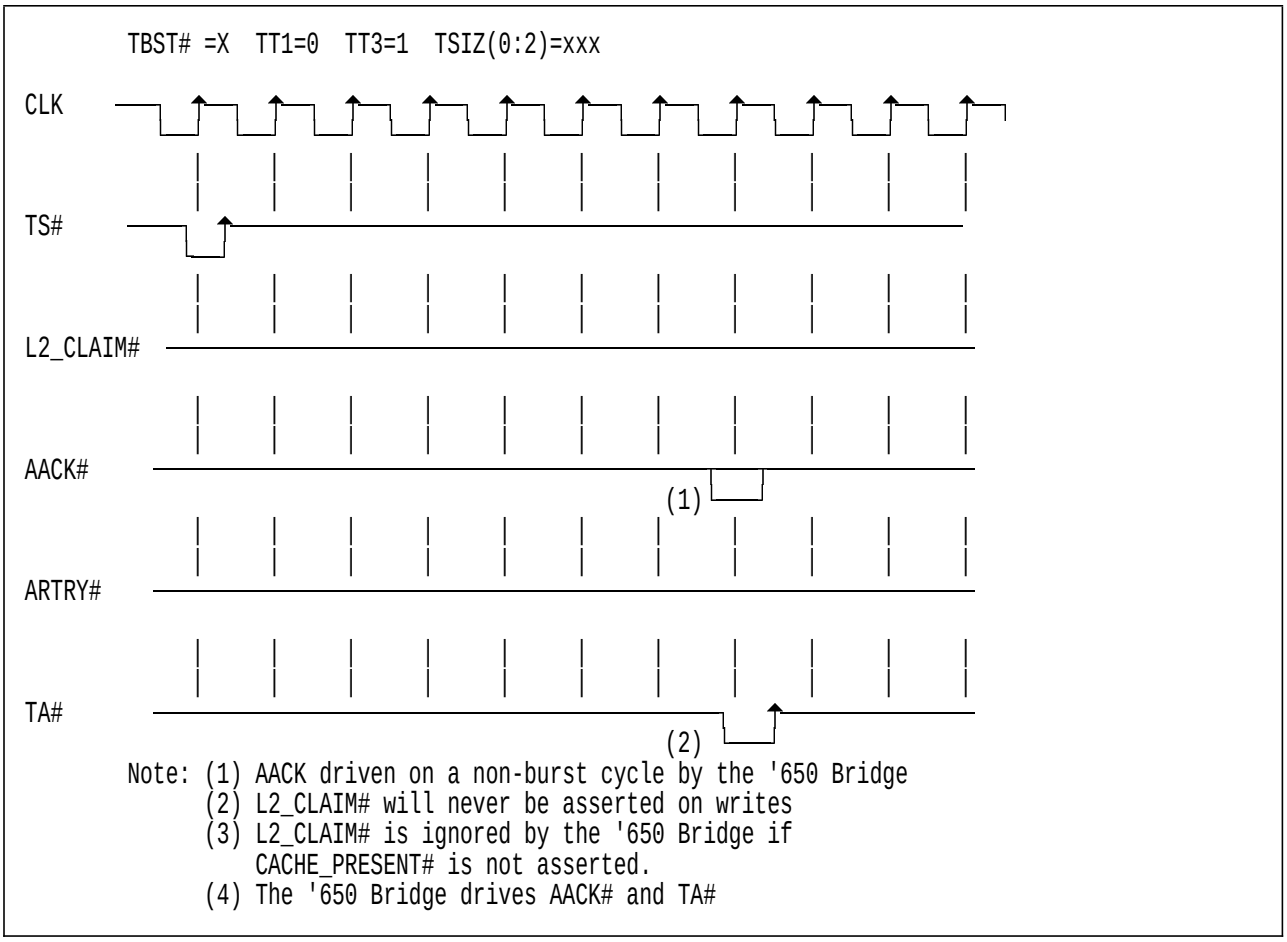


Figure 32. WT L2 Response to Write Cycles

6.7.2.6 Copy-Back L2 (CB L2) Protocol

The CB L2 (also called Write-Back or Store-In) is used to provide the PowerPC processor faster access to memory for both reads and writes. The writes that are cached are from L1 cast-outs. Any writes with kill (due to a snoop hit operation) will not be cached by the CB L2.

Since the CB L2 does cache L1 cast-outs, the performance of these cache cast-outs cycles will be improved (PowerPC processor-to-memory writes).

Coherency is enforced by the MI protocol. That is, data can be in one of three states:

VM Valid Modified

VC Valid Clean

I Invalid

Note: All cachable write operations are “hits” in a CB L2. However, the term “miss” is used to imply that the address of the transfer is to a location not currently stored in the L2, while a hit is to a line marked as valid (either clean or dirty).

The data in the CB L2 is either valid and clean (data is equal to the data in memory), valid and modified (main memory contains stale data), or invalid.

Any cache sector in the CB L2 can be changed from invalid to valid on a burst read or write operation. In order for the data to be stored in the CB L2, the following must be true:

- The cache must be enabled (L2_UPDATE_INH# not asserted)
- CI# must be de-asserted
- L2_MISS_INH# must be de-asserted
- The bus transaction must be a burst read or write that misses the L2
- The bus transaction must not be a write with kill that is a result of a snoop push

On reads, if CI# is asserted and the access is a hit, the CB L2 will ignore the CI# input. If the cycle is a write, and CI# is asserted, the CB L2 will invalidate on a hit, and ignore (not update) on misses.

In addition, if the WT# (write-through) bit is asserted, the CB L2 will cache the data, but NOT assert L2_CLAIM# on writes (entry marked as valid-clean).

Once the data is in the cache, the PowerPC processor can read the data from the cache in any valid byte combinations. Writes of less than 8 bytes that hit in the cache will cause the line to be marked invalid unless the cache includes byte addressability.

6.7.2.6.1 CB L2 READ Responses: The CB L2 maintains coherency in a manner similar to the PowerPC processor.

All processor address bus cycles are snooped whether the address originates with the processor or the ’650 bridge. If any snoop cycle hits the cache, the CB L2 determines whether the snoop hit is caused by the PowerPC processor or the ’650 bridge.

If the read access is from the PowerPC processor, and it hits in the CB L2, then the CB L2 will assert L2_CLAIM# and pace the cycle with AACK and TA.

If the read access is an I/O snoop cycle from the ’650 bridge, the CB L2 will determine if it is a hit. If it is a hit, it will then determine if the data is modified. If the data is not modified, it will do nothing. The CB L2 cache must not assert L2_CLAIM# during any I/O snoop (a snoop originated by the ’650 bridge).

If the data is modified, it will assert ARTRY#. The clock after it asserts ARTRY#, the CB L2 must assert L2_BR# to the ’650 Bridge.

If the PowerPC processor also asserts ARTRY# on the same cycle as the CB L2, the ’650 Bridge will grant the bus to the processor. The CB L2 must sample the 60X_BR# on the clock after ARTRY#. If the 60X_BR# is active, the CB L2 must de-assert its bus request on the following clock. (The CB L2 will update its data when the PowerPC processor pushes its data and therefore the data in the L2 is superfluous.)

If the L2_BR# was already active before the snoop cycle, and the current snoop cycle does not require the CB L2 to write to memory, the CB L2 must de-assert its bus request the cycle after ARTRY#. The CB L2 can then re-assert its bus request[1].

6.7.2.6.2 CB L2 WRITE Responses: The CB L2 must fully decode the TT bits and the CI# and WT# bits. On any write misses that are marked as either write through, cache inhibited, or writes with kill, the CB L2 will not assert L2_CLAIM#.

The CB L2 will not update its cache on write misses of less than 32 bytes (TBST# must be active).

---

[1] Note that, in an implementation which is a single-processor system, the arbiter does not need to see only the one BR# active from the processor. As mentioned earlier, if both the L2_BR# and the 60X_BR# are active, the arbiter will always grant the next cycle to the PowerPC processor. The L2_BR# will then be sampled after the processor cycle is complete to determine if it also needs the bus.

The CB L2 should buffer any data that is written or read by the PowerPC processor that is to be cached by the L2 but requires a cache push operation by the L2 to precede its writing into the cache. (This situation happens when the processor reads or writes a cache line in the L2, there is a miss, and the data stored at that tag location is dirty.) This buffer can be of variable depth. If the buffer is full (from previous operation(s)), and the current cycle is cachable but requires a push, the CB L2 MUST not assert L2_CLAIM# or ARTRY#. Instead, the CB L2 should just not cache these cycles. This is required in order to maintain the greatest bandwidth from memory to the PowerPC processor and vice versa[2].

6.7.2.6.2.1 PowerPC Processor Writes: The CB L2 cache will cache data on the following write transactions:

- A burst write that hits a clean line
- A burst write that hits a dirty line
- A burst write that misses a line (only if the buffer is not full)
- A non-burst write that hits a clean line
- A non-burst write that hits a dirty line
- A non-burst write that misses a line (only if the buffer is not full)

On a burst write that is a hit to the same address that is already cached, the CB L2 will assert L2_CLAIM# and overwrite the previous data (whether modified or not).

If the address is a hit that is to a different address than that in the L2 cache, and the currently cached data is clean, the cycle completes as before.

If the cycle is not a burst, then the CB L2 will determine whether or not the cycle hits in the cache. If the cycle hits in the cache to either a clean or a dirty sector of the same address, the CB L2 will assert L2_CLAIM# and update the required byte(s). If the cycle would overwrite dirty data to another address, the CB L2 will NOT assert L2_CLAIM#. That is, the CB L2 cannot have any entry of less than 32 bytes cached.

If the write is a miss and the buffer is full, the CB L2 may do either of the following:

- Assert ARTRY# and BR_L2#
- Nothing (preferred)

6.7.2.6.2.2 I/O Snoop Writes: All processor address bus cycles are snooped whether the address originates with the processor or the ’650 bridge. The CB L2 determines whether the snoop is caused by the PowerPC processor or the ’650 bridge. The CB L2 must not assert L2_CLAIM# on any I/O snoop.

If the write access is an I/O snoop cycle from the ’650 bridge, the CB L2 will determine if it is a hit. If it is a miss, it will do nothing. If it is a hit, the CB L2 will then determine if the data is modified. If the data is not modified, it will will mark that line as invalid.

If the data is modified, it will assert ARTRY#. The clock after it asserts ARTRY#, the CB L2 must assert L2_BR# to the ’650 Bridge.

If the PowerPC processor also asserts ARTRY# on the same cycle as the CB L2, the ’650 Bridge will grant the bus to the processor. The CB L2 must sample the 60X_BR# on the clock after ARTRY#. If the 60X_BR# is active, the CB L2 must de-assert its bus request on the following clock.

---

[2] The reason for not asserting ARTRY# on a CB L2 pipeline (buffer) full condition is that the PowerPC processor will be delayed by at least 20 cycles since the L2 would need to write out the data, and then the processor would need to re-read (write) the data from (to) memory.

If the L2_BR# was already active before the snoop cycle, and the current snoop cycle does not require the CB L2 to write to memory, the CB L2 must de-assert its bus request the cycle after ARTRY#. The CB L2 can then re-assert its bus request[3].

6.7.2.7 CB L2 Timing Diagrams

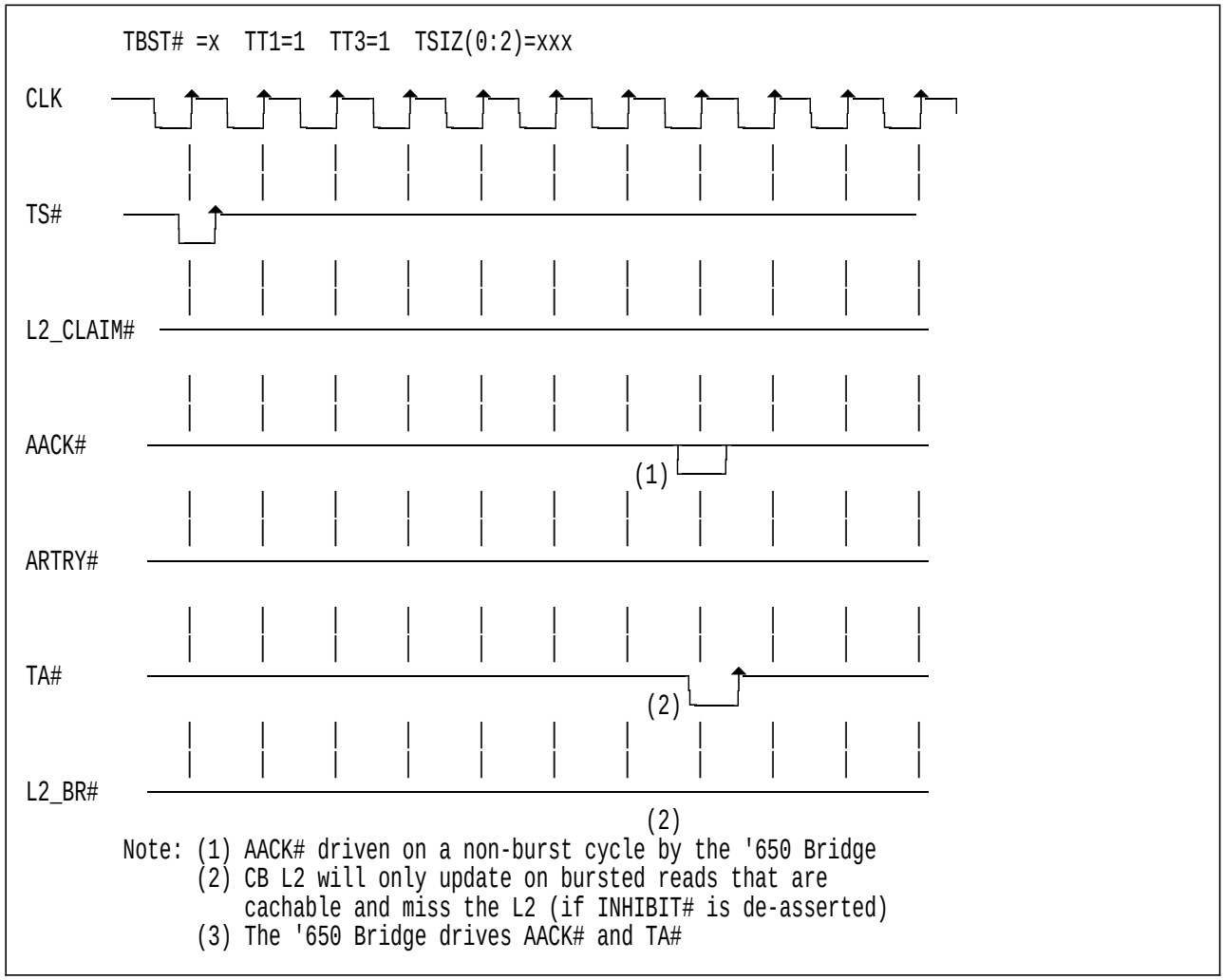


Figure 33. CB L2 Response to Read Misses

[3] Note that, in an implementation which is a single-processor system, the arbiter does not need to see only the one BR# active from the processor. As mentioned earlier, if both the L2_BR# and the 60X_BR# are active, the arbiter will always grant the next cycle to the PowerPC processor. The L2_BR# will then be sampled after the processor cycle is complete to determine if it also needs the bus.

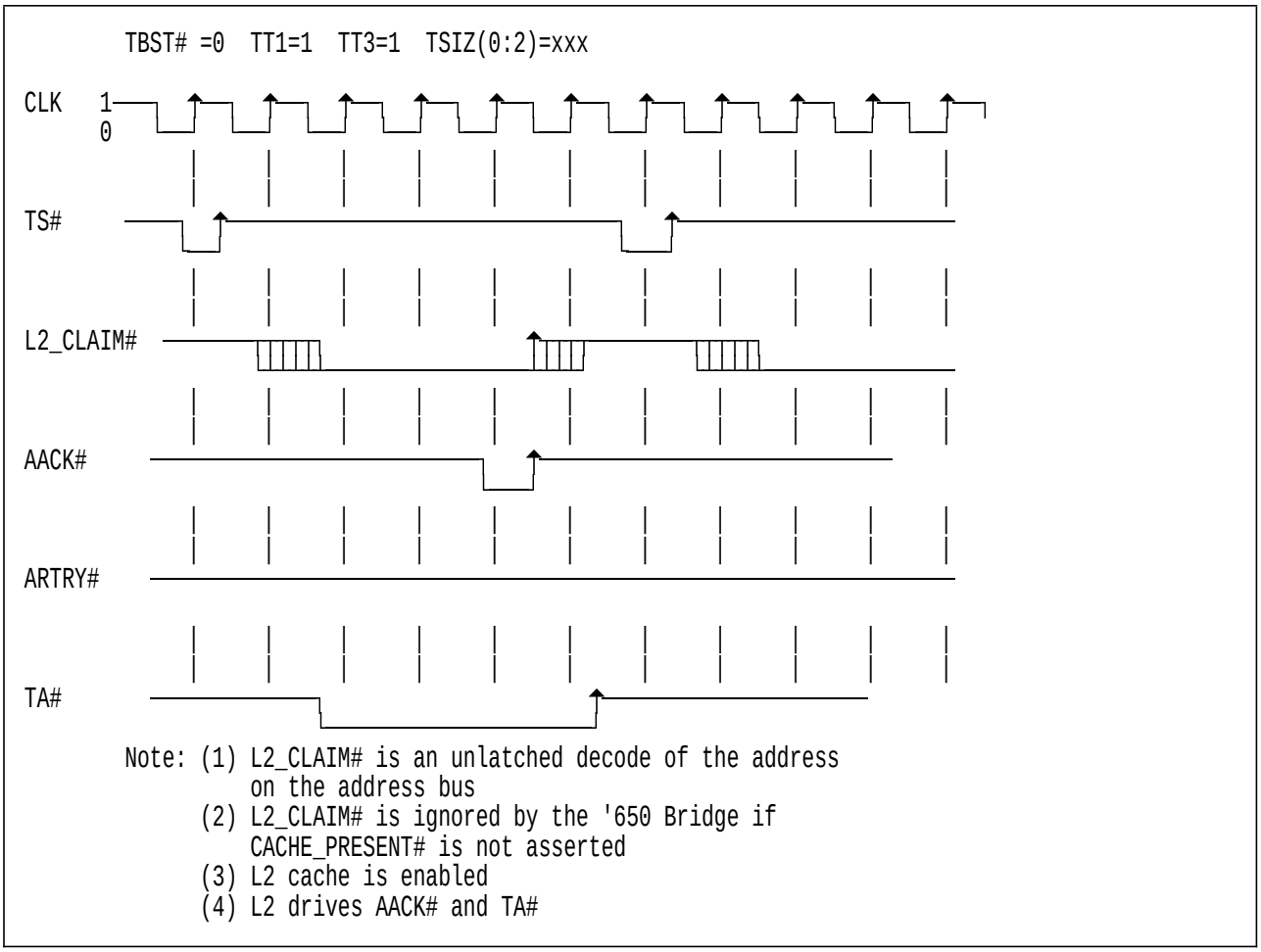


Figure 34. CB L2 Response to Burst READ Hits

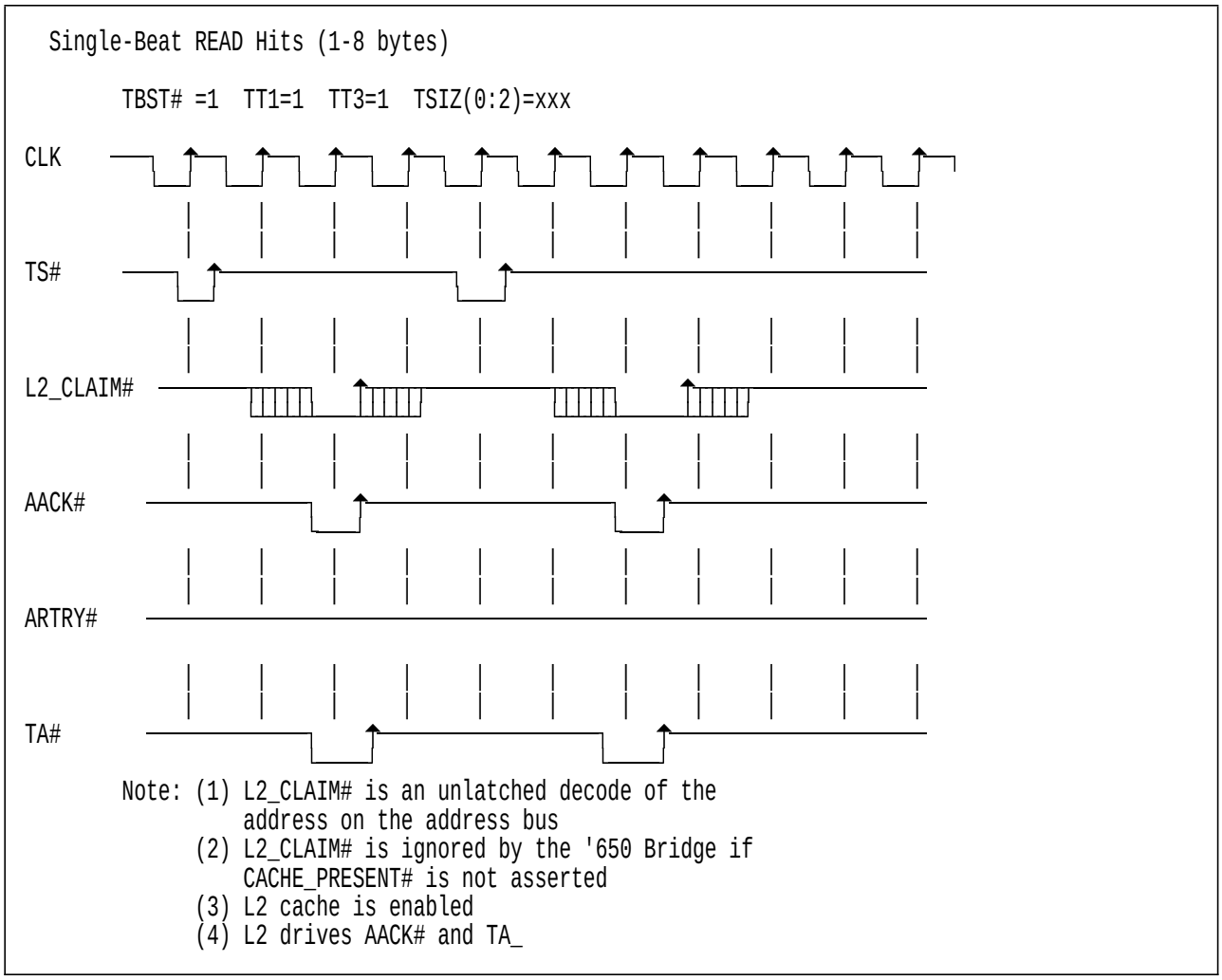


Figure 35. CB L2 Response to Single-Beat READ Hits (1-8 bytes)

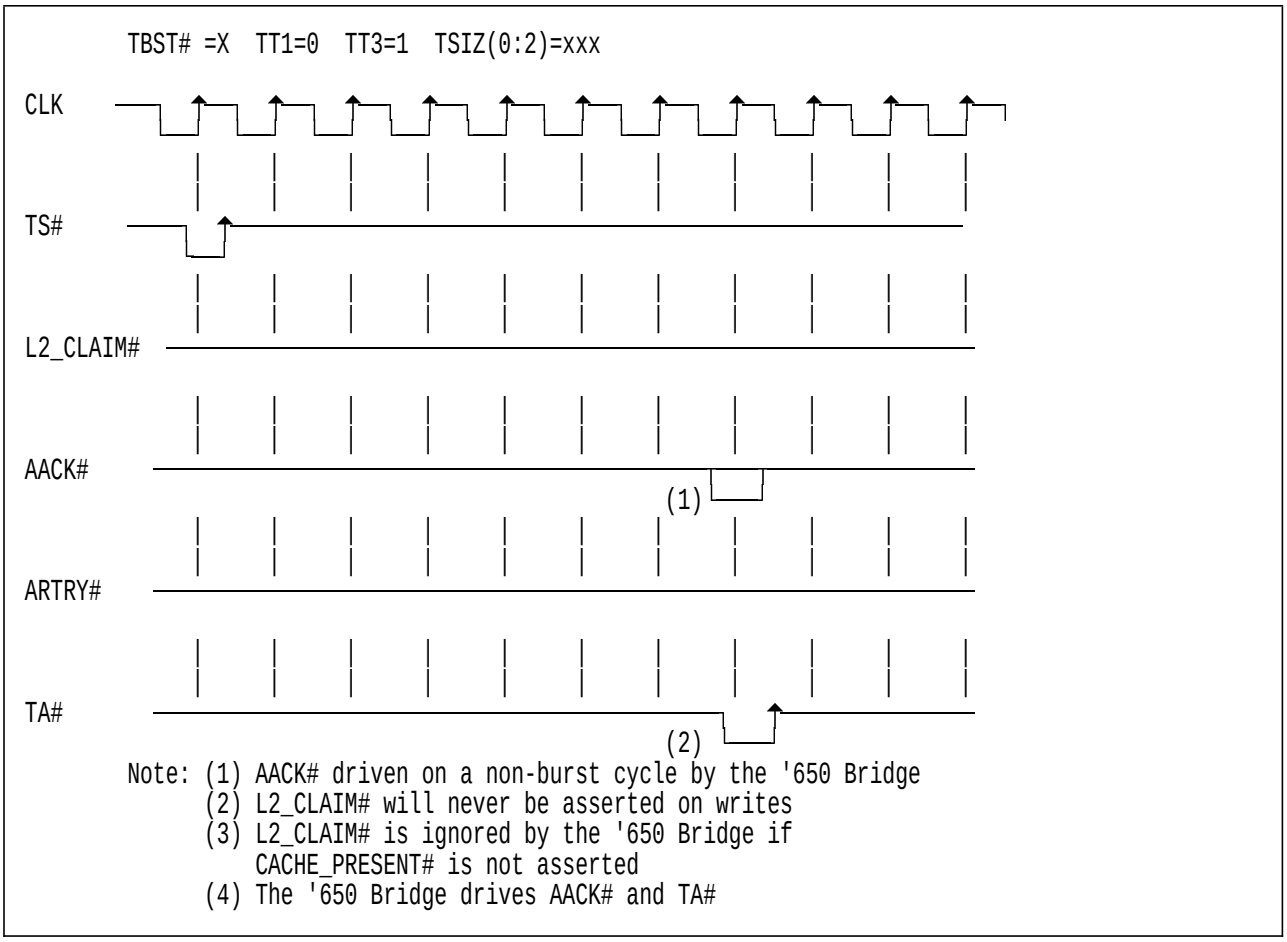


Figure 36. CB L2 Response to Write Cycles

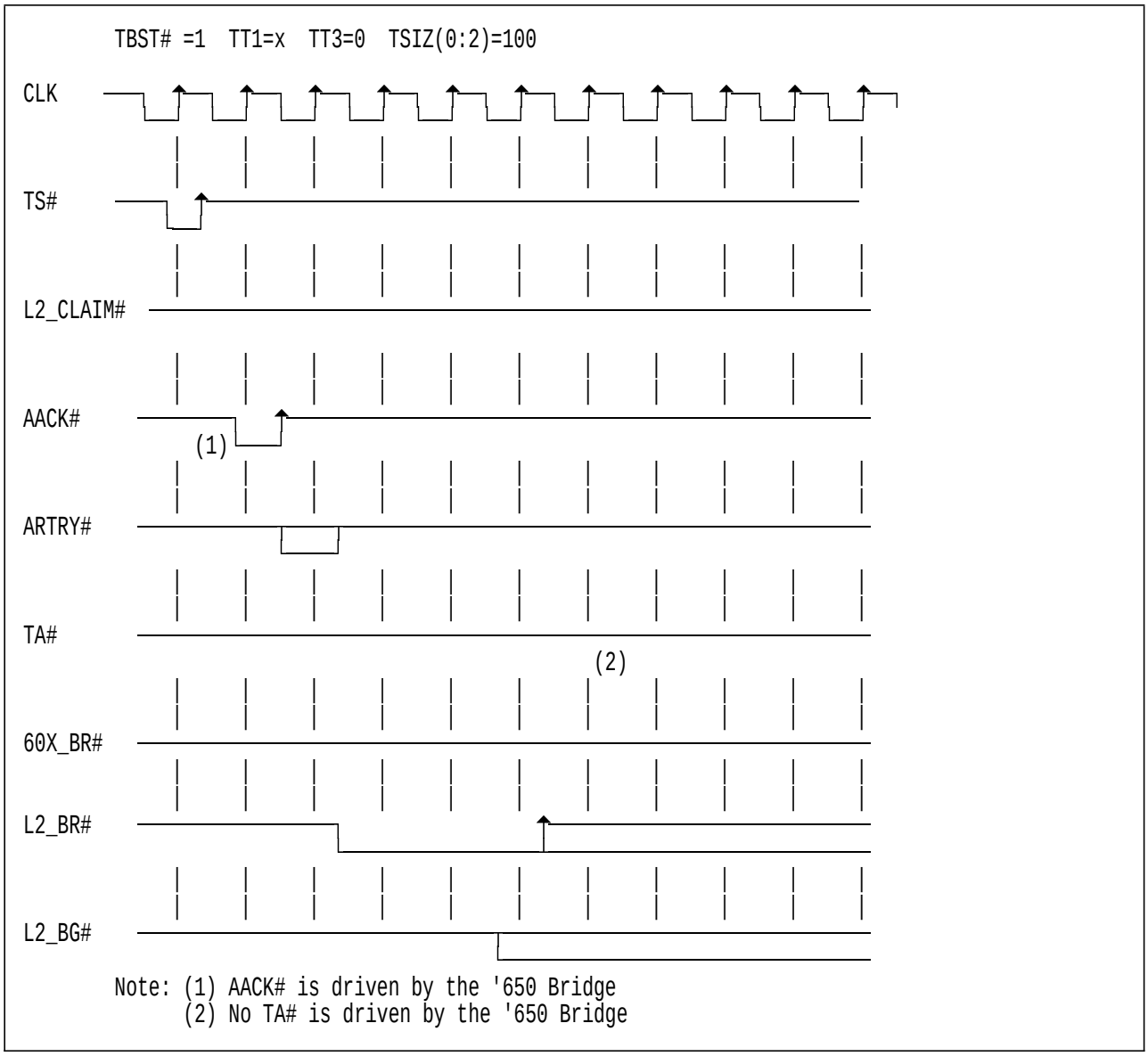


Figure 37. WT L2 Response to I/O Snoop Hits without PowerPC Processor Snoop Hit

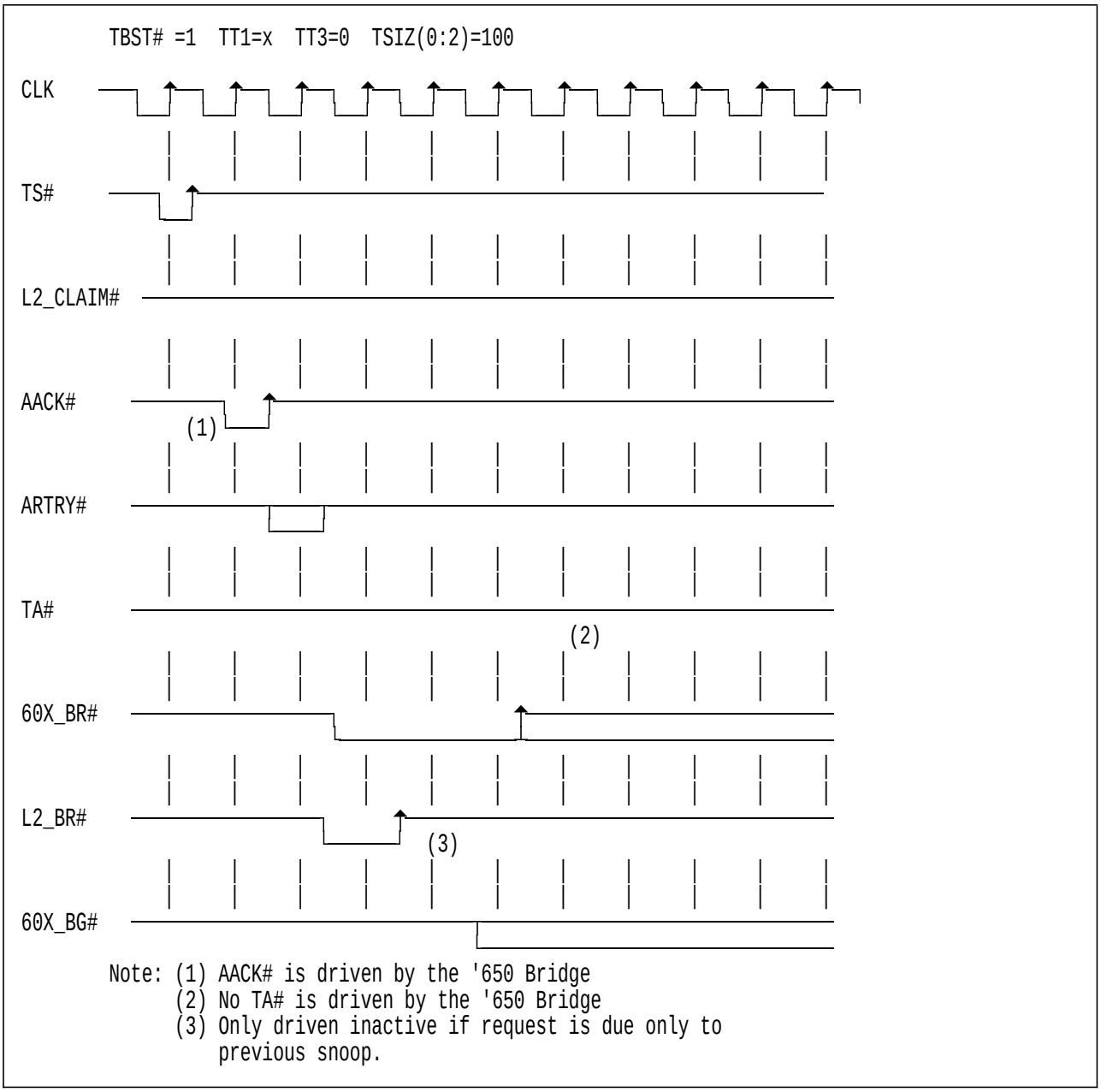
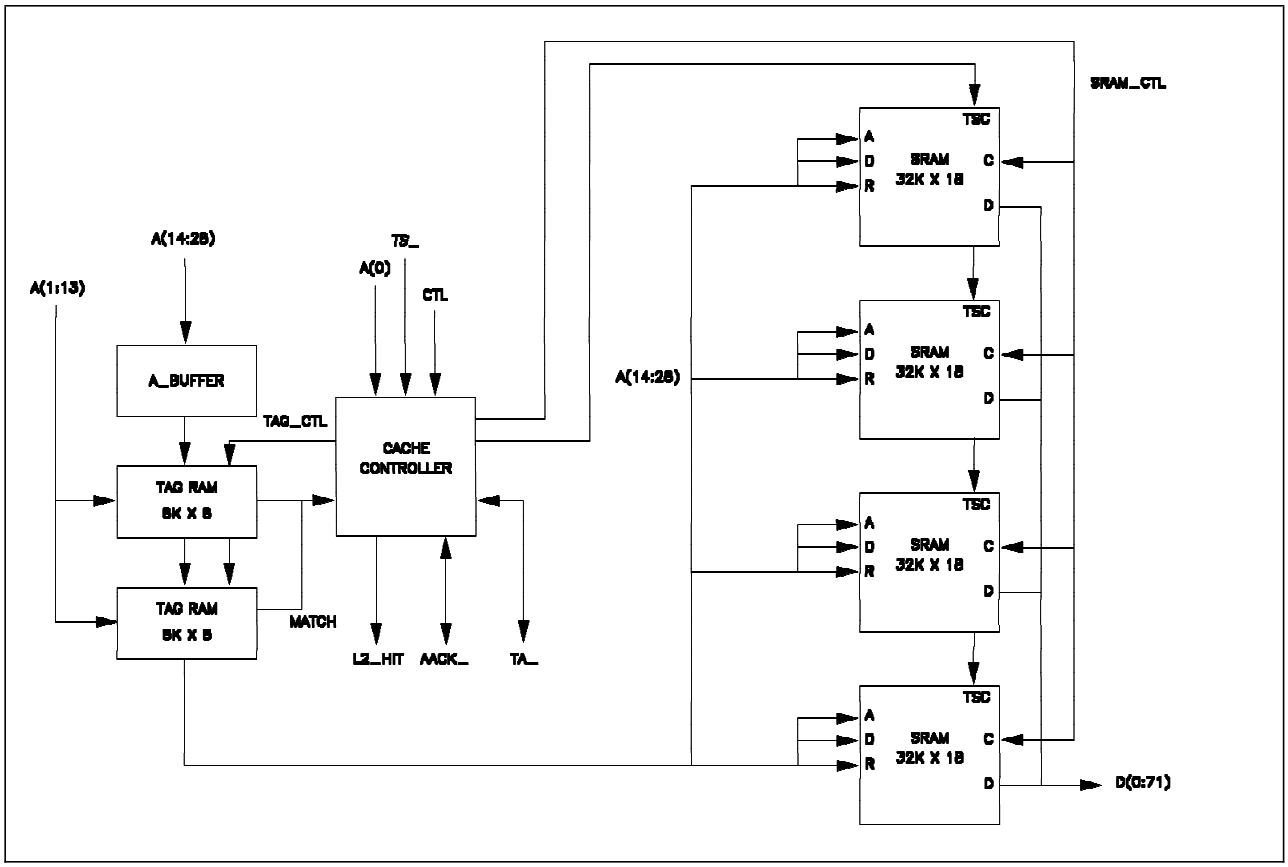


Figure 38. CB L2 Response to I/O Snoop Hits with PowerPC Processor Snoop Hit

6.7.3 L2 Cache Design

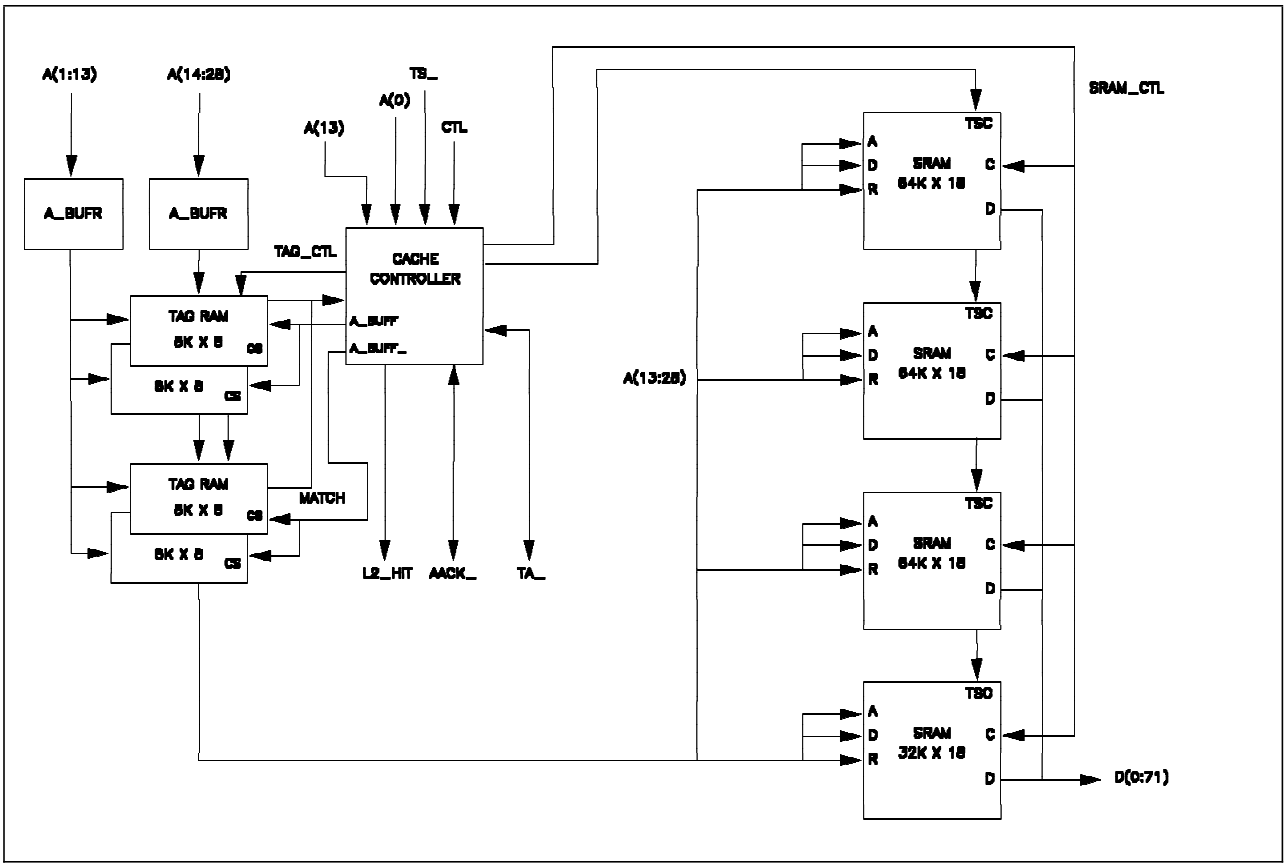
This section defines the design of an L2 Cache. Figure 39 gives a block diagram of the structure for a
/ 256-KB cache, and Figure 40 gives a block diagram for a 512-KB cache. For both designs, the cache con-
/ troller may be an IBM 8184995, and the Tag RAM may be an IDT 71B74 or equivalent. For the 256-KB
/ cache, the SRAMs may be Motorola MCM67M518, which is a 32Kx18 BurstRam, or an equivalent
/ product. The SRAMs for the 512-KB cache may be Motorola MCM67M618, which is a 64Kx18
/ BurstRam, or an equivalent product.



/ Figure 39. 256-KB L2 Block Diagram

The control signals for this cache controller are as follows:

Signals	*Description*
VCC (4 pins)	POWER. Plus-5-volt power to controller.
GND (4 pins)	POWER. GND to controller.
A_IN	INPUT. Address In. Address input for Tag RAM bank select (used only for 512-KB L2).
A_EN	INPUT. Address Buffer Enable. Active (high) enables A_BUFF and A_BUFF_. Inactive (low) disables A_BUFF (low) and A_BUFF_ (high).
A_BUFF	OUTPUT. Address Buffer Output. Non-inverted buffer of A_IN when A_EN is active (high). Low when A_EN is inactive (low).
A_BUFF_	OUTPUT. Address Buffer Output. Inverted buffer of A_IN when A_EN is active (high). High when A_EN is inactive (low).
ARTRY_HIT_	OUTPUT. L2 Hit with ARTRY_ timings. This signal is a synchronous hit signal which goes active (low) after clock 2. The signal can be sampled by the system on clock 3. This signal is valid for one clock cycle only.
OE_	INPUT. When active (low), all outputs of the controller will be driven. When inactive (high), all outputs are tri-stated.
BUS_CLOCK0	INPUT. Bus Clock. Card I/O. This clock drives all state machine changes. All changes occur on the rising edge of this clock.



/ Figure 40. 512-KB L2 Block Diagram

HRESET_ INPUT. Reset. When active (low), this signal resets the controller to its initial states and will generate the control signals to flush the cache Tag RAM.

A0 INPUT. Card I/O. This MSb of the address bus determines if the address is below 2 GB. Used as a qualifier in determining if the L2 will respond to the address presented on the PowerPC processor bus.

L2_CACHE_FLUSH_ INPUT. Card I/O. Same as the card.

L2_CACHE_DIS_ INPUT. Card I/O. Same as the card.

L2_CACHE_INH_ INPUT. Card I/O. Same as the card.

CI_ INPUT. Card I/O. Same as the card.

MATCH INPUT. Active (high) signal from the cache Tag RAM indicates that the address presented on the PowerPC processor bus has been cached in the L2. (This is a “hit.”) Inactive (low) signal indicates that the address is not in the L2 cache. (This is a “miss.”)

L2_CACHE_HIT_ OUTPUT. Card I/O. Same as the card.

TA_ INPUT. PowerPC Processor Transfer Acknowledge. Used to determine when data placed on the data bus is valid.

TA_OUT_ OUTPUT. Active (low) during cache reads to indicate that valid data has been placed on the PowerPC processor bus. This signal MUST be buffered external to the controller with an open-collector (or tri-statable) buffer.

AACK_ INPUT. PowerPC Processor Address Acknowledge. Active during reads of the cache to signal the system that a new address can be placed on the bus. Tri-stated when not active.

TSIZ(2..0) INPUT. PowerPC Processor Transfer Size. Card I/O. Same as the card.

TBST_ INPUT. PowerPC Processor Transfer Burst. Card I/O. Same as the card.

TS_ INPUT. PowerPC Processor Transfer Start. Card I/O. Same as the card.

TT(4..0) INPUT. PowerPC Processor Transfer Type. Card I/O. Same as the card.

TAG_VALID OUTPUT. Is active (high) when entry for Tag RAM is valid. Inactive (low) during a Tag RAM write will invalidate the tag. During a read of the Tag RAM, an active (high) will force the Tag RAM to match only validated memory locations.

TAG_WE_ OUTPUT. Writes the data on the inputs to the Tag RAM when the signal transitions from active (low) to inactive (high).

TAG_RESET_ OUTPUT. When active (low), all tags in the Tag RAM will be reset and invalidated.

SRAM_TSC_ OUTPUT. When active (low), the Burst SRAMs latch address.

SRAM_S1_ OUTPUT. When active (low), the Burst SRAMs are enabled.

SRAM_BAA_ OUTPUT. When active (low), the address in the Burst SRAMs is incremented on the next rising clock edge.

SRAM_W_ OUTPUT. When active (low), the data on the inputs to the Burst SRAMs will be stored at the location pointed to by the address on the next rising edge of the clock.

SRAM_G_ OUTPUT. When active (low), data will be driven by the Burst SRAM on the data lines.

6.7.4 Upgrade Slot Pin Definition

This section defines the pins for the upgrade slot.Figure 41 shows a diagram of this 2x101 Micro Channel-style connector. It consists of two rows of 101 pins. The 101 pins include the 10 pins from VA10 through VA1, 49 pins from A1 through A49, and 42 pins from A52 through A93.

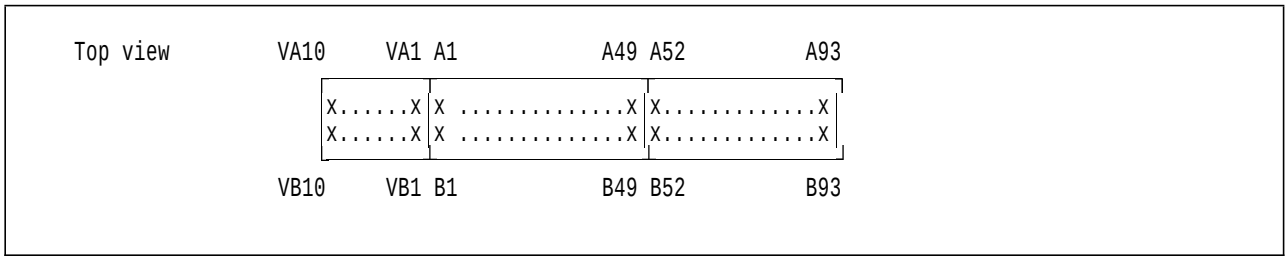


Figure 41. Upgrade/L2 Cache Connector

Table 23 (Page 1 of 4). Upgrade Processor/L2 Cache Connector Pin Assignments

Pin	Function	Pin	Function
A1	L2_CLAIM#	B1	L2_TAG_CLR#
A2	60X_A(9)	B2	DATA_BUS_GNT#, BG_60X#
A3	GROUND	B3	60X_A(10)
A4	60X_A(11)	B4	60X_A(12)

Table 23 (Page 2 of 4). Upgrade Processor/L2 Cache Connector Pin Assignments			
Pin	**Function**	**Pin**	**Function**
A5	60X_A(13)	B5	GROUND
A6	60X_A(14)	B6	60X_A(15)
A7	+5 VOLTS	B7	60X_A(16)
A8	60X_A(18)	B8	60X_A(17)
A9	60X_A(19)	B9	+5 VOLTS
A10	60X_A(0)	B10	60X_A(20)
A11	GROUND	B11	60X_A(1)
A12	60X_A(2)	B12	60X_A(3)
A13	60X_A(4)	B13	GROUND
A14	60X_A(5)	B14	60X_A(6)
A15	+5 VOLTS	B15	60X_A(7)
A16	60X_A(21)	B16	60X_A(8)
A17	60X_A(22)	B17	+5 VOLTS
A18	60X_A(23)	B18	60X_A(24)
A19	GROUND	B19	60X_A(25)
A20	60X_A(26)	B20	60X_A(27)
A21	60X_A(28)	B21	GROUND
A22	60X_A(29)	B22	60X_A(30)
A23	+5 VOLTS	B23	60X_A(31)
/ A24	GROUND	B24	GROUND
/ A25	TRANSFER TYPE (4), TT(4)	B25	GROUND
A26	TRANSFER TYPE (2), TT(2)	B26	TRANSFER TYPE (3), TT(3)
A27	TRANSFER TYPE (0), TT(0)	B27	TRANSFER TYPE (1), TT(1)
A28	TRANSFER SIZE (1), TSIZ(1)	B28	TRANSFER SIZE (2), TSIZ(2)
A29	TRANSFER_BURST#, TBST#	B29	TRANSFER SIZE (0), TSIZ(0)
A30	TRANSFER_ACKNOWLEDGE#, TA#	B30	GROUND
A31	GROUND	B31	ADDRESS_ACKNOWLEDGE#, AACK#
A32	DATA_RETRY#, DRTRY#	B32	ADDRESS_RETRY#, ARTRY#
A33	TRANSFER_ERR_ACK#, TEA#	B33	EXT_ADDR_TRANS_START#, XATS#
A34	GROUND	B34	GROUND
A35	BUS_REQUEST_60X#, BR_60X#	B35	DATA_PARITY_ERROR#, DPE#
A36	BUS_REQUEST_L2#, BR_L2#	B36	BUS_GRANT_60X#, BG_60X#
A37	+5 VOLTS	B37	BUS_GRANT_L2#, BG_L2#
A38	TRANSFER_START#, TS#	B38	60X_INTERRUPT#, INT_60X
A39	GROUND	B39	+5 VOLTS
/ A40	GROUND	B40	GROUND
A41	GROUND	B41	BUS_CLOCK_0
A42	BUS_CLOCK_1	B42	GROUND
A43	GROUND	B43	BUS_CLOCK_2
A44	60X_D(31)	B44	+5 VOLTS
/ A45	60X_D(30)	B45	GROUND
A46	60X_D(28)	B46	60X_D(29)
A47	GROUND	B47	60X_D(27)
A48	60X_D(25)	B48	60X_D(26)

Table 23 (Page 3 of 4). Upgrade Processor/L2 Cache Connector Pin Assignments			
Pin	**Function**	**Pin**	**Function**
A49	60X_D(24)	B49	GROUND
A50	<key>	B50	<key>
A51	<key>	B51	<key>
A52	60X_D(22)	B52	60X_D(23)
A53	+5 VOLTS	B53	60X_D(21)
A54	60X_D(19)	B54	60X_D(20)
A55	60X_D(18)	B55	+5 VOLTS
A56	60X_D(16)	B56	60X_D(17)
A57	GROUND	B57	60X_D(15)
A58	60X_D(13)	B58	60X_D(14)
A59	60X_D(12)	B59	GROUND
A60	60X_D(10)	B60	60X_D(11)
A61	GROUND	B61	60X_D(9)
A62	60X_D(7)	B62	60X_D(8)
A63	60X_D(6)	B63	+5 VOLTS
A64	60X_D(5)	B64	60X_D(4)
A65	+5 VOLTS	B65	60X_D(3)
A66	60X_D(1)	B66	60X_D(2)
A67	60X_D(0)	B67	GROUND
A68	60X_D(62)	B68	60X_D(63)
A69	60X_D(61)	B69	+5 VOLTS
A70	60X_D(59)	B70	60X_D(60)
A71	GROUND	B71	60X_D(58)
A72	60X_D(56)	B72	60X_D(57)
A73	60X_D(55)	B73	GROUND
A74	60X_D(53)	B74	60X_D(54)
A75	+5 VOLTS	B75	60X_D(52)
A76	60X_D(50)	B76	60X_D(51)
A77	60X_D(49)	B77	GROUND
A78	60X_D(47)	B78	60X_D(48)
A79	GROUND	B79	60X_D(46)
A80	60X_D(44)	B80	60X_D(45)
A81	60X_D(43)	B81	+5 VOLTS
A82	60X_D(42)	B82	60X_D(41)
A83	+5 VOLTS	B83	60X_D(40)
A84	60X_D(38)	B84	60X_D(39)
A85	60X_D(37)	B85	GROUND
A86	60X_D(35)	B86	60X_D(36)
A87	+5 VOLTS	B87	60X_D(34)
A88	60X_D(32)	B88	60X_D(33)
A89	GROUND	B89	+5 VOLTS
A90	60X_DATA_PARITY_(6)	B90	60X_DATA_PARITY_(7)
A91	60X_DATA_PARITY_(4)	B91	60X_DATA_PARITY_(5)
A92	60X_DATA_PARITY_(2)	B92	60X_DATA_PARITY_(3)

Table 23 (Page 4 of 4). Upgrade Processor/L2 Cache Connector Pin Assignments			
Pin	**Function**	**Pin**	**Function**
A93	60X_DATA_PARITY_(0)	B93	60X_DATA_PARITY_(1)
VA1	GROUND	VB1	FULL_SPEED
VA2	+5 VOLTS	VB2	60X_CACHE_INHIBIT#, CI#
VA3	60X_WRITE_THROUGH#, WT#	VB3	60X_GLOBAL#, GBL#
VA4	60X_SHARED#, SHD#	VB4	BCLK_SPEED1
VA5	BCLK_SPEED0	VB5	HARD_RESET#, HRESET#
VA6	SOFT_RESET#, SRESET#	VB6	UPGRADE_PROCESSOR_PRESENT#
VA7	L2_CACHE_PRESENT#	VB7	+5 VOLTS
VA8	L2_CACHE_256K	VB8	L2_CACHE_CB (COPY-BACK)
VA9	L2_UPDATE_INH#	VB9	L2_MISS_INH_#
VA10	TRANSFER_CODE_(1), TC(1)	VB10	TRANSFER_CODE_(0), TC(0)

Appendix A. Implementation Examples

The following subsections give diagrams and brief descriptions of example implementations for a portable, an energy-managed workstation, a medialess workstation, a technical workstation, a server, and a symmetric
/ multiprocessor. A description of a diagnostic strategy is also included.

A.1 Portable

This appendix gives a basic hardware overview of a PowerPC Reference Platform-compliant portable system. Figure 42 shows a schematic diagram of the components of this portable system. This portable
| configuration uses the PowerPC 603 processor, and provides one serial port, one parallel port, various audio
| jacks, built-in microphone and speakers, one keyboard/mouse port, one video graphics port, one SCSI-2
| port, and two type-2 PCMCIA sockets.

A.1.1 Subsystems

Major subsystems are connected to the internal +5.0-volt, 32-bit PCI bus. A memory controller, PCI-ISA Bridge, SCSI controller, power management controller, and video graphics controllers are located on this bus.

Native and extended I/O controllers, Real-Time Clock, business audio, and PCMCIA are located on the internal ISA bus.

A.1.2 PowerPC 603 Processor

| The PowerPC 603 processor is configured to have 32 address bits and and can be configured to have 32 data
| bits. Internally, the 603 has an 8-KB instruction cache and an 8-KB data cache.

The 603 processor has a number of differences from the 601. Please consult the 603 documentation for details. Two changes in particular should be noted: the 603 does not snoop the internal instruction cache, and the 603 Endian switching instructions are different than the 601 instructions.

The following process will switch the system from Big-Endian to Little-Endian mode when used in the Ref-
/ erence Implementation with a PowerPC processor other than the PowerPC 601 microprocessor. Because
/ this process must execute during transition periods when the processor and system components are in dif-
/ ferent Endian modes, care must be taken to assure that interrupts are not taken or that data and instructions
/ do not remain in cache in the wrong Endian order. The instructions to perform this switch must not span a
/ page boundary. The process is outlined below:

/ a) Enable address translation
b) Flush all system caches
c) Disable interrupts
/ d) Enter supervisor state
/ e) Invalidate the instruction cache
f) Set the processor and system state to Little-Endian (see Figure 43)

Note: Processor and system are now in Little-Endian mode. All instructions must be in Little-Endian order.

g) Put interrupt handlers and processor data structures in Little-Endian format
h) Enable interrupts
i) Start the Little-Endian operating system initialization

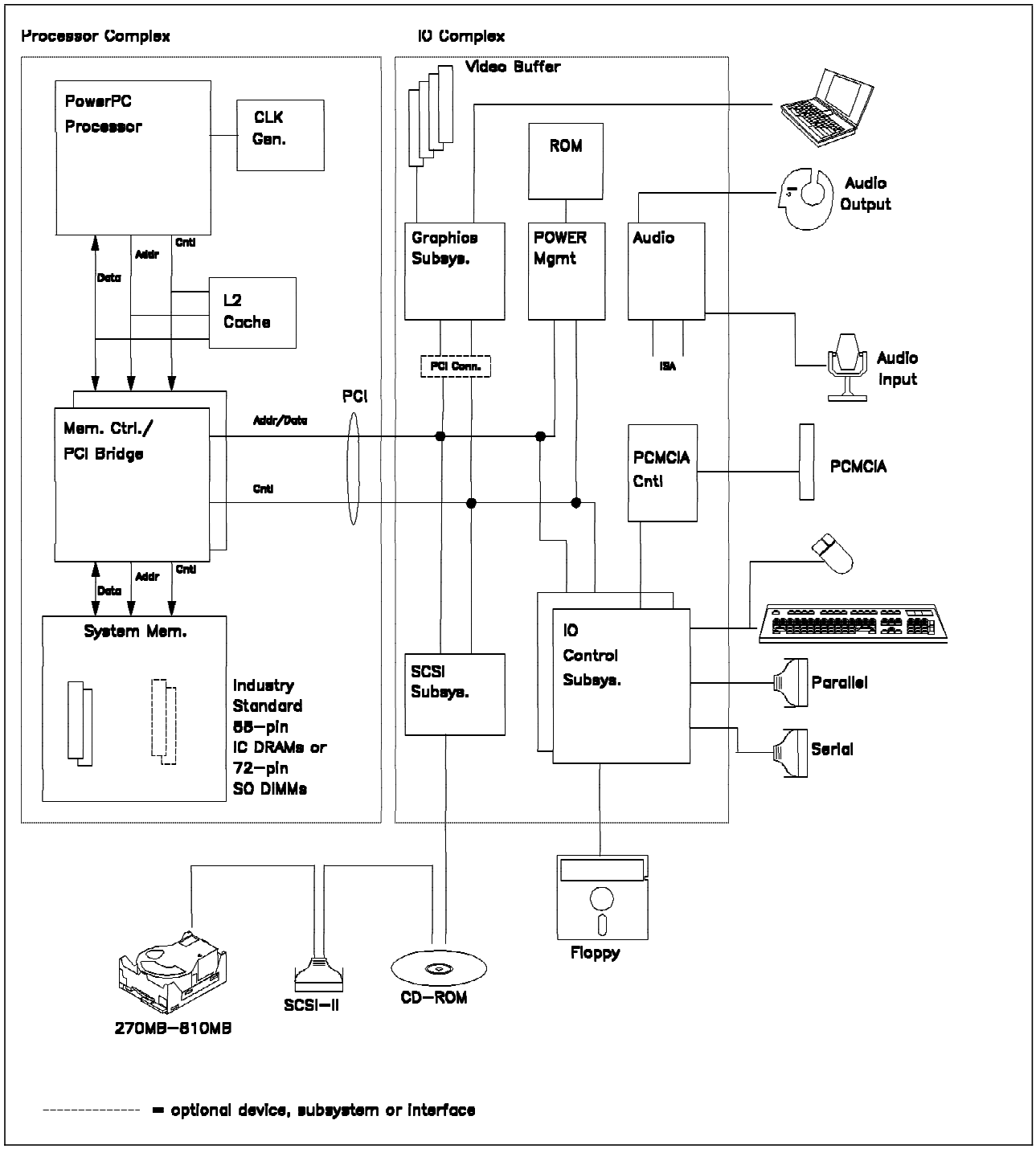


Figure 42. PowerPC Reference Platform Recommended Portable System

```
/   ;Set the MSR ILE bit (bit 15) here.  It is not copied from SRR1.
       x00  mfmsr      R9   ;Load current MSR register
       x04  ori  R9,R9,MSR_LE    ;Set the LE bit in the MSR (bit 31)
       x08  mtsrr1     R9   ;Store the target MSR in SRR1
/      xxC  bl   x     ;Address of next instruction in Link Register
/   x:
/      x10  mfspr R2,8      ;Link register to R2
/      x14  addi R2,R2,y-x  ;Calculate instruction after RFI
/      x18  mtsrr0     R2   ;Set branch address in SRR0
       x1C  rfi        ;Return from interrupt to set LE mode
                  ;The RFI performs the necessary synchronization actions
    y:
       x20  stb  R5,0(R29)  ;Set system components to LE
/   ;Register R5 has the data and R29 has the address for the Endian control port
/   ;Endian control port at X'8000 0092' must be addressed at
/   ;X'8000 0095' because processor is modifying addresses now
    ;To this point all instructions are in Big-Endian Format
/   ;Include a string of palindromic instructions to pass time until the system
/   ;completes the switch.  Twenty five are suggested based on a 66 MHz processor
/      x24  addi R0,R1,0x138     ;Instructions which work LE or BE
/                        ;This instruction generates 0x38010138
/      x28  addi R0,R1,0x138
/      x2C  addi R0,R1,0x138
/      ...
/      xyy  addi R0,R1,0x138
/   ;Start of Little Endian instructions
```

Figure 43. PowerPC Instruction Stream to Switch Endian Modes

/ This switching process is only applicable for PowerPC processors in a design with a Bi-Endian memory as / shown in Figure 56.

/ If the design uses the Big-Endian memory approach as shown in Figure 57, then this process must be modi- / fied. The Little-Endian portion of the operating system loader and any of the rest of the operating system / which was brought into memory before the system was switched to Little-Endian mode does not reside in / memory in the correct format. This information was brought into memory in the true Little-Endian format / as it existed on the media. It must exist in memory in the “PowerPC Little-Endian” form. This form has / the bytes reversed within each doubleword. After the system is in Little-Endian mode, further input from / media of Little-Endian information will arrive in system memory in the correct form. Examples of / approaches to accomplish this byte reversal are as follows:

/ • Before the processor switches to Little-Endian mode, load and store each Little-Endian doubleword, / doing a byte reversal in the process.
/ • Swap out the Little-Endian material while in the Big-Endian mode and swap it back in after the system / is in Little-Endian mode and before the processor has been switched to Little-Endian mode. The / instructions to perform the swap-in must have been byte reversed.
/ • During the install process, byte reverse each doubleword of Little-Endian information on the media. / This format cannot be moved to an implementation with Bi-Endian memory.

A.1.3 Memory

The memory is composed of base memory and optional upgradable memory. The system board has socket(s) for JEIDA 88-pin IC DRAM cards or 72-pin SODIMMs (small outline dual inline memory modules) for upgrade memory.

Both base and optional memories are +5-volt, 32-bit and support data parity but do not support ECC. In power-down modes, the memory is refreshed at a slow rate to sustain data.

A.1.4 PCI Bridge/Memory Controller (PCIB/MC)

The portable system uses a custom-designed controller that bridges between the 603, memory, and PCI buses.

A.1.5 Power Management Controller

The portable system uses a custom-designed controller that performs low-level power management and PCI bus arbitration. System ROM is connected through this controller to the PCI bus.

A.1.6 Flash ROM -- 512 KB (AMD Am29F040-120)

This component contains the POST and boot code. It is recommended that the system board speed, native I/O complement, and like information be programmed into this device. There is no other source of such information built into the system board. Algorithms for programming this data into Flash before or during the manufacturing process are possible.

After power on, the processor fetches the initial code from this device. A maximum of 16 MB for System ROM and registers is allowed in the Reference Implementation memory map, but the system board utilizes a 512-KB device for the System ROM.

A.1.7 SCSI Controller -- NCR 53C810

This 8-bit SCSI-2 controller supports 32-bit word data burst transfers with variable burst lengths as a PCI master device. A high-level SCSI processor requires very little 603 overhead in operations. A 50-pin single-ended SCSI connector is available. On the SCSI bus, a hard disk drive pack and CD-ROM drive are connected. The bus supports data transfer speeds of up to 5 MB/s and 10 MB/s in asynchronous and synchronous modes, respectively.

A.1.8 PCI-ISA Bridge -- Intel 82378ZB System I/O

In addition to ISA bus bridging, this controller provides the following system services:

- Two 83C37 DMA controllers
- 82C54 timer
- Two 8259 interrupt controllers
- System control ports

The PCI-ISA bridge positively decodes PCI cycles for selected addresses, and subtractively decodes unclaimed PCI cycles for the internal ISA bus.

The BIOS Timer address range is relocated to avoid conflicts with the EEPROM, which resides in ISA address range 0078h through 007Ch. The PCI bus arbitration is achieved by the power management controller, but not by the PCI-ISA bridge, on the portable system.

A.1.9 Native I/O Controller -- National Semiconductor PC87322 Super I/O

This ISA device contains a floppy disk controller, serial port controllers, and a parallel port controller.

The floppy disk controller is downward compatible to µPD765 and N82077. Through a 26-pin port, an external floppy disk drive is supported.

On this portable system, one of two serial port controllers is used and connected to a 9-pin D-sub connector. The controllers are compatible to PC16550s.

The bi-directional parallel port controller is compatible to IEEE 1284, and is connected to a 25-pin D-sub connector.

A.1.10 Extended I/O Controller

The portable system uses an I/O controller, which supports variable I/O functions. Most of system control registers are contained in this controller.

A PS/2-compatible auxiliary device controller provides two channels of serial interfaces: one for alphanumeric devices, the other for pointing devices. The serial interface for alphanumeric devices is directly and solely connected to the universal micro control unit (UMCU). The UMCU is equivalent to an 84-, 85-, or 89-key space-saving keyboard. The UMCU has an additional serial interface for external alphanumeric devices.

A serial interface is connected to pointing devices. A keyboard/mouse port provides serial interfaces for external alphanumeric devices and pointing devices.

The extended I/O controller supports a PS/2 Type-A EEPROM (64x16) for storing security passwords, vital product data, and some configuration information.

A.1.11 Real-Time Clock -- Dallas Semiconductor DS1585S

This ISA device has a Real-Time Clock in a 128-byte CMOS address space and an 8-KB NVRAM space.

A.1.12 Business Audio -- Crystal Semiconductor CS4231

Business Audio is provided through the Crystal Semiconductor CS4231 stereo audio integrated circuit. This device is the same IC used in the Reference Implementation and is described in Section 6.3.4, “Audio Subsystem.” One microphone and a pair of stereo speakers are provided as built-in features. Also supported are four 3.5-mm jacks for:

- Stereo earphones
- Stereo microphone input
- Stereo line in
- Stereo line out

Also, the conventional (Timer 2) PC speaker functions, UMCU (Universal Micro Control Unit) beeps, PCMCIA audio, and CD audio functions are supported for audio application and alert.

A.1.13 PCMCIA Controller -- Ricoh RF5C366C

This ISA device supports PCMCIA 2.0 sockets. The sockets can provide +5 volts and +12 volts as Vpp, and support +5-volt signal interfaces.

A.1.14 LCD Display

The LCD display has a minimum 640x480 resolution.

A Western Digital 90C24A2 graphics controller is connected to the internal PCI bus through a bus bridge and supports 1 MB of Video RAM. The graphics subsystem supports an 18-bit RAMDAC that provides for connection through a standard 15-pin D-sub connector of an external CRT display.

A.1.15 L2 Cache

A look-aside L2 cache is mounted on the PowerPC 603 bus.

A.2 Energy-Managed Workstation

This appendix gives a basic hardware overview of a PowerPC Reference Platform-compliant energy-managed workstation. Figure 44 shows a diagram of the components of this energy-managed workstation. The basic hardware consists of a system board, power supply, diskette drive, SCSI disk drive, CD-ROM, cables, keyboard, monitor, and mechanical package. This energy-managed workstation configuration will use the PowerPC 603 processor, and will provide a SCSI external port, two serial ports, one parallel port, business audio with a stereo microphone jack, stereo earphone jack, line in and line out jacks, video graphics, keyboard and mouse jacks, and four PCMCIA slots. There are no ISA or PCI expansion card sockets. Base memory support is 16 MB with expansion capability to a maximum of 80 MB of System Memory. L2 cache or processor upgrades are not described, but provisions for them are included in the configuration. Video is provided by a PCI-based S3-928 daughter card. The subsections below define the major components of this system.

A.2.1 System Board

The system board contains most of the electronics for the system. Major subsystems are connected to main memory through the PCI bus. The memory controller, System I/O, and SCSI are located on this bus. The video is attached to the PCI bus through a special socket, allowing for future upgrades.

The board is designed with a space-saving layout, using both a CPU card and a base I/O card. The boards require +5 volts and +3.3 volts to power most of the components. PCMCIA Flash programming features which require +12 volts are also supported by the power supply.

A.2.2 PowerPC 603 Processor

The PowerPC 603 processor is available in operating frequencies of 50, 66 and 75 MHz. There are 32 address bits and 64 or 32 data bits. It can be configured as either a 32-bit data device or a 64-bit data device via a pin. The 603 contains a 16-KB internal cache which is split between instruction and data. The 603 package uses 240 pins and requires +3.3-volt power.

The PowerPC 603 processor has a number of differences from the PowerPC 601 processor. Please consult the 603 documentation for details. Two changes in particular should be noted: the PowerPC 603 does not snoop the internal instruction cache, and for the PowerPC 603 and any PowerPC processor other than the PowerPC 601, the Endian switching instructions are different than those shown in the desktop Reference Implementation. Refer to Section A.1.2, “PowerPC 603 Processor,” for suggested Endian switching logic.

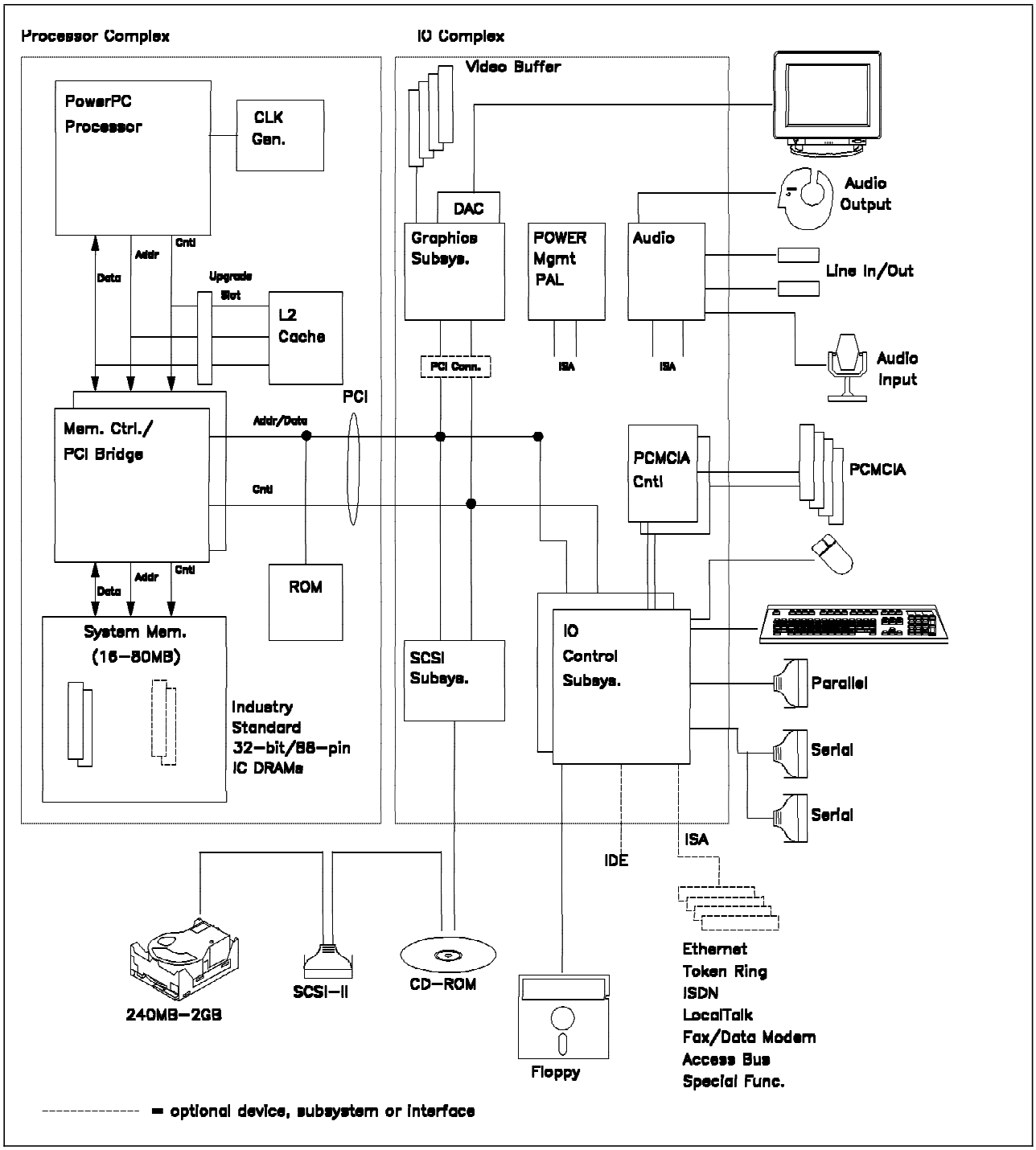


Figure 44. PowerPC Reference Platform Recommended Energy-Managed Workstation

A.2.3 Memory Subsystem

The memory subsystem is composed of base memory and optional upgradable memory. Base memory consists of 16 MB parity memory soldered directly on the system board. Upgrade memory consists of 2 JEIDA 88-pin IC DRAM card sockets. The memory controller has the following features:

- Does not support ECC memories
- Supports only 4-, 8-, 16- and 32-MB IC DRAM Cards

- Supports only non-interleaved memory access operation
- I/O Memory on the PCI bus is not cachable by the processor

A.2.4 PCI Bridge/Memory Controller (PCIB/MC)

The energy-managed system uses a custom-designed memory/bridge controller which provides the following functions:

- Processor bus cycles converted to the following:
  - System Memory cycles
  - PCI I/O cycles
  - PCI memory cycles
  - PCI interrupt acknowledgement
  - PCI configuration cycles
  - Address translation consistent with the memory map of the Reference Implementation
- PCI master access to System Memory
- Processor bus snoop cycles as a result of PCI System Memory access
- System Memory programmability for flexible memory device support

A.2.5 System I/O Bridge Chip -- SIO (Intel 82378ZB)

This function is provided by an Intel 82378ZB. It provides a PCI-to-ISA bus bridge where the native I/O and ISA attachments reside and it provides system services such as DMA and interrupt control. Its major functions are listed below.

- Bridge between PCI and ISA
  - Eight- or 16-bit ISA devices
  - Twenty-four-bit addressing on ISA
  - Partially decodes Native I/O addresses
  - Unclaimed PCI memory address below 16 MB -> ISA
  - Unclaimed PCI I/O address below 64 KB -> ISA
  - Powers up to an “open” condition
  - Generates ISA clock, programmable /3 or /4 divide ratio
  - Allows ISA mastering and has programmable decodes which map ISA memory cycles to the PCI bus
  - Has 32-bit posted memory write data buffer, no I/O buffering
- Seven-channel DMA between ISA devices and PCI memory
  - Eight- or 16-bit devices on ISA (and PCMCIA) bus only
  - Thirty-two-bit addressing at DMA
  - Function of two 83C37’s
  - Eight-byte bidirectional buffer for DMA data
  - Supports Scatter-Gather DMA operations
  - Supports Guaranteed Access Time Mode for DMA and ISA masters
- Timer block -- function of an 82C54
- Interrupt controller -- function of two 8259’s
- PCI bus arbiter -- unused in the energy-managed system
- Functions as PCI slave during programming and ISA slave cycles, as bus master during DMA (or ISA master cycles)

A.2.6 Native I/O Controller -- National PC87322 SUPER I/O

Control is provided via the ISA bus. This component contains:

- Floppy disk controller -- software compatible with DP8473, 765A, and NS82077.
- Two serial ports -- software compatible with INS8250N-B, PC16550A, and PC16450. The serial ports contain FIFOs.
- One parallel port -- bidirectional, EPP compatible.
- IDE interface -- unused on the energy-managed system.

A.2.7 SCSI NCR 53C810

This component attaches directly to the PCI bus on the system board and supports the following features:

- Eight-bit SCSI-2 interface
- Variable block size and scatter/gather data transfers
- Thirty-two-bit word data bursts with variable burst lengths
- Full 32-bit PCI bus master
- Sixty-four-byte FIFO buffer

A.2.8 Real-Time Clock (RTC) -- Dallas Semiconductor DS1585S

Functions of the RTC component include:

- RTC Function (PC compatible)
- Eight-KB Non-Volatile RAM
- Separate, replaceable battery
- Power management wake alarm interrupt

A.2.9 Keyboard/Mouse -- Intel 8042AH

This component contains the keyboard and mouse controls and is connected to the X bus (a buffered 8-bit subset of the ISA bus).

A.2.10 Flash ROM -- 512 KB (AMD Am29F040-120) / EPROM Alternative

This component contains the POST and boot code. It is recommended that the system board speed, native I/O complement, and like information be programmed into this device. There is no other source of such information built into the system board. Algorithms for programming this data into Flash before or during the manufacturing process are possible.

After power on, the processor fetches the initial code from this device. A maximum of 16 MB for System ROM and Registers is allowed in the Reference Implementation memory map, but the system board utilizes a 512-KB device for the System ROM.

A jumper is provided which allows use of an EPROM (AMD 27C040-150JC -- 512 KB) in place of the Flash for systems that do not wish to bear the cost of Flash. The jumper must be present on all boards. Alternatively, 256-KB EPROMs may also be utilized.

A.2.11 Clock Generation

The 603 processor has a very simple clock requirement; therefore, the system board clock circuits are much simpler than in the Reference Implementation described in Section 6.2.5, “Clock Generation.”

The PowerPC 603 processor requires a single system clock input. This input sets the frequency of operations for the bus interface. Internally, the 603 uses a phase-lock loop (PLL) circuit to generate a master clock for all of the CPU circuits. This PLL circuit is locked to the clock input and may be set as an integer multiple of the clock input.

A.2.12 Business Audio

Business audio is provided through the Crystal Semiconductor CS4231 stereo audio integrated circuit. This is the same IC used in the Reference Implementation and is described in section 6.3.4, “Audio Subsystem.” Conventional (Timer 2) PC speaker functions are also provided, as well as support for audio from the PCMCIA connectors. Also supported are four rear-mounted 3.5 mm jacks for:

- Stereo earphones (the speaker is muted when an earphone jack is inserted into the connector)
- Stereo microphone input
- Stereo line in
- Stereo line out

There is also a system-board-mounted connector (with cable) for direct playback from the CD-ROM.

A.2.13 Timer 2 Audio Support

The Timer 2 signal is summed with the outputs of the CS4231 audio IC at the operational amplifier which drives the speaker. This provides the capability of supporting standard audio. Software may elect to directly drive Timer 2 audio or to emulate Timer 2 audio through the CS4231 audio IC.

A.2.14 Processor Time Base Support

The 603 processor’s Time Base function operates at one-fourth of the processor bus clock rate, which may be either 25 or 33 MHz. This Time Base and decrementer function is distinct from the calendar and RTC function normally found in PCs.

A.2.15 L2 Cache and Upgrade Processor Support

The system board has support for an upgrade socket, but no upgrade boards are defined for an L2 cache or processor upgrade for this system. The 603 bus signals and various control signals such as presence detect signals and cache control signals are wired to the socket. Support is included on the system board for:

- Write-through or copy-back cache cards. (Copy-back cache support is tentative at this writing.)
- Two cache type ID bits in a system board register.
- Various cache controls such as flush and disable.
- Arbitration for the upgrade processor or copy-back cache.

A.2.16 I/O Decoder

This component resides on the X bus. It receives partial decode signals from the SIO chip and further decodes these to produce chip selects for various components. It also contains most of the system registers and implements password protection.

A.2.17 Bi-Endian Support

The 603 processor normally operates with Big-Endian (BE) byte significance. It has a mode of operation designed to more efficiently process code written with Little-Endian (LE) byte significance. The system board has hardware to support Bi-Endian operation. The system defaults to BE mode at power on. The / mode may be switched, but the hardware is not optimized for frequent mode switching. Refer to A.1.2, / "PowerPC 603 Processor," for a description of the Endian switching approach.

When the system is in Big-Endian mode, data is stored in memory with BE ordering. When the system is in Little-Endian mode, data is stored in memory with LE ordering.

A.2.18 PCMCIA Controller -- Intel 82365SL

This configuration supports four PCMCIA option slots by using two Intel 82365SL PCMCIA controllers. Each 82365SL controls two PCMCIA slots. The system board has address/data buffers and power control as required in the PCMCIA specification. PCMCIA cards may be hot-plugged. The system board provides for two front and two rear PCMCIA slots. Each slot may contain either a type 1, 2, or 3 card.

This configuration supports propagation of the RING INDICATE signal from the controller to the system board for the "wake up" function. This system also propagates the AUDIO signal to support PCMCIA cards having audio output.

Note: The system board does not currently support +3.3-volt card operation.

A.2.19 Power Management Controller

A custom-designed power management controller is used in this configuration and performs the following functions:

- Dynamic control of the processor clock
- System clock control
- Peripheral power and mode control
- I/O activity monitor
- Programmable power management interrupt
- System status save and restore
- System power management status control

Power management for the energy-managed workstation can put the system into one of four states. These states are diagrammed in Figure 45 and are described as follows:

- OPERATIONAL
  - Default State
  - Maximum system performance
  - All system resources are turned on and available for immediate use
  - 100% power consumption rate
- STAND-BY
  - Active power management on local device controls
  - Potentially reduced system performance at a given instance
  - Devices are powered on demand
- SUSPEND
  - Maximum system power savings after a short period of inactivity while maintaining the capability to instantaneously return to operations
  - Most devices are either in a low-power mode or actually powered off
  - The processor may have the clock stopped

  - System state information is maintained along with System Memory contents
  - Most interrupts return the system to operational
- HIBERNATION
  - The system enters HIBERNATION after a longer period of inactivity or a HIBERNATION event has occurred
  - Most devices are powered off
  - System state information is stored on the system hardfile for later retrieval
  - Resuming operation will take longer than SUSPEND due to the restoration of the system state from the system hardfile
  - Maximum power savings while still being able to wake up by some external event
- OFF
  - The AC power is physically removed from the system

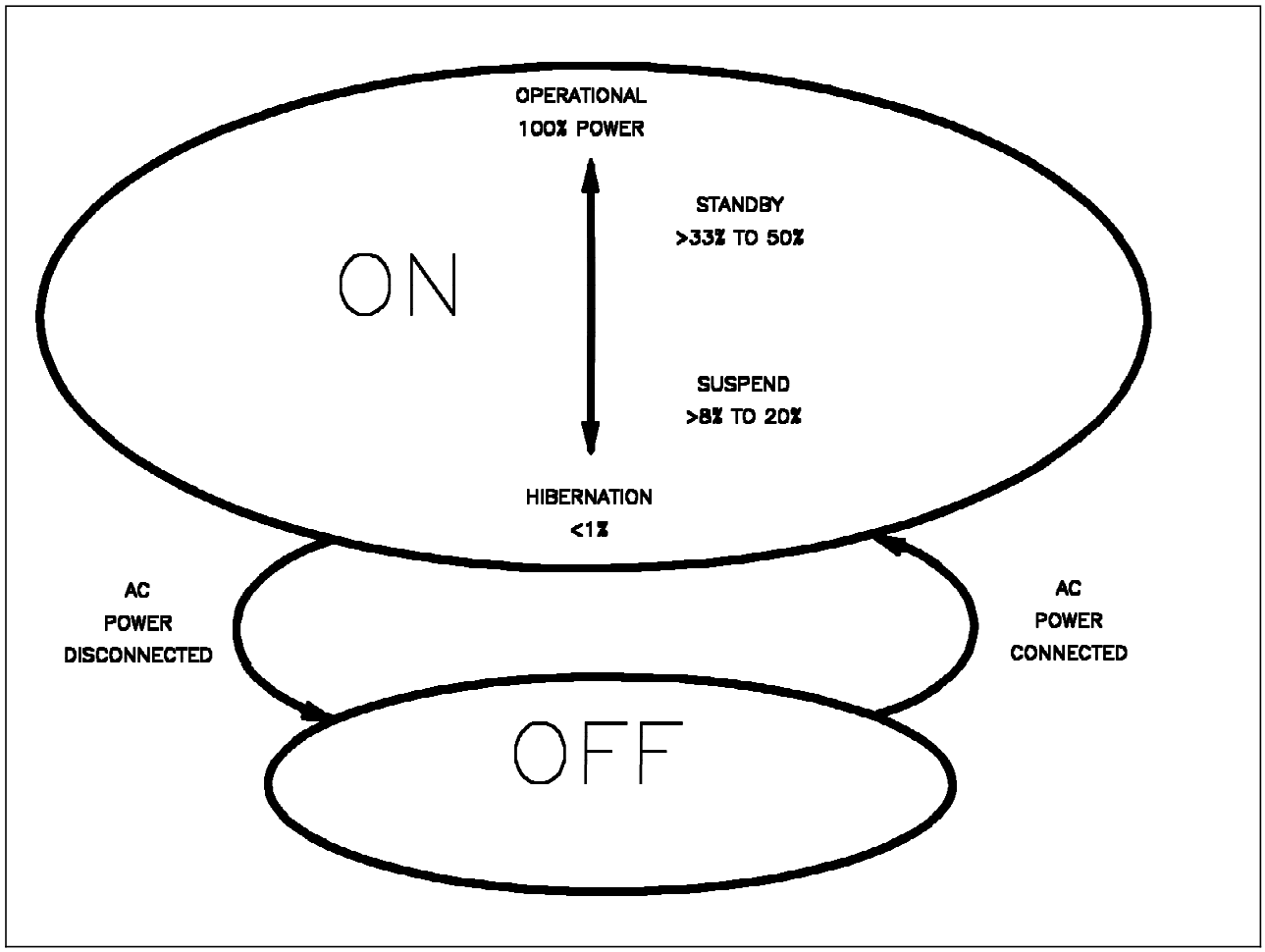


Figure 45. Energy-Managed Workstation's Power States

A.3 Medialess

The medialess configuration shown in Figure 46 is a simple variation on the Reference Implementation described in Section 6.0, "Reference Implementation." This configuration is different in two ways from the Reference Implementation:

a) The medialess system has no data storage capability. The SCSI subsystem, hardfile, CD-ROM, and floppy drive shown in Figure 18 would be removed from the Reference Implementation to form a medialess configuration.

b) The medialess system requires a network connection. The medialess system must be used attached to a network. The network would supply storage for the operating system including its boot-up code, for applications, and for data. An Ethernet or Token Ring adaptor would be placed on one of the expansion buses. The network would supply the operating system software, application software, permanent storage for the user's data and any temporary storage for paging.

The software which runs on the medialess system must accommodate this environment. The System ROM and the boot-up portion of the hosted operating system software must support boot from a network. The operating system must support the medialess operation and contain drivers for the network interface. Diagnostic software must be available through the network, or supplied by maintenance personnel (e.g. a floppy drive installed when needed for diagnostics).

A variation of the medialess system is a "dataless" system. In this system, a hardfile is placed in the configuration and used for system support (e.g. paging), but most of the software and data reside on the network. This configuration has performance advantages over a pure medialess configuration.

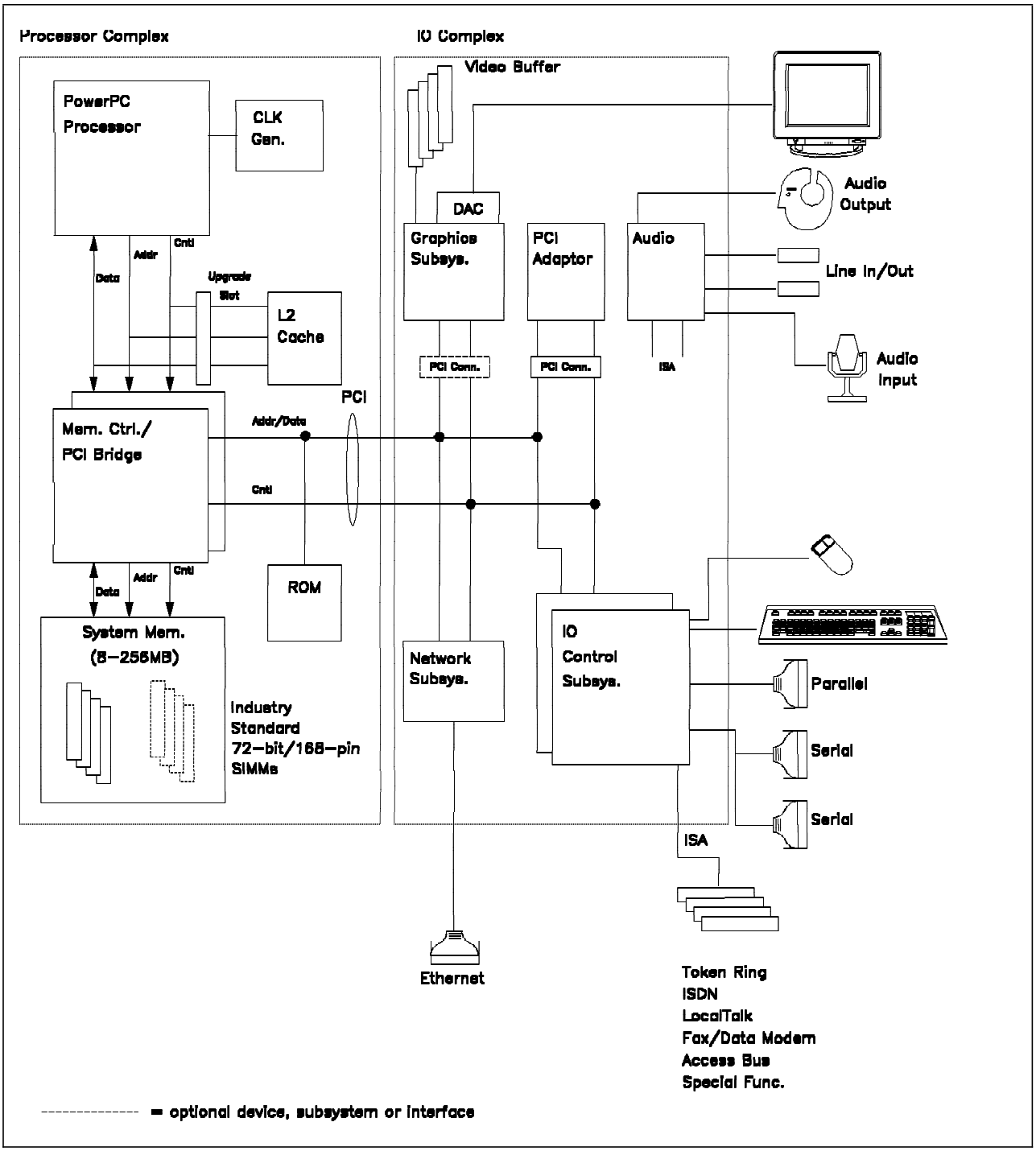


Figure 46. Recommended PowerPC Reference Platform Medialess System

A.4 Technical Workstation

The Technical Workstation configuration as shown in Figure 47 is a higher-performance variation of the Reference Implementation. It is intended to support graphics-intensive work, including 3D graphics. This configuration can differ from the Reference Implementation in several ways: increased system memory, an incorporated L2 cache, a second PCI bus, additional graphics resolution, or additional PCI adaptors.

The system memory for the technical workstation should be augmented. The memory should be upgradable to at least 128 MB. The memory system should implement ECC memory. The minimum system memory should be:

- 16 MB for an entry-level technical workstation
- 64 MB for an entry-level 3D graphics workstation

This memory system could be augmented by installing a copy-back L2 cache in the upgrade slot. For performance reasons, the graphics subsystem should be attached on a second PCI bus or attached directly on the local PowerPC processor bus. As shown in the figure, the graphics subsystem is attached to one PCI bus labeled “PCI 1.” A second PCI bus supporting the remainder of the I/O subsystems is labeled “PCI 2.” In this approach the memory controller and bus bridge would have to be changed from that used in the Reference Implementation (Section 6.2.3, “PCI Bridge and Memory Controller (PCIB/MC)”). If instead the graphics adaptor was placed on the processor bus, then the bus bridge and memory controller could be the same as the Reference Implementation, but the processor bus would have an additional load. The graphics system should support at least 1024x768 pixels with 8 bits of color information. Upgrades should be available for 1280x1024 pixels with 16 or 24 bits of color information. Two or three PCI adaptors are recommended. The number of adaptors depends upon the number of other loads connected to the PCI and the planned use of the workstation.

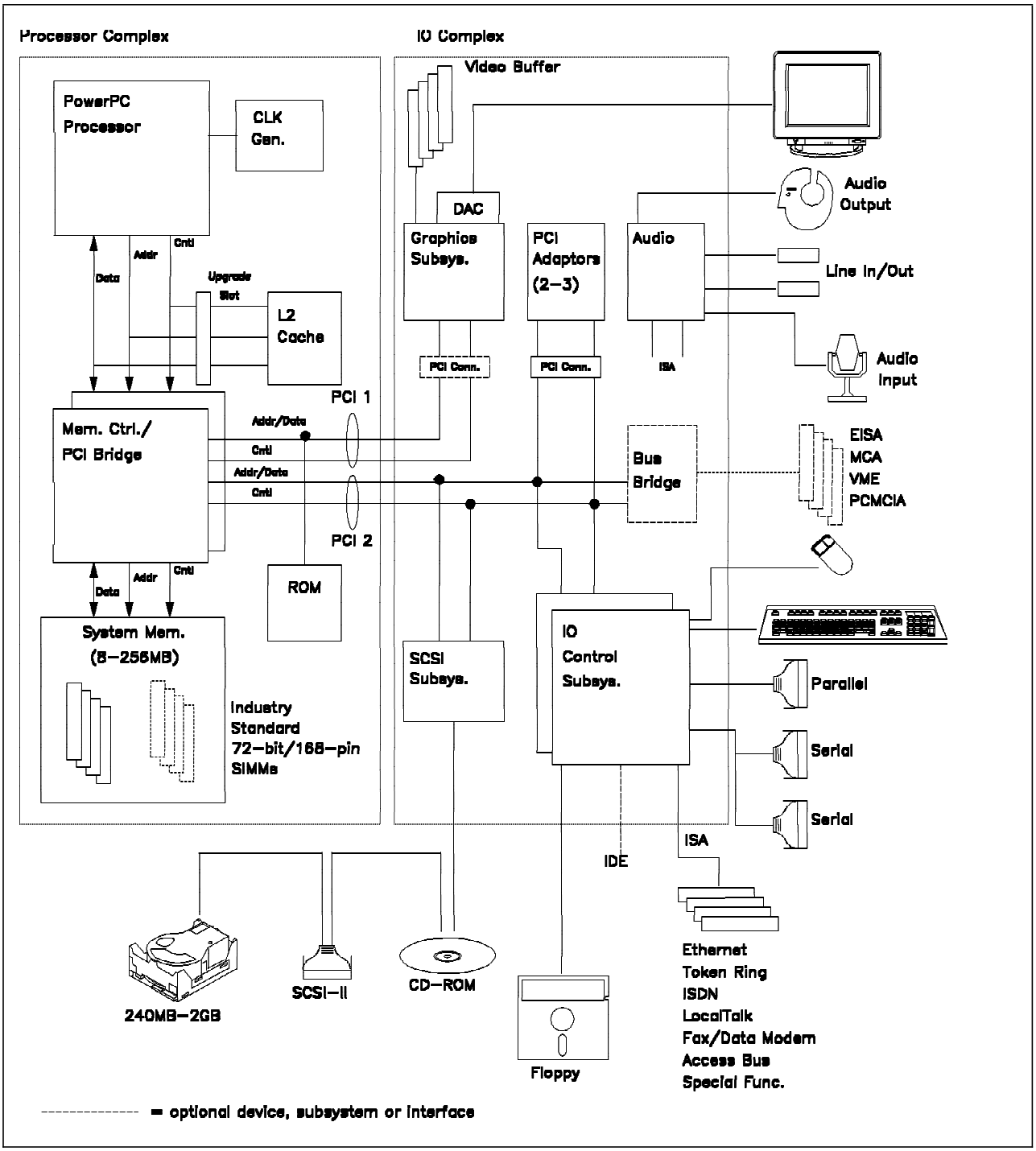


Figure 47. Recommended PowerPC Reference Platform Technical Workstation

A.5 Server

The server configuration section previously published in this specification has been deleted. A more comprehensive description of a server platform using a PowerPC processor will be included in a future version of this specification.

A.6 Symmetric Multiprocessor

The two-way multiprocessor configuration shown in Figure 48 is a variation of the Reference Implementation. The configuration differs from the Reference Implementation by having more processors, more L2 caches, more System Memory, multiple PCI buses, and additional facilities for interrupt handling and inter-processor communication.

/ Section 3.13, "Multiprocessor Considerations," lists the requirements and some of the recommendations for
/ interrupts, inter-processor communications, processor synchronization and memory coherence for multi-
/ processor systems. This section includes some additional recommendations and examples.

For multiprocessor environments, the *PowerPC Reference Platform Specification* intends to focus primarily on symmetric multiprocessor (SMP) systems. Figure 48 shows two processors connected through L2 caches to a system bus shared with the System Memory controller and PCI bus bridge. If the system bus is similar in design to the PowerPC processor bus, peak data bandwidths of greater than 400 MB/sec are possible. With an efficient L2 cache and memory system design, the system bus utilization for the two processors is manageable. SMP designs utilizing more than two processors may require greater system bus bandwidth and a more efficient memory system design (i.e. 16-byte data bus, GTL signal levels, 100-MHz bus clock, split address and data transactions, synchronous DRAM, interleaved memory, etc.).

In many system environments requiring multiple processors, additional performance is required in the I/O
/ subsystems. Often, the size and complexity of the I/O subsystem scales nearly linearly with the number of
/ processors. By using multiple bus bridges attached to System Memory and to the system bus, I/O operations can better utilize the available memory bandwidth. To optimize I/O-to-memory performance, bursting data transfers should be used, special data buffering within the bus bridge should be used, and key
/ I/O subsystems should have bus master capability so that they may manage data transfers independent of the system's processors.

Error detection, error isolation, error reporting and error recovery are important considerations in a multiprocessor system. Errors occurring in one processor or L2 cache should not be allowed to propagate to System Memory or to other processors or L2 caches. I/O adaptors and bus bridges should detect and report errors. ECC memory is recommended. Software should attempt to recover from some types of errors.

Multiprocessor configurations offer significantly greater availability than uniprocessors and are better suited for many applications than clusters of uniprocessors. The recommended recovery action, in the event of a processor or L2 cache failure, is to reset and reinitialize the failing processor. With appropriate software support, data retained in main memory and/or on disk can be used to roll back transactions, reestablish terminal sessions, re-dispatch tasks and re-try I/O operations associated with the failing CPU. It is important that the system continue to run even if some number of tasks are lost.

The following are some recommended attributes for a two-way SMP for a commercial database or file server environment:

- 1- to 4-MB L2 caches, in-line, copy-back, MESI protocol (per processor). One in-line L2 cache per processor is recommended because connecting two processors to one lookaside L2 cache would exceed the capacity of the processor bus in some, if not all, workloads.
- 64-MB minimum System Memory, expandable to 1 GB, ECC, synchronous DRAM support.
- An intelligent SCSI-2 subsystem which can efficiently handle a RAID-5 hard disk array.
- An intelligent communications adaptor with buffering (Ethernet, Token Ring, ATM, etc.).
- Multiple PCI buses attached through a single Memory Controller/ PCI Bridge subsystem.
- Enhanced Interrupt Controller with programmable redirection and priority capabilities. Must also handle inter-processor interrupts.
- All other system features recommended in the "Hardware Configuration" and "Reference Implementation" sections of this document.

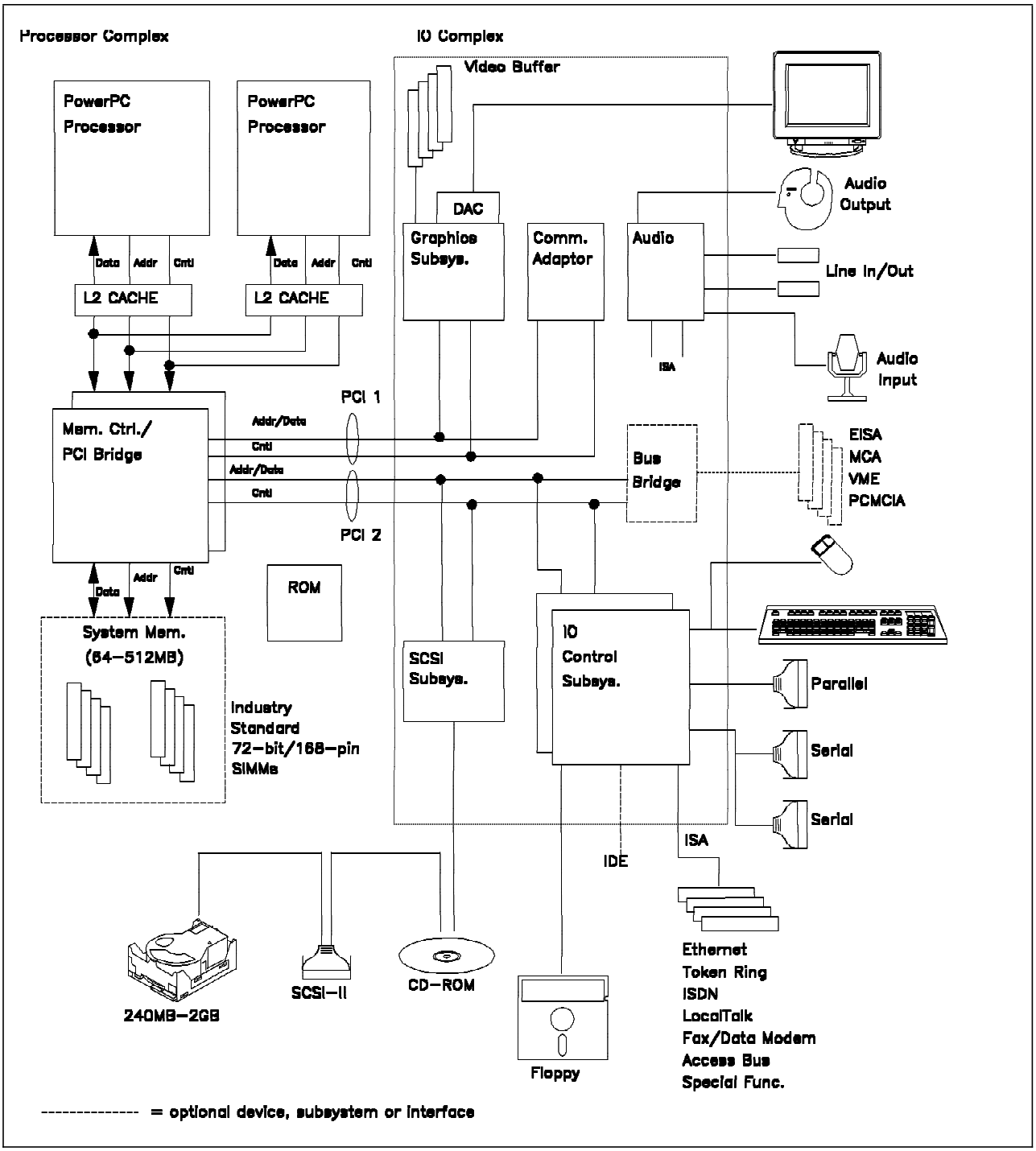
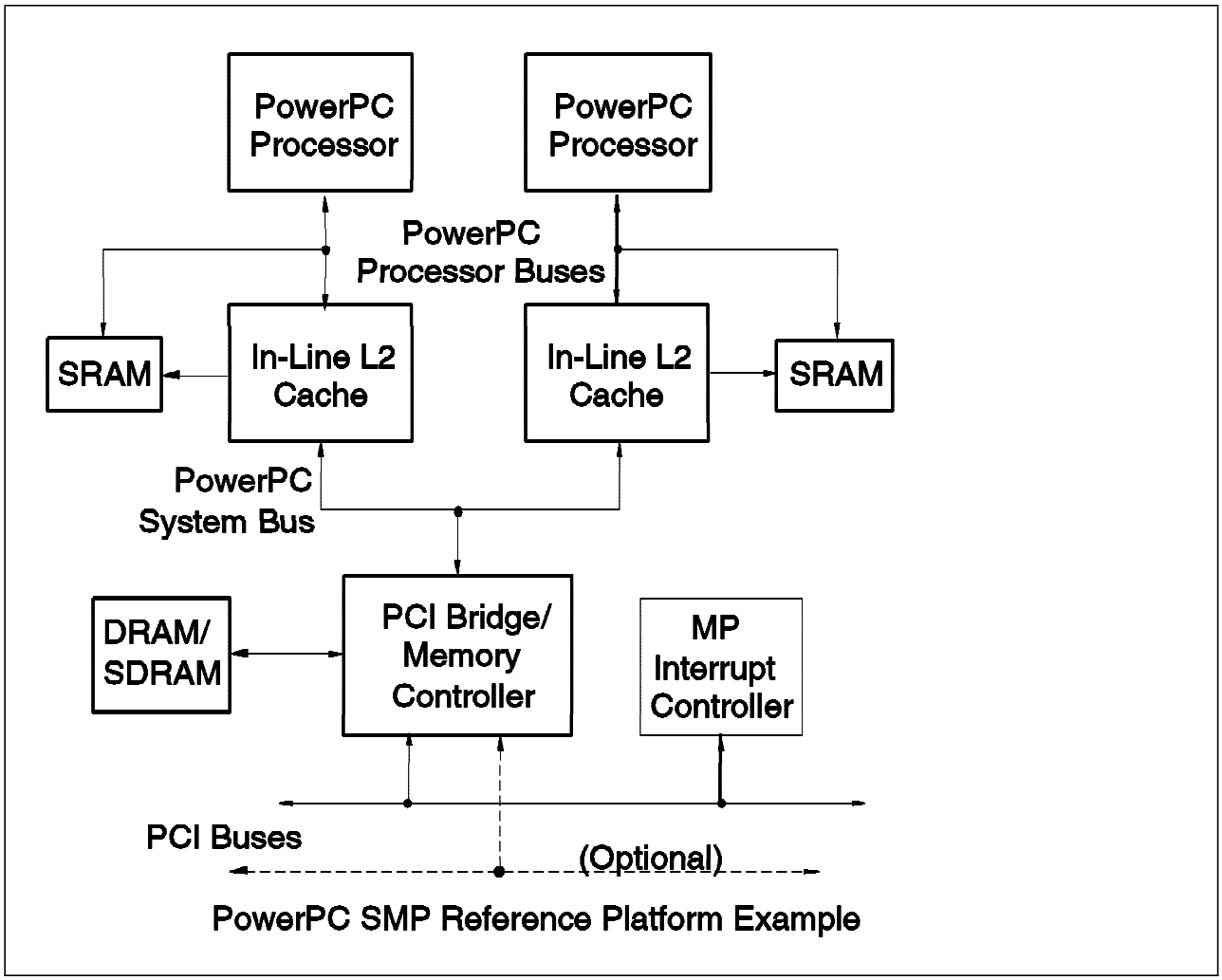


Figure 48. PowerPC Reference Platform Symmetric Multiprocessor System

/ A.6.1 SMP Overview

/ Figure 49 provides some additional detail on how an SMP system might be constructed. The in-line L2
/ cache, the memory controller/PCI host bridge, and the MP interrupt controller are each discussed in more
/ detail in the following sections. These sections briefly describe the basic components of each of these subsys-
/ tems along with general design guidelines.



/ Figure 49. SMP Overview -- Processor and Memory Subsystems

/ A.6.2 In-Line L2 Cache

/ Figure 50 is an example of an in-line, copy-back L2 cache for PowerPC based systems. (Please see Section / 6.7, “Upgrade Slot Definition” for a description of look-aside L2 caches.) Some of the features of an in-line / L2 cache include:

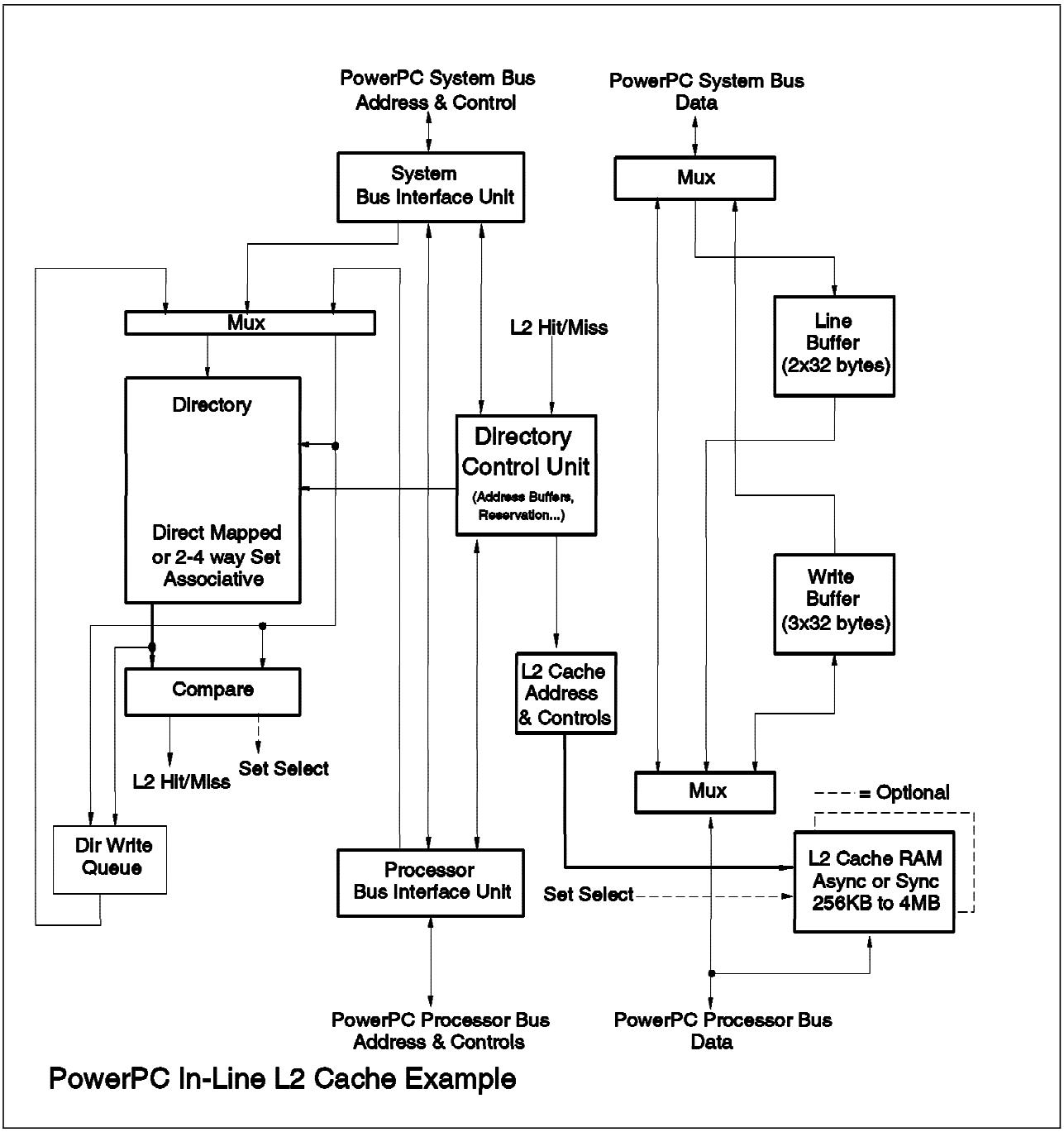
/ • Write Buffers: Three 32-byte write buffers are typical. (Each line in the 601’s cache contains two / 32-byte sectors. Each line in the 604’s caches contains one 32-byte sector.) If an L2 miss causes a line / to be cast out of the L2 cache, the old sector(s), if modified, are stored in the write buffers until they are / forwarded to the memory controller. Normally, the first sector of the old line being cast out from L2 / can be transferred to the memory controller before the first data from the new line arrives.

/ The write buffers are also used to hold write data from processor store requests which miss L2. These / processor store requests are typically tagged “write-through” or “cache-inhibited.”

/ The third write buffer is typically dedicated to hold the sector pushed out of L2 due to a snoop opera- / tion.

/ • Line Buffers: Two 32-byte line buffers are typical. The line buffers are used for memory read oper- / ations. If data arrives from memory and the processor bus is unavailable, the data is written into the line / buffers. If data arrives from memory and the processor bus is available, the line buffers are bypassed.

/ The write buffers and the line buffers include address and controls and participate in bus snooping. / Snoop requests which hit on these buffers (pipeline collisions) are normally retried.



/ Figure 50. In-Line L2

/ • Directory: The directory, or Tag RAM, keeps track of which lines/sectors are held in the L2 cache.
/ System bus requests which hit on the L2 cache are forwarded to the L1 cache, if necessary.
/ • Reservation Address Register: The reservation address register within the L2 cache tracks the reserva-
/ tion, if any, held by the processor. The reservation address register is set by the Read Atomic and *lwarx*
/ Reservation Set requests on the processor bus. The reservation is cleared when the reservation (RSRV_)
/ output signal from the processor is not asserted. System bus requests which hit on the reservation
/ address register are forwarded to the processor whether or not the line is valid in the L2 cache. Reserva-
/ tions are on a 32-byte boundary.
/ • Address pipelining: Address pipelining allows a new bus transaction to begin before the current trans-
/ action has finished by overlapping the data transfers associated with a previous request with the address

transfer for subsequent requests. In-line L2 caches typically support one level of address pipelining for each of the processor and system buses.

- Split transactions: Split transactions allow one bus master to have mastership of the address bus while another bus master has mastership of the data bus. In-line L2 caches typically support split transactions for both the processor and system buses.
- TLB Invalidate and *tlbsync* requests from the processor bus are forwarded to the system bus and vice-versa.
- *Sync* and *eieio* requests from the processor bus are forwarded to the system bus after the write buffers are flushed.
- The L2 cache is normally a superset of the L1 cache (except for certain special cases such as lines marked "write-through.") When a line is LRU'd out of L2, the corresponding line in L1, if any, is pushed out of L1 or invalidated, as needed.
- Snoop filtering: The L2 cache directory typically includes an inclusion bit to indicate whether a line currently resident in the L2 cache is also resident in the L1 cache. If a snoop request hits on the L2 directory and the inclusion bit in that directory entry is off (indicating the line is not in the L1 cache) there is no need to snoop the L1 cache. This speeds up snoop responses and cache-to-cache transfers.
- L2 cache line size: 32- or 64-byte line sizes are typical.

Typical configurations for L2 caches include:

- A direct-mapped 512-KB or larger L2 with an integrated cache control/directory and discrete synchronous SRAM chips.
- A 2-way set-associative 256-KB or larger integrated cache-control/directory/RAM single-chip cache.

L2 cache performance observations:

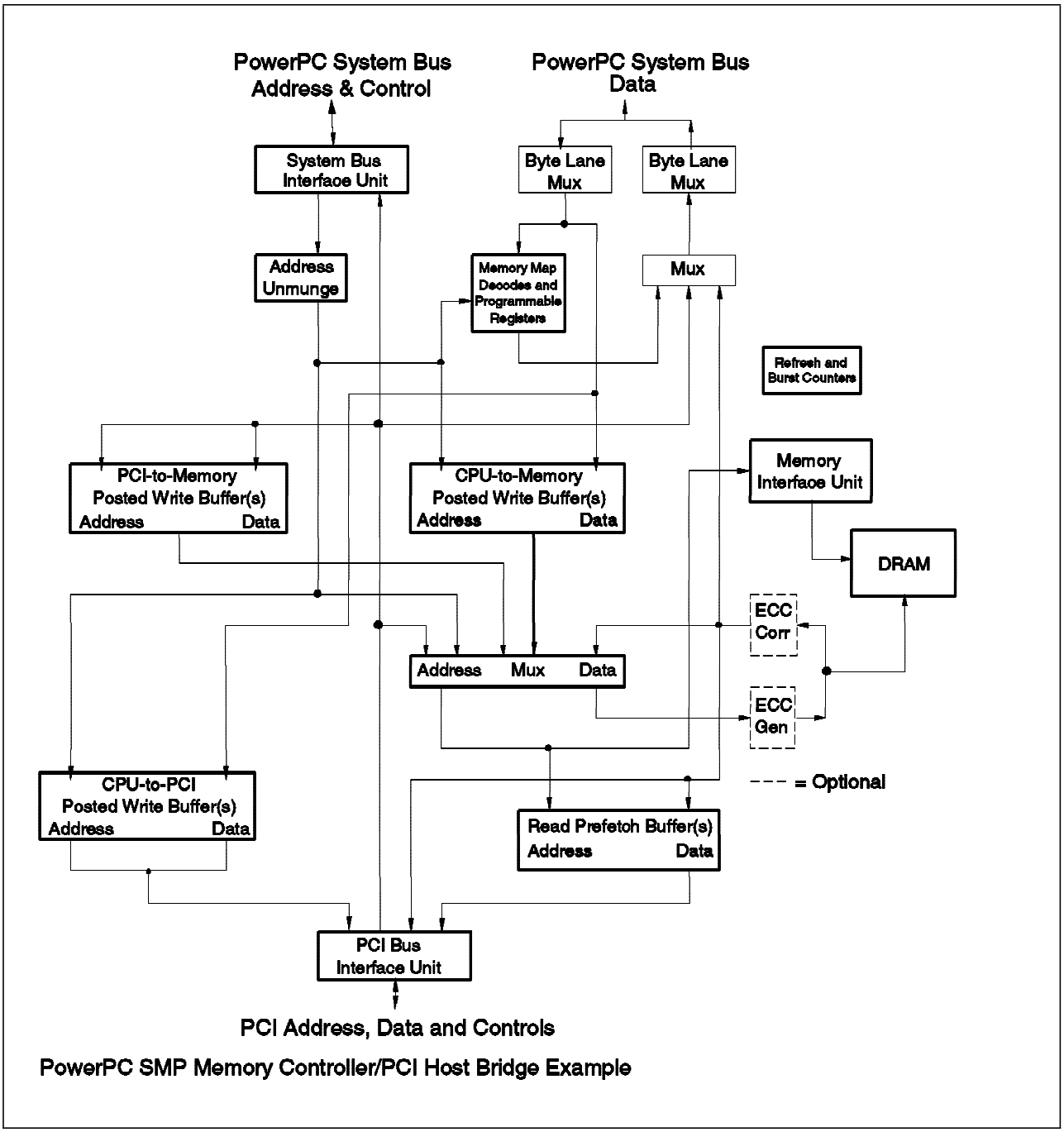
- Benchmark YOUR cache and YOUR memory subsystem design! A well-designed L2 cache can improve processor performance by 25-35%.
- Run the processor bus and the L2 cache at the highest possible frequency (66 - 80 MHz is typical).
- L2 cache read access times are normally stated in terms of number of cycles from transfer start (TS#) on the PowerPC processor bus to the first data beat (DH, DL) and number of cycles between data beats. For systems using synchronous SRAMs, 2-1-1-1 and 3-1-1-1 are typical L2 cache read access times (at 66 MHz). For systems using asynchronous SRAMs, 2-2-2-2 and 3-2-2-2 are typical L2 cache read access times. Avoid design trade-offs which add additional cycles (wait states) to the L2 cache read access times.
- For small L2 caches (256 KB), a multi-way set-associative L2 cache of size x performs almost as well as a direct-mapped L2 cache of size 2x.
- Use write buffers and line buffers to pipeline requests and free up the processor bus as quickly as possible.
- Use split transactions to overlap address-only requests (e.g. snoop requests) with data transfer.

A.6.3 Memory Controller/PCI Host Bridge

Figure 51 is an example of an integrated memory controller/PCI host bridge for PowerPC based systems. Features include:

- CPU-to-memory posted write buffer. One 32-byte or one 64-byte CPU-to-memory posted write buffer is typical (should match L2 line size).
- PCI-to-memory posted write buffer: one or two 32-byte PCI-to-memory posted write buffers per PCI bus are typical.

The posted write buffers allow write requests on the system or PCI bus to be queued while main memory is busy servicing a previous write or read request. The posted write buffers also help permit simultaneous and independent operation of the system and PCI buses.



/ Figure 51. Memory Controller/PCI Host Bridge

/ Strict ordering must be preserved between the posted write buffers. All snoop activity must be resolved
/ before data from either the processor or PCI is written into the posted write buffers. For example, if a
/ PCI write request to System Memory hits on a line which is modified in a processor's cache, the line
/ from the processor must be pushed out of the processor's cache and placed into the CPU-to-memory
/ posted write buffer before the PCI request is placed into the PCI-to-memory posted write buffer (the
/ PCI request is typically re-tried until the data is pushed out of the processor's cache to the
/ CPU-to-memory posted write buffer). These requests should be written to System Memory in the order
/ they were placed in the posted write buffers.

/ There is no indication on the PowerPC processor bus whether a cache block write request is the result
/ of a snoop operation or some internal condition (e.g. LRU) within the processor or L2 cache. If the
/ memory controller/PCI host bridge (or the L2 cache) needs to determine that a cache block write