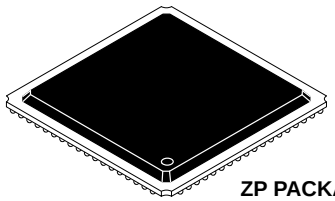
MPC2604GA

Advance Information

Integrated Secondary Cache for PowerPC™ Microprocessors

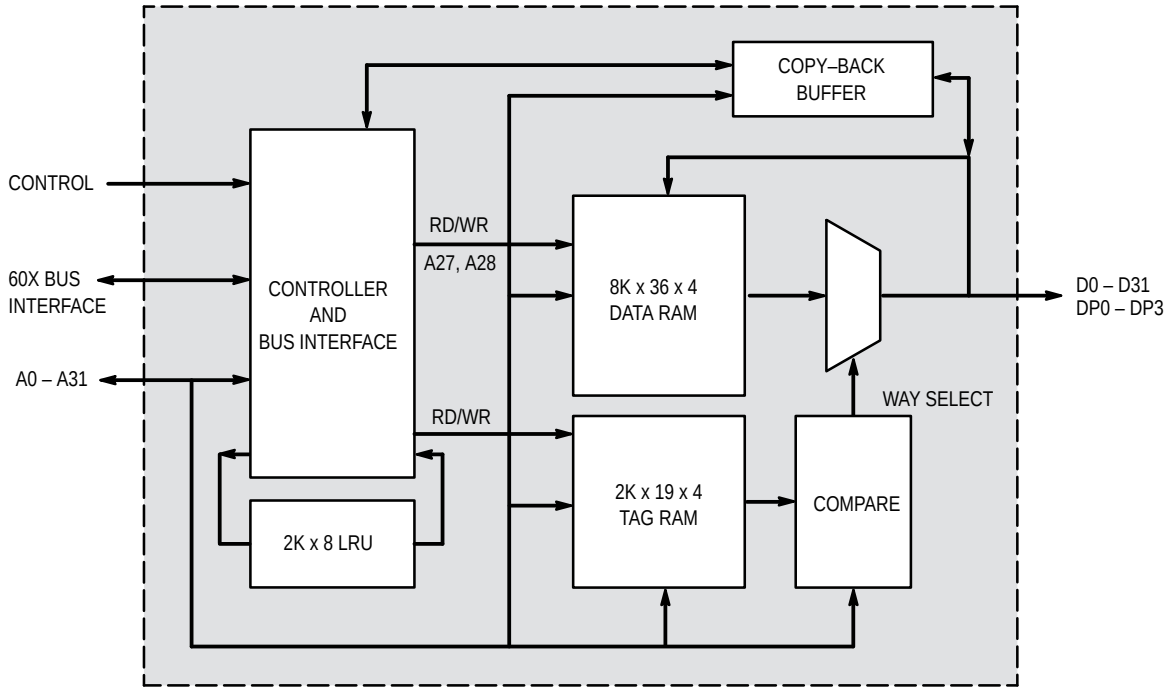


ZP PACKAGE
PBGA
CASE 1103–01

The MPC2604GA is an integrated look–aside cache with copy–back or write–through capability designed for all PowerPC 60X applications. Using 0.5 µm technology along with standard cell logic technology, the MPC2604GA integrates data, tag, host interface, and Least Recently Used (LRU) memory with a cache controller to provide the highest performance Level 2 cache solution available for the 64 bit PowerPC bus.

- 66 MHz Zero Wait State Performance (2–1–1–1 Burst)
- 4–Way Set Associative Cache Design
- 32K x 36 Data Memory Array
- 8K x 19 Tag Array
- Least Recently Used (LRU) Cache Control Logic
- Copy–Back or Write–Through Modes of Operation
- Copy–Back Buffer for Improved Performance
- Single 5 V Power Supply with 3.3 V Compatible I/O
- Two or Four Chip Cache Solution (Two Chips = 256K, Four Chips = 512K Bytes)
- Single Clock Operation
- Compliant with Proposed IEEE Standard 1149.2 Test Access Port (JTAG)
- Supports Four–Processor Multiprocessor Systems in a Shared Cache Configuration
- High Board Density 25mm PBGA Package

BLOCK DIAGRAM



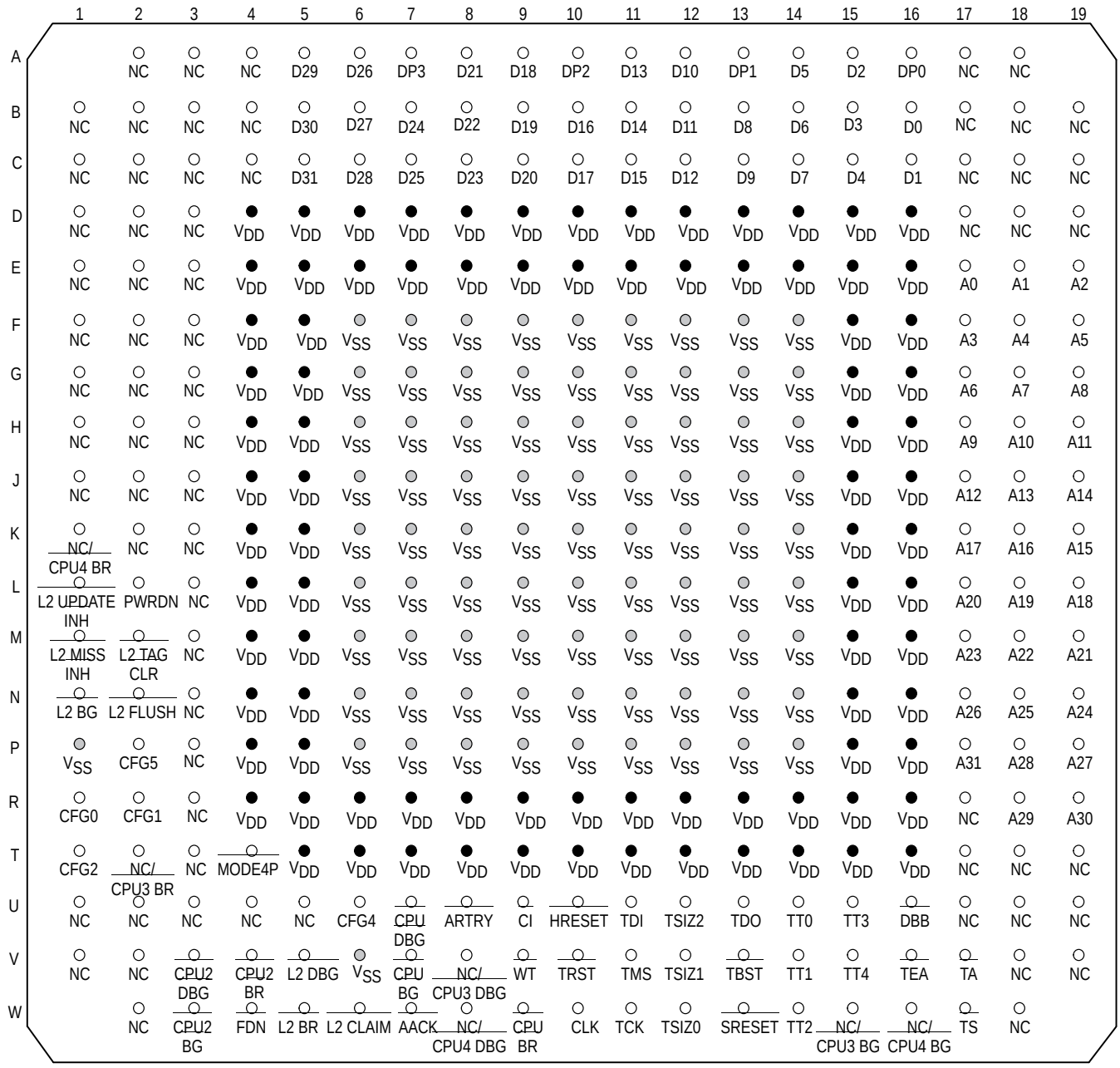
PowerPC is a trademark of IBM Corporation

This document contains information on a new product. Specifications and information herein are subject to change without notice.

REV 1
7/1/96 This data sheet describes the MPC2604GA revision 2.0

MOTOROLA

PIN ASSIGNMENTS



TOP VIEW (X–RAY VIEW)

PIN DESCRIPTIONS

Pin Locations	Pin Name	Type	Description
17E–17P, 18E–18R, 19E–19R	A0 – A31	I/O	Address inputs from processor. Can also be outputs for processor snoop addresses. A0 is MSB. A31 is LSB.
7W	AACK	I/O	Address acknowledge input/output.
8U	ARTRY	I/O	Address retry status I/O. Generated when a read or write snoop to a dirty processor cache line has occurred.
1R, 2R, 1T, 6U, 2P	CFG0 CFG1 CFG2 CFG4 CFG5	I	Configuration inputs. These must be tied to either V_{DD} or V_{SS}. CFG0 Defines Data Bus Side 0 Data Bus Low 1 Data Bus High CFG1 Defines Cache Size 0 256K 1 512K CFG2 Cache Line Selector For 512K Cache Size 0 Even Cache Lines 1 Odd Cache Lines CFG4 AACK Driver Enable 0 Disable AACK Driver 1 Enable AACK Driver CFG5 Snoop Data Tenure Selector 0 Supports Snoop Data Tenure 1 Does not Support Snoop Data Tenure
9U	CI	I/O	Cache inhibit I/O.
10W	CLK	I	Clock input. This must be the same as the processor clock input.
7V	CPU BG	I	CPU bus grant input.
3W	CPU2 BG	I	MPC2604GA logically ORs this signal with CPU BG. Used in multiprocessor configuration as the second CPU BG.
15W	NC/CPU3 BG	I	MPC2604GA logically ORs this signal with CPU BG. Used in multiprocessor configuration as the third CPU BG. This pin is a no connect when MODE4P is high.
16W	NC/CPU4 BG	I	MPC2604GA logically ORs this signal with CPU BG. Used in multiprocessor configuration as the fourth CPU BG. This pin is a no connect when MODE4P is high.
9W	CPU BR	I	CPU bus request input.
4V	CPU2 BR	I	MPC2604GA logically ORs this signal with CPU BR. Used in multiprocessor configuration as the second CPU BR.
2T	NC/CPU3 BR	I	MPC2604GA logically ORs this signal with CPU BR. Used in multiprocessor configuration as the third CPU BR. This pin is a no connect when MODE4P is high.
1K	NC/CPU4 BR	I	MPC2604GA logically ORs this signal with CPU BR. Used in multiprocessor configuration as the fourth CPU BR. This pin is a no connect when MODE4P is high.
7U	CPU DBG	I	CPU data bus grant input from arbiter.
3V	CPU2 DBG	I	MPC2604GA logically ORs this signal with CPU DBG. Used in multiprocessor configuration as the second CPU DBG.
8V	NC/CPU3 DBG	I	MPC2604GA logically ORs this signal with CPU DBG. Used in multiprocessor configuration as the third CPU DBG. This pin is a no connect when MODE4P is high.
8W	NC/CPU4 DBG	I	MPC2604GA logically ORs this signal with CPU DBG. Used in multiprocessor configuration as the fourth CPU DBG. This pin is a no connect when MODE4P is high.
5A, 6A, 8A, 9A, 11A, 12A, 14A, 15A, 5B–16B, 5C–16C	D0 – D31	I/O	Data bus input and output. D0 is MSB. D31 is LSB.
16U	DBB	I/O	Data bus busy. Used as input when processor is master, driven as an output after a qualified L2 DBG when MPC2604GA is the bus master. Note: To operate in Fast L2 mode, this pin must be tied high.
16A, 13A, 10A, 7A	DP0 – DP3	I/O	Data bus parity input and output.
4W	FDN	I/O	Flush done I/O used for communication between other MPC2604GA devices. Must be tied together between all MPC2604GA parts along with a pullup resistor.

10U	HRESET	I	Hard reset input from processor bus. This is an asynchronous input that must be low for at least 16 clock cycles to ensure the MPC2604GA is properly reset.
1N	L2 BG	I	Bus grant input from arbiter.
5W	L2 BR	I/O	Bus request I/O. Normally used as an output.
6W	L2 CLAIM	O	L2 Memory Claim output. Used to claim the bus for processor initiated memory operations that hit the L2 cache. L2 CLAIM goes true (low) before the rising edge of CLK following TS true.
5V	L2 DBG	I	Data bus grant input. Comes from system arbiter, used to start data tenure for bus operations where MPC2604GA is the bus master.
2N	L2 FLUSH	I	Causes cache to write back dirty lines and clears all tag valid bits.
1M	L2 MISS INH	I	Prevents line fills on misses when asserted.
2M	L2 TAG CLR	I	Invalidates all tags and holds cache in a reset condition.
1L	L2 UPDATE INH	I	Cache disable. When asserted, the MPC2604GA will not respond to signals on the local bus and internal states do not change.
4T	MODE4P	I	When tied low, enables arbitration pins for four–processor configuration.
2L	PWRDN	I	Provides low power mode. Disables internal clock tree and prevents address and data transitions into the RAM array.
13W	SRESET	I	Soft reset input from processor bus.
17V	TA	I/O	Transfer acknowledge status I/O from processor bus.
13V	TBST	I/O	Transfer burst status I/O from processor bus. Used to distinguish between burstable and non–burstable memory operations.
11W	TCK	I	Test clock input for IEEE 1149.2 boundary scan (JTAG).
11U	TDI	I	Test data input for IEEE 1149.2 boundary scan (JTAG).
13U	TDO	O	Test data output for IEEE 1149.2 boundary scan (JTAG).
16V	TEA	I	Transfer error acknowledge status input from processor bus.
11V	TMS	I	Test mode select for IEEE 1149.2 boundary scan (JTAG).
10V	TRST	I	Test reset input for IEEE 1149.2 boundary scan (JTAG).
17W	TS	I/O	Transfer start I/O from processor bus (can also come from any bus master on the processor bus). Signals the start of either a processor or bus master cycle.
12U, 12V, 12W	TSIZ0–2	I/O	Transfer size I/O from processor bus.
15V, 15U, 14W, 14V, 14U	TT0–4	I/O	Transfer type I/O from processor bus.
9V	WT	I/O	Write through status input from processor bus. When tied to ground, the MPC2604GA will operate in write–through mode only (no copy–back).
2A–4A, 17A, 18A, 1B–4B, 17B–19B, 1C–4C, 17C–19C, 1D–3D, 17D–19D, 1E–3E, 1F–3F, 1G–3G, 1H–3H, 1J–3J, 2K, 3K, 3L, 3M, 3N, 3P, 3R, 17R, 3T, 17T–19T, 1U–5U, 17U–19U, 1V, 2V, 18V, 19V, 2W, 18W	NC	—	No connection: There is no connection to the chip.
4D–16D, 4E–16E, 4F, 5F, 15F, 16F, 4G, 5G, 15G, 16G, 4H, 5H, 15H, 16H, 4J, 5J, 15J, 16J, 4K, 5K, 15K 16K, 4L, 5L, 15L, 16L, 4M, 5M, 15M, 16M, 4N, 5N, 15N, 16N, 4P, 5P, 15P, 16P, 4R–16R, 5T–16T	V_{DD}	Supply	Power supply: 5.0 V ± 5%.
6F–14F, 6G–14G, 6H–14H, 6J–14J, 6K–14K, 6L–14L, 6M–14M, 6N–14N, 1P, 6P–14P, 6V	V_{SS}	Supply	Ground.

ABSOLUTE MAXIMUM RATINGS (Voltages Referenced to V_{SS} = 0 V)

Rating	Symbol	Value	Unit
Power Supply Voltage	V_{DD}	– 0.5 to + 7.0	V
Voltage Relative to V_{SS}	V_{in}, V_{out}	– 0.5 to V_{DD} + 0.5	V
Output Current (per I/O)	I_{out}	± 20	mA
Power Dissipation (T_A = 70°C)	P_D	3.5	W
Temperature Under Bias	T_{bias}	– 10 to + 85	°C
Operating Temperature	T_J	20 to + 110	°C
Storage Temperature	T_{stg}	– 55 to + 125	°C

NOTE: Permanent device damage may occur if ABSOLUTE MAXIMUM RATINGS are exceeded. Functional operation should be restricted to RECOMMENDED OPERATING CONDITIONS. Exposure to higher than recommended voltages for extended periods of time could affect device reliability.

This device contains circuitry to protect the inputs against damage due to high static voltages or electric fields; however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum rated voltages to this high–impedance circuit.

DC OPERATING CONDITIONS AND CHARACTERISTICS

(T_J = 20 to + 110°C, Unless Otherwise Noted)

RECOMMENDED OPERATING CONDITIONS (Voltages Referenced to V_{SS} = 0 V)

Parameter	Symbol	Min	Typ	Max	Unit
Supply Voltage (Operating Voltage Range)	V_{DD}	4.75	5.0	5.25	V
Input High Voltage	V_{IH}	2.0	—	V_{DD} + 0.3	V
Input Low Voltage	V_{IL}	– 0.5*	—	0.8	V

* V_{IL}(min) = – 2.0 V ac (pulse width ≤ 20 ns).

DC CHARACTERISTICS

Parameter	Symbol	Min	Typ	Max	Unit
Input Leakage Current (All Inputs, V_{in} = 0 to V_{DD})	$I_{lkg(I)}$	—	—	± 1.0	μA
Output Leakage Current (High–Z State, V_{out} = 0 to V_{DD})	$I_{lkg(O)}$	—	—	± 1.0	μA
AC Supply Current (I_{out} = 0 mA, All inputs = V_{IL} or V_{IH}, V_{IL} = 0 V, and V_{IH} ≥ 3.0 V, Cycle Time = 15 ns, max value assumes a constant burst read hit, with 100% bus utilization, and 100% hit rate; typ value, 50% bus utilization, 90% hit rate.)	I_{CCA}	—	290	640	mA
AC Quiescent Current (I_{out} = 0 mA, All inputs = V_{IL} or V_{IH}, V_{IL} = 0 V and V_{IH} ≥ 3.0 V, Cycle Time = 15 ns, All Other Inputs DC)	I_Q	—	—	65	mA
Power Down Current	I_{PD}	—	—	40	mA
Output Low Voltage (I_{OL} = + 8.0 mA)	V_{OL}	—	—	0.4	V
Output High Voltage (I_{OH} = – 4.0 mA)	V_{OH}	2.4	—	—	V
Output High Voltage (I_{OH} = – 20 μA)	V_{OH}	—	—	3.3	V

CAPACITANCE (f = 1.0 MHz, dV = 3.0 V, T_A = 25°C, Periodically Sampled Rather Than 100% Tested)

Parameter	Symbol	Typ	Max	Unit
Input Capacitance	C_{in}	4	6	pF
Output Capacitance	C_{out}	6	8	pF
Input/Output Capacitance	$C_{I/O}$	8	10	pF

AC OPERATING CONDITIONS AND CHARACTERISTICS

(T_J = 20 to + 110°C, Unless Otherwise Noted)

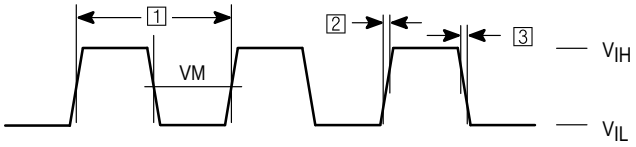
Input Timing Measurement Reference Level . . . . . 1.5 V
Input Pulse Levels . . . . . 0 to 3.0 V
Input Rise/Fall Time . . . . . 2 ns

Output Measurement Timing Level . . . . . 1.5 V
Output Load . . . . . 50Ω Termination to 1.5 V

AC CLOCK SPECIFICATIONS

Parameter	Timing Reference	MPC2604GA–66		MPC2604GA–50		Unit	Notes
		Min	Max	Min	Max		
Frequency of Operation		—	66.67	—	50	MHz	
Clock Cycle Time	[1]	15	—	20	—	ns	2
Clock Rise and Fall Time	[2], [3]	1.0	2.0	1.0	2.0	ns	1
Clock Duty Cycle Measured at 1.5 V		40	60	40	60	%	
Clock Short–Term Jitter (Cycle to Cycle)		—	± 50	—	± 50	ps	2

NOTES:
1. Rise and fall times for the clock input are measured from 0.4 V to 2.4 V.
2. This parameter is sampled and not 100% tested.



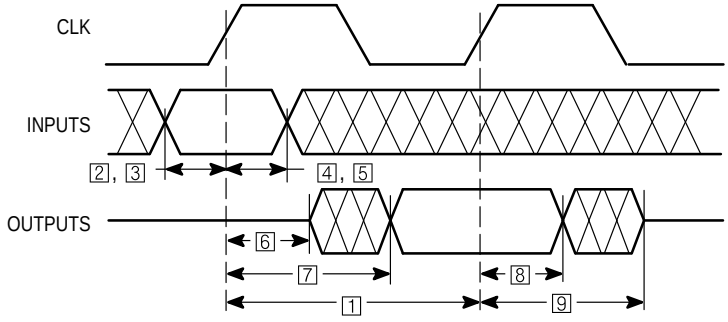
VM = Midpoint Voltage (1.5 V)

Clock Input Timing Diagram

AC SPECIFICATIONS

Parameter	Timing Reference	MPC2604GA–66		MPC2604GA–50		Unit	Notes
		Min	Max	Min	Max		
Clock Cycle Time	[1]	15	—	20	—	ns	
Address/Transfer Start Setup TS	[2]	4.5	—	5	—	ns	1
Setup Time for All Other Inputs	[3]	4.5	—	6	—	ns	1, 3
Clock to Address/Transfer Start Inputs Invalid (Input Hold)	[4]	1	—	1	—	ns	1, 2
Clock to All Other Inputs Invalid (Input Hold)	[5]	1	—	1	—	ns	1, 3
Clock to Output Driven	[6]	2	9	2	15	ns	4
Clock to TA, L2 CLAIM, D0 – D31, DP0 – D3, A0 – D31 Valid	[7]	2	10	2	15	ns	
Clock to All Other Outputs Valid	[7]	2	9	2	15	ns	
Clock to Output Invalid	[8]	2	—	2	—	ns	4
Clock to all Outputs High–Z	[9]	2	12	2	12	ns	4

NOTES:
1. All input specifications are measured from the TTL level (0.8 or 2.0 V) of the signal in question to the 1.4 V level of the rising edge of the input clock. Both input and output timings are measured at the pin.
2. Address/transfer start input signals are composed of the following: A0 – A31, $\overline{TS}$.
3. All other input signals are composed of the following: AACK, ARTRY, CPU BG, CPU2 BG, CPU BR, CPU2 BR, CI, DBB, CPU DBG, CPU2 DBG, D0 – D31, DP0 – D3, L2 BG, L2 DBG, TA, TBST, TEA, TS, TSIZ0 – TSIZ2, TT0 – TT4, WT.
4. This parameter is sampled and not 100% tested.



PC2604GA RESPONSE TO 60X TRANSFER ATTRIBUTES

TT0 – TT4	TBST	CI	WT	Tag Status	MPC2604GA Response	Notes
X1X10	0	1	X	Miss	Line–fill (Processor read miss)	1, 2, 3
X1X10	0	1	X	Hit	L2 CLAIM, AACK, TA (Processor read hit)	6
X1010	1	0	X	Hit Clean	Paradox – Invalidate the line (Processor n–cacheable read hit clean line)	
X1010	1	0	X	Hit Dirty	Paradox – ARTRY, L2 BR, then write back data, invalidate the line (Processor n–cacheable read hit dirty line)	
00110	0	1	X	Miss	Line–fill except right after a snoop hit to processor (Processor write miss)	1, 3, 4, 5
00110	0	1	1	Hit	L2 CLAIM, AACK, TA except after a snoop hit to processor (Processor write hit)	4, 5
00X10	X	1	0	Hit Clean	Cache update (Processor write through WT hit clean)	
00110	0	1	0	Hit Dirty	Cache update, clear dirty bit	
00010	1	1	0	Hit Dirty	Paradox – ARTRY, L2 BR, write back data, keep valid, clear dirty bit	
X0010	1	0	X	Hit Clean	Paradox – Invalidate the line (Processor n–cacheable write hit clean line)	
X0010	1	0	X	Hit Dirty	Paradox – ARTRY, L2 BR, then write back data, invalidate the line (Processor n–cacheable SB write hit dirty line)	
00100	X	X	X	Hit Clean	Invalidate tag (Flush block address–only)	
00100	X	X	X	Hit Dirty	ARTRY, L2 BR, write back data, invalidate tag (flush block address–only)	
00000	X	X	X	Hit Clean	No action (clean block address–only)	
00000	X	X	X	Hit Dirty	ARTRY, L2 BR, write back data, reset dirty bit (Clean block address–only)	
01100	X	X	X	Hit	Invalidate tag (kill block address–only)	

NOTES:

1. If a line fill is going to replace a dirty line and the Cast Out Buffer (COB) is full, the line fill will be canceled (unless the line fill is a write which hits in the COB. In this case the line fill will occur.)
2. If a burst read misses the cache but hits the COB, the MPC2604GA will supply the data from the COB, but not perform a line fill.
3. If ARTRY is asserted during a line fill to replace a dirty line, the line fill will be canceled, the to–be–replaced line will recover its old tag (valid, dirty, tag field), and the COB goes back to an invalid condition, even if the line fill is a burst write to the line in the COB.
4. If a processor burst write occurs right after a snoop write that was a cache hit, the MPC2604GA will invalidate the line. If the snoop was a cache miss, the MPC2604GA will not perform a write allocate.
5. If a processor burst write occurs right after a snoop read that was a cache hit, the MPC2604GA will update the cache and clear the dirty bit. If the snoop was a cache miss, the MPC2604GA will perform a write allocate.
6. If ARTRY is asserted during a read hit, the MPC2604GA will abort the process.

PC2604GA RESPONSE TO CHIPSET TRANSFER ATTRIBUTES

TT0 – TT4	Tag Status	MPC2604GA Response	Notes
00100 X0010 X1110	Hit Clean	Invalidate line	
00100 X0010 X1110	Hit Dirty	ARTRY and L2 BR write back data, invalidate line	1
00000 X1010	Hit Clean	No action	
00000 X1010	Hit Dirty	ARTRY and L2 BR, write back data, reset dirty bit	1
0110X 00110	Hit	Invalidate (Kill block)	

NOTE:

1. In all snoop push cases, BR is sampled the cycle after the ARTRY window. If BR is asserted in this cycle, L2 BR will be immediately negated and an assertion of L2 BG will be ignored.

TRANSFER ATTRIBUTES GENERATED FOR L2 COPYBACK

TT0 – TT4	TBST	CI	WT
00010	0	1	1

SYSTEM USAGE AND REQUIREMENTS

The MCP2604GA is a high–performance look–aside cache for PowerPC systems. A look–aside cache is defined as a cache that resides on the same bus as the processor, the memory controller, the DMA bridge, and the arbiter. The advantage of a look–aside cache is that, when the processor makes a memory request, the cache adds no delay to the memory controller's response time in the event that the request cannot be satisfied by the cache. However, there are certain system requirements that must be met before a look–aside cache can be used.

Comprehension of $\overline{\text{L2 CLAIM}}$

Because the memory controller sees every memory request that is issued by the processor, there must be a mechanism for the cache to inform the memory controller that it has detected a cache hit and that it will satisfy the processor's request. The MPC2604GA has a signal called L2 CLAIM that is asserted whenever a cache hit is detected. Any memory controller with which the MPC2604GA is to be used must have the ability to monitor this signal.

Pipeline Depth

The 60X bus allows pipelining of transactions such that a new transaction can be initiated before a previous transaction has fully completed. The level of pipelining that exists on the bus is defined by how many new data transactions have been initiated while the original transaction is still being processed. By this definition the MPC2604GA can only work in a one level deep pipeline. In the presence of transactions for which it has asserted L2 CLAIM, the MPC2604GA can control the level of pipelining by delaying its assertion of AACK. However, for transactions that it cannot control, the MPC2604GA is dependent upon the memory controller to control pipeline depth. Thus, another system requirement for the use of the MPC2604GA is the use of a memory controller that only allows one level deep of pipelining on the 60X bus.

Bus Mastering

Bus mastering is only a requirement for systems which seek to use the MPC2604GA as a copyback, as opposed to a write–through, cache. The requirement is that the system arbiter must have the ability to allow the MPC2604GA to become a bus master. Specifically, the system arbiter must be able to recognize assertions of L2 BR and must have the ability to assert L2 BG and L2 DBG.

These are the only requirements above and beyond what should already exist in a PowerPC system. All other necessary control signals are signals that are required for the processor to communicate with the memory controller, the DMA bridge, and the arbiter.

CONFIGURATION PINS

The MPC2604GA has four configuration pins: CFG0, CFG1, CFG2, CFG4, and CFG5.

CFG0

A given MPC2604GA device only has 32 data pins, plus parity. Because the PowerPC 60X bus is defined to be a 64–bit bus, it is necessary to use two MPC2604GA devices to create a cache. Since the processor can write quantities of less than 8 bytes (64 bits) in a transaction, and because each MPC2604GA device is exactly the same, there must be a mechanism to indicate which device is connected to the high half of the data bus (D0 – D31) and which device is connected to the low half of the data bus (D32 – D63). CFG0 is the means to this end. CFG0 should be tied high on the device that is connected to the high half of the data bus, and CFG0 should be tied low on the device that is connected to the low half of the data bus.

CFG1, CFG2

When used as a two chip solution, the MPC2604GA implements a 256KB cache. It has the ability to be used in a four chip solution to create a 512KB cache. In this configuration, two independent pairs are created that each cache different addresses. Specifically, one pair, the even pair, will only cache addresses with A26 = 0. The odd pair will only cache addresses with A26 = 1. CFG1 defines whether a 256KB cache is being used (CFG1 = 0) or a 512KB cache is being used (CFG1 = 1). In the case where CFG1 = 1, CFG2 defines whether a given device is a member of the even pair or the odd pair. Thus, when implementing a four chip configuration, each of the devices should have CFG1 tied high, and each device should have a different combination of CFG0 and CFG2:

- The even high device should have CFG0 = 1 and CFG2 = 0.
- The even low device should have CFG0 = 0 and CFG2 = 0.
- The odd high device should have CFG0 = 1 and CFG2 = 1.
- The odd low device should have CFG0 = 0 and CFG2 = 1.

When CFG1 = 0, CFG2 is not used by the MPC2604GA.

CFG4

When the MPC2604GA asserts $\overline{\text{L2 CLAIM}}$ to signal to the memory controller that a cache hit has been detected, it is taking control of the address and data tenures of the transaction (see **60X Bus Operation** and **Memory Coherence**). This means that the MPC2604GA will assert AACK to end the address tenure, and it will assert TA as needed for the data tenure. Tying CFG4 low implements a custom feature that prevents the MPC2604GA from asserting AACK to end transactions for which it has asserted L2 CLAIM. In systems that tie CFG4 low it is necessary for the memory controller to assert AACK for all transactions. It is expected that most systems will tie CFG4 high.

CFG5

Many core logic chipsets are designed such that the DMA bridge and the memory controller are resident in the same device. In such systems there is internal communication between these two functional units. Bus transactions generated by the DMA bridge are solely for the purpose of keeping the system coherent. They are not explicit requests from memory that have data tenures associated with them. How-

ever, some chipsets are designed with the memory controller and the DMA bridge partitioned into different devices. In systems such as these, transactions generated by the DMA bridge are true memory requests that have data tenures associated with them. These are called snoop data tenures. Because these two types of systems are fundamentally different, the MPC2604GA must know in which type of system it is resident in order to respond properly to the different types of transactions. For systems that do not have snoop data tenures, CFG5 must be tied high. For systems that do use snoop data tenures, CFG5 must be tied low.

RESET/INITIALIZATION

To ensure proper initialization and system functionality, the HRESET pin of the MPC2604GA should be connected to the same signal that is used to reset the processor. When HRESET is negated, the MPC2604GA commences an internal initialization sequence to clear all of the valid bits in the cache. The sequence takes approximately 4000 clock cycles. During this time the MPC2604GA will not participate in any bus transaction that occurs. All transactions are, however, monitored so that, regardless of when the initialization sequence completes, the MPC2604GA is prepared to take action on the next transaction initiated by the processor.

At some point after this 4000 cycle sequence, the MPC2604GA will detect its first cache hit. At this time the system will experience its first assertion of L2 CLAIM. If the memory controller must be configured via software to comprehend assertions of L2 CLAIM, this configuration operation must have completed by this time. For systems that cannot guarantee that this requirement is met, it is necessary to disable the MPC2604GA until such time as this configuration can be guaranteed. Disabling the MPC2604GA can be accomplished by asserting L2 UPDATE INH sometime during reset and negating it when it is deemed safe for caching to commence.

60X BUS OPERATION

All transactions have what is called an address tenure. An address tenure is a set number of bus cycles during which the address bus and its associated control signals are being used for the transaction at hand. In general, there are two types of transactions. Those that only have address tenures, called address–only transactions. And those that require the use of the data bus and therefore will have a data tenure. These transactions are called data transactions. This section describes how address and data tenures are defined as viewed by the MPC2604GA.

Address Tenures

Address tenures on the 60X bus are fairly well defined. They start with an assertion of TS by a device that has been granted the bus by the system arbiter. This device is called the bus master for this transaction. At the same time that TS is asserted, the bus master also drives the address and all other relevant control signals that define the transaction. TS is only asserted for one cycle but all other signals are held valid by the bus master until some other device asserts AACK. The device that asserts AACK becomes the slave to this transaction. Typically, the slave is the memory controller, although for transactions that are cache hits the MPC2604GA becomes the slave by driving L2 CLAIM.

Transactions can be aborted by any device on the bus by asserting ARTRY. ARTRY may be asserted at any time after TS is asserted, but must be held through the cycle after AACK is asserted. This cycle is referred to as the ARTRY window, since it is the cycle in which all devices sample ARTRY to determine if the address tenure has completed successfully.

If an address tenure is not aborted by an assertion of ARTRY, then the next bus master is free to assert TS, the cycle after the ARTRY window to start a new address tenure. If ARTRY is asserted in the ARTRY window, all devices that are not asserting ARTRY must negate their bus request in the following cycle. This next cycle is called the BR window. The purpose of this protocol is to give immediate bus mastership to the device that asserted ARTRY with the expectation that that device will take this opportunity to clean up whatever circumstances caused it to assert ARTRY. Typically, this involves writing data back to memory to maintain coherence in the system.

Data Tenures

Data tenures are more complicated to define than address tenures. They require two conditions to start: an assertion of TS that initiates a data transaction and a qualified assertion of the bus master's data bus grant. For a data bus grant to be considered qualified, no device on the bus may be asserting DBB in the cycle that the data bus grant is asserted.

Data transactions come in two types: single–beat transactions and burst transactions. The type is determined by the state of TBST during the address tenure of the transaction. If the bus master asserts TBST, the transaction is a burst transaction and will require four assertions of TA in order to complete normally. If TBST is negated during the address tenure, the transaction only requires one assertion of TA, thus the name single–beat.

Which device drives the data bus during a data transaction depends upon whether the transaction is a read or a write. For a read transaction, the slave device drives the data bus. For a write transaction, the master drives the data bus. In all data transactions, the slave device asserts TA to indicate that either valid data is present on the bus, in the case of a read; or that it is reading data off the data bus, in the case of a write. The master device asserts DBB the cycle after it has been granted the data bus and keeps it asserted until the data tenure has completed.

A data tenure can be aborted in two different ways. The address tenure for the transaction can be aborted by an assertion of ARTRY. Or, the slave device may assert TEA to indicate that some error condition has been detected. Either event will prematurely terminate the data tenure.

Data Streaming

For the majority of data transactions there must be a wait state between the completion of one data tenure and the start of the next. This turnaround cycle avoids the contention on the data bus that would occur if one device starts driving data before another device has had a chance to turn off its data bus drivers. When a cache read hit is pipelined on top of another cache read hit, there is no need for this turnaround cycle since the same device will be driving the data bus for both data tenures. The 60X bus has the ability to remove this unnecessary wait state and allow back–to–back cache read hits to stream together. This ability is only enabled if the sys-

tem is put into Fast L2 mode. Note that not all PowerPC processors support Fast L2 mode.

One of the requirements for taking advantage of this data streaming capability is that the system arbiter must be sophisticated enough to identify situations in which streaming may occur. Upon recognizing these situations, it must assert the processor's data bus grant in the cycle coincident with the fourth assertion of TA of the first cache read, so that the data tenure for the second cache read may commence in the next cycle.

Because it only recognizes qualified assertions of CPU DBG, the MPC2604GA must not be aware of the processor's assertions of DBB. This means that the DBB pin of the MPC2604GA must be tied to a pullup resistor rather than connected to the system DBB to which all other devices are connected. This forces the system arbiter to a level of sophistication such that it only supplies qualified data bus grants and thus the DBB signal is unnecessary to the whole system.

Data Bus Parking

The MPC2604GA has the ability to respond to a processor read or write hit starting in the cycle after the processor has asserted TS. This is referred to as a 2–1–1–1 response. However, even though the MPC2604GA has this ability, it is dependent upon the system to allow this quick of a response to occur. As discussed above, a data tenure cannot start until the master has been given a qualified bus grant. In order for the data tenure to start the cycle after TS is asserted, the data bus must be granted in the cycle coincident with the assertion of TS. At bus speeds of 66 MHz it is extremely difficult for an arbiter to detect an assertion of TS and itself assert CPU DBG in the same cycle. In order to realistically allow this situation to occur, CPU DBG must be asserted independent of the processor's assertion of TS.

Data bus parking is a system feature whereby the processor always has a qualified data bus grant when the data bus is idle. It is also a requirement for systems which seek to take advantage of the 2–1–1–1 response time capabilities of the MPC2604GA. This feature is typically present in arbiters that have the level of sophistication necessary to support data streaming. But it is also a feature of systems that do not even have a data bus arbiter. In these systems the data bus grant of every device in the system is tied to ground. The assertion of DBB by the current data bus master effectively removes the qualified data bus grant of all devices in the system, including its own. Note that in systems that have no data bus arbiter that it is impossible to take advantage of data streaming.

There is another caveat associated with data bus parking. Care must be taken when using data bus parking along with Fast L2 mode. In normal bus mode when the processor reads data off the bus, it will wait one cycle before passing the data on to internal functional units. The purpose of this one cycle waiting period is to check for an assertion of DRTRY, which invalidates the data that has been already read. One of the advantages of running the processor in Fast L2 mode is that this internal processor wait state is removed.

A problem will arise, however, if the processor is given data the cycle after TS is asserted, as is possible with the MPC2604GA, and the transaction is aborted by some other device asserting ARTRY. Because the processor will not sample $\overline{\text{ARTRY}}$ until two cycles after the assertion of $\overline{\text{TS}}$, the data read off the bus will have already been forwarded to the internal functional units. Thus, incorrect results may occur in the system.

To avoid this situation in a system that seeks to run Fast L2 mode with the data bus parked, there must be a guarantee that ARTRY will never be asserted for cache read hits. This is a further requirement to be imposed upon the DMA bridge and the memory controller. If this guarantee cannot be made, the data bus cannot be parked when running in Fast L2 mode.

Processor Reads

When the processor issues a read transaction, the MPC2604GA does a tag lookup to determine if this data is in the cache. If there is a cache hit and CI is not asserted, the MPC2604GA will assert L2 CLAIM and supply the data to the processor when the data tenure starts.

If the processor issues a cache–inhibited read ($\overline{\text{CI}}$ asserted) and the MPC2604GA detects a cache hit to a non–dirty, or clean, cache line, the line will be marked invalid. If the cache–inhibited read hits a dirty line, the MPC2604GA will assert ARTRY and write the dirty line back to memory.

If the read misses in the cache, the MPC2604GA will perform a linefill only if it is a burst read and it is not marked cache–inhibited. During a linefill, the MPC2604GA stores the data present on the bus as it is supplied by the memory controller.

Processor Writes

The conditions for asserting $\overline{\text{L2 CLAIM}}$ for processor writes are almost the same as for processor reads. There must be a cache hit and CI must not be asserted. In addition, however, WT must not be asserted. Single beat writes that are marked either write–through or cache–inhibited that hit in the cache cause the MPC2604GA to assert ARTRY and write the dirty line back to memory.

Transaction Pipelining

As explained in **Pipeline Depth**, the MPC2604GA can only handle one level of pipelining on the bus. Since the assertion of L2 CLAIM gives it the ability to assert AACK, the MPC2604GA has the ability to control this pipeline depth for transactions that are cache hits by delaying its assertion of AACK.

Pipelined cache hits are transactions that hit in the cache but occur while there is still an outstanding data transaction on the bus. The timing of the assertion of AACK for a pipelined cache hit is dependent upon the completion of the previous transaction. For explanation purposes, the previous transaction will be referred to as transaction one. The pipelined cache hit will be referred to as transaction two.

If transaction one is a cache hit, the MPC2604GA will be the slave device for the transaction. Since, for burst operations, the MPC2604GA always asserts TA for four consecutive clock cycles, the end of the data tenure for transaction one will be at a deterministic clock cycle. In this case, AACK for transaction two can be asserted coincident with the last assertion of TA for transaction one. If transaction one is not a cache hit, the MPC2604GA will wait until after the data tenure for transaction one has completed before asserting AACK to complete the address tenure of transaction two.

MEMORY COHERENCE

When a processor brings data into its on–chip cache and modifies it, a situation has arisen in which the main memory now contains irrelevant, or stale, data. Given that most systems support some form of DMA there must exist a means by which the processor is forced to write this modified, or dirty, data back to main memory. The DMA bridge is responsible for generating bus transactions to ensure that main memory locations accessed by DMA operations do not contain stale data. These transactions, called snoops, come in three different categories, each of which will be discussed below.

Snoops cause the processor and the MPC2604GA to check to see if they have dirty copies of the memory location specified in the snoop transaction. If either device does have a dirty copy it will assert ARTRY and make use of the opportunity presented in the BR window to write this data back to main memory.

Situations can arise where a cache line is dirty in both the processor's L1 cache and in the MPC2604GA. In cases such as these, snoop transactions should cause the processor to write its data back to memory since it is by definition more recent than the data in the MPC2604GA. Since ARTRY is a shared signal and it cannot be determined which devices are driving it, the MPC2604GA samples CPU BR in the BR window to determine if the snoop hit a dirty line in the L1 cache. If CPU BR is asserted during this window, the MPC2604GA will defer to the processor.

Snoop Reads

A snoop read causes dirty data to be written back to memory but allows both the L1 and L2 to keep a valid copy. In cases where the snoop hits a dirty cache line in the processor, the MPC2604GA will update its contents as the processor writes the data back to main memory.

Snoop reads can be implemented in two ways. One is that the DMA bridge can issue a clean transaction (TT[0:4] = 00000). The other is that the DMA bridge can do a read transaction (TT[0:4] = x1010). If the DMA bridge does a read transaction, the MPC2604GA determines that it is a snoop read rather than a processor read by the state of CPU BG the cycle before TS was asserted. If the processor was not granted the bus then the transaction had to have been issued by the DMA bridge and is therefore a snoop read.

Snoop Writes

Snoop writes also cause dirty data to be written back to main memory. The difference from a snoop read is that the cache line must then be invalidated in both the processor's cache and in the L2 cache. When the processor writes data back to memory in response to a snoop write, the MPC2604GA will not cache the data as it appears on the bus. If a valid copy resides in the cache, the MPC2604GA will invalidate it.

Again there are multiple transactions that can be used by the DMA bridge to implement a snoop write. It can issue a flush transaction (TT[0:4] = 00100), a read with intent to modify (TT[0:4] = x1110), or a write with flush (TT[0:4] = 00010). As with snoop reads, the MPC2604GA distinguishes between processor issued data transactions and snoop transactions by the state of CPU BG in the cycle previous to the assertion of TS.

Snoop Kills

Kills are snoops that cause cache entries to be immediately invalidated, regardless of whether they are dirty. This saves time if the DMA operation is going to modify all the data in the cache line. To implement a snoop kill the DMA bridge can issue a kill transaction (TT[0:4] = 01100) or a write with kill (TT[0:4] = 00110).

FOUR CHIP IMPLEMENTATION

The MPC2604GA has the ability to cascade two pairs of devices together to implement a 512KB cache. The **CFG1, CFG2** section discusses the details of configuring the devices for this type of implementation. The four chip implementation adds a level of complexity to the functionality of a copyback cache. This is due to the fact that in this situation there are essentially two independent caches that are sharing common arbitration signals for bus mastership. This section will discuss situations that can cause contention over these arbitration signals and how the MPC2604GA was designed to avoid these situations.

Multiple Castouts

Because each pair of MPC2604GAs has its own castout buffer (COB), it is possible for situations to arise in which both pairs need to do copyback operations. Under normal circumstances the two pairs will enter castout conditions at different times. In these cases, when a pair determines that it needs to do a castout, the L2 BR signal is first sampled. If L2 BR is already asserted then it is clear that the other pair is also in a castout situation. The late pair will wait until L2 BR is negated before continuing in its attempt to perform its castout.

Because of the BR window protocol associated with assertions of ARTRY, it is possible for a situation to arise where pair two is waiting for pair one to do its castout before asserting L2 BR. If there is an assertion of ARTRY by a device other than pair one, the pair one is required to negate L2 BR in the BR window. In order to prevent pair two from interpreting pair one's negation of L2 BR as an indication that pair one has completed its castout, a simple arbitration mechanism is used. Both pairs have a simple one–bit counter that is synchronized such that both counters always have the same value. For the purposes of performing a castout operation, the even pair can only assert L2 BR if the counter is zero. The odd pair can only assert L2 BR if the counter is one. This simple mechanism prevents both pairs from asserting L2 BR in the same cycle and therefore not being cognizant of the other pair's need to perform a castout.

Snoop Hit Before Castout

The other situation that can cause problems with a shared bus request occurs when a snoop hits a dirty line in one of the MPC2604GA pairs. If pair one has a cache line in its COB, it will assert L2 BR so that it may perform a castout operation. If a snoop hits a dirty line in pair two, it will assert both ARTRY and L2 BR so that it can write the snoop data back to main memory. When pair one detects that ARTRY has been asserted, it needs to be made aware that pair two needs to request the bus. Otherwise, at the same time that pair two is asserting L2 BR, pair one will attempt to conform to the BR window protocol and negate L2 BR. This situation is avoided by pair one sampling FDN when it detects that

ARTRY has been asserted. If FDN is asserted at the same time as ARTRY is asserted, pair one will recognize that pair two is asserting ARTRY. Pair one will then high–Z L2 BR so that there will not be contention when pair two is asserting L2 BR.

MULTIPROCESSING

The MPC2604GA can be used as a common cache for up to four processors. For each processor there is a bus request, bus grant, and data bus grant signal pin on the MPC2604GA. Each of these pins needs to be connected to the respective processor's arbitration signals in the system.

The MPC2604GA treats multiple processors as one processor. Thus, the same restrictions on pipelining depth are true with regard to how many processor transactions can be outstanding at any one time. There can only be one data transaction from ANY processor pipelined on top of a current data transaction that was issued by ANY processor.

The data tenures for all processors must be performed in the same order as the address tenures on a system–wide basis. If processor one makes a request and then processor two makes a request, processor one's data tenure must precede processor two's data tenure. Note that this is not a 60X bus restriction, but rather a restriction necessary for proper operation of the MPC2604GA.

The MPC2604GA keeps coherent with the L1 caches of multiple processors as defined by the MESI (Modified–Exclusive–Shared–Invalid) protocol without actually implementing the protocol. This is possible for two reasons. Since the MPC2604GA is a look–aside cache, all transactions are monitored by all devices on the bus. Also, the MPC2604GA cannot, on its own, modify data. Thus, if one processor requests exclusive access to a cache line, it is not necessary for the MPC2604GA to invalidate its copy of the data, as would be required under the MESI protocol. If a second processor requests the same data, the transaction will cause the first processor to assert ARTRY. This will prevent the MPC2604GA from supplying stale data to the second processor.

As discussed in **Data Bus Parking**, care must be taken when parking the data bus in Fast L2 mode. By the nature of MP systems running under the MESI protocol there will be assertions of ARTRY to abort cache read hits. Thus, in an MP system, the data bus cannot be parked to any processor if the system is to be run in Fast L2 mode.

ASYNCHRONOUS SIGNALS

The MPC2604GA supports four asynchronous control signals. These signals were originally defined in the PowerPC Reference Platform (PReP) specification. Because these signals are defined to be asynchronous, the MPC2604GA must synchronize them internally. This process takes eight clock cycles. Thus, to guarantee recognition by the MPC2604GA, assertions of any one of these signals must last a minimum of eight clock cycles.

L2 FLUSH

When L2 FLUSH is asserted, the MPC2604GA initiates an internal sequence that steps through every cache line present. Valid lines that are clean are immediately marked invalid. Valid lines that are dirty must be written back to main memory.

To keep memory up to date, the MPC2604GA must still monitor all transactions on the bus. Any transaction that is not a processor burst write will cause the MPC2604GA to assert ARTRY. Burst writes cause the MPC2604GA to do a lookup on the affected address and mark the line invalid if it is present.

Because the MPC2604GA must still monitor all transactions, it cannot use the tag RAM for the flush sequence unless there is a guarantee that no new transaction will be initiated on the bus. The only way to ensure that no new transactions will occur is for the MPC2604GA to be granted the bus. Thus, upon entering the sequence initiated by the assertion of L2 FLUSH, the MPC2604GA will assert L2 BR. As soon as L2 BG is asserted, the MPC2604GA can start stepping through the tag RAM entries.

L2 FLUSH need not be held asserted for the flush sequence to complete. Once started the sequence will run to completion unless overridden by an assertion of HRESET.

L2 MISS INH

When L2 MISS INH is asserted, the MPC2604GA will not load any new data into the cache. The data already present will remain valid and the MPC2604GA will respond to cache hits. This condition only lasts as long as L2 MISS INH is asserted. When L2 MISS INH is negated, the MPC2604GA will start to bring new data into the cache when there are cache misses.

L2 TAG CLR

When L2 TAG CLR is asserted, the MPC2604GA will invalidate all entries in the cache. This internal sequence is the same as the one initiated by an assertion of HRESET. During this sequence, the MPC2604GA will not participate in any bus transaction. However, it will keep track of all bus transactions so that when the sequence is finished, the MPC2604GA can immediately participate in the next bus transaction.

As is the case with assertions of L2 FLUSH, an assertion of L2 TAG CLR need not be held for the duration of the sequence. Once asserted the sequence will run to completion regardless of the state of L2 TAG CLR.

L2 UPDATE INH

When L2 UPDATE INH is asserted, the MPC2604GA is disabled from all activity. Bus transactions continue to be monitored so that as soon as L2 UPDATE INH is negated, the MPC204GA can participate in the next transaction.

Reference Application Note AN1265, "Configuring the MPC2604GA Integrated L2 Cache with the MPC106."

READ HIT/WRITE HIT

Figure 1 shows a read hit from an idle bus state. The MPC2604GA asserts L2 CLAIM the cycle after TS to inform the memory controller that there is a cache hit and the cache will control the rest of the transaction. L2 CLAIM is held through the cycle after AACK is asserted. Since there are no active data tenures from previous transactions, the MPC2604GA asserts AACK the cycle after TS is asserted. Note there must be a qualified assertion of $\overline{\text{CPU DBG}}$ in the same cycle as the assertion of TS for the MPC2604GA to respond with TA in the next cycle. CPU DBG does not affect the timing of L2 CLAIM or AACK.

The write hit timing is virtually the same. The only difference is the processor drives the data instead of the MPC2604GA.

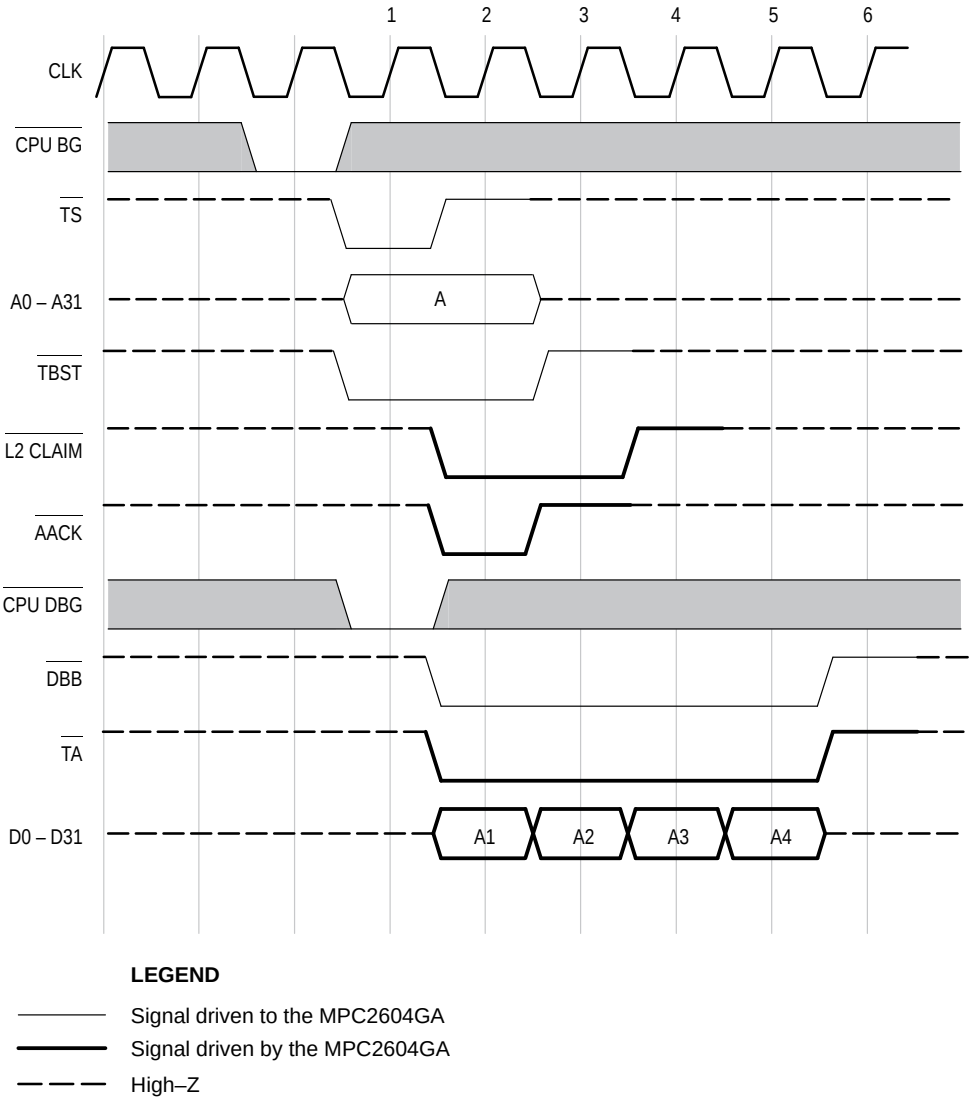


Figure 1. Burst Read (or Write) Hit

MULTIPLE READ/WRITE HITS (Normal Bus Mode)

Figure 2 is an illustration of MPC2604GA pipeline depth limit with multiple read hits. The MPC2604GA supports only one level of address pipelining for data transfer. Therefore, it must hold off on its assertion of AACK for a pipelined TS until the data tenure for the first $\overline{TS}$ is done. The MPC2604GA asserts AACK at the same time as the fourth TA for data tenures that it controls.

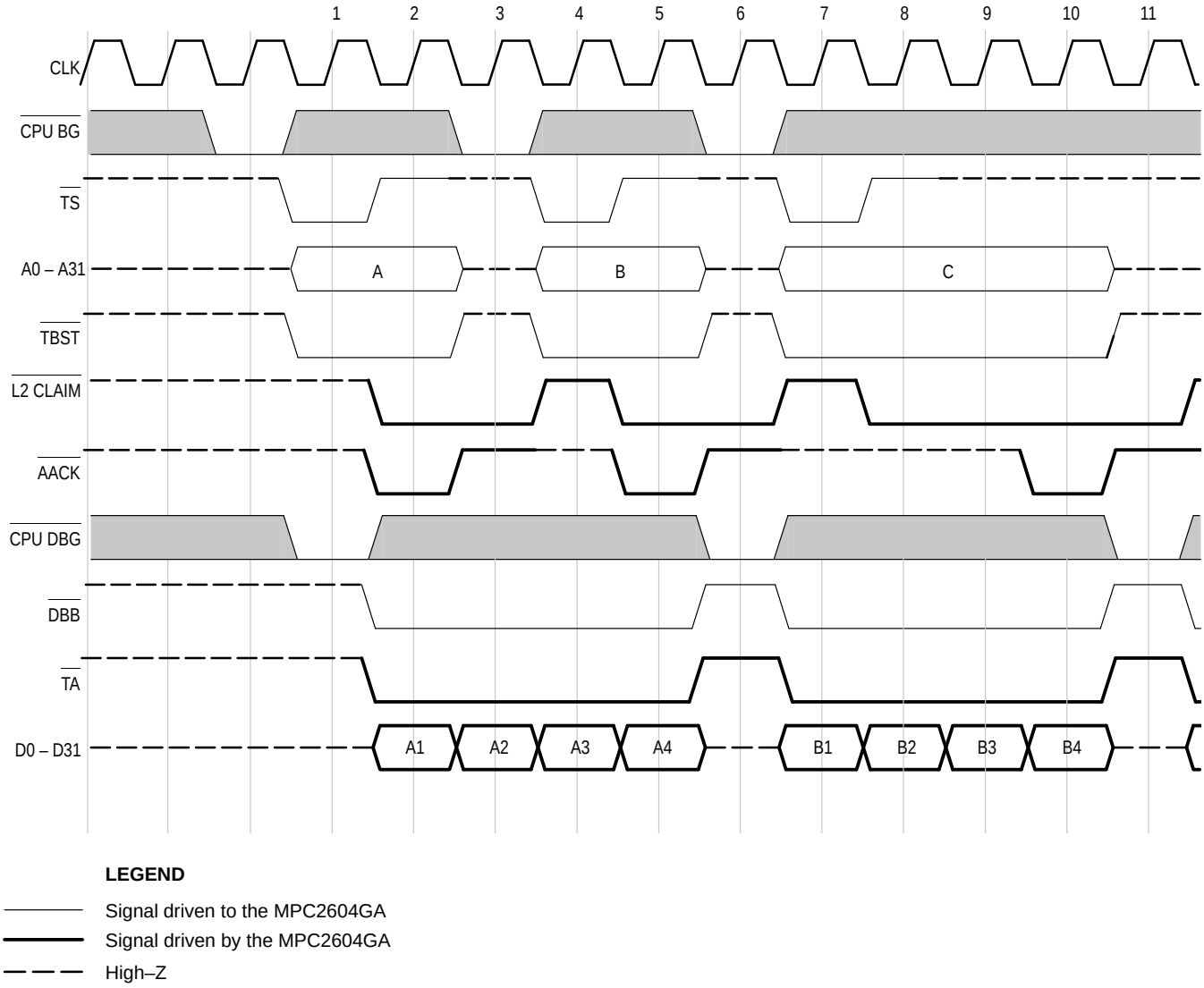


Figure 2. Multiple Burst Read (or Write) Hits

READ MISS (Normal Bus Mode)

Figure 3 is an illustration of MPC2604GA pipeline depth with a read miss followed by a read hit.

For illustration purposes the read miss is shown as a 3–1–1–1 response from memory. AACK for the second access is not driven true until the cycle after the fourth TA of the read miss. This is because the MPC2604GA is not in control of TA for the first access and must, therefore, wait until the first access' data tenure is complete before it can drive AACK true for the read hit.

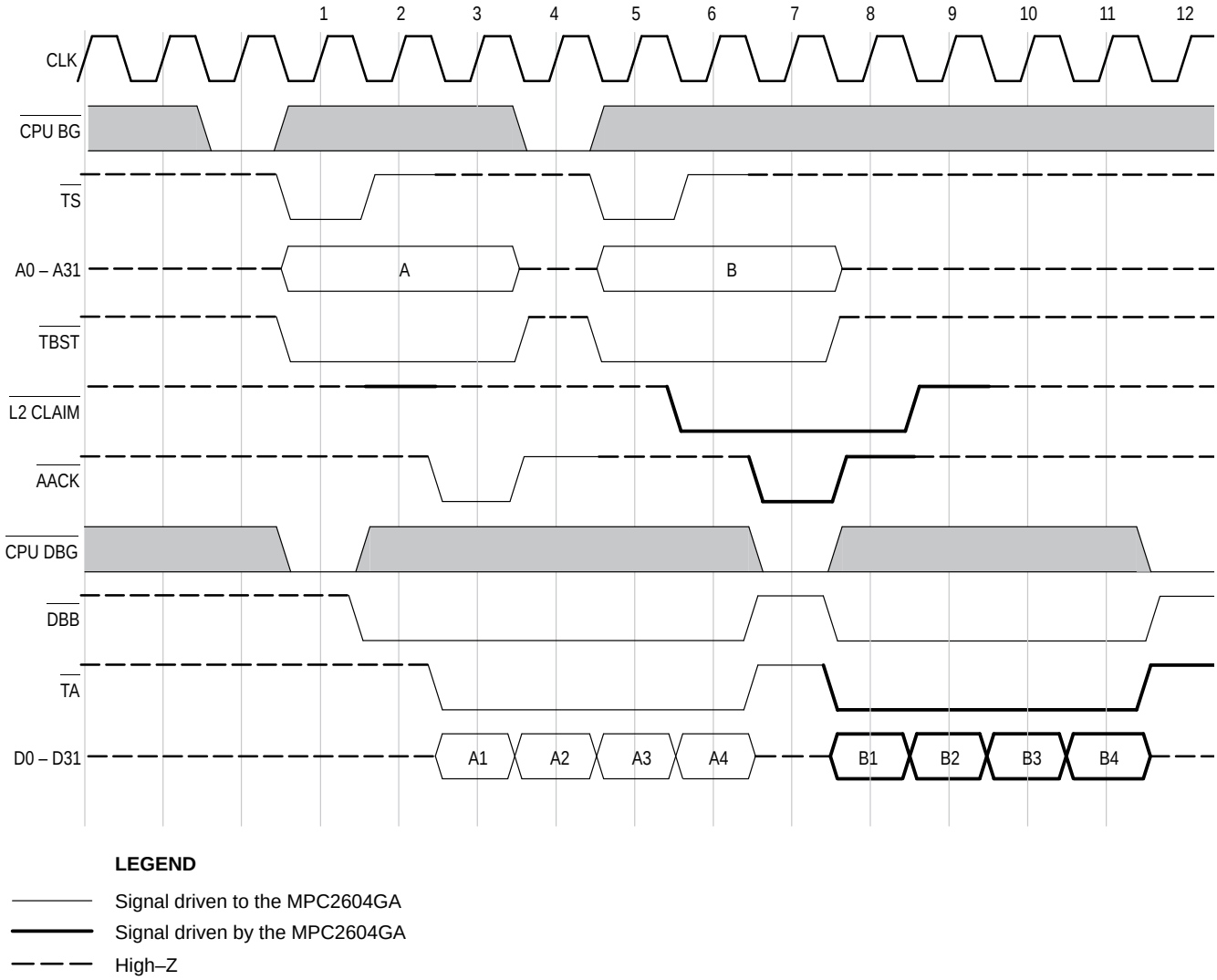


Figure 3. Read Miss Followed by a Burst Read Hit for MPC603/604

MULTIPLE READ HITS (Fast L2 Mode)

Back to back pipelined burst read hits for the MPC604 in Fast L2 mode, also called data streaming mode, are shown in Figure 4. Note that CPU DBG is negated except for the cycles coincident with the fourth TA of each data tenure. This is a requirement for data streaming. Note also that DBB is not shown. For proper operation in Fast L2 mode the DBB pin of the MPC2604GA must be tied to a pull–up resistor.

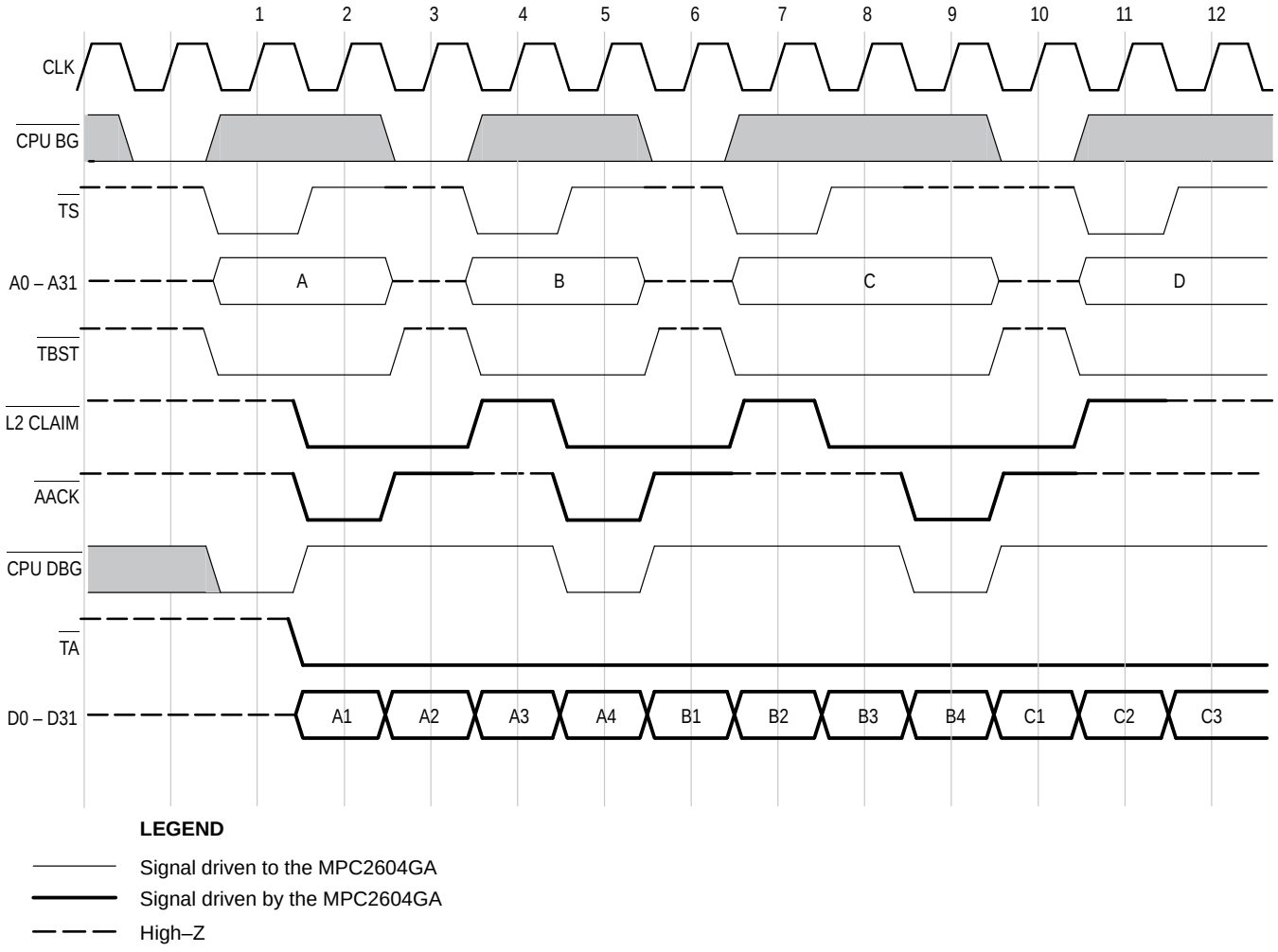


Figure 4. Multiple Burst Read Hits in Fast L2 Mode

WRITE THROUGH BURST WRITE HIT

Figure 5 shows the fastest possible burst write hit to a write–through mode L2 cache line, read miss or write miss processing that replaces a clean line. For these operations MPC2604GA will not assert any signals on the 60X bus. A cache line is considered write through if WT is asserted by the processor when it asserts $\overline{TS}$.

The speed at which a write–through operation completes is solely dependent on the memory controller. The timing shown here assumes that the memory controller has a write buffer that can accept data this quickly.

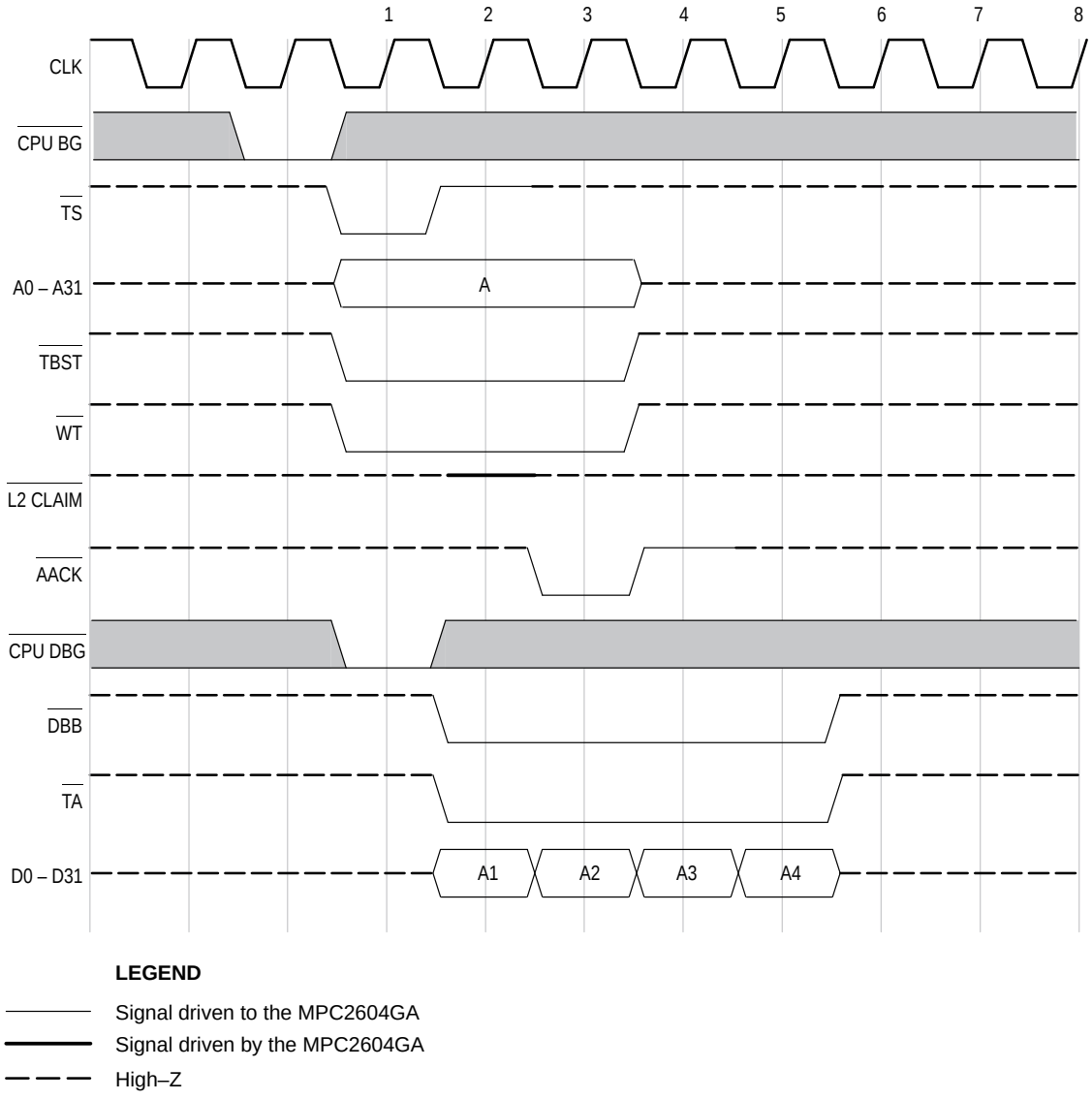
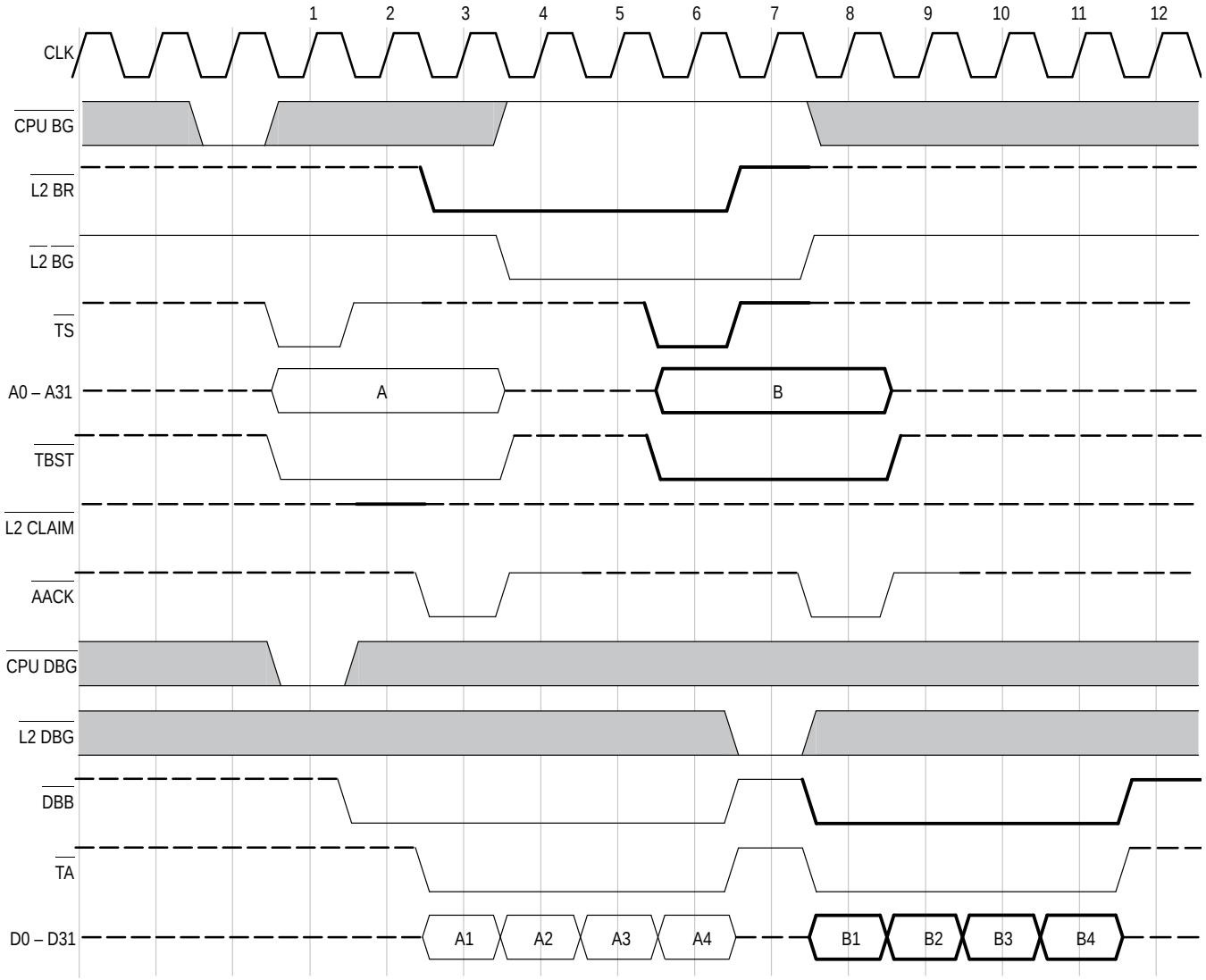


Figure 5. Fastest Possible Write Through Burst Write Hit for MPC603/604

READ/WRITE MISS

Figure 6 is an illustration of a processor read or write miss that causes the MPC2604GA to replace a dirty line. L2 BR is asserted two clocks after TS. The dirty data to be replaced is moved into the internal Cast Out Buffer (COB) at the same time the new data is written into the cache. Note that the copyback operation occurs after the processor request is satisfied. In addition, no delay is added to the processor transaction. It proceeds as fast as the memory controller will allow.



LEGEND

Signal driven to the MPC2604GA

Signal driven by the MPC2604GA

High–Z

Figure 6. Read or Write Miss Followed by Castout

READ/WRITE SNOOP HIT (Dirty L2 Line)

Figure 7 is an illustration of a read or write snoop to a cache line that is dirty in the L2, but is not dirty in the processor's cache. When a snoop hits a dirty line, the MPC2604GA will assert ARTRY through the cycle following the assertion of AACK. This cycle is called the ARTRY window. Note that the MPC2604GA also asserts L2 BR at the same time it asserts ARTRY. Because the snoop could also have hit a dirty line in the processor's cache, the MPC2604GA samples the processor's $\overline{BR}$ signal the cycle following the ARTRY window. This cycle is called the BR window. If the processor's BR signal is not asserted, the MPC2604GA will start sampling L2 BG, the cycle after the BR window.

Note that the MPC2604GA cannot do a 2–1–1–1 copy back burst. The earliest that it can handle the first assertion of TA is two cycles after its assertion of TS.

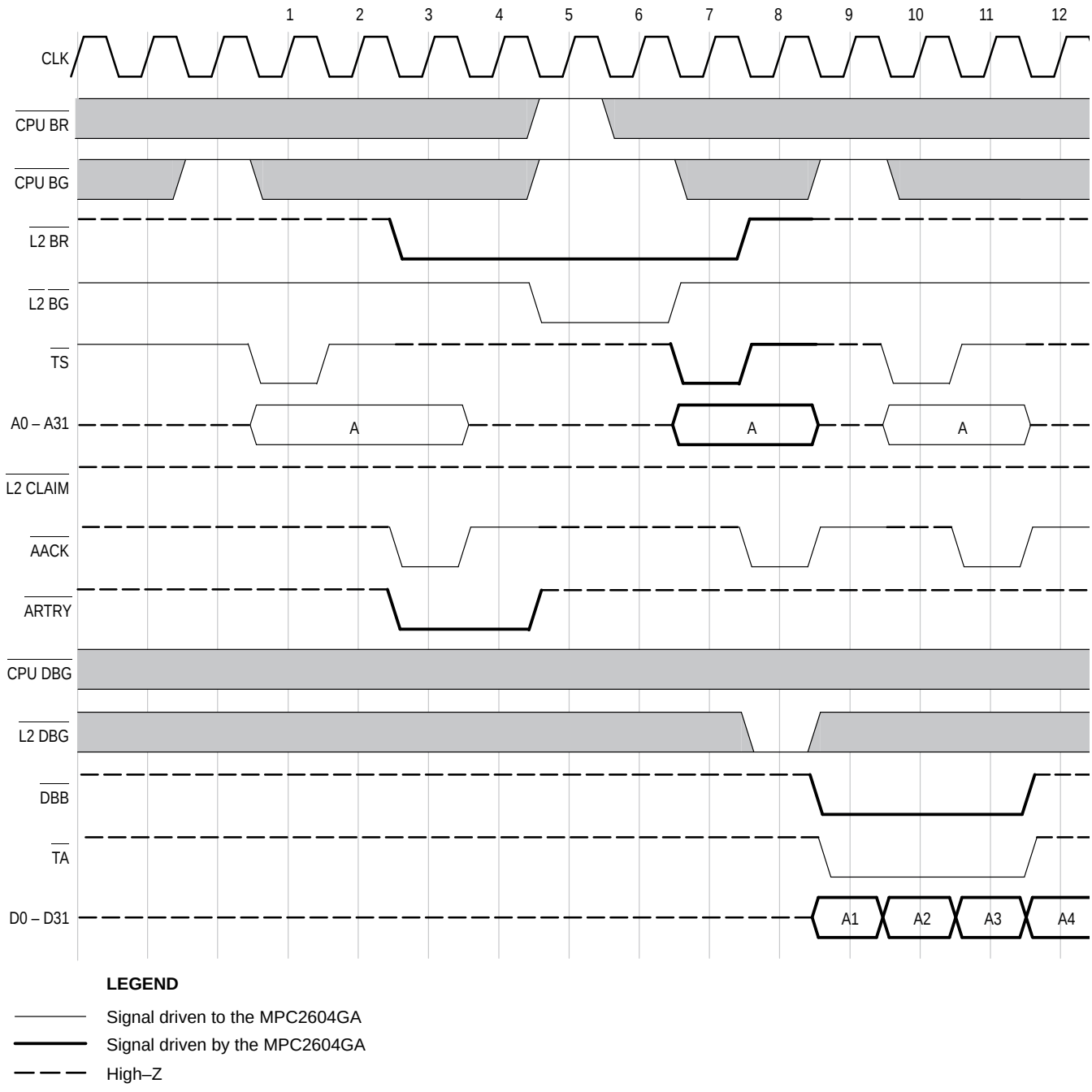


Figure 7. Read or Write Snoop Hit to Dirty L2 Cache Line and Clean Processor Cache Line

READ/WRITE SNOOP HIT (Dirty L2 and Processor Line)

An illustration of PowerPC read or write snoop hit to a dirty L2 cache line is shown in Figure 8. The processor has a dirty copy of the cache line. In this case both the processor and the MPC2604GA assert ARTRY. This situation is detected by sampling CPU BR in the BR window, as described in the previous example. If CPU BR is asserted in the BR window, the MPC2604GA will negate $\overline{\text{L2 BR}}$. It will also ignore assertions of L2 BG. This allows the processor to write back its dirty cache line, at which time the MPC2604GA will either update or invalidate its copy depending on whether it is a snoop read or snoop write.

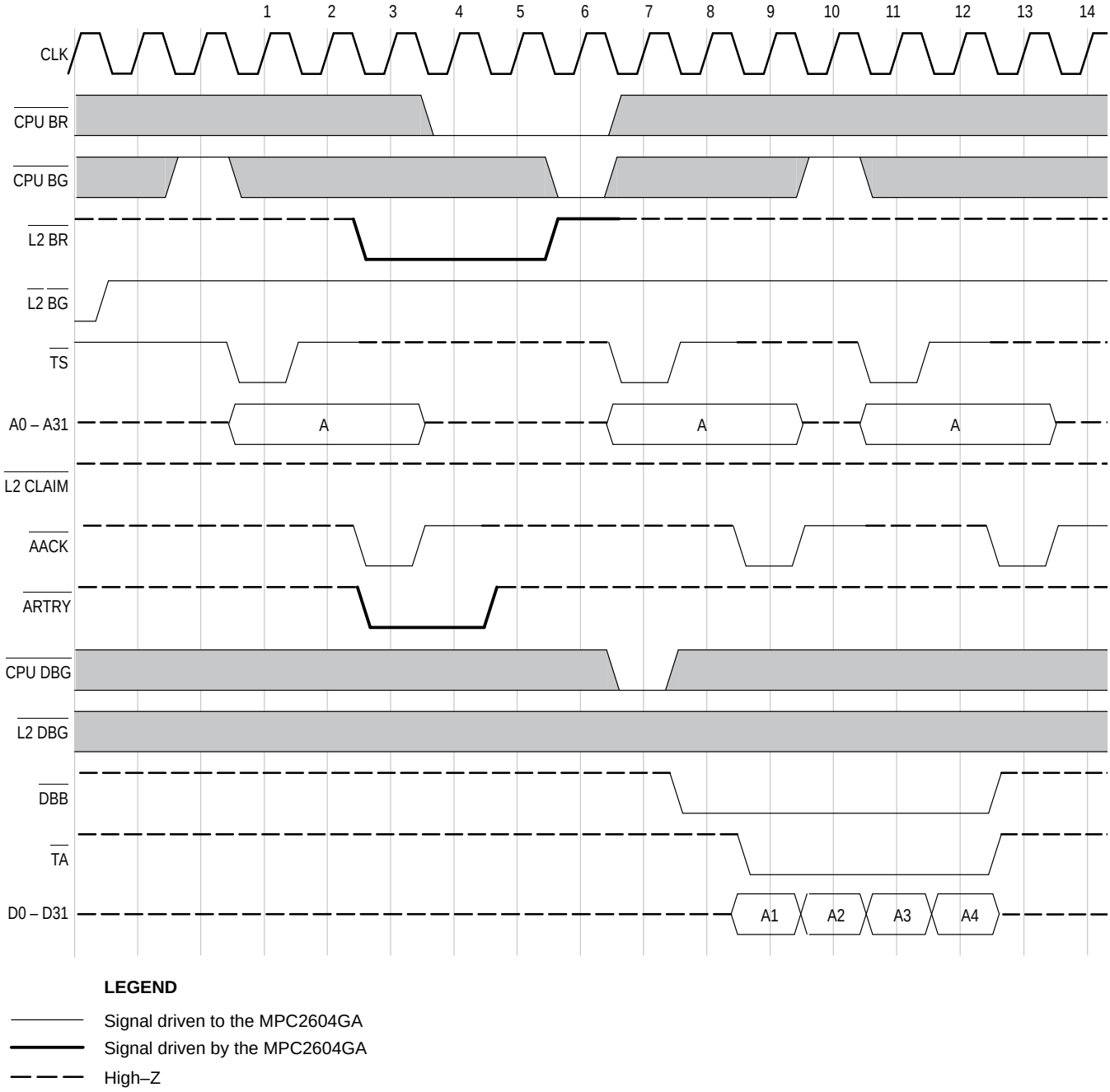


Figure 8. Read or Write Snoop Hit to Dirty L2 Cache Line and Dirty Processor Cache Line

READ HIT/WRITE HIT (Without $\overline{\text{CPU DBG}}$ Parked)

Most of the previous examples have assumed $\overline{\text{CPU DBG}}$ is asserted in the same cycle that the processor asserts TS. This implies CPU DBG is parked. In some systems it may not be desirable or possible to park CPU DBG. Figure 9 shows the response for a read hit from the MPC2604GA is gated by the assertion of CPU DBG. The fastest response possible in a system that does not park CPU DBG is 3–1–1–1.

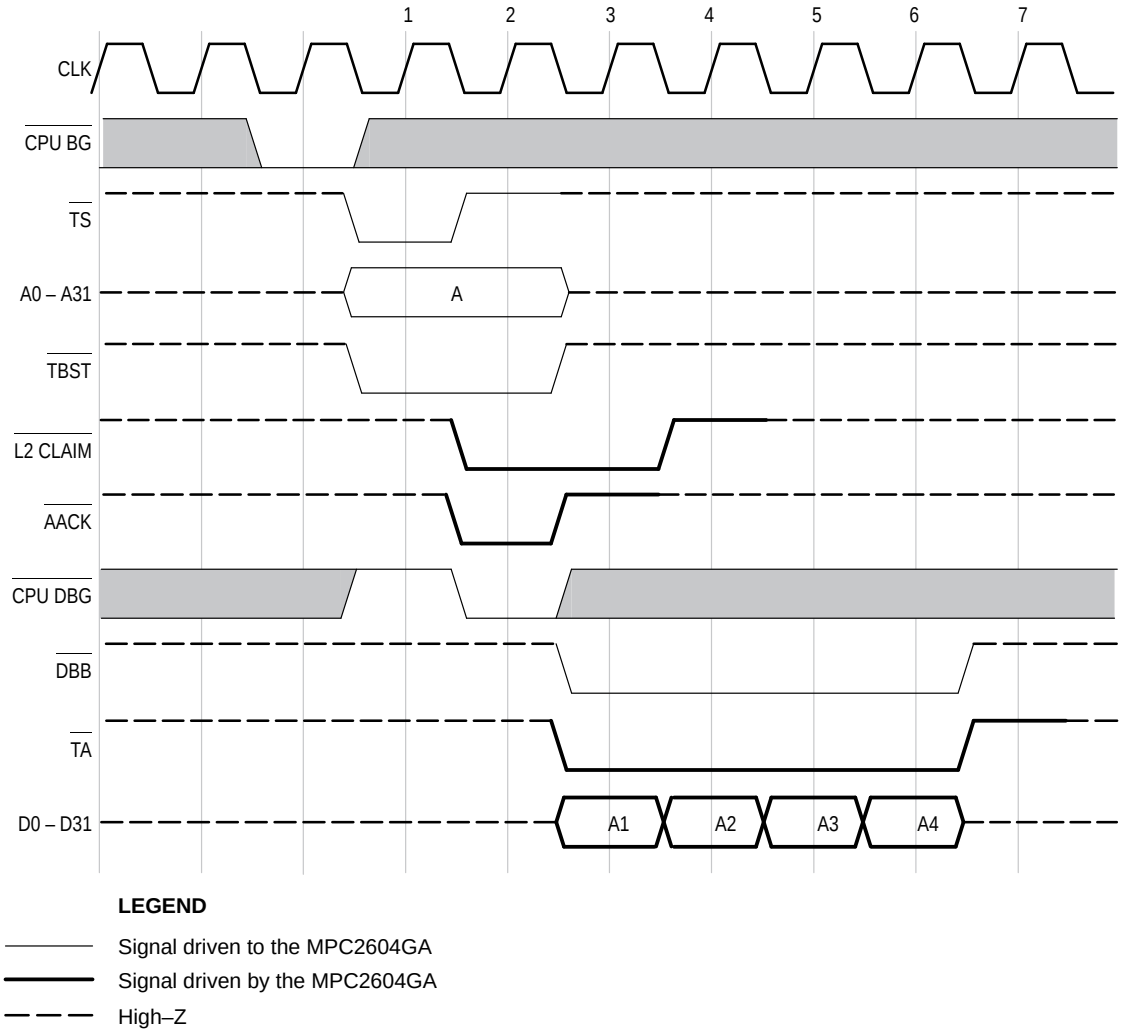


Figure 9. Burst Read (or Write) Hit Without $\overline{\text{CPU DBG}}$ Parked

JTAG

AC OPERATING CONDITIONS AND CHARACTERISTICS FOR THE TEST ACCESS PORT (IEEE 1149.2 (Proposed))

(T_A = 0 to + 70°C, Unless Otherwise Noted)

Input Timing Measurement Reference Level . . . . . 1.5 V
Input Pulse Levels . . . . . 0 to 3.0 V
Input Rise/Fall Time . . . . . 3 ns

Output Measurement Timing Level . . . . . 1.5 V
Output Load . . . . . 50Ω Termination to 1.5 V

TAP CONTROLLER TIMING

Parameter	Symbol	MPC2604GA–66		MPC2604GA–50		Unit	Notes
		Min	Max	Min	Max		
Cycle Time	t_{CK}	30	—	30	—	ns	
Clock High Time	t_{CKH}	12	—	12	—	ns	
Clock Low Time	t_{CKL}	12	—	12	—	ns	
Clock Low to Output Valid	t_A	5	9	5	9	ns	
Clock Low to Output High–Z	t_{CKZ}	0	9	0	9	ns	1
Clock Low to Output Active	t_{CKX}	0	9	0	9	ns	2, 3
Setup Times: TMS TDI	t_s t_{sd}	2	—	2	—	ns	
Hold Times: TMS TDI	t_h t_{hd}	2	—	2	—	ns	

NOTES:
1. TDO will High–Z from a clock low edge depending on the current state of the TAP state machine.
2. TDO is active only in the SHIFT–IR and SHIFT–DR state of the TAP state machine.
3. Transition is measured ± 500 mV from steady–state voltage. This parameter is sampled and not 100% tested.

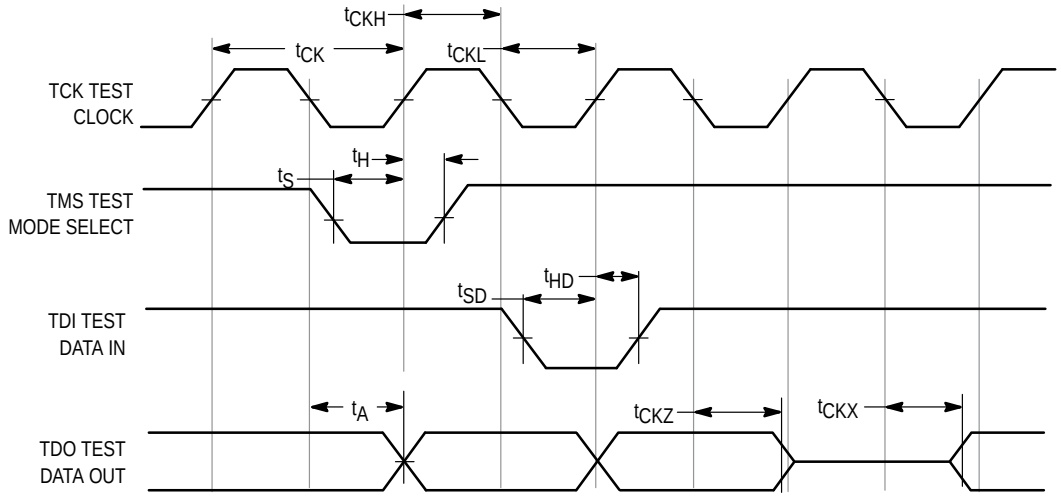


Figure 10. TAP Controller Timing

TEST ACCESS PORT DESCRIPTION

INSTRUCTION SET

A four pin IEEE Standard 1149.2 Test Port (JTAG) is included on this device. When the TAP (Test Access Port) controller is in the SHIFT–IR state, the instruction register is placed between TDI and TDO. In this state, the desired instruction would be serially loaded through the TDI input. The TAP instruction set for this device are as follows.

STANDARD INSTRUCTIONS

Instruction	Code (Binary)	Description
BYPASS	1111*	Bypass instruction
SAMPLE/PRELOAD	0010	Sample and/or preload instruction
EXTEST	0000	Extest instruction
HIGHZ	1001	High–Z all output pins while bypass register is between TDI and TDO
CLAMP	1100	Clamp output pins while bypass register is between TDI and TDO

* Default state at power–up.

TAP STANDARD INSTRUCTION SET

NOTE

The descriptions in this section are not intended to be used without the supporting IEEE 1149.1 – 1995 Standard.

SAMPLE/PRELOAD TAP INSTRUCTION

The SAMPLE/PRELOAD TAP instruction is used to allow scanning of the boundary–scan register without causing interference to the normal operation of the chip logic. The 123–bit boundary–scan register contains bits for all device signal and clock pins and associated control signals. This register is accessible when the SAMPLE/PRELOAD TAP instruction is loaded into the TAP instruction register in the SHIFT–IR state. When the TAP controller is then moved to the SHIFT–DR state, the boundary–scan register is placed between TDI and TDO. This scan register can then be used prior to the EXTEST instruction to preload the output pins with desired values so that these pins will drive the desired state when the EXTEST instruction is loaded. As data is written into TDI, data also streams out TDO which can be used to pre–sample the inputs and outputs.

SAMPLE/PRELOAD would also be used prior to the CLAMP instruction to preload the values on the output pins that will be driven out when the CLAMP instruction is loaded.

EXTEST TAP INSTRUCTION

The EXTEST instruction is intended to be used in conjunction with the SAMPLE/PRELOAD instruction to assist in testing board level connectivity. normally, the SAMPLE/PRELOAD instruction would be used to preload all output pins. The EXTEST instruction would then be loaded. During EXTEST, the boundary–scan register is placed between TDI and TDO in the SHIFT–DR state of the TAP controller. Once the EXTEST instruction is loaded, the TAP controller would then be moved to the Run–Test/Idle state. In this state, one cycle of TCK would cause the preloaded data on the output pins to be driven while the values on the input pins would be sampled. Note the TCK, not the clock pin (CLK), is used as the clock input while CLK is only sampled during EXTEST. After one clock cycle of TCK, the TAP controller would then be moved to the SHIFT–DR state where the sampled values would be shifted out of TDO (and new values would be shifted in TDI). These values would normally be compared to expected values to test for board connectivity.

CLAMP TAP INSTRUCTION

The CLAMP instruction is provided to allow the state of the signals driven from the output pins to be determined from the boundary–scan register while the bypass register is selected as the serial path between TDI and TDO. The signals driven from the output pins will not change while the CLAMP instruction is selected. EXTEST could also be used for this purpose, but CLAMP shortens the board scan path by inserting only the bypass register between TDI and TDO. To use CLAMP, the SAMPLE/PRELOAD instruction would be used first to scan in the values that will be driven on the output pins when the CLAMP instruction is active.

HIGHZ TAP INSTRUCTION

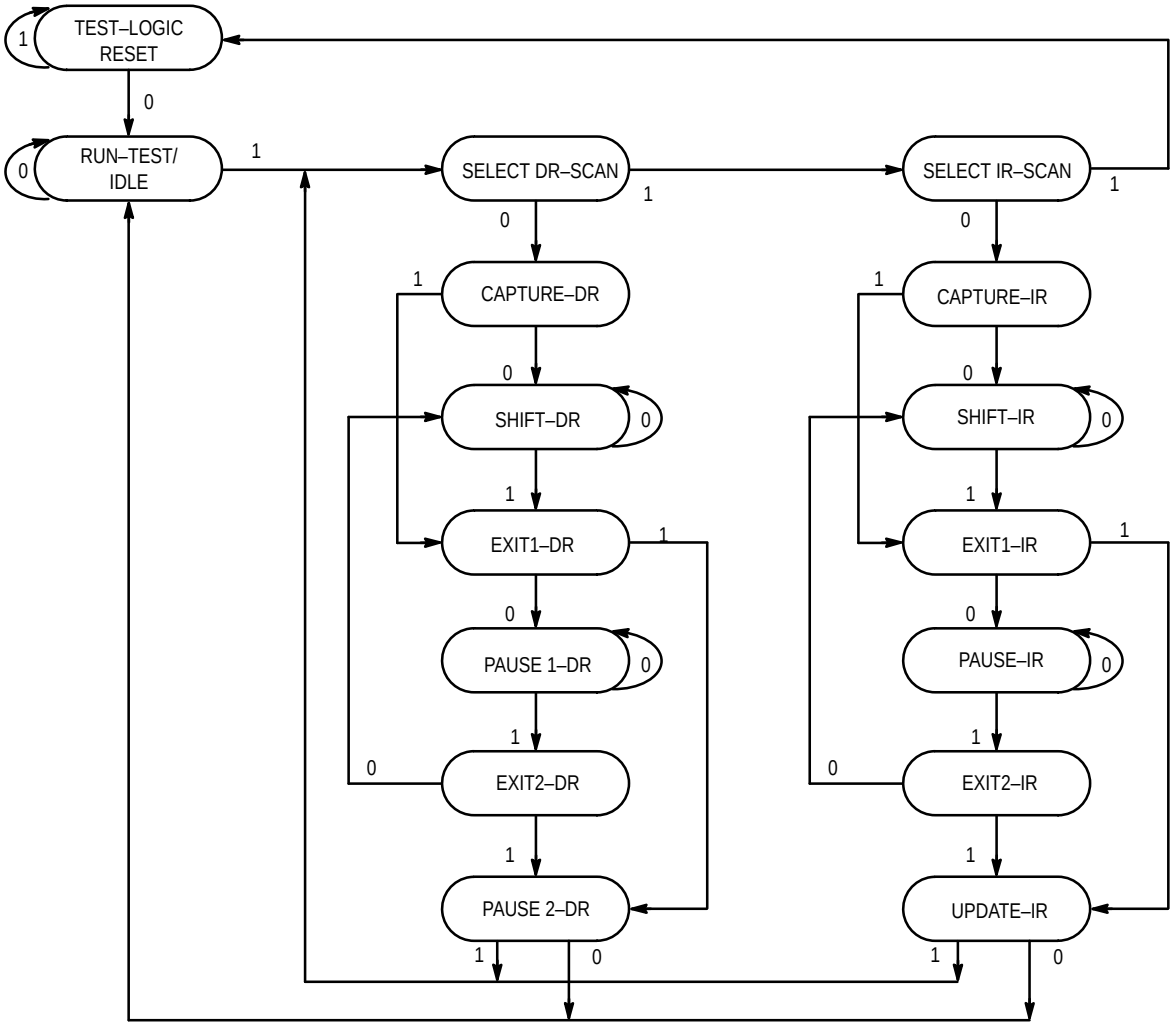
The HIGHZ instruction is provided to allow all the outputs to be placed in an inactive drive state (High–Z). During the HIGHZ instruction the bypass register is connected between TDI and TDO.

BYPASS TAP INSTRUCTION

The BYPASS instruction is the default instruction loaded at power up. This instruction will place a single shift register between TDI and TDO during the SHIFT–DR state of the TAP controller. This allows the board level scan path to be shortened to facilitate testing of other devices in the scan path.

DISABLING THE TEST ACCESS PORT AND BOUNDARY SCAN

It is possible to use this device without utilizing the four pins used for the test access port. To circuit disable the device, TCK must be tied to V_{SS} to preclude mid level inputs. Although TDI and TMS are designed in such a way that an undriven input will produce a response equivalent to the application of a logic 1, it is still advisable to tie these inputs to V_{DD} through a 1K resistor. TDO should remain unconnected.



NOTE: The value adjacent to each state transition represents the signal present at TMS at the rising edge of TCK.

Figure 11. TAP Controller State Diagram

BOUNDARY SCAN ORDER

BIT NUMBER

The order of the boundary scan chain. Bit 0 is the closest to TDO.

BIT/PIN NAME

The name of the physical pin. For an output enable cell, this is the name of the corresponding output enable.

BIT/PIN TYPE

Input — Input only pin.
I/O — Bi–directional pin that can be put into high–Z state.
Output Enable — Boundary scan cell to hold the output enable state of other I/O pads. Output enable does not correspond to a physical pin. To set an I/O to an input, the output enable must have a 1. To set an I/O to an output, the output enable must have a 0. Note that these internal output enables are active low.

OUTPUT ENABLE

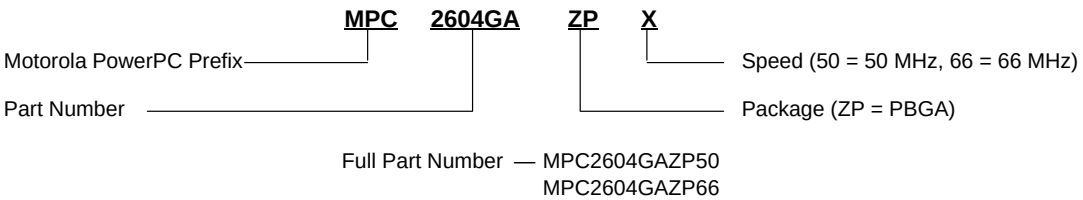
The name of the output enable cell that determines if the cell is enabled or in the high–Z state. If the pin type is input or output enable, this entry will be empty.

Bit Number	Bit/Pin Name	Bit/Pin Type	Output Enable
0	D13	I/O	DOE
1	D14	I/O	DOE
2	D15	I/O	DOE
3	D10	I/O	DOE
4	D11	I/O	DOE
5	D12	I/O	DOE
6	DP1	I/O	DOE
7	D8	I/O	DOE
8	D9	I/O	DOE
9	D5	I/O	DOE
10	D6	I/O	DOE
11	D7	I/O	DOE
12	D2	I/O	DOE
13	D3	I/O	DOE
14	D4	I/O	DOE
15	DP0	I/O	DOE
16	D0	I/O	DOE
17	D1	I/O	DOE
18	DP2	I/O	DOE
19	D17	I/O	DOE
20	D16	I/O	DOE
21	D18	I/O	DOE
22	D19	I/O	DOE
23	D20	I/O	DOE
24	D21	I/O	DOE
25	D22	I/O	DOE
26	D23	I/O	DOE
27	DP3	I/O	DOE
28	D24	I/O	DOE
29	D25	I/O	DOE
30	D26	I/O	DOE
31	D27	I/O	DOE
32	D28	I/O	DOE
33	D29	I/O	DOE
34	D30	I/O	DOE
35	D31	I/O	DOE
36	CPU4 BR	Input	
37	L2 UPDATE INH	Input	
38	PWRDN	Input	
39	L2 MISS INH	Input	
40	L2 TAG CLR	Input	
41	L2 BG	Input	
42	L2 FLUSH	Input	
43	CFG5	Input	
44	CFG0	Input	
45	CFG1	Input	
46	CFG2	Input	
47	CPU3 BR	Input	
48	MODE4P	Input	
49	CPU2 DBG	Input	
50	CPU2 BG	Input	
51	CPU2 BR	Input	
52	FDN	I/O	FDNOE
53	L2 DBG	Input	
54	L2 BR	I/O	L2BROE
55	L2 CLAIM	I/O	L2CLAIMOE
56	CPU DBG	Input	
57	CPU BG	Input	
58	AACK	I/O	AACKOE
59	ARTRY	I/O	ARTRYOE
60	CPU3 DBG	Input	
61	CPU4 DBG	Input	
62	CI	I/O	AOE
63	WT	I/O	AOE
64	CPU BR	Input	

Bit Number	Bit/Pin Name	Bit/Pin Type	Output Enable
65	HRESET	Input	
66	TSIZ0	I/O	AOE
67	TSIZ1	I/O	AOE
68	TSIZ2	I/O	AOE
69	SRESET	Input	
70	TBST	I/O	AOE
71	TT2	I/O	AOE
72	TT1	I/O	AOE
73	TT0	I/O	AOE
74	CPU3 BG	Input	
75	TT4	I/O	AOE
76	TS	I/O	AOE
77	TA	I/O	TAOE
78	TT3	I/O	AOE
79	CPU4 BG	Input	
80	TEA	Input	
81	DBB	I/O	DBBOE
82	TAOE	Output Enable	
83	SHDOE	Output Enable	
84	L2CLAIMOE	Output Enable	
85	L2BROE	Output Enable	
86	FDNOE	Output Enable	
87	DBBOE	Output Enable	
88	DOE	Output Enable	
89	ARTRYOE	Output Enable	
90	AACKOE	Output Enable	
91	A31	I/O	AOE

Bit Number	Bit/Pin Name	Bit/Pin Type	Output Enable
92	A30	I/O	AOE
93	A29	I/O	AOE
94	A28	I/O	AOE
95	A27	I/O	AOE
96	A26	I/O	AOE
97	A25	I/O	AOE
98	A24	I/O	AOE
99	A23	I/O	AOE
100	A22	I/O	AOE
101	A21	I/O	AOE
102	A20	I/O	AOE
103	A19	I/O	AOE
104	A18	I/O	AOE
105	A17	I/O	AOE
106	A16	I/O	AOE
107	A15	I/O	AOE
108	A14	I/O	AOE
109	A13	I/O	AOE
110	A12	I/O	AOE
111	A11	I/O	AOE
112	A10	I/O	AOE
113	A9	I/O	AOE
114	A8	I/O	AOE
115	A7	I/O	AOE
116	A6	I/O	AOE
117	A5	I/O	AOE
118	A4	I/O	AOE
119	A3	I/O	AOE
120	A2	I/O	AOE
121	A1	I/O	AOE
122	A0	I/O	AOE
123	AOE	Output Enable	

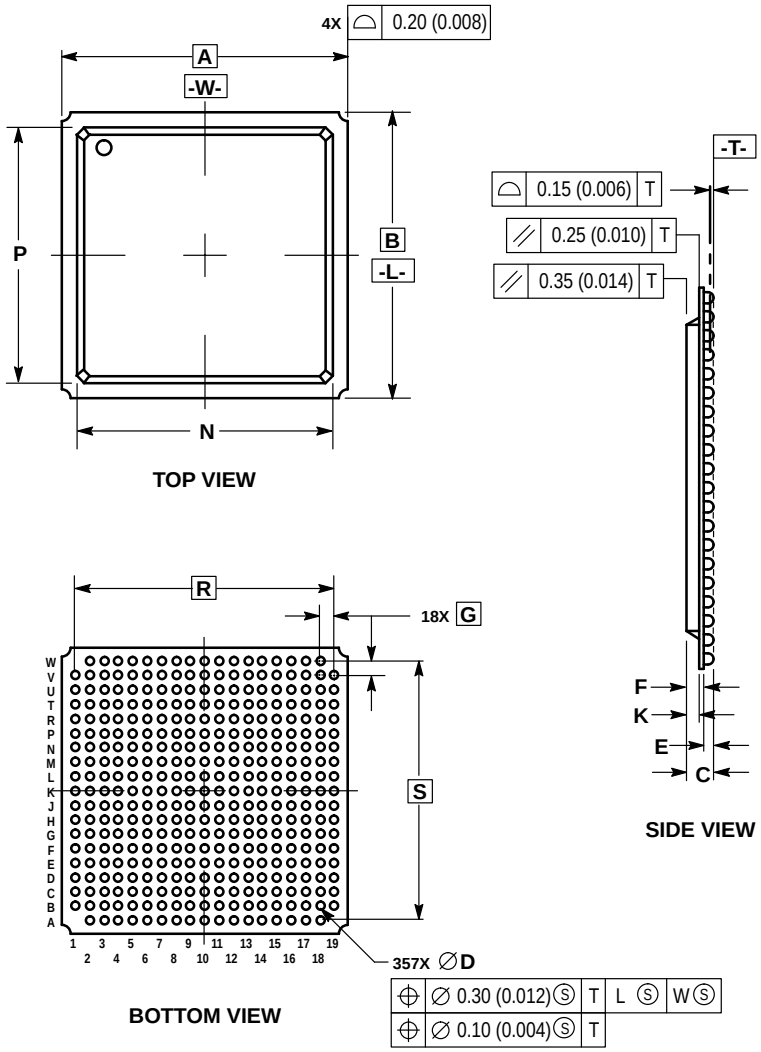
ORDERING INFORMATION
(Order by Full Part Number)



Full Part Number — MPC2604GAZP50
MPC2604GAZP66

PACKAGE DIMENSIONS

ZP PACKAGE
PBGA
CASE 1103–01



NOTES:
4. DIMENSIONING AND TOLERANCING PER ANSI Y14.5M, 1982.
5. CONTROLLING DIMENSION: MILLIMETER.

	MILLIMETERS		INCHES	
DIM	MIN	MAX	MIN	MAX
A	25.00 BSC		0.984 BSC	
B	25.00 BSC		0.984 BSC	
C	—	2.05	—	0.081
D	0.60	0.90	0.024	0.035
E	0.50	0.70	0.020	0.028
F	0.95	1.35	0.037	0.053
G	1.27 BSC		0.05 BSC	
K	0.70	0.90	0.028	0.035
N	22.40	22.60	0.882	0.890
P	22.40	22.60	0.882	0.890
R	22.86 BSC		0.900 BSC	
S	22.86 BSC		0.900 BSC	

How to reach us:
USA / EUROPE / Locations Not Listed: Motorola Literature Distribution; P.O. Box 20912; Phoenix, Arizona 85036. 1–800–441–2447 or 602–303–5454

MFAX: RMFAX0@email.sps.mot.com – TOUCHTONE 602–244–6609
INTERNET: http://Design–NET.com

JAPAN: Nippon Motorola Ltd.; Tatsumi–SPD–JLDC, 6F Seibu–Butsuryu–Center, 3–14–2 Tatsumi Koto–Ku, Tokyo 135, Japan. 03–81–3521–8315

ASIA / PACIFIC: Motorola Semiconductors H.K. Ltd.; 8B Tai Ping Industrial Park, 51 Ting Kok Road, Tai Po, N.T., Hong Kong. 852–26629298