



# ***PowerPC™***

## Application Note

### **PowerPC™ 60x Bus Implementation Differences**

This document details differences in implementation of the PowerPC 60x Bus with the MPC750, MPC7400, MPC7410, and MPC7450 microprocessors.

The 60x Bus is a high-performance bus specification with separate address and data buses, each with its own set of arbitration and control signals. This allows for the decoupling of the data tenure from the address tenure of a transaction and provides for a wide range of system bus implementations, including the following:

- non-pipelined bus operation
- pipelined bus operation
- split transaction operation

## **1.1 I/O Voltage Level**

### **1.1.1 MPC750**

The 60x Bus input/output voltage levels for the MPC750 are set at 3.3V and are not configurable.

### **1.1.2 MPC7400/MPC7410 and MPC7450**

The 60x Bus input/output voltage levels for the MPC7400, the MPC7410, and the MPC7450 are specified in Table 1.

Table 1. 60x Bus I/O Voltage Configuration

Pin <sup>1</sup>	Connection <sup>2</sup>	MPC7400	MPC7410	MPC7450
BVSEL	HRESET	2.5V I/O	2.5V I/O	2.5V I/O
	GND	1.8V I/O	1.8V I/O	1.8V I/O
	OVDD	3.3V I/O (ipu)	2.5V I/O (ipu)	2.5V I/O (ipu <sup>3</sup> )
	$\overline{\text{HRESET}}$	3.3V I/O	not supported	not supported

<sup>1</sup> Configurations are determined by sampling the respective input pin before and after reset negation.

<sup>2</sup> Connecting an input to  $\overline{\text{HRESET}}$  provides the “low during hreset, high after hreset” encoding.

<sup>3</sup> ipu = internal pullup; default if unconnected.

## 1.2 Signal Differences

Signal differences are described in Table 2.

Table 2. Signal Differences

Description	MPC750	MPC7400/MPC7410	MPC7450
60x bus mode	default	EMODE = VDD	BMODE $\overline{0}$ = VDD BMODE $\overline{1}$ = VDD
Address bus	A[0:31]	A[0:31]	A[0:35]
Address parity	AP[0:3]	AP[0:3]	AP[0:4]
Address bus busy	$\overline{\text{ABB}}$ (input/output) (weak pull-up required)	ABB (output only)	not supported
Transaction burst	TBST (input/output)	TBST (output only)	TBST (output only)
Cache Inhibited	$\overline{\text{CI}}$ (output only)	$\overline{\text{CI}}$ (input/output) (weak pull-up recommended)	$\overline{\text{CI}}$ (output only)
Write through	WT (output Only)	WT (input/output) (weak pull-up recommended)	WT (output only)
Data bus busy	$\overline{\text{DBB}}$ (Input/output) (weak pull-up required)	DBB (output only)	not supported
Data bus write only	DBWO	DBWO	not supported
Data bus disable	DBDIS	not supported	not supported
Data retry	DRTRY	not supported	not supported
Reservation	RSRV	RSRV	not supported
TLBI synchronize	TLBISYNC	not supported	not supported

## 1.3 Extended Addressing

### 1.3.1 MPC750 and MPC7400/MPC7410

The MPC750 and the MPC7400/MPC7410 support a 32-bit address bus with four bits of odd parity.

### 1.3.2 MPC7450

The MPC7450 supports both a 32-bit addressing mode and a 36-bit extended addressing mode. The MPC7450 can be configured to support extended addressing by setting the XAEN bit of HID0 (bit 14).

When extended physical addressing is disabled, the MPC7450 will drive the four most significant bits to zeroes. Note that the four most significant bits are still sampled and should be actively pulled to zero if they are not being used in a system. For compatibility between the MPC7450 and previous microprocessors when extended physical addressing is disabled, note the following:

- connect the MPC7450 A[0:3] to b'0000'
- connect the MPC7450 A[4:35] to the MPC750/MPC7400/MPC7410 A[0:31]
- connect the MPC7450 AP[0] to b'1'
- connect the MPC7450 AP[1:4] to the MPC750/MPC7400/MPC7410 AP[0:3]

When extended physical addressing is enabled, the MPC7450 drives a 36-bit physical address with five bits of odd parity.

- AP[0] contains odd parity for A[0:3].
- AP[1] contains odd parity for A[4:11].
- AP[2] contains odd parity for A[12:19].
- AP[3] contains odd parity for A[20:27].
- AP[4] contains odd parity for A[28:35].

The boot address/reset vector is the same as on previous chips (0x0FFF00100) because extended addressing is only enabled when translation is enabled and the reset vector is mapped in real mode.

## 1.4 Bus Request

All processors may issue  $\overline{BR}$  for a non-window-of-opportunity transaction and then de-assert  $\overline{BR}$  without running a transaction. This may occur if the transaction is cancelled internally.

### 1.4.1 MPC7400/MPC7410

The MPC7400/MPC7410 may issue  $\overline{BR}$  for a non-window of opportunity transaction and not be able to accept immediately a qualified bus grant if the limit of outstanding address tenures has been reached. This situation resolves itself if forward progress can be made on the system data bus. Note that it may be necessary to re-arbitrate the  $\overline{BG}$  every cycle to allow for a higher priority request when the current  $\overline{BR}$  was not made in the window of opportunity.

## 1.5 $\overline{\text{ABB}}$ and Address Bus Arbitration

### 1.5.1 MPC750

The MPC750 defines the  $\overline{\text{ABB}}$  pin as both input and output.

### 1.5.2 MPC7400/MPC7410

The MPC7400/MPC7410 defines the  $\overline{\text{ABB}}$  pin as output only and also generates an internal *abb* that tracks outstanding transactions. The processor relies on the system arbiter to provide grants for the address bus only when the bus is available and the grant may be accepted. Note that this may have implications if the system arbiter is parking the address bus and counting on  $\overline{\text{ABB}}$  as an input to perform arbitration.

### 1.5.3 MPC7450

The MPC7450 does not include an  $\overline{\text{ABB}}$  pin but does generate an internal *abb* that tracks outstanding transactions. The processor relies on the system arbiter to provide grants for the address bus only when the bus is available and the grant may be accepted. Note that this may have implications if the system arbiter is parking the address bus and counting on  $\overline{\text{ABB}}$  as an input to perform arbitration.

## 1.6 Address Transfer Attributes

Differences between the MPC750, the MPC7400/MPC7410, and the MPC7450 with respect to the setting of the WIMG bits are as follows:

- $\overline{\text{WT}}$ . On all processors, this signal reflects the write-through status for a transaction as determined by the MMU address translation. It is also asserted for burst writes due to **dcbf** (Flush) and **dcbst** (Clean) instructions and for snoop pushes; it is negated for **ecowx** transactions. The MPC750 and the MPC7450 also use the  $\overline{\text{WT}}$  signal during read transactions to indicate the transaction is an instruction fetch ( $\overline{\text{WT}} = 1$ ) or not an instruction fetch ( $\overline{\text{WT}} = 0$ ).
- $\overline{\text{CI}}$ . On all processors, this signal reflects the cache-inhibited status for a transaction as determined by the MMU address translation unless the L1 cache is disabled. It is always asserted for **eciwx/ecowx** bus transactions independent of the address translation.
- $\overline{\text{GBL}}$ . On all processors, this signal indicates that the transaction is global and should be snooped by other masters (output) or must be snooped by the processor (input). The  $\overline{\text{GBL}}$  bit is asserted for **sync**, **tlbsync**, **tlbie**, and **eieio** instructions. The  $\overline{\text{GBL}}$  bit is always deasserted for castouts, snoop pushes, and **eciwx/ecowx** instructions.
- The MPC750 can be configured to override the M bit from translation and always treat instruction fetches as non-global. This mode is selected by clearing the IFEM bit (bit 23) of HID0.
- The MPC7400/MPC7410 also asserts  $\overline{\text{GBL}}$  for Kill transactions due to **dcba** and **dcbz** instructions that reach the 60x Bus.

## 1.7 Instruction Fetch Differentiation

### 1.7.1 MPC750 and MPC7450

The MPC750 and the MPC7450 use the  $\overline{WT}$  signal during read transactions to indicate that the transaction is an instruction fetch ( $\overline{WT} = 1$ ) or not an instruction fetch ( $\overline{WT} = 0$ ).

### 1.7.2 MPC7400/MPC7410

The MPC7400/MPC7410 can be configured to differentiate instruction fetches from data fetches by setting the IFTT bit (bit 23) of HID0. The TT code for all D-side reads will be changed from Read (TT=01010) to Read Atomic (TT=11010). I-side reads will continue to be identified as Read (TT=01010).

## 1.8 Address Tenure Termination

### 1.8.1 MPC750

The MPC750 requires no minimum processor-clock-to-bus-clock ratio to process a snoop request.

### 1.8.2 MPC7400/MPC7410

The MPC7400/MPC7410 requires a minimum of two processor cycles to process a snoop and generate a response after latching  $\overline{TS}$  and associated transfer attributes.

### 1.8.3 MPC7450

The MPC7450 requires a minimum of five processor cycles to process a snoop and to generate a response after latching  $\overline{TS}$  and associated transfer attributes. As a result, if the system bus is running faster than one-fifth the processor frequency, the system must extend the address tenure of all transactions that will be snooped by a MPC7450 by delaying assertion of  $\overline{AACK}$ . For Core:Bus frequency multiples of 2:1 and 2.5:1,  $\overline{AACK}$  must be delayed a minimum of two bus cycles. For Core:Bus frequency multiples of 3:1, 3.5:1, 4:1, and 4.5:1,  $\overline{AACK}$  must be delayed a minimum of one bus cycle.

## 1.9 Enveloped Transactions

### 1.9.1 MPC750 and MPC7400/MPC7410

The MPC750 and the MPC7400/MPC7410 support enveloped transactions where  $\overline{AACK}$  is delayed long enough so that the entire data tenure is contained within the address tenure.

### 1.9.2 MPC7450

The MPC7450 does not support enveloped transactions.

## 1.10 Address Retry

### 1.10.1 MPC7450

In the MPC7450, a retry response (assertion of the  $\overline{\text{ARTRY}}$  signal) indicates the address tenure should be rerun at a later time. There should be no assumptions about internal effects that the retried operation had on the MPC7450.

Some assertions of  $\overline{\text{ARTRY}}$  by the MPC7450 indicate that the snooping processor requires access to the bus to eliminate the retry condition but is unable to use the window of opportunity to do so (for example, for TLBSYNC). The system must perform fair arbitration to allow all retrying processors to clear such stall conditions. Continuing to give arbitration priority to the last master after an  $\overline{\text{ARTRY}}$  may cause a deadlock condition.

## 1.11 Window of Opportunity

### 1.11.1 MPC750

If the MPC750 has a snoop copyback to perform and can perform it, the MPC750 asserts  $\overline{\text{BR}}$  in the window of opportunity. However, in some cases, the processor may not be able to accept a qualified bus grant until several clocks after the window of opportunity. Therefore, to perform a snoop copyback, if the system bus arbiter asserts  $\overline{\text{BG}}$  to the processor during the cycle after the window of opportunity (such as the third cycle after  $\overline{\text{AACK}}$ ), it must keep  $\overline{\text{BG}}$  asserted until it recognizes the assertion of  $\overline{\text{TS}}$  indicating that the processor has started the snoop copyback transaction. The MPC750 is not guaranteed to hold  $\overline{\text{BR}}$  asserted if  $\overline{\text{BG}}$  has been parked.

### 1.11.2 MPC7400/MPC7410

If the MPC7400/MPC7410 has a snoop copyback to perform and can perform it, the MPC7400/MPC7410 asserts  $\overline{\text{BR}}$  in the window of opportunity. However, in some cases, the processor may not be able to accept a qualified bus grant until many clocks after the window of opportunity. This is true if the processor is providing data from the off-chip L2, if the internal data transaction queue is full, or if the processor owns a cache line because the address tenure of a bus transaction has been performed but does not yet have possession of the data because the data tenure of the bus transaction has not been performed.

In some circumstances, forward progress on the data bus must be maintained to allow the address bus to make forward progress. Therefore, to perform a snoop copyback, if the system bus arbiter asserts  $\overline{\text{BG}}$  to the processor during the cycle after the window of opportunity (such as the third cycle after  $\overline{\text{AACK}}$ ), it should keep  $\overline{\text{BG}}$  asserted until it recognizes the negation of  $\overline{\text{BR}}$  or the assertion of  $\overline{\text{TS}}$  indicating that the processor has started the snoop copyback transaction. During those waiting clocks, the processor will keep  $\overline{\text{BR}}$  asserted to keep the bus arbiter informed. Failure to grant the bus to the MPC7400/MPC7410 until it has asserted  $\overline{\text{TS}}$  for the snoop copyback will leave the snoop copyback transaction pending.

If another master is granted the bus and performs a global address transaction to the same address, the MPC7400/MPC7410 will assert  $\overline{\text{ARTRY}}$  and  $\overline{\text{SHD}}$  for that transaction and assert  $\overline{\text{BR}}$  during the window of opportunity. If another master is granted the bus and performs a global address transaction to a different address, the MPC7400/MPC7410 will assert  $\overline{\text{ARTRY}}$  without  $\overline{\text{SHD}}$  for that transaction and back off  $\overline{\text{BR}}$  during the window of opportunity for that address. At this point, the arbiter must use fair arbitration to ensure that the MPC7400/MPC7410 has an opportunity to empty to the address bus with a qualified  $\overline{\text{BG}}$ .

### 1.11.3 MPC7450

If the MPC7450 has a snoop copyback to perform and can perform it, the MPC7450 asserts  $\overline{BR}$  in the window of opportunity. However, in some cases, the processor may not be able to accept a qualified bus grant until several clocks after the window of opportunity. Therefore, to perform a snoop copyback, if the system bus arbiter asserts  $\overline{BG}$  during the window of opportunity, it must keep  $\overline{BG}$  asserted until it recognizes the negation of  $\overline{BR}$  or the assertion of  $\overline{TS}$  indicating that the processor has started the snoop copyback transaction. During those waiting clocks, the processor keeps  $\overline{BR}$  asserted to keep the bus arbiter informed. Depending on the system operation, failure to do this may cause the processor to miss the window of opportunity and possibly block its internal snoop copyback queue when a new snoop transaction is captured. At this point the arbiter must use fair arbitration to ensure that the snoop queue has an opportunity to empty to the address bus with a qualified  $\overline{BG}$ .

## 1.12 $\overline{DBB}$ and Data Bus Arbitration

### 1.12.1 MPC750

The MPC750 defines the  $\overline{DBB}$  pin as both input and output.

### 1.12.2 MPC7400/MPC7410

The MPC7400/MPC7410 defines the  $\overline{DBB}$  pin as output only and also generates an internal *dbb* that tracks its own outstanding transactions. The processor relies on the system arbiter to provide grants for the data bus only when the bus is available and the grant may be accepted. Note that this may have implications if the system arbiter is parking the data bus and counting on  $\overline{DBB}$  as an input to perform arbitration.

### 1.12.3 MPC7450

The MPC7450 does not include an  $\overline{DBB}$  pin. The processor generates an internal *dbb* that tracks its own outstanding transactions. The processor relies on the system arbiter to provide grants for the data bus only when the bus is available and the grant may be accepted. Note that this may have implications if the system arbiter is parking the data bus and counting on  $\overline{DBB}$  as an input to perform arbitration.

## 1.13 Data Bus Disable

### 1.13.1 MPC750

The MPC750 supports a  $\overline{DBDIS}$  input. When asserted during a write transaction, this input indicates the processor must release the data bus and the data bus parity to high impedance in the following cycle. The data tenure remains active,  $\overline{DBB}$  remains asserted, and the transfer termination signals are still monitored by the MPC750.  $\overline{DBDIS}$  is ignored during read transactions.

### 1.13.2 MPC7400/MPC7410 and MPC7450

The MPC7400/MPC7410 and the MPC7450 do not support data bus disable.

## 1.14 $\overline{\text{DRTRY}}$ Mode

### 1.14.1 MPC750

The MPC750 supports the  $\overline{\text{DRTRY}}$  mode of operation described in the 60x Bus Specification.

### 1.14.2 MPC7400/MPC7410 and MPC7450

The MPC7400/MPC7410 and the MPC7450 do not support the  $\overline{\text{DRTRY}}$  mode of operation described in the 60x Bus Specification. This has implications for the earliest assertion of  $\overline{\text{TA}}$  as described in the following section.

## 1.15 Earliest Assertion of $\overline{\text{TA}}$

### 1.15.1 MPC750

If an MPC750 system is in  $\overline{\text{DRTRY}}$  mode, system logic must ensure the first (or only) assertion of  $\overline{\text{TA}}$  for a data transfer does not occur sooner than the cycle before the snoop response window (one cycle after  $\overline{\text{TS}}$ ). If an MPC750 system is in no- $\overline{\text{DRTRY}}$  mode, system logic must ensure the first (or only) assertion of  $\overline{\text{TA}}$  for a data transfer does not occur sooner than the first cycle of the snoop response window (two cycles after  $\overline{\text{TS}}$ ). This guarantees a relationship between  $\overline{\text{TA}}$  and  $\overline{\text{ARTRY}}$  so that, in case of an address retry, the BIU discards the data in the chip before it can be forwarded to the cache and load/store unit. Typically, the external memory controller also detects the  $\overline{\text{ARTRY}}$  address tenure and aborts the read or write operation in progress. If this  $\overline{\text{TA}}/\overline{\text{ARTRY}}$  relationship is not met, the MPC750 may enter an undefined state.

### 1.15.2 MPC7400/MPC7410

In an MPC7400/MPC7410 system, the system chipset logic must ensure the first (or only) assertion of  $\overline{\text{TA}}$  for a data transfer does not occur sooner than the first cycle of the snoop response window (two cycles after  $\overline{\text{TS}}$ ). This guarantees the relationship between  $\overline{\text{TA}}$  and  $\overline{\text{ARTRY}}$  so that, in case of an address retry, the BIU discards the data before it can be forwarded internally to the cache and load/store unit. Typically, the external memory controller also detects the  $\overline{\text{ARTRY}}$  address tenure and aborts the read or write operation in progress. If this  $\overline{\text{TA}}/\overline{\text{ARTRY}}$  relationship is not met, the MPC7400/MPC7410 may enter an undefined state.

### 1.15.3 MPC7450

In an MPC7450 system, the system chipset logic must ensure the first (or only) assertion of  $\overline{\text{TA}}$  for a data transfer does not occur sooner than the last cycle of the snoop response window (cycle after  $\overline{\text{AACK}}$ ). This guarantees the relationship between  $\overline{\text{TA}}$  and  $\overline{\text{ARTRY}}$  so that, in case of an address retry, the BIU discards the data before it can be forwarded internally to the cache and load/store unit. Typically, the external memory controller also detects the  $\overline{\text{ARTRY}}$  address tenure and aborts the read or write operation in progress. If this  $\overline{\text{TA}}/\overline{\text{ARTRY}}$  relationship is not met, the MPC7450 may enter an undefined state.

Note that when  $\overline{\text{AACK}}$  is not delayed, the snoop response window is a single cycle, and the second cycle after  $\overline{\text{TS}}$  and the cycle after  $\overline{\text{AACK}}$  are coincident.



## 1.16 $\overline{\text{TEA}}$ and $\overline{\text{ARTRY}}$

### 1.17 MPC750

The MPC750 gives  $\overline{\text{TEA}}$  precedence over  $\overline{\text{ARTRY}}$  when both occur simultaneously. This means that the transaction will be terminated immediately.

#### 1.17.1 MPC7400/MPC7410 and MPC7450

The MPC7400/MPC7410 and the MPC7450 give  $\overline{\text{ARTRY}}$  precedence over  $\overline{\text{TEA}}$  when both occur simultaneously. This means that the transaction will be repeated until the  $\overline{\text{ARTRY}}$  condition is resolved.

## 1.18 Outstanding Data Tenures

### 1.18.1 MPC750

The MPC750 can support up to two outstanding data tenures. These can include any combination of one instruction fetch, one load/store operation, two castouts, or one snoop push.

#### 1.18.2 MPC7400/MPC7410

The MPC7400/MPC7410 can support up to six outstanding data tenures before receiving a qualified  $\overline{\text{DBG}}$  for any transaction. Note that once a qualified  $\overline{\text{DBG}}$  has been received, one more transaction can be queued. Outstanding tenures can include any combination of load/store operations, castouts, or snoop pushes but are limited to a single instruction fetch.

#### 1.18.3 MPC7450

The MPC7450 can support up to sixteen outstanding data tenures before receiving a qualified  $\overline{\text{DBG}}$  for any transaction. Note that once a qualified  $\overline{\text{DBG}}$  has been received, one more transaction can be queued. Fifteen of the outstanding tenures can include any combination of two instruction fetches, five loads, one store, nine castouts/single-beat stores, or ten snoop pushes. The final data tenure is always reserved for snoop pushes.

## 1.19 $\overline{\text{DBWO}}$ and Data Transaction Reordering

### 1.19.1 MPC750 and MPC7400/MPC7410

The MPC750 and the MPC7400/MPC7410 support a limited data reordering mechanism using the  $\overline{\text{DBWO}}$  pin. When asserted along with a qualified data bus grant,  $\overline{\text{DBWO}}$  will indicate that the processor should run the data transaction for the oldest outstanding write operation. If no outstanding write transaction is pending, the processor defaults to the oldest outstanding transaction.

#### 1.19.2 MPC7450

The MPC7450 does not support data transaction reordering using  $\overline{\text{DBWO}}$ . In addition, DTI[0-3] must always be driven negated in 60x mode.

## 1.20 MEI Coherency

### 1.20.1 MPC750

The MPC750 supports only a three-state coherency protocol which marks cache lines as modified, exclusive, or invalid. No shared state is available. Table 3 provides transaction types for basic operations. Implications for how reservations are maintained due to the lack of a shared state and a  $\overline{\text{SHD}}$  pin are described in Section 1.22, “Reservation Handling.” The MPC750 also uses the  $\overline{\text{TBST}}$  pin as an input to identify single-beat cache-inhibited or write-through transactions and treats them as read with no intent to cache.

### 1.20.2 MPC7400/MPC7410

The MPC7400/MPC7410 can be configured to support a three-state coherency protocol by setting  $\text{MSSCR0}[0] = \text{b}'0'$ . This disables handling of the shared state. Table 3 provides transaction types for basic operations. Note that the MPC7400/MPC7410 does not issue a RWITM transaction type for cacheable-load or instruction-fetch requests. In a multi-master environment, the  $\overline{\text{SHD}}$  pin must be enabled and sampled for correct operation by setting  $\text{MSSCR0}[1] = \text{b}'1'$ .

### 1.20.3 MPC7450

The MPC7450 does not support a three-state coherency protocol.

## 1.21 MESI Coherency

Table 3 provides transaction types for basic operations. Also note the following:

- The MPC750 does not support a four-state coherency protocol.
- The MPC7400/MPC7410 can be configured to support a four-state coherency protocol by setting  $\text{MSSCR0}[0] = \text{b}'1'$ .
- The MPC7450 only supports a four-state coherency protocol.

**Table 3. Basic Operation Transaction Types**

Operation	MPC750	MPC7400/MPC7410	MPC7450
Cacheable instruction fetch	RWITM	READ	READ
Cache inhibited instruction fetch	READ (8-byte)	READ (8-byte)	READ (32-byte burst)
Cacheable load	RWITM	READ	READ
Cache inhibited load	READ	READ	READ
Cache inhibited altivec load	not supported	READ	alignment interrupt
Cacheable lwarx	RWITM atomic	READ atomic	READ atomic
Cache Inhibited lwarx	READ atomic	READ atomic	data storage interrupt
Cacheable Store	RWITM	RWITM	RWITM
Cache Inhibited or Write Through Store	Wr w/ Flush	Wr w/ Flush	Wr w/ Flush
Cache Inhibited or Write Through AltiVec Store	not supported	Wr w/ Flush	alignment interrupt
Cacheable stwcx.	RWITM atomic	RWITM atomic	RWITM atomic

Table 3. Basic Operation Transaction Types (Continued)

Operation	MPC750	MPC7400/MPC7410	MPC7450
Cache Inhibited stwcx.	Wr w/ Flush atomic	Wr w/ Flush atomic	data storage interrupt
Write Through stwcx.	Wr w/ Flush atomic	data storage interrupt	data storage interrupt

## 1.22 Reservation Handling

### 1.22.1 MPC750

The MPC750 clears the reservation in the scenarios shown in Table 4.

Table 4. MPC750 Reservation Clear Scenarios

Transaction Type	Address	GBL
Wr w/ Kill	match	any
Wr w/ Flush	match	asserted
Wr w/ Flush atomic	any	asserted
Kill	match	asserted

### 1.22.2 MPC7400/7410 and MPC7450

The MPC7400/MPC7410 and the MPC7450 clear the reservation in the scenarios shown in Table 5.

Table 5. MPC7400/MPC7410/MPC7450 Reservation Clear Scenarios

Transaction Type	Address	GBL
Wr w/ Kill	match	asserted
Wr w/ Flush	match	asserted
Wr w/ Flush atomic	match	asserted
Kill	match	asserted
RWITM	match	asserted
RWITM atomic	match	asserted

## 1.23 TLBI Synchronize

### 1.23.1 MPC750

The MPC750 supports a  $\overline{\text{TLBISYNC}}$  input. This input causes the MPC750 to halt after executing the tlbsync instruction.


### 1.23.2 MPC7400/7410 and MPC7450

The MPC7400/MPC7410 and the MPC7450 do not support the  $\overline{\text{TLBISYNC}}$  input.

DigitalDNA is a trademark of Motorola, Inc.

The PowerPC name, the PowerPC logotype, and PowerPC 603e are trademarks of International Business Machines Corporation used by Motorola under license from International Business Machines Corporation.

Information in this document is provided solely to enable system and software implementers to use PowerPC microprocessors. There are no express or implied copyright licenses granted hereunder to design or fabricate PowerPC integrated circuits or integrated circuits based on the information in this document.

Motorola reserves the right to make changes without further notice to any products herein. Motorola makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Motorola assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. "Typical" parameters which may be provided in Motorola data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters, including "Typicals" must be validated for each customer application by customer's technical experts. Motorola does not convey any license under its patent rights nor the rights of others. Motorola products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Motorola product could create a situation where personal injury or death may occur. Should Buyer purchase or use Motorola products for any such unintended or unauthorized application, Buyer shall indemnify and hold Motorola and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Motorola was negligent regarding the design or manufacture of the part. Motorola and  are registered trademarks of Motorola, Inc. Motorola, Inc. is an Equal Opportunity/Affirmative Action Employer.

**How to reach us:**

**USA/EUROPE/Locations Not Listed:** Motorola Literature Distribution; P.O. Box 5405, Denver, Colorado 80217. 1-303-675-2140 or 1-800-441-2447

**JAPAN:** Motorola Japan Ltd.; SPS, Technical Information Center, 3-20-1, Minami-Azabu, Minato-ku, Tokyo 106-8573 Japan. 81-3-3440-3569

**ASIA/PACIFIC:** Motorola Semiconductors H.K. Ltd.; Silicon Harbour Centre, 2 Dai King Street, Tai Po Industrial Estate, Tai Po, N.T., Hong Kong. 852-26668334

**Technical Information Center:** 1-800-521-6274

**HOME PAGE:** <http://www.motorola.com/semiconductors>

**Document Comments:** FAX (512) 933-2625, Attn: RISC Applications Engineering

**World Wide Web Addresses:** <http://www.motorola.com/PowerPC>  
<http://www.motorola.com/NetComm>  
<http://www.motorola.com/ColdFire>



**MOTOROLA**

AN2097/D