

PowerPC™ 60x BUS Implementation Differences

by *Freescale Semiconductor, Inc.*
Austin, TX

This document details differences in implementation of the 60x bus on the MPC750, MPC7400, MPC7410, and MPC7450 PowerPC™ 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. Therefore, the data tenure can be decoupled from the address tenure of a transaction, with a wide range of system bus implementations, including the following:

- Non-pipelined bus operation
- Pipelined bus operation
- Split transaction operation

1 I/O Voltage Level

Differences among the MPC750, the MPC7400/MPC7410, and the MPC7450 with respect to I/O voltage level are as follows:

1.1 MPC750

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

Contents

1	I/O Voltage Level	1
2	Signal Differences	2
3	Extended Addressing	3
4	Bus Request	3
5	ABB and Address Bus Arbitration	4
6	Address Transfer Attributes	4
7	Instruction Fetch Differentiation	5
8	Address Tenure Termination	5
9	Enveloped Transactions	5
10	Address Retry	5
11	Window of Opportunity	6
12	DBB and Data Bus Arbitration	7
13	Data Bus Disable	7
14	DRTRY Mode	7
15	Earliest Assertion of TA	7
16	TEA and ARTRY	8
17	Outstanding Data Tenures	8
18	DBWO and Data Transaction Reordering	9
19	MEI Coherency	9
20	MESI Coherency	9
21	Reservation Handling	10
22	TLBI Synchronize	11
23	Revision History	11

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 ¹	Connection ²	MPC7400	MPC7410	MPC7450
BVSEL	HRESET	2.5 V I/O	2.5V I/O	2.5V I/O
	GND	1.8 V I/O	1.8V I/O	1.8V I/O
	OVDD	3.3 V I/O (ipu) ³	3.3V I/O (ipu) ^{4, 5}	2.5V I/O (ipu) ³
	$\overline{\text{HRESET}}$	3.3 V I/O	3.3V I/O ⁶	not supported

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

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

³ ipu = internal pullup; default if unconnected

⁴ MPC7410RXnnnLE (Rev 1.4) and later only. Previous revisions do not support 3.3 V OVDD; having BVSEL = OVDD selects the 2.5 V threshold.

⁵ ipu = internal pullup; on MPC7410RXnnnLE (Rev 1.4) and later, default if unconnected is 3.3 V threshold. On MPC7410RXnnnLD (Rev 1.3) and earlier, default if unconnected is 2.5 V threshold.

⁶ MPC7410RXnnnLE (Rev 1.4) and later only. Previous revisions do not support BVSEL = $\overline{\text{HRESET}}$.

2 Signal Differences

Signal differences are described in [Table](#) .

Table 2. Signal Differences

Description	MPC750	MPC7400/MPC7410	MPC7450
60x bus mode	default	$\overline{\text{EMODE}} = \text{VDD}$	$\overline{\text{BMODE0}} = \text{VDD}$ $\overline{\text{BMODE1}} = \text{VDD}$
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)	$\overline{\text{ABB}}$ (output only)	not supported
Transaction burst	$\overline{\text{TBST}}$ (input/output)	$\overline{\text{TBST}}$ (output only)	$\overline{\text{TBST}}$ (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	$\overline{\text{WT}}$ (output Only)	$\overline{\text{WT}}$ (input/output) (weak pull-up recommended)	$\overline{\text{WT}}$ (output only)
Data bus busy	$\overline{\text{DBB}}$ (Input/output) (weak pull-up required)	$\overline{\text{DBB}}$ (output only)	not supported
Data bus write only	$\overline{\text{DBWO}}$	$\overline{\text{DBWO}}$	not supported
Data bus disable	$\overline{\text{DBDIS}}$	not supported	not supported

Table 2. Signal Differences (continued)

Description	MPC750	MPC7400/MPC7410	MPC7450
Data retry	$\overline{\text{DRTRY}}$	not supported	not supported
Reservation	$\overline{\text{RSRV}}$	$\overline{\text{RSRV}}$	not supported
TLBI synchronize	$\overline{\text{TLBISYNC}}$	not supported	not supported

3 Extended Addressing

3.1 MPC750 and MPC7400/MPC7410

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

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 drives 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 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 enabled only when translation is enabled and the reset vector is mapped in real mode.

4 Bus Request

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

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

5 \overline{ABB} and Address Bus Arbitration

Differences among the MPC750, the MPC7400/MPC7410, and the MPC7450 with respect to address bus arbitration are as follows:

- MPC750. Defines the \overline{ABB} pin as both input and output.
- MPC7400/MPC7410. Defines the \overline{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{ABB} as an input to perform arbitration.
- MPC7450. Does not include an \overline{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{ABB} as an input to perform arbitration.

6 Address Transfer Attributes

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

- \overline{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{WT} signal during read transactions to indicate the transaction is an instruction fetch ($\overline{WT} = 1$) or not an instruction fetch ($\overline{WT} = 0$).
- \overline{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{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{GBL} bit is asserted for **sync**, **tlbsync**, **tlbie**, and **eiemo** instructions. The \overline{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{GBL} for Kill transactions due to **dcba** and **dcbz** instructions that reach the 60x bus.

7 Instruction Fetch Differentiation

Differences among the MPC750, the MPC7400/MPC7410, and the MPC7450 with respect to instruction fetching are as follows:

- MPC750 and 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 ($\overline{WT} = 0$).
- 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 changes from Read (TT = 01010) to Read Atomic (TT = 11010). I-side reads continue to be identified as Read (TT = 01010).

8 Address Tenure Termination

Differences among the MPC750, the MPC7400/MPC7410, and the MPC7450 with respect to address tenure termination are as follows:

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

9 Enveloped Transactions

Differences among the MPC750, the MPC7400/MPC7410, and the MPC7450 with respect to enveloped transactions are as follows:

- MPC750 and MPC7400/MPC7410. Support enveloped transactions where \overline{AACK} is delayed long enough that the entire data tenure is contained within the address tenure.
- MPC7450. Does not support enveloped transactions.

10 Address Retry

In the MPC7450, a retry response (assertion of the \overline{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{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 $\overline{TLBSYNC}$). 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{ARTRY} may cause a deadlock condition.

11 Window of Opportunity

Differences among the MPC750, the MPC7400/MPC7410, and the MPC7450 with respect to window of opportunity are detailed in the following subsections.

11.1 MPC750

If the MPC750 has a snoop copyback that it can perform, it 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} to the processor during the cycle after the window of opportunity (such as the third cycle after \overline{AACK}), it must keep \overline{BG} asserted until it recognizes the assertion of \overline{TS} indicating that the processor has started the snoop copyback transaction. The MPC750 is not guaranteed to hold \overline{BR} asserted if \overline{BG} is parked.

11.2 MPC7400/MPC7410

If the MPC7400/MPC7410 has a snoop copyback that it can perform, it 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 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{BG} to the processor during the cycle after the window of opportunity (such as the third cycle after \overline{AACK}), it should 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 holds \overline{BR} asserted to keep the bus arbiter informed. Failure to grant the bus to the MPC7400/MPC7410 until it asserts \overline{TS} for the snoop copyback leaves 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 asserts \overline{ARTRY} and \overline{SHD} for that transaction and asserts \overline{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 asserts \overline{ARTRY} without \overline{SHD} for that transaction and backs off \overline{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{BG} .

11.3 MPC7450

If the MPC7450 has a snoop copyback that it can perform, it 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 holds \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 $\overline{\text{BG}}$.

12 $\overline{\text{DBB}}$ and Data Bus Arbitration

Differences among the MPC750, the MPC7400/MPC7410, and the MPC7450 with respect to data bus arbitration are as follows:

- MPC750. Defines the $\overline{\text{DBB}}$ pin as both input and output.
- MPC7400/MPC7410. Defines the $\overline{\text{DBB}}$ 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 can be accepted. Note that this may have implications if the system arbiter is parking the data bus and counting on $\overline{\text{DBB}}$ as an input to perform arbitration.
- MPC7450. Does not include a $\overline{\text{DBB}}$ 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 can be accepted. Note that this may have implications if the system arbiter is parking the data bus and counting on $\overline{\text{DBB}}$ as an input to perform arbitration.

13 Data Bus Disable

Differences among the MPC750, the MPC7400/MPC7410, and the MPC7450 with respect to data bus disable are as follows:

- MPC750. Supports a $\overline{\text{DBDIS}}$ 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, $\overline{\text{DBB}}$ remains asserted, and the transfer termination signals are still monitored by the MPC750. $\overline{\text{DBDIS}}$ is ignored during read transactions.
- MPC7400/MPC7410 and MPC7450. Do not support data bus disable.

14 $\overline{\text{DRTRY}}$ Mode

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

15 Earliest Assertion of $\overline{\text{TA}}$

Differences among the MPC750, the MPC7400/MPC7410, and the MPC7450 with respect to earliest assertion of $\overline{\text{TA}}$ are as follows:

- MPC750. In $\overline{\text{DRTRY}}$ mode, system logic must ensure that 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{TS}). If an MPC750 system is in no- \overline{DRTRY} mode, system logic must ensure that the first (or only) assertion of \overline{TA} for a data transfer does not occur sooner than the first cycle of the snoop response window (two cycles after \overline{TS}). This guarantees a relationship between \overline{TA} and \overline{ARTRY} so that, for 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{ARTRY} address tenure and aborts the read or write operation in progress. If this $\overline{TA}/\overline{ARTRY}$ relationship is not met, the MPC750 may enter an undefined state.

- MPC7400/MPC7410. The system chipset logic must ensure that the first (or only) assertion of \overline{TA} for a data transfer does not occur sooner than the first cycle of the snoop response window (two cycles after \overline{TS}). This guarantees the relationship between \overline{TA} and \overline{ARTRY} so that, for 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{ARTRY} address tenure and aborts the read or write operation in progress. If this $\overline{TA}/\overline{ARTRY}$ relationship is not met, the MPC7400/MPC7410 may enter an undefined state.
- MPC7450. The system chipset logic must ensure that the first (or only) assertion of \overline{TA} for a data transfer does not occur sooner than the last cycle of the snoop response window (cycle after \overline{AACK}). This guarantees the relationship between \overline{TA} and \overline{ARTRY} so that, for 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{ARTRY} address tenure and aborts the read or write operation in progress. If this $\overline{TA}/\overline{ARTRY}$ relationship is not met, the MPC7450 may enter an undefined state. Note that when \overline{AACK} is not delayed, the snoop response window is a single cycle, and the second cycle after \overline{TS} and the cycle after \overline{AACK} are coincident.

16 \overline{TEA} and \overline{ARTRY}

The MPC750 gives \overline{TEA} precedence over \overline{ARTRY} when both occur simultaneously so that the transaction terminates immediately. The MPC7400/MPC7410 and the MPC7450 give \overline{ARTRY} precedence over \overline{TEA} when both occur simultaneously so that the transaction repeats until the \overline{ARTRY} condition is resolved.

17 Outstanding Data Tenures

Differences among the MPC750, the MPC7400/MPC7410, and the MPC7450 with respect to outstanding data tenures are as follows:

- MPC750. Can support up to two outstanding data tenures, including any combination of one instruction fetch, one load/store operation, two castouts, or one snoop push.
- MPC7400/MPC7410. Can support up to six outstanding data tenures before receiving a qualified \overline{DBG} for any transaction. Note that after a qualified \overline{DBG} is 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.
- MPC7450. Can support up to sixteen outstanding data tenures before receiving a qualified \overline{DBG} for any transaction. Note that after a qualified \overline{DBG} is 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.

18 DBWO and Data Transaction Reordering

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, $\overline{\text{DBWO}}$ indicates 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. 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.

19 MEI Coherency

Differences among the MPC750, the MPC7400/MPC7410, and the MPC7450 with respect to MEI coherency are as follows:

- MPC750. Supports only a three-state coherency protocol that 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 21, “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.
- MPC7400/MPC7410. Can be configured to support a three-state coherency protocol by setting MSSCR0[0] = 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 MSSCR0[1] = b'1'.
- MPC7450. Does not support a three-state coherency protocol.

20 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 MSSCR0[0] = b'1'.
- The MPC7450 supports only 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

Table 3. Basic Operation Transaction Types (continued)

Operation	MPC750	MPC7400/MPC7410	MPC7450
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
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

21 Reservation Handling

21.1 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

21.2 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

22 TLBI Synchronize

The MPC750 supports a $\overline{\text{TLBISYNC}}$ input that causes the MPC750 to halt after executing the **tlbsync** instruction. The MPC7400/MPC7410 and the MPC7450 do not support the $\overline{\text{TLBISYNC}}$ input.

23 Revision History

Table 6. Revision History

Revision Number	Changes
0.0	Initial release
0.1	Nontechnical reformatting
0.2	Updated Table 1 for 3.3V OVDD support on MPC7410 Rev 1.4 and later.
0.3	Updated Table 1 with additional footnote changes.
1	Nontechnical reformatting and rebranding the document for Freescale.

How to Reach Us:

Home Page:

www.freescale.com

email:

support@freescale.com

USA/Europe or Locations Not Listed:

Freescale Semiconductor
Technical Information Center, CH370
1300 N. Alma School Road
Chandler, Arizona 85224
1-800-521-6274
480-768-2130
support@freescale.com

Europe, Middle East, and Africa:

Freescale Halbleiter Deutschland GmbH
Technical Information Center
Schatzbogen 7
81829 Muenchen, Germany
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)
support@freescale.com

Japan:

Freescale Semiconductor Japan Ltd.
Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku
Tokyo 153-0064, Japan
0120 191014
+81 3 5437 9125
support.japan@freescale.com

Asia/Pacific:

Freescale Semiconductor Hong Kong Ltd.
Technical Information Center
2 Dai King Street
Tai Po Industrial Estate,
Tai Po, N.T., Hong Kong
+800 2666 8080
support.asia@freescale.com

For Literature Requests Only:

Freescale Semiconductor
Literature Distribution Center
P.O. Box 5405
Denver, Colorado 80217
1-800-441-2447
303-675-2140
Fax: 303-675-2150
LDCForFreescaleSemiconductor@hibbertgroup.com

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

Freescale Semiconductor reserves the right to make changes without further notice to any products herein. Freescale Semiconductor makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale Semiconductor 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 Freescale Semiconductor 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. Freescale Semiconductor does not convey any license under its patent rights nor the rights of others. Freescale Semiconductor 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 Freescale Semiconductor product could create a situation where personal injury or death may occur. Should Buyer purchase or use Freescale Semiconductor products for any such unintended or unauthorized application, Buyer shall indemnify and hold Freescale Semiconductor 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 Freescale Semiconductor was negligent regarding the design or manufacture of the part.

Freescale™ and the Freescale logo are trademarks of Freescale Semiconductor, Inc. The Power Architecture and Power.org word marks and the Power and Power.org logos and related marks are trademarks and service marks licensed by Power.org. All other product or service names are the property of their respective owners.

© Freescale Semiconductor, Inc., 2001, 2006.

Document Number: AN2097

Rev. 1
10/2006

