Application Note

AN1795/D
Rev. 1.1, 6/2003

Designing PowerPC™ MPC7400 Systems

This application note describes differences between the 60x bus and the native bus mode of the MPC7400 processor (a new bus interface that is derived from the 60x bus). It also briefly describes the 360-pin MPC7400 processor that is pin-compatible with the MPC750 microprocessor. This document assumes that the reader has a working knowledge of the MPC750 microprocessor and the 60x bus protocol. The MPC7400 is a PowerPC™ microprocessor.

The MPC7400 provides a mode switch (via the $\overline{\text{EMODE}}$ signal) that allows either the MPC7400 native bus or 60x bus operation. The MPC7400 native bus mode includes several additional features that allow it to provide higher memory bandwidth than the 60x bus. The following list summarizes 60x bus interface features:

- 32-bit address bus (plus 4 bits of odd parity)
- 64-bit data bus (plus 8 bits of odd parity)
- Support for a 3-state MEI coherency protocol similar to the MPC750
- Support for a 4-state MESI protocol similar to the MPC604 processors
- On-chip snooping to maintain data cache coherency for MP applications
- Support for address-only transfers
- Support for limited out-of-order transactions
- TTL compatible interface

In addition to the 60x bus features, to gain increased performance, the MPC7400 native bus mode has the following features:

- Increased address bus bandwidth by eliminating dead cycles under some circumstances
- Full data streaming for burst reads and writes
- Increased levels of address pipelining
- Support for full out-of-order transactions
- Support for data intervention in MP systems (5-state MERSI)
- Support for up to seven outstanding transactions (six pending plus one data tenure in progress).

1 MPC7400 Native Bus Mode Signals

The MPC7400 native bus mode protocol defines several new signals not present in the 60x bus protocol. Also, there are MPC7400 signals not supported by the MPC7400 native bus mode protocol. These signal differences are summarized in Table 1.

Table 1. Signal Summary

60x Bus Signals not in MPC7400 Native Bus Mode	60x Bus Signals Expanded for MPC7400 Native Bus Mode	New MPC7400 Native Bus Mode Signals
Address Bus Busy $\overline{\text{ABB}}$	Data Bus Write Only $\overline{\text{DBWO}}$	Address Monitor AMON
Data Bus Busy $\overline{\text{DBB}}$	Shared $\overline{\text{SHD}}$	Data Monitor DMON
Data Retry $\overline{\text{DRTRY}}$		Hit $\overline{\text{HIT}}$
Extended Transfer Protocol $\overline{\text{XATS}}$		Data Ready $\overline{\text{DRDY}}$
Transfer Code TC[0:1]		Enhanced Mode $\overline{\text{EMODE}}$
Cache Set Element CSE[0:1]		L2 Address L2A17,L2A18
Address Parity Error $\overline{\text{APE}}$		Check $\overline{\text{CHK}}$
Data Parity Error $\overline{\text{DPE}}$		Bus Voltage Select BVSEL
		L2 Voltage Select L2VSEL

The three types of signals in Table 1 (shown in the column headings) are discussed in the following three sections.

1.1 60x Bus Signals Not in the MPC7400 Native Bus Mode

Several signals defined in the 60x bus protocol are no longer required by the MPC7400 native bus mode protocol. Most of these signals are not implemented in the MPC7400, however, new signals provide similar functionality for compatibility reasons. These signals are identified and described below:

1.1.1 Address Bus Busy and Data Bus Busy ($\overline{\text{ABB}}$ and $\overline{\text{DBB}}$)

The MPC7400 does not use the $\overline{\text{ABB}}$ or $\overline{\text{DBB}}$ signals as inputs. The MPC7400 tracks its own outstanding transactions, and will rely on the system arbiter to provide grants for the address and data buses only when the bus is available and the grant may be accepted.

For compatibility with 60x system arbiters, the MPC7400 will generate $\overline{\mathrm{ABB}}$ and $\overline{\mathrm{DBB}}$ signals as outputs in the form of AMON and DMON respectively. This means that the system arbiter must not assume that a master knows that the bus is busy with a transaction for another master.

An MPC7400 native bus mode system arbiter must synthesize its own $\overline{\mathrm{ABB}}$ and $\overline{\mathrm{DBB}}$ signals internally because the processor is not required to generate them.

1.1.2 Data Retry ($\overline{\mathrm{DRTRY}}$)

The data retry input does not exist in the MPC7400 native bus mode specification, so the $\overline{\mathrm{DRTRY}}$ signal is not supported.

1.1.3 Extended Transfer Protocol ($\overline{\mathrm{XATS}}$)

Extended transfer protocol, used for accesses to direct-store segments, is not supported by the native bus mode of the MPC7400 processor interface.

1.1.4 Transfer Code (TC[0:1])

The transfer code signals have been removed from the MPC7400 native bus mode interface. The information provided by these signals in the 60x bus during read operations was code versus data. This information is now provided on the write-through ($\overline{\mathrm{WT}}$) signal during read operations.

1.1.5 Cache Set Element (CSE[0:1])

The cache set element signals have been removed because the MPC7400 does not support snoop-filtering devices. Note: Snoop filtering devices filter system coherency traffic.

1.1.6 Address Parity Error and Data Parity Error ($\overline{\mathrm{APE}}$ and $\overline{\mathrm{DPE}}$)

The address parity error and data parity error signals have been removed from the MPC7400.

1.2 60x Bus Signals Expanded for MPC7400 Native Bus Mode

The MPC7400 native bus mode support of full out-of-order transactions and increased address bus bandwidth is realized through the expanded definitions of $\overline{\mathrm{DBWO}}$ and $\overline{\mathrm{SHD}}$. The $\overline{\mathrm{DBWO}}$ signal was expanded to DTI for support of full out-of order transactions. The $\overline{\mathrm{SHD}}$ expansion to $\overline{\mathrm{SHD0}}$ and $\overline{\mathrm{SHD1}}$ allows for increased levels of address pipelining. Both of these expanded signals are discussed below.

1.2.1 Data Bus Write Only ($\overline{\mathrm{DBWO}}$) to Data Transfer Index (DTI[0–2])

The 60x bus transaction reordering scheme was implemented with the $\overline{\mathrm{DBWO}}$ signal. The MPC7400 native bus mode can be configured to support a generalized reordering scheme using the new 3-bit data transfer index (DTI) signal.

DTI is a signal from the system arbiter to the MPC7400 that supports reordered data tenures. This signal can be bused or point-to-point. It must be driven valid by the system arbiter on the cycle before a data bus grant ($\overline{\mathrm{DBG}}$). It is sampled each cycle by the MPC7400, and is qualified by the assertion of $\overline{\mathrm{DBG}}$ on the following cycle.

The data transfer index is a pointer into the MPC7400's queue of outstanding transactions, indicating which transaction is to be serviced by the subsequent data tenure. Note that this protocol is a generalization of the $\overline{\mathrm{DBWO}}$ protocol in which the assertion of $\overline{\mathrm{DBWO}}$ indicated that the first write operation in the queue was to be serviced. For example, DTI = '000' means that the oldest transaction is to be serviced, DTI = '001' means the second oldest transaction is to be serviced, etc., up to DTI = '101' meaning the 6th oldest transaction is to be serviced. Note that since the MPC7400 only supports six outstanding data transactions a maximum setting for DTI of '101' is allowed.

Data tenure reordering can be disabled by setting DTI[0-2] to b'000'. This will always select the oldest transaction in the outstanding transaction queue.

1.2.2 Shared ($\overline{\mathrm{SHD}}$) to Shared ($\overline{\mathrm{SHD0}}$, $\overline{\mathrm{SHD1}}$)

The MPC7400 native bus mode interface allows a given master to drive a new address tenure every other cycle, so the shared signal must be able to be driven every other cycle too. But, since it must be actively negated and might be driven by multiple masters at any given time, electrical requirements dictate that two versions of the $\overline{\mathrm{SHD}}$ signal be implemented. When signaling a snoop response of shared, the MPC7400 must assert $\overline{\mathrm{SHD0}}$ unless $\overline{\mathrm{SHD0}}$ was asserted in any of the 3 cycles prior to the snoop response window for the current transaction. In that case, the MPC7400 must assert $\overline{\mathrm{SHD1}}$. This way $\overline{\mathrm{SHD0}}$ or $\overline{\mathrm{SHD1}}$ can be three-stated, driven negated, then three-stated again before it will need to be reasserted. When the MPC7400 is a bus master, the MPC7400 must consider the snoop response to be shared if either $\overline{\mathrm{SHD0}}$ or $\overline{\mathrm{SHD1}}$ is asserted.

1.3 New MPC7400 Native Bus Mode Signals

The MPC7400 native bus mode's support for data intervention in microprocessor systems and full data streaming for burst reads and writes is realized through the addition of two new signals—$\overline{\mathrm{HIT}}$ and $\overline{\mathrm{DRDY}}$. Other new signals include support for enabling the MPC7400 native bus mode, larger L2 cache sizes, entering a diagnostics mode, and I/O voltage configuration. These new signals are discussed below.

1.3.1 Hit ($\overline{\mathrm{HIT}}$)

The $\overline{\mathrm{HIT}}$ signal is a point-to-point signal output from the processor or local bus slave to the system arbiter. This signal is a snoop response valid in the address retry ($\overline{\mathrm{ARTRY}}$) window (the cycle after an address acknowledge ($\overline{\mathrm{AACK}}$)) that indicates the MPC7400 will supply intervention data. That is, the MPC7400 has found the data in its cache that has been requested by another master's bus transaction. Instead of asserting $\overline{\mathrm{ARTRY}}$ and flushing the data to memory, the MPC7400 asserts $\overline{\mathrm{HIT}}$ to indicate that it can supply the data directly to the other master.

Like other snoop responses, $\overline{\mathrm{HIT}}$ can be driven as soon as the second cycle after $\overline{\mathrm{TS}}$. If $\overline{\mathrm{AACK}}$ is delayed, the response needs to be held until the cycle after $\overline{\mathrm{AACK}}$.

The MPC7400 implements the optional protocol of native bus mode of the MPC7400 processor to communicate to the system whether or not the intervention data needs to be forwarded to memory. If the MPC7400 intervenes with shared or exclusive data rather than modified data, it can indicate this to the processor by asserting the $\overline{\mathrm{HIT}}$ signal for a second cycle after $\overline{\mathrm{AACK}}$. If the data is modified, the MPC7400 negates $\overline{\mathrm{HIT}}$ on the second cycle after $\overline{\mathrm{AACK}}$, and the system will "snarf" the data and forward it to memory. (Snarfing is when a device provides data specifically for another device and a third device reads the data also.)

Note that it is possible for the MPC7400 to assert both $\overline{\text{ARTRY}}$ and $\overline{\text{HIT}}$ simultaneously for the same snoop response. When simultaneously asserted, $\overline{\text{ARTRY}}$ supersedes $\overline{\text{HIT}}$.

1.3.2 Data Ready ($\overline{\text{DRDY}}$)

The $\overline{\text{DRDY}}$ signal is a point-to-point signal from the MPC7400 to the system arbiter. It is a data bus request indicating to the arbiter that data for an outstanding intervention transaction previously signaled with a $\overline{\text{HIT}}$ is ready. The arbiter will respond by granting the data bus to all devices participating in the transaction.

1.3.3 Enhanced Mode ($\overline{\text{EMODE}}$)

The assertion of the $\overline{\text{EMODE}}$ signal at hard reset's ($\overline{\text{HRESET}}$) negation will select the MPC7400 native bus mode, otherwise 60x bus mode is selected. After the negation of $\overline{\text{HRESET}}$, if $\overline{\text{EMODE}}$ is asserted it selects address bus drive mode [1] (only for native bus mode of the MPC7400 processor mode). $\overline{\text{EMODE}}$ negation selects normal address bus drive mode.

1.3.4 Additional L2 Address Signals (L2A17, L2A18)

The L2 cache interface of the MPC7400 provides an 18-bit address bus that controls a maximum of 2 Mbytes of external SRAM memory. The L2A17 address pin allows for 2 Mbytes SRAM addressing. In the MPC7400, the L2A18 address pin (which could allow 4 Mbytes addressable L2 cache) is not supported.

1.3.5 Check ($\overline{\text{CHK}}$)

This signal is for the MPC7400 testing purposes and supports three modes of operation:

- A post power-on-reset internal memory test and initialization can be selected by tying $\overline{\text{CHK}}$ to $\overline{\text{HRESET}}$.
- Tying $\overline{\text{CHK}}$ low enables engineering diagnostic mode.
- For normal operation tie $\overline{\text{CHK}}$ high.

1.3.6 Bus Voltage Select (BVSEL)/L2 Voltage Select (L2VSEL)

The MPC7400 provides several I/O voltages to support both compatibility with existing systems and migration to future systems. The MPC7400's core voltage must always be provided at 1.8V. Voltage to the L2 and processor I/O pins is provided through a separate sets of supply pins according to the configurations shown in Table 2 The voltage configuration for each bus is selected by sampling the state of the voltage select pins before and after the negation of $\overline{\text{HRESET}}$.

Table 2. I/O Voltages

BVSEL Pin	L2VSEL Pin	Processor Interface Voltage (V)	L2 Interface Voltage (V)
0	0	1.8	1.8
0	1	1.8	3.3
0	$\overline{\text{HRESET}}$	1.8	2.5

[1]Address Bus Drive mode causes the MPC7400 to drive the address bus whenever $\overline{\text{BG}}$ is asserted independent of whether the MPC7400 has a bus transaction to run or not.

Table 2. I/O Voltages

BVSEL Pin	L2VSEL Pin	Processor Interface Voltage (V)	L2 Interface Voltage (V)
1	0	3.3	1.8
1	1	3.3	3.3
1	$\overline{\text{HRESET}}$	3.3	2.5

1.3.7 Bus Monitor Signals (AMON, DMON)

The AMON and DMON signals are outputs from the MPC7400. AMON replaces the functionality of the 60x bus's $\overline{\text{ABB}}$ signal. Likewise, DMON replaces the $\overline{\text{DBB}}$ signal of the 60x bus.

1.4 MPC7400 Pin Locations

Table 3 summarizes the pin differences between the MPC7400 processor and the MPC750.

Table 3. New MPC7400 Signal Locations

Pin	MPC750 Signal	MPC7400 Signal
D01 H06 G01	$\overline{\text{DBWO}}$ $\overline{\text{DRTRY}}$ $\overline{\text{DBDIS}}$	DTI[0] DTI[1] DTI[2]
A03	TLBISYNC	$\overline{\text{EMODE}}$
B03	No-connect	$\overline{\text{SHD0}}$
B04	No-connect	$\overline{\text{SHD1}}$
K09	No-connect	$\overline{\text{DRDY}}$
B05	No-connect	$\overline{\text{HIT}}$
K19	No-connect	L2A17
W19	No-connect	L2ASPARE
W01	No-connect	BVSEL [1]
A19	No-connect	L2VSEL[1]
K11	No-connect	CHK[1]

[1] BVSEL, L2VSEL, and CHK signals are either connected to Vdd, Vss, or $\overline{\text{HRESET}}$, depending on configuration.

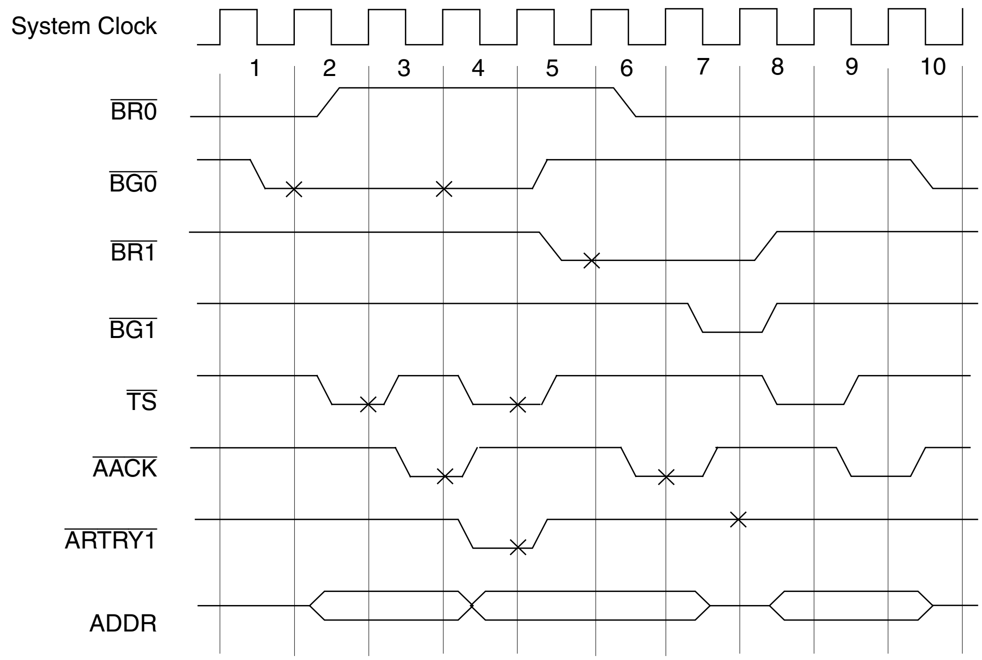
2 Transaction Timing Changes

The following sections describe the transaction timing changes between the 60x bus mode and the MPC7400 native bus mode.

2.1 Address Tenure Timing Changes

The 60x bus requires an idle cycle between address tenures. The MPC7400 native bus removes this restriction for address tenures by allowing back-to-back address tenures to be initiated by the same bus master. The master must ensure that it is still the bus master by checking the bus grant ($\overline{\text{BG}}$) signal on the cycle in which the end of the first transaction occurs (the cycle when the address acknowledge ($\overline{\text{AACK}}$) is sent to the master).

Because the native bus mode of the MPC7400 processor protocol allows new address tenures to begin without a dead cycle in between, a new tenure can begin (via the transfer start ($\overline{\text{TS}}$) signal) on the same cycle that another device asserts the address retry ($\overline{\text{ARTRY}}$) signal for the tenure that had just ended. If this happens, the system and all bus devices must recognize that the second $\overline{\text{TS}}$ is implicitly retried as well. Both behaviors (back-to-back address tenures and $\overline{\text{ARTRY}}$ assertion) are shown in Table 1.



Cycle 1: The master has requested the bus and receives a qualified bus grant

Cycle 2: The master begins the address tenure by driving TS and the address

Cycle 3: The system responds with AACK, ending the address tenure. The master receives another (parked) address bus grant.

Cycle 4: The master begins a new address tenure by driving TS and a new address. Some snooping device, however, asserts ARTRY for the first transaction. Bus grant remains parked to processor 0.

Cycle 5: The system delays AACK for the second transaction for some reason. BG0 is negated to allow the retrying processor to request the bus. Processor 1 takes advantage of this “window of opportunity” and requests the bus to perform a push of the data that caused the retry.

Cycle 6: The system asserts AACK to terminate the second address tenure. Since the “window of opportunity” has passed, processor 0 requests the address bus again to retry its transaction. But the arbiter may not rearbitrate and grant the address bus to processor 0 before the push requested in the window of opportunity.

Cycle 7: Even though this cycle would be the snoop response window for the second address tenure, no processor may assert ARTRY, because that transaction was canceled by the ARTRY. (If AACK had not been delayed, an assertion of ARTRY here could run into contention with the snooper that would be driving ARTRY negated from the snoop response window of the first address tenure.) A bus grant is given to processor 1 to perform its push.

Cycle 8: Processor 1 begins its snoop push.

Cycle 9: The snoop push address tenure is acknowledged and terminated

Cycle 10: The arbiter now grants processor 0 the address bus to retry its transaction.

Figure 1. . Address Tenure Example

2.2 Data Tenure Timing Changes

The 60x bus is also required to have an idle cycle between data tenures. The MPC7400 native bus mode removes this requirement by allowing data streaming from one data tenure to the next. Data streaming allows burst data tenures from a single source to be driven back-to-back without a dead cycle in between. A dead cycle must be placed between two adjacent data tenures in which the data is driven from different agents. For example, if the first data transfer is a processor read from memory and the second data transfer is a processor write to memory, data streaming is not allowed. In addition, data streaming from one data transfer to a second is only allowed if the first transfer is a multiple beat transfer. Streaming from a multiple beat transaction to a single beat transaction is illegal.

3 Revision History

Table 4. Revision History

Revision Number	Changes
0.0	Initial release
1.0	Updates incorporated in document
1.1	Nontechnical reformatting

THIS PAGE INTENTIONALLY LEFT BLANK

THIS PAGE INTENTIONALLY LEFT BLANK

How to Reach Us:

Home Page:
www.freescale.com

E-mail:
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 or +1-480-768-2130
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 or +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 or 303-675-2140
Fax: 303-675-2150
LDCForFreescaleSemiconductor@hibbertgroup.com

AN1795/D