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

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 (ABB and DBB)

The MPC7400 does not use the ABB or 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 \( \text{ABB} \) and \( \text{DBB} \) signals as outputs in the form of \( \text{AMON} \) and \( \text{DMON} \) 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 \( \text{ABB} \) and \( \text{DBB} \) signals internally because the processor is not required to generate them.

### 1.1.2 Data Retry (DRTRY)

The data retry input does not exist in the MPC7400 native bus mode specification, so the DRTRY signal is not supported.

### 1.1.3 Extended Transfer Protocol (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 (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 (APE and 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 \( \text{DBWO} \) and \( \text{SHD} \). The \( \text{DBWO} \) signal was expanded to \( \text{DTI} \) for support of full out-of-order transactions. The \( \text{SHD} \) expansion to \( \text{SHD0} \) and \( \text{SHD1} \) allows for increased levels of address pipelining. Both of these expanded signals are discussed below.

### 1.2.1 Data Bus Write Only (DBWO) to Data Transfer Index (DTI[0–2])

The 60x bus transaction reordering scheme was implemented with the \( \text{DBWO} \) 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 (DBG). It is sampled each cycle by the MPC7400, and is qualified by the assertion of 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 DBWO protocol in which the assertion of 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 (SHD) to Shared (SHD0, 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 SHD signal be implemented. When signaling a snoop response of shared, the MPC7400 must assert SHD0 unless 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 SHD1. This way SHD0 or 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 SHD0 or 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—HIT and 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 (HIT)

The 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 (ARTRY) window (the cycle after an address acknowledge (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 ARTRY and flushing the data to memory, the MPC7400 asserts HIT to indicate that it can supply the data directly to the other master.

Like other snoop responses, HIT can be driven as soon as the second cycle after TS. If AAACK is delayed, the response needs to be held until the cycle after AAACK.

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 HIT signal for a second cycle after AAACK. If the data is modified, the MPC7400 negates HIT on the second cycle after AAACK, 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 ARTRY and HIT simultaneously for the same snoop response. When simultaneously asserted, ARTRY supersedes HIT.

### 1.3.2 Data Ready (DRDY)

The 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 HIT is ready. The arbiter will respond by granting the data bus to all devices participating in the transaction.

### 1.3.3 Enhanced Mode (EMODE)

The assertion of the EMODE signal at hard reset’s (HRESET) negation will select the MPC7400 native bus mode, otherwise 60x bus mode is selected. After the negation of HRESET, if EMODE is asserted it selects address bus drive mode¹ (only for native bus mode of the MPC7400 processor mode). 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 (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 CHK to HRESET.
- Tying CHK low enables engineering diagnostic mode.
- For normal operation tie 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 HRESET.

<table>
<thead>
<tr>
<th>BVSEL Pin</th>
<th>L2VSEL Pin</th>
<th>Processor Interface Voltage (V)</th>
<th>L2 Interface Voltage (V)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1.8</td>
<td>1.8</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1.8</td>
<td>3.3</td>
</tr>
<tr>
<td>0</td>
<td>HRESET</td>
<td>1.8</td>
<td>2.5</td>
</tr>
</tbody>
</table>

¹Address Bus Drive mode causes the MPC7400 to drive the address bus whenever BG is asserted independent of whether the MPC7400 has a bus transaction to run or not.
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 ABB signal. Likewise, DMON replaces the 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

<table>
<thead>
<tr>
<th>Pin</th>
<th>MPC750 Signal</th>
<th>MPC7400 Signal</th>
</tr>
</thead>
<tbody>
<tr>
<td>D01</td>
<td>DBW0</td>
<td>DTI[0]</td>
</tr>
<tr>
<td>H06</td>
<td>DTRY</td>
<td>DTI[1]</td>
</tr>
<tr>
<td>G01</td>
<td>DBDIS</td>
<td>DTI[2]</td>
</tr>
<tr>
<td>A03</td>
<td>TLBISYNC</td>
<td>EMODE</td>
</tr>
<tr>
<td>B03</td>
<td>No-connect</td>
<td>SHD0</td>
</tr>
<tr>
<td>B04</td>
<td>No-connect</td>
<td>SHDT</td>
</tr>
<tr>
<td>K09</td>
<td>No-connect</td>
<td>DRDY</td>
</tr>
<tr>
<td>B05</td>
<td>No-connect</td>
<td>HIT</td>
</tr>
<tr>
<td>K19</td>
<td>No-connect</td>
<td>L2A17</td>
</tr>
<tr>
<td>W19</td>
<td>No-connect</td>
<td>L2ASPARE</td>
</tr>
<tr>
<td>W01</td>
<td>No-connect</td>
<td>BVSEL</td>
</tr>
<tr>
<td>A19</td>
<td>No-connect</td>
<td>L2VSEL</td>
</tr>
<tr>
<td>K11</td>
<td>No-connect</td>
<td>CHK</td>
</tr>
</tbody>
</table>

*BVSEL, L2VSEL, and CHK signals are either connected to Vdd, Vss, or 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 (BG) signal on the cycle in which the end of the first transaction occurs (the cycle when the address acknowledge (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 (TS) signal) on the same cycle that another device asserts the address retry (ARTRY) signal for the tenure that had just ended. If this happens, the system and all bus devices must recognize that the second TS is implicitly retried as well. Both behaviors (back-to-back address tenures and ARTRY assertion) are shown in Table 1.
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>
<thead>
<tr>
<th>Revision Number</th>
<th>Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>Initial release</td>
</tr>
<tr>
<td>1.0</td>
<td>Updates incorporated in document</td>
</tr>
<tr>
<td>1.1</td>
<td>Nontechnical reformatting</td>
</tr>
</tbody>
</table>
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.