PrimeCell™ MultiPort Memory Controller (PL172)

Revision: r2p4

Technical Reference Manual

ARM®

Copyright © 2002-2006 ARM Limited. All rights reserved.
ARM DDI 0215E

PrimeCell MultiPort Memory Controller (PL172)
Technical Reference Manual

Copyright © 2002-2006 ARM Limited. All rights reserved.

Release Information

The following changes have been made to this document.

Change history

Date	Issue	Confidentiality	Change
January 2002	A	Non-Confidential	First release.
July 2002	B	Non-Confidential	Second release.
March 2003	C	Non-Confidential	Third release for r2p2.
22 March 2004	D	Non-Confidential	Fourth release for r2p3. The description of the **HCLK**:**MPMCCLKOUT** ratio, set in the MPMCConfig Register is clarified. See *Configuration Register* on page 3-10.
27 April 2006	E	Non-Confidential	Fifth release for r2p4. The description of the size of the row address synchronous memory parts that the controller supports is corrected. See *Features* on page 1-2. The description of the **MPMCDQMOUT[3:0]** signal is corrected. See *Pad interface and control signals* on page A-11.

Proprietary Notice

Words and logos marked with ® or ™ are registered trademarks or trademarks of ARM Limited in the EU and other countries, except as otherwise stated below in this proprietary notice. Other brands and names mentioned herein may be the trademarks of their respective owners.

Neither the whole nor any part of the information contained in, or the product described in, this document may be adapted or reproduced in any material form except with the prior written permission of the copyright holder.

The product described in this document is subject to continuous developments and improvements. All particulars of the product and its use contained in this document are given by ARM in good faith. However, all warranties implied or expressed, including but not limited to implied warranties of merchantability, or fitness for purpose, are excluded.

This document is intended only to assist the reader in the use of the product. ARM Limited shall not be liable for any loss or damage arising from the use of any information in this document, or any error or omission in such information, or any incorrect use of the product.

Confidentiality Status

This document is Non-Confidential. The right to use, copy and disclose this document may be subject to license restrictions in accordance with the terms of the agreement entered into by ARM and the party that ARM delivered this document to.

Product Status

The information in this document is final, that is for a developed product.

Web Address

`http://www.arm.com`

Copyright © 2002-2006 ARM Limited. All rights reserved.

Copyright © 2002-2006 ARM Limited. All rights reserved.

Contents
PrimeCell MultiPort Memory Controller (PL172) Technical Reference Manual

Copyright © 2002-2006 ARM Limited. All rights reserved.

 Copyright © 2002-2006 ARM Limited. All rights reserved.

List of Tables
PrimeCell MultiPort Memory Controller (PL172) Technical Reference Manual

Copyright © 2002-2006 ARM Limited. All rights reserved.

 Copyright © 2002-2006 ARM Limited. All rights reserved.

List of Figures
PrimeCell MultiPort Memory Controller (PL172) Technical Reference Manual

Copyright © 2002-2006 ARM Limited. All rights reserved.

 Copyright © 2002-2006 ARM Limited. All rights reserved.

Preface

This preface introduces the *PrimeCell MultiPort memory Controller (PL172) Technical Reference Manual*. It contains the following sections:

Copyright © 2002-2006 ARM Limited. All rights reserved.

About this manual

This document is the technical reference manual for the PrimeCell MPMC (PL172).

Product revision status

The r*n*p*n* identifier indicates the revision status of the product described in this manual, where:

r*n* Identifies the major revision of the product.

p*n* Identifies the minor revision or modification status of the product.

Intended audience

This manual is written for hardware and software engineers implementing *System-on-Chip* (SoC) designs. It provides information to enable designers to integrate the peripheral into a target system as quickly as possible.

Using this manual

This manual is organized into the following chapters:

Chapter 1 *Introduction*

Read this chapter for an introduction to the PrimeCell MPMC (PL172).

Chapter 2 *Functional Overview*

Read this chapter for a description of the major functional blocks of the PrimeCell MPMC (PL172).

Chapter 3 *Programmer's Model*

Read this chapter for a description of the PrimeCell MPMC (PL172) registers and programming details.

Chapter 4 *Programmer's Model for Test*

Read this chapter for a description of the logic in the PrimeCell MPMC (PL172) for functional verification and production testing.

Appendix A *Signal Descriptions*

Read this appendix for descriptions of the PrimeCell MPMC (PL172) signals.

 Copyright © 2002-2006 ARM Limited. All rights reserved.

Conventions

Conventions that this manual can use are described in:

- *Typographical*
- *Signals*
- *Numbering* on page xiv.

Typographical

The typographical conventions are:

italic	Highlights important notes, introduces special terminology, denotes internal cross-references, and citations.
bold	Highlights interface elements, such as menu names. Denotes ARM processor signal names. Also used for terms in descriptive lists, where appropriate.
`monospace`	Denotes text that you can enter at the keyboard, such as commands, file and program names, and source code.
`mono`space	Denotes a permitted abbreviation for a command or option. You can enter the underlined text instead of the full command or option name.
`monospace italic`	Denotes arguments to monospace text where the argument is to be replaced by a specific value.
`monospace bold`	Denotes language keywords when used outside example code.
< and >	Angle brackets enclose replaceable terms for assembler syntax where they appear in code or code fragments. They appear in normal font in running text. For example: • `MRC p15, 0 <Rd>, <CRn>, <CRm>, <Opcode_2>` • The Opcode_2 value selects which register is accessed.

Signals

The signal conventions are:

Signal level	The level of an asserted signal depends on whether the signal is active-HIGH or active-LOW. Asserted means HIGH for active-HIGH signals and LOW for active-LOW signals.
Prefix A	Denotes *Advanced eXtensible Interface* (AXI) global and address channel signals.

Copyright © 2002-2006 ARM Limited. All rights reserved.

Prefix B	Denotes AXI write response channel signals.
Prefix C	Denotes AXI low-power interface signals.
Prefix H	Denotes *Advanced High-performance Bus* (AHB) signals.
Prefix n	Denotes active-LOW signals except in the case of AHB or *Advanced Peripheral Bus* (APB) reset signals.
Prefix P	Denotes APB signals.
Prefix R	Denotes AXI read channel signals.
Prefix W	Denotes AXI write channel signals.
Suffix n	AHB **HRESETn** and APB **PRESETn** reset signals.

Numbering

The numbering convention is:

<size in bits>'<base><number>

This is a Verilog method of abbreviating constant numbers. For example:

- 'h7B4 is an unsized hexadecimal value.
- 'o7654 is an unsized octal value.
- 8'd9 is an eight-bit wide decimal value of 9.
- 8'h3F is an eight-bit wide hexadecimal value of `0x3F`. This is equivalent to b00111111.
- 8'b1111 is an eight-bit wide binary value of b00001111.

Further reading

This section lists publications by ARM Limited.

ARM Limited periodically provides updates and corrections to its documentation. See `http://www.arm.com` for current errata sheets, addenda, and the ARM Limited Frequently Asked Questions list.

ARM publications

This manual contains information that is specific to the PrimeCell MultiPort Memory Controller (PL172). See the following documents for other relevant information:

- *AMBA® Specification (Rev 2.0)* (ARM IHI 0011)

Copyright © 2002-2006 ARM Limited. All rights reserved.

- *ARM PrimeCell MultiPort Memory Controller (PL172) Design Manual* (PL172 DDES 0000)
- *PrimeCell MultiPort Memory Controller (PL172) Integration Manual* (ARM DII 0078)
- *AMBA Design Kit Technical Reference Manual* (ARM DDI 0243)
- *ARM PrimeCell External Bus Interface (PL220) Technical Reference Manual* (ARM DDI 0249).

Copyright © 2002-2006 ARM Limited. All rights reserved.

Feedback

ARM Limited welcomes feedback on the PrimeCell MPMC (PL172) and its documentation.

Feedback on this product

If you have any comments or suggestions about this product, contact your supplier giving:

- the product name
- a concise explanation of your comments.

Feedback on this manual

If you have any comments on this manual, send email to `errata@arm.com` giving:

- the title
- the number
- the relevant page number(s) to which your comments apply
- a concise explanation of your comments.

ARM Limited also welcomes general suggestions for additions and improvements.

Copyright © 2002-2006 ARM Limited. All rights reserved.

Chapter 1
Introduction

This chapter introduces the PrimeCell *MultiPort Memory Controller* (MPMC) (PL172). It contains the following sections:

Copyright © 2002-2006 ARM Limited. All rights reserved.

1.1 About the PrimeCell MultiPort Memory Controller (PL172)

The PrimeCell *MultiPort Memory Controller* (MPMC) is an *Advanced Microcontroller Bus Architecture* (AMBA) (Rev 2.0) compliant *System-on-Chip* (SoC) peripheral that is developed, tested, and licensed by ARM Limited. It connects to the *Advanced High-performance Bus* (AHB).

1.1.1 Features

The MPMC provides the following features:

- AMBA 32-bit AHB compliancy.
- Dynamic memory interface support including SDRAM, JEDEC low-power SDRAM, and Micron SyncFlash.
- Asynchronous static memory device support including RAM, ROM, and Flash, with or without asynchronous page mode.
- Designed to operate with cached processors with write through (ARM7x), or copy back (ARM9x and ARM10x) caches.
- Designed to operate with uncached, and tightly coupled memory processors.
- Low transaction latency.
- Read and write buffers to reduce latency and to improve performance, particularly for uncached processors.
- Four AHB interfaces for accessing external memory.
- 8-bit, 16-bit, and 32-bit wide static memory support.
- 16-bit and 32-bit wide chip select SDRAM memory support.
- 16-bit wide chip select Micron SyncFlash memory support.

 Copyright © 2002-2006 ARM Limited. All rights reserved.

- Static memory features include:
  - asynchronous page mode read
  - programmable wait states
  - bus turnaround delay
  - output enable, and write enable delays
  - extended wait.
- Four chip selects for synchronous memory and four chip selects for static memory devices.
- Software controllable **HCLK** to **MPMCCLKOUT** ratio.
- Power-saving modes dynamically control SDRAM **MPMCCKEOUT** and **MPMCCLKOUT**.
- Dynamic memory self-refresh mode supported by a *Power Management Unit* (PMU) interface or by software.
- Controller supports 2K, 4K, and 8K row address synchronous memory parts. That is typical 512Mb, 256Mb, 128Mb, and 16Mb parts, with 8, 16, or 32 DQ bits per device.
- Two reset domains enable dynamic memory contents to be preserved over a soft reset.
- A separate AHB interface to program the MPMC. This enables the MPMC registers to be situated in memory with other system peripheral registers.
- Locked AHB transactions supported.
- Support for all AHB burst types.
- Little and big-endian support.
- Support for the *External Bus Interface* (EBI) that enables the memory controller pads to be shared.
- Integrated *Test Interface Controller* (TIC).
- PrimeCell ID support.

Note

Synchronous static memory devices, synchronous burst mode, are not supported.

Copyright © 2002-2006 ARM Limited. All rights reserved.

1.2 Supported dynamic memory devices

This section provides examples of dynamic memory devices that are supported by the MPMC:

- *Examples of JEDEC SDRAM devices*
- *Examples of Micron synchronous flash type devices* on page 1-5
- *Examples of JEDEC low-power SDRAM devices* on page 1-5.

Note

This is not an exhaustive list of supported devices.

1.2.1 Examples of JEDEC SDRAM devices

The MPMC supports the following SDRAM devices:

- 16Mb devices:
  - Micron MT48LC1M16A1S
  - Samsung K4S160822D-G/F
- 64Mb devices:
  - Micron MT48LC2M32B2-6
  - Micron MT28S4M162C-10
  - Samsung K4S641632C
  - Elpida VDP4564323-10
  - Hitachi HM5264805F-75
- 128Mb devices:
  - Micron MT48LC4M32B2
  - Micron MT48LC16M8A2
  - Micron MT48LC8M16A2
- 256Mb devices:
  - Elpida VPP45256163-10
  - Micron MT48LC16M16A2-8E
  - Hitachi HM522532F-B6
  - Hitachi HM5225805B-75
- 512Mb device:
  - Elpida HM5257805B-A6.

 Copyright © 2002-2006 ARM Limited. All rights reserved.

1.2.2 Examples of Micron synchronous flash type devices

The MPMC supports the following 64Mb devices:

- Micron MT28S4M16LC-10
- Micron MT28S4M16LC-12.

1.2.3 Examples of JEDEC low-power SDRAM devices

The MPMC supports the following JEDEC low-power SDRAM devices:

- 64Mb Micron MT48LC2M32LFFC-8
- 128Mb Infineon HYB25L128160AC.

Copyright © 2002-2006 ARM Limited. All rights reserved.

1.3 Supported static memory devices

This section provides examples of static memory devices that are supported by the MPMC:

- *Examples of ROM devices*
- *Examples of page mode ROM devices*
- *Examples of SRAM devices*
- *Examples of flash devices*
- *Examples of page mode flash devices*.

Note

This is not an exhaustive list of supported devices.

1.3.1 Examples of ROM devices

The MPMC supports the 128Mb Samsung K3N9V100M-YC.

1.3.2 Examples of page mode ROM devices

The MPMC supports the 128Mb Samsung K3P9V100M-YC.

1.3.3 Examples of SRAM devices

The MPMC supports the following devices:

- 256Kb IDT IDT71V256SA20Y
- 256Kb Micron MT28F004b5-672
- 1Mb Micron MTSC2568-12
- 4Mb Samsung K6F8016R6M
- 4Mb Samsung K6R4016CK-12
- 8Mb Samsung K6T8016C3M-70
- 8Mb Samsung K6F8008R2M.

1.3.4 Examples of flash devices

The MPMC supports the 4Mb Micron MT28F004B5.

1.3.5 Examples of page mode flash devices

The MPMC supports the 8Mb Intel 28F800F3 and the 4Mb Intel E28F320J3A110.

 Copyright © 2002-2006 ARM Limited. All rights reserved.

1.4 Product revisions

This section describes changes to the functionality of the product in:
- *r1p0-r2p2*
- *r2p2-r2p3* on page 1-10
- *r2p3-r2p4* on page 1-10.

1.4.1 r1p0-r2p2

The changes between releases r1p0 and r2p2 are described in:
- *Functional changes*
- *Performance optimizations* on page 1-10
- *Power optimizations* on page 1-10.

Functional changes

This section describes the following functional changes:
- *Addition of EBI side band signals*
- *Addition of MPMCSTCS1PB signal* on page 1-8
- *CKE set low on power-on reset* on page 1-8
- *Pad interface updates* on page 1-8
- *Static memory address updates* on page 1-8
- *AHB register port signal updates* on page 1-8
- *MPMCDynamicConfig Register* on page 1-9
- *MPMCStaticConfig Registers* on page 1-9
- *MPMCConfig Register* on page 1-9
- *MPMCPeriphId Registers* on page 1-9
- *MpmcDynamicControl Register* on page 1-9.

Addition of EBI side band signals

The EBI side band signals have been added to support the EBI:
- **MPMCEBIGNT**
- **MPMCBACKOFF**
- **MPMCEBIREQ**.

For backwards compatibility, connect the signals as follows:
- **MPMCEBIGNT** = HIGH
- **MPMCBACKOFF** = LOW
- **MPMCEBIREQ** unconnected.

Copyright © 2002-2006 ARM Limited. All rights reserved.

Addition of MPMCSTCS1PB signal

This signal enables you to set the polarity of static memory chip select 1 byte lane using a tie off.

For backwards compatibility, connect **MPMCSTCS1PB** so that it is LOW.

CKE set low on power-on reset

CKE is set LOW on power-on reset to support the ASIC being powered down while the SDRAM is in self-refresh mode.

The **MPMCCKEOUT** signal is LOW on power-on reset.

Pad interface updates

The controller has been extended to support both command or clock delayed modes with the addition of the MPMCDynamicReadConfig Register. See *Dynamic Memory Read Configuration Register* on page 3-15.

Revision r1p0 supported clock delay mode only. This is the power-on reset configuration for r2p1.

Program the added MPMCDynamicReadConfig Register to select command delay mode.

The following signal is added to support command delay mode:

- **MPMCCLKDELAY**.

For backwards compatibility, connect the signal as follows:

- **MPMCCLKDELAY** <= **MPMCCLK**.

Static memory address updates

You can right-shift the static memory address when accessing 16 or 32-bit static memory chip selects to reduce pin count.

Addition of the following signal enables backwards compatibility mode:

- **MPMCREL1CONFIG**.

For backwards compatibility, connect **MPMCREL1CONFIG** so that it is HIGH.

AHB register port signal updates

The AHB register port signal is updated to support register updates.

HRDATAREG width is changed and the following signals are removed:

- **HWDATAREG29TO28**

 Copyright © 2002-2006 ARM Limited. All rights reserved.

- **HWDATAREG26**
- **HWDATAREG24TO22**.

See *AHB register signals* on page A-2

MPMCDynamicConfig Register

The row width, column width, and number of bank fields from the MPMCDynamicConfig Register are removed. This information is now generated from the address mapping information.

MPMCStaticConfig Registers

The MPMCStaticConfig Registers correctly indicate the value of the **MPMCCS30POL**, **MPMCSTCS1MW[1:0]**, and **MPMCSTCS1PB** signals as selected by the tie-off pins. See *Static Memory Configuration Registers 0-3* on page 3-31.

MPMCConfig Register

The MPMCConfig Register correctly indicates the value of the **MPMCBIGENDIAN** signal as selected by the tie-off pin. See *Configuration Register* on page 3-10.

MPMCPeriphId Registers

These registers are updated to indicate the new revision number and to enable additional configurations. See *Peripheral Identification Registers* on page 3-40.

MpmcDynamicControl Register

An output clock control field has been added to the MpmcDynamicControl Register. See *Dynamic Memory Control Register* on page 3-11.

MPMCConfig Register drain write buffers field

The MPMCConfig Register drain write buffers field has been removed because the write buffers are flushed as soon as possible.

Copyright © 2002-2006 ARM Limited. All rights reserved.

Performance optimizations

The changes made to improve performance are described in:

- *Long AHB bursts*
- *Additional pipelining in memory accesses*
- *Buffer size increase*
- *Reduced transaction latency*.

Long AHB bursts

Long AHB bursts, eight and 16, are not rearbitrated.

Additional pipelining in memory accesses

There is additional pipelining in memory accesses to maximize the SDRAM memory bandwidth.

Buffer size increase

The buffer size has been increased from four, 4 x32-bit buffers to four, 16 x 32-bit buffers to reduce transaction latency and to maximize memory bandwidth.

Reduced transaction latency

Transaction time has been reduced to minimize latency.

Power optimizations

Pad interface signal toggling has been minimized to reduce power consumption.

1.4.2 r2p2-r2p3

It is no longer possible to restart the system using hard reset when it has locked up. If you attempt to do this then data might be lost or not retained.

1.4.3 r2p3-r2p4

There is no change to the functionality described in this manual. See the engineering errata that accompanies the product deliverables for more information.

 Copyright © 2002-2006 ARM Limited. All rights reserved.

Chapter 2
Functional Overview

This chapter describes the major functional blocks of the MPMC. It contains the following sections:

Copyright © 2002-2006 ARM Limited. All rights reserved.

2.1 MPMC functional description

Figure 2-1 shows a block diagram of the MPMC.

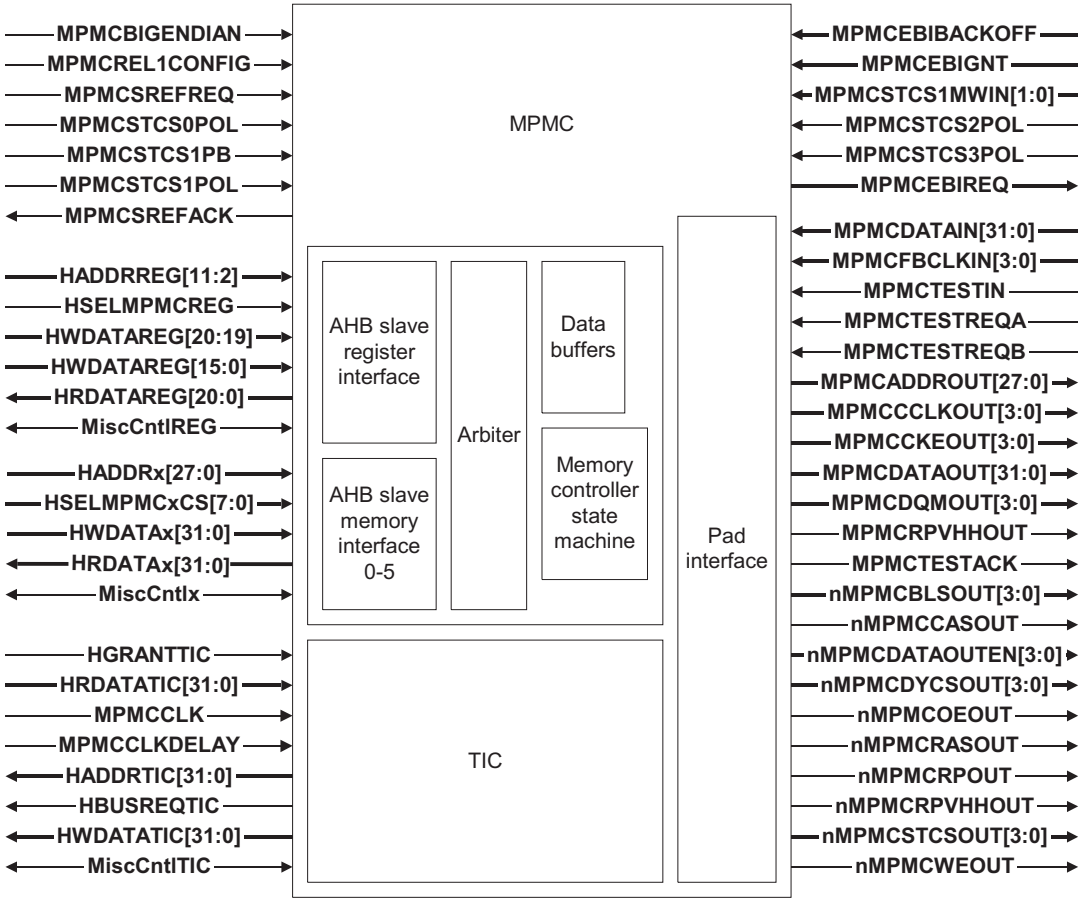

Figure 2-1 MPMC block diagram

Note

In Figure 2-1 the letter **x** in the AHB slave memory interface signals denotes a number between 0-3.

The MPMC block optimizes and controls external memory transactions.

The following sections describe the functions of the MPMC blocks:

- *AHB slave register interface* on page 2-3

 Copyright © 2002-2006 ARM Limited. All rights reserved.

- *AHB slave memory interfaces*
- *Data buffers* on page 2-4
- *Arbiter* on page 2-6
- *Memory controller state machine* on page 2-6
- *Pad interface* on page 2-7
- *Test Interface Controller (TIC)* on page 2-7.

Note

For 32-bit wide chip selects data is transferred to and from dynamic memory in SDRAM bursts of four. For 16-bit wide chip selects SDRAM bursts of eight are used.

2.1.1 AHB slave register interface

The AHB slave register interface block enables you to program the registers of the MPMC. This module also contains most of the registers and performs the majority of the register address decoding. This interface must be connected to the ARM processor AHB bus to enable you to program the MPMC.

Memory transaction endianness and transfer width

To eliminate the possibility of endianness problems, all data transfers to and from the registers of the MPMC must be 32 bits wide.

Note

If an access is attempted with a size other than a word (32 bits), it causes an ERROR response on **HRESP** and the transfer is terminated.

2.1.2 AHB slave memory interfaces

The AHB slave memory interfaces enable devices to access the external memories. The memory interfaces are prioritized, with interface 0 having the highest priority. Having more than one memory interface enables high-bandwidth peripherals direct access to the MPMC, without data having to pass over the main system bus.

Note

- All AHB burst types are supported, enabling the most efficient use of memory bandwidth.
- The AHB interfaces do not generate SPLIT and RETRY responses.

Copyright © 2002-2006 ARM Limited. All rights reserved.

Memory transaction endianness

The Endian mode (N) bit in the MPMCConfig Register determines the endianness of the data transfers to and from the external memories.

Note

The memory controller must be idle, see the busy field of the MPMCStatus Register, before endianness is changed, so that the data is transferred correctly.

Memory transaction size

Memory transactions can be 8, 16, or 32 bits wide. Any access attempted with a size greater than a word (32 bits) causes an ERROR response on **HRESP** and the transfer is terminated.

Write protected memory areas

Write transactions to write-protected memory areas generate an ERROR response on **HRESP** and the transfer is terminated.

2.1.3 Data buffers

The AHB interfaces use read and write buffers to improve memory bandwidth and reduce transaction latency. The MPMC contains four 16-word buffers. The buffers are not tied to a particular AHB interface, and you can use them as read buffers, write buffers, or a combination of both. The buffers are allocated automatically. Because of the way the buffers are designed they are always coherent for reads and writes, and across AHB memory interfaces.

The buffers are always enabled for dynamic memory. You can enable or disable them for static memory using the MPMCStaticConfig Registers.

Write buffers

Write buffers:

- Merge write transactions so that the number of external transactions are minimized.
- Buffer data until the MPMC can complete the write transaction improving AHB write latency.

 Copyright © 2002-2006 ARM Limited. All rights reserved.

- Convert all dynamic memory write transactions into quadword bursts on the external memory interface. This enhances transfer efficiency for dynamic memory.
- Reduce external memory traffic. This improves memory bandwidth and reduces power consumption.

Write buffer operation:

- If the buffers are enabled, an AHB write operation writes into the *Least Recently Used* (LRU) buffer if empty.
- If the LRU buffer is not empty the contents of the buffer are flushed to memory to make space for the AHB write data.
- If a buffer contains write data it is marked as dirty, and its contents are written to memory before the buffer can be reallocated.

The write buffers are flushed whenever:

- the memory controller state machine is not busy performing accesses to external memory
- the memory controller state machine is not busy performing accesses to external memory, and an AHB interface is writing to a different buffer.

Note

For dynamic memory the smallest buffer flush is a quadword of data. For static memory the smallest buffer flush is a byte of data.

Read buffers

Read buffers:

- Buffer read requests from memory. Future read requests that hit the buffer read the data from the buffer rather than memory, reducing transaction latency.
- Convert all read transactions into quadword bursts on the external memory interface. This enhances transfer efficiency for dynamic memory.
- Reduce external memory traffic. This improves memory bandwidth and reduces power consumption.

Read buffer operation:

- If the buffers are enabled, and the read data is contained in one of the buffers, the read data is provided directly from the buffer.

Copyright © 2002-2006 ARM Limited. All rights reserved.

- If the read data is not contained in a buffer, the LRU buffer is selected. If the buffer is dirty (contains write data), the write data is flushed to memory. When an empty buffer is available the read command is posted to the memory. While the memory controller is waiting for the data to be returned the memory controller can re-arbitrate to enable additional memory transactions to be processed. When the first data item is returned from memory the read data is provided to the respective AHB port. Other AHB ports can access the data in the buffer when the read transaction has completed.

A buffer filled by performing a read from memory is marked as not-dirty, not containing write data, and its contents are not flushed back to the memory. If a subsequent AHB transfer performs a write that hits the buffer, that buffer is overwritten and marked as dirty.

2.1.4 Arbiter

The arbiter arbitrates between the AHB slave memory interfaces. AHB interface 0 has the highest access priority, and AHB interface 3 has the lowest priority. For more information see *Arbitration* on page 2-16.

2.1.5 Memory controller state machine

The memory controller state machine comprises two functional blocks:

- a static memory controller
- a dynamic memory controller.

The dynamic memory controller holds up to two requests in its internal buffer. It prioritizes and rearranges accesses to maximize memory bandwidth and minimize transaction latency.

For example, if AHB interfaces 3 and 2 simultaneously request a data transfer from dynamic memory, to different memory banks, and the port 2 request address is to a closed page, but port 3 address is for an already open page, the following sequence occurs:

1. The ACT command is sent to open the SDRAM row specified by the AHB interface 2 address.
2. The AHB interface 3 access is completed.
3. The AHB interface 2 access is completed.

The access priority is modified to take into account the ease of getting data to complete each transfer, but the access priority is always biased to the highest priority AHB interface.

 Copyright © 2002-2006 ARM Limited. All rights reserved.

2.1.6 Pad interface

The pad interface block provides the interface to the pads. The pad interface uses feedback clocks, **MPMCFBCLKIN[3:0]**, to resynchronize SDRAM read data from the off-chip to on-chip domains.

Figure 2-2 shows a block diagram of the pad interface.

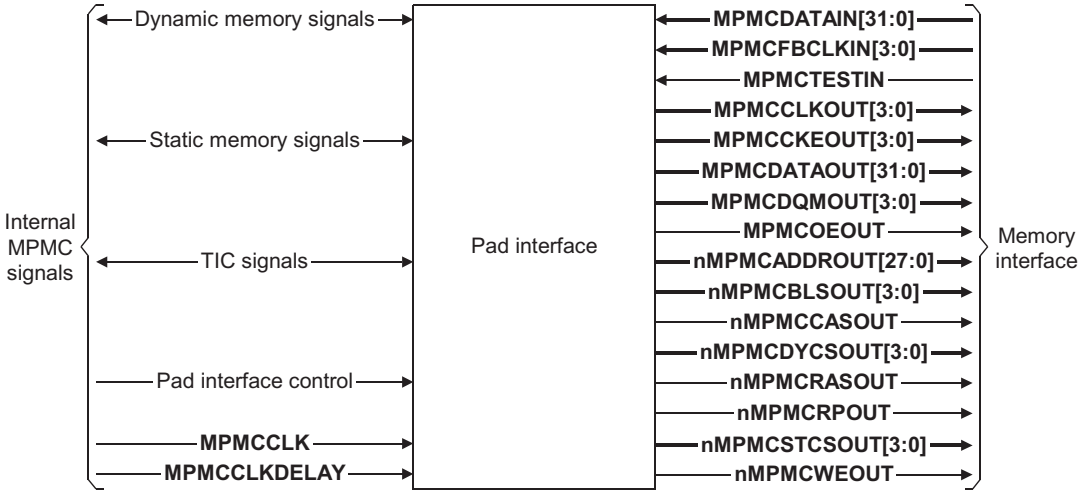

Figure 2-2 Pad interface block diagram

2.1.7 Test Interface Controller (TIC)

The TIC enables TIC device testing. For full details about the TIC, see the *AMBA Specification*. The AHB master interface must be placed on the same AHB interface as the ARM processor. Figure 2-3 on page 2-8 shows a block diagram of the TIC.

Copyright © 2002-2006 ARM Limited. All rights reserved.

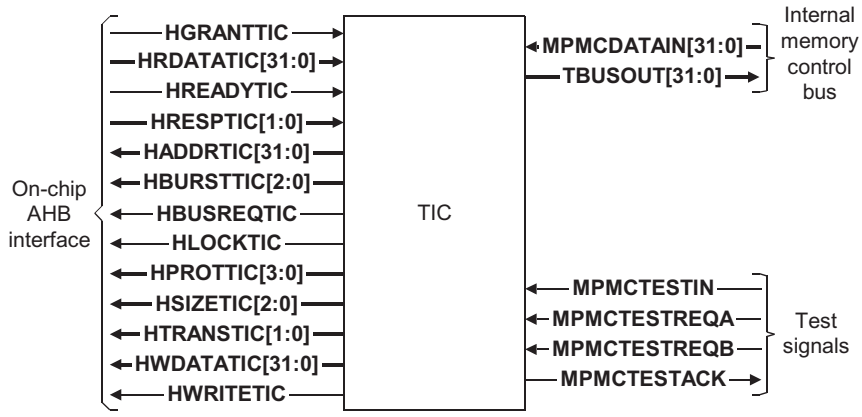

Figure 2-3 TIC block diagram

Copyright © 2002-2006 ARM Limited. All rights reserved.

2.2 Overview of an example MPMC system

Figure 2-4 shows the MPMC in an example system.

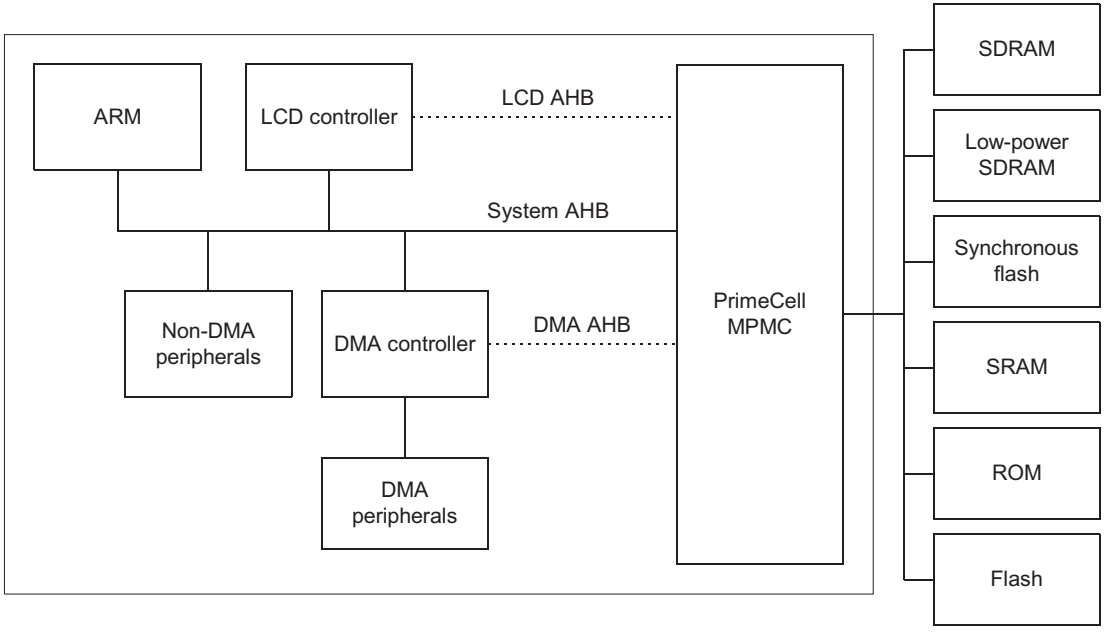

Figure 2-4 MPMC in an example system

The example system uses two types of buses:

- external
- internal (AHB).

2.2.1 External bus

The off-chip bus that contains data, address, and control signals, connects the ASIC or ASSP to the external memory.

Note

- Connecting a large number of memory devices externally impacts on performance.
- Only one memory device can be accessed at a time.

Copyright © 2002-2006 ARM Limited. All rights reserved.

2.2.2 Internal bus

The on-chip bus enables communication between the on-chip peripherals. The MPMC appears as a standard slave on the on-chip bus and controls the memory on the external bus.

Providing multiple AHB interfaces improves system performance by enabling several access requests to be presented to the memory controller at the same time. This enables the MPMC to pipeline many of the operations, for example, bank activate and precharge, and so reduce the average system access latency and improve utilization of external memory. The use of multiple AHB interfaces also improves system performance by removing heavy DMA traffic from the main AHB bus.

 Copyright © 2002-2006 ARM Limited. All rights reserved.

2.3 Low-power operation

In many systems, the contents of the memory system have to be maintained during low-power sleep modes. The MPMC provides two features to enable this:

- Dynamic memory refresh over soft reset.
- A mechanism to place the dynamic memories into self-refresh mode.

The dynamic memories can enter self-refresh mode:

- Automatically in hardware using a *Power Management Unit* (PMU). This is typically present to control the safe transition between the following modes:
  - power-up
  - reset
  - normal
  - sleep.
- Manually in software by setting the SREFREQ bit in the MPMCDynamicControl Register and polling the SREFACK bit in the MPMCStatus Register.

You can use the PMU to enable self-refresh mode to be entered automatically. To do this, the PMU asserts the **MPMCSREFREQ** signal when self-refresh mode is to be entered. The memory controller then closes any open memory banks and puts the external memory into self-refresh mode. The MPMC then asserts the **MPMCSREFACK** signal to indicate to the PMU that self-refresh mode is entered. The system must ensure that the memory subsystem is idle before asserting **MPMCSREFREQ**. Any transactions to memory that are generated while the memory controller is in self-refresh mode are rejected and an error response is generated (**HRESP** = ERROR). Deasserting **MPMCSREFREQ** returns the memory to normal operation. See the memory data sheet for refresh requirements.

Note

- If **MPMCSREFREQ** is not required this signal must be tied LOW.
- Static memory can be accessed as normal when the SDRAM memory is in self-refresh mode.

2.3.1 Low-power SDRAM deep-sleep mode

The MPMC supports JEDEC low-power SDRAM deep-sleep mode. Deep-sleep mode can be entered by setting the deep-sleep mode bit, DP, in the MPMCDynamicControl Register. The device is then put into a low-power mode where the device is powered down and no longer refreshed. All data in the memory is lost.

Copyright © 2002-2006 ARM Limited. All rights reserved.

2.3.2 Low-power SDRAM partial array refresh

The MPMC supports JEDEC low-power SDRAM partial array refresh. You can program partial array refresh by initializing the SDRAM memory device appropriately. When the memory device is put into self-refresh mode only the memory banks specified are refreshed. The memory banks that are not refreshed lose their data contents.

 Copyright © 2002-2006 ARM Limited. All rights reserved.

2.4 Locked accesses

Locked accesses on the AHB bus, transactions where **HMASTLOCK** is HIGH, are processed appropriately. When the memory controller performs a locked transfer the memory controller does not re-arbitrate to another AHB port until the lock transfer is completed.

Copyright © 2002-2006 ARM Limited. All rights reserved.

2.5 Burst types

All AHB burst types are supported.

Note

INCR transfers are split internally into four-word bursts.

Copyright © 2002-2006 ARM Limited. All rights reserved.

2.6 Busy transfer type

When an AHB master generates busy cycles (AHB HTRANS=BUSY) the memory controller waits until busy is inactive before completing the transfer. This increases transfer latency. The memory controller does not re-arbitrate to another AHB port if busy goes active.

Copyright © 2002-2006 ARM Limited. All rights reserved.

2.7 Arbitration

The memory controller re-arbitrates at the end of an AHB burst transfer. AHB port 0 has the highest priority and AHB port 3 has the lowest priority.

Re-arbitration occurs:

- when a read or write request has been posted to the memory controller
- if an AHB port is unable to post a request it is backed off and the AHB is re-arbitrated
- at the end of an AHB burst transfer.

If an INCR8, WRAP8, INCR16, or WRAP16 AHB burst is broken by the AHB arbiter the AHB memory port performing the transfer has the highest priority for the subsequent request. To maintain arbitration priority in the memory controller it is recommended that the AHB arbiter re-arbitrates on a burst boundary.

Note

- It is recommended that the AHB arbiter re-arbitrates on a burst boundary, because this leads to the most efficient bus utilization.
- AHB protocol does not permit AHB masters to break a defined length burst transfer, INCR4, WRAP4, INCR8, WRAP8, INCR16, and WRAP16. An AHB master can only break an undefined length burst transfer (INCR).
- It is not possible for an AHB burst to cross an SDRAM column, because AHB bursts must not cross a 1KB boundary, and the smallest SDRAM column length supported is 1KB long.

2.7.1 Re-arbitration occurrence

The re-arbitration occurrence is the time when the memory controller re-arbitrates. The following lists when re-arbitration does not occur:

- When a locked transfer, as defined by the AHB **HLOCK** signal, has started the AHB memory port is not re-arbitrated until the locked transfer has completed and the AHB lock signal is deasserted.
- When an AHB burst transfer has started the AHB memory port is not re-arbitrated until the burst has completed. The longest AHB transfer is 16 AHB transfers long (INCR16/WRAP16).

 Copyright © 2002-2006 ARM Limited. All rights reserved.

2.7.2 Re-arbitration priority

The arbitration in MPMC happens in multiple levels:

- AHB level
- memory controller level.

The arbitration scheme used at each level is described in the following sections:

- *Priority at the AHB/buffer level*
- *Priority at the memory controller level*.

Priority at the AHB/buffer level

A request from highest priority AHB port that has been allocated a buffer is submitted to the memory controller (AHB port 0 is the highest priority port, AHB port 3 is the lowest priority port).

If there is not a buffer available for highest priority AHB, or the buffer allocated is a locked buffer from a previous transaction, or the memory controller is unable to accept a new command, the AHB port is backed off from arbitration until the reason for backoff is deasserted.

Auto-refresh requests are passed through to the memory controller when they are raised.

Priority at the memory controller level

Auto-refresh has the highest priority.

When a request has been raised, if State Machine 0 is free, the request is assigned to State Machine 0, otherwise it is assigned to State Machine 1.

When the access requests are raised to the memory controller, DASM0 (Dynamic memory Data-Access-State Machine 0) has higher priority over DASM1(Dynamic Memory Data-Access-State Machine 1).

Copyright © 2002-2006 ARM Limited. All rights reserved.

2.8 Worst-case transaction latency

The worst-case transaction latency for the highest priority AHB memory port is 108 clock cycles. Lower priority AHB memory ports can be locked out indefinitely if a higher priority AHB memory port continually performs memory requests. These values are based on the assumptions given in the following sections:

- *Worst-case transaction latency for the highest priority AHB memory port*
- *System factors effecting worst-case latency* on page 2-19.

2.8.1 Worst-case transaction latency for the highest priority AHB memory port

The following assumptions were made for calculating the worst-case transaction latency:

- System factors have been ignored in these calculations. For information on system factors see *System factors effecting worst-case latency* on page 2-19.
- SDRAM memory latency values are:
  - precharge = 3
  - active = 3
  - CAS = 3
  - Auto-refresh (t_{RFC}) = 7.
- The SDRAM memory chip selects have a 32-bit wide data bus.
- There are no devices connected to the static memory chip selects.

Note

Using SDRAM memory chip selects with a 16-bit wide data bus, SDRAM memory with larger latency values, or using slow static memory devices in a system can impact the worst-case latency.

For the MPMC (PL172) the worst-case latency scenario is as follows:

- All MPMC buffers are fully filled with 16 words of write data. The write data is to unopened SDRAM memory rows.
- An INCR16 read is submitted to a lower priority port. The read transfer hits the memory controller buffer.
- An auto-refresh is generated after the INCR16 read is submitted.
- The highest priority AHB memory port, AHB 0, performs a read access, a few cycles after the INCR16 read access starts.

 Copyright © 2002-2006 ARM Limited. All rights reserved.

The read data is to unopened SDRAM memory rows.

Before the first data from the highest priority AHB memory port is returned the following transactions occur:

1. The low-priority INCR16 reads 16 data items from the buffer. This transaction takes 16 cycles.
2. The SDRAM auto-refresh is generated. This transaction takes 15 cycles.
3. The highest priority read transaction causes write buffer to be flushed. The write buffer is to an unopened SDRAM memory row. This transaction takes 35 cycles.
4. The highest priority read transaction is performed. The read is to an unopened SDRAM memory row. The first data is returned after 18 cycles.

The worst-case transaction latency for highest priority AHB memory port is 16+15+35+18=84 cycles.

2.8.2 System factors effecting worst-case latency

This section describes the system factors that can affect the memory controller worst-case latency to a memory request.

MPMC AHB memory port priority

The memory controller AHB memory ports are prioritized. If a master connected to a HIGH priority port performs continuous transactions, lower priority ports are not able to access the bus until the higher priority port has completed its transactions.

AHB bus

If there are multiple AHB masters on the same AHB bus, a master can only receive ownership of the bus, and submit transaction, when the AHB arbiter grants the bus to the master. The arbitration strategy in the AHB arbiter then affects the worst-case latency for a system with multiple masters on an AHB bus.

AHB masters

When an AHB master generates busy (AHB HTRANS=BUSY) cycles, the memory controller waits until busy is inactive before completing the transfer. This increases transfer latency. The memory controller does not re-arbitrate to another AHB port if busy goes active.

Copyright © 2002-2006 ARM Limited. All rights reserved.

If an AHB master performs locked (AHB HMASTLOCK=LOCK) cycles, the memory controller cannot re-arbitrate until the locked transaction has completed. This can then increase worst-case transfer latency, because the memory controller is not able to re-arbitrate to a higher priority port until the locked access has completed.

Memory devices

If slow SDRAM memories and/or a 16-bit SDRAM chip select is used, transactions take longer to complete, and that affects transfer latency.

If slow static memory devices are used, and/or an 8-bit or 16-bit static memory chip select data bus is used, transactions take longer to complete, and that affects transfer latency.

Pad multiplexing

If the memory bus is multiplexed externally, for example by using an EBI, the worst-case transfer latency is affected because the external bus is shared by multiple devices.

 Copyright © 2002-2006 ARM Limited. All rights reserved.

2.9 Memory bank select

Eight independently-configurable memory chip selects are supported, with a separate AMBA AHB select, **HSELMPMCxCS[7:0]**, to select the appropriate chip select:

- **HSELMPMCxCS** selects 0 to 3 are used to select static memory devices
- **HSELMPMCxCS** selects 4 to 7 are used to select dynamic memory devices.

Table 2-1 shows the relationship between the AHB select, **HSELMPMCxCS[7:0]**, and the memory chip selects. In the table an x refers to the AHB port number.

Table 2-1 Memory bank selection

AHB select	Chip select	Memory device	Memory chip select
HSELMPMCxCS[0]	0	Static Mem 0	**nMPMCSTCSOUT[0]**
HSELMPMCxCS[1]	1	Static Mem 1	**nMPMCSTCSOUT[1]**
HSELMPMCxCS[2]	2	Static Mem 2	**nMPMCSTCSOUT[2]**
HSELMPMCxCS[3]	3	Static Mem 3	**nMPMCSTCSOUT[3]**
HSELMPMCxCS[4]	4	Dynamic Mem 0	**nMPMCDYCSOUT[0]**
HSELMPMCxCS[5]	5	Dynamic Mem 1	**nMPMCDYCSOUT[1]**
HSELMPMCxCS[6]	6	Dynamic Mem 2	**nMPMCDYCSOUT[2]**
HSELMPMCxCS[7]	7	Dynamic Mem 3	**nMPMCDYCSOUT[3]**

The system AHB decoder determines the address range allocated to a chip select.

Note

The largest amount of memory permitted for a single chip select is 256MB.

Copyright © 2002-2006 ARM Limited. All rights reserved.

2.10 Memory map

The MPMC provides hardware support for booting from external nonvolatile memory. During booting the nonvolatile memory must be located at address `0x00000000` in memory. When the system is booted, the SRAM or SDRAM memory can be remapped to address `0x00000000` by modifying the address map in the AHB decoder.

2.10.1 Power-on reset memory map

On power-on reset, memory chip select 1 is mirrored onto memory chip select 0 and chip select 4. Therefore, any transactions to memory chip select 0 or chip select 4 (or chip select 1) access memory chip select 1. Clearing the address mirror bit (M) in the MPMCControl Register disables address mirroring, and the memory chip select 0, chip select 4, and memory chip select 1 can then be accessed as normal.

2.10.2 Chip select 1 memory configuration

You can use several input signals to configure the memory width and chip select polarity of static memory chip select 1. This enables you to boot from chip select 1.

The configuration tie-off signals are:

- **MPMCSTCS1MW[1:0]**, memory width select
- **MPMCSTCS1POL**, chip select polarity
- **MPMCSTCS1PB**, byte lane static polarity.

2.10.3 Boot from flash, SRAM remapped after boot

The system set up is:

- chip select 1 is connected to the boot flash device
- chip select 0 is connected to the SRAM to be remapped to `0x00000000` after boot.

The boot sequence is as follows:

1. At power on the reset chip select 1 is mirrored into chip select 0 (and chip select 4). The following signals are configured so that the nonvolatile memory device can be accessed:
   - **MPMCSTCS1MW[1:0]**
   - MPMCSTCS1POL (also **MPMCSTCS0POL**, **MPMCSTCS2POL**, **MPMCSTCS3POL**, and **MPMCSTCS1PB**).
2. When the power-on reset (**nPOR**) and AHB reset (**HRESETn**) go inactive, the processor starts booting from `0x00000000` in memory.

 Copyright © 2002-2006 ARM Limited. All rights reserved.

3. The software programs the optimum delay values in the flash memory so that the boot code can run at full speed.
4. The code branches to chip select 1 so that the code can continue executing from the nonremapped memory location.
5. The appropriate values are programmed into the MPMC to configure chip select 0.
6. The address mirroring is disabled by clearing the Address Mirror (M) field in the MPMCControl Register.
7. The ARM reset and interrupt vectors are copied from flash memory to SRAM that can then be accessed at address `0x00000000`.
8. More boot, initialization, or application code is executed.

2.10.4 Example of a boot from flash, SDRAM remapped after boot

The system set up is:

- chip select 1 is connected to the boot flash device
- chip select 4 is connected to the SDRAM to be remapped to `0x00000000` after boot.

The boot sequence is as follows:

1. At power on the reset chip select 1 is mirrored into chip select 4 (and chip select 0). The following signals are configured so that the nonvolatile memory device can be accessed:
   - **MPMCSTCS1MW[1:0]**
   - **MPMCSTCS1POL** (also **MPMCSTCS0POL**, **MPMCSTCS2POL**, **MPMCSTCS3POL**, and **MPMCSTCS1PB**).
2. When the power-on reset (**nPOR**) and AHB reset (**HRESETn**) go inactive, the processor starts booting from `0x00000000` in memory.
3. The software programs the optimum delay values in the flash memory so that the boot code can run at full speed.
4. The code branches to chip select 1 so that the code can continue executing from the nonremapped memory location.
5. The appropriate values are programmed into the MPMC to configure chip select 4 and the memory device is initialized.
6. The address mirroring is disabled by clearing the Address Mirror (M) field in the MPMCControl Register.

Copyright © 2002-2006 ARM Limited. All rights reserved.

7. The ARM reset and interrupt vectors are copied from flash memory to SDRAM that can then be accessed at address `0x00000000`.
8. More boot, initialization, or application code is executed.

2.10.5 Memory aliasing

Memory aliasing is not recommended. This is because internally the memory controller uses a 28-bit address (enabling up to 256MB of memory per chip select). The chip select or MPMC buffer is only selected when the appropriate AHB **HSELMPMCxCS[7:0]** signal goes active. If the region of memory region provided by the AHB decoder is larger than the memory connected to the chip select, the memory is aliased many times in the memory region. However, because the memory controller buffers always use a 28-bit address, this can cause coherency issues if aliased addressing is used.

2.10.6 AHB port address map

It is recommended that the addresses for accessing the memory for each of the AHB ports are similar. Because the memory controller uses a 28-bit address internally, the buffers use the full 28-bit address to determine the area of memory that is accessed. If the addresses are not similar, coherency issues might occur.

Note

You can tie off the high-order address bits appropriately to get around the coherency issues.

2.10.7 Unused AHB HADDRx address bits

If some of the **HADDRx** address bits are not required, the unused address bits must be tied LOW.

 Copyright © 2002-2006 ARM Limited. All rights reserved.

2.11 Sharing memory interface signals

You can share the memory interface signals, and the memory controller primary input and output signals, with other peripherals by using an *External Bus Interface* (EBI) block.

Copyright © 2002-2006 ARM Limited. All rights reserved.

 Copyright © 2002-2006 ARM Limited. All rights reserved.

Chapter 3
Programmer's Model

This chapter describes the MPMC registers and provides information for programming the microcontroller. It contains the following sections:

Copyright © 2002-2006 ARM Limited. All rights reserved.

3.1 About the programmer's model

The base address of the MPMC is not fixed, but is determined by the AHB decoder, and can be different for any particular system implementation. However, the offset of any particular register from the base address is fixed. The registers of the MPMC can only be accessed using AHB register interface port of the memory controller.

The external memory is accessed using the AHB memory interface ports. Addresses are not fixed, but are determined by the AHB decoder, and can be different for any particular system implementation. The **HSELMPMC[3:0]CS[7:0]** signals select transfers to the external memories of the MPMC.

Note

[3:0] indicates the AHB port number, and [7:0] indicates the chip select to be accessed.

Copyright © 2002-2006 ARM Limited. All rights reserved.

3.2 Summary of registers

Table 3-1 shows the MPMC registers in base offset order.

Note

For reset value some bits are reset by AHB reset, **HRESETn**, and some by power-on reset, **nPOR**.

Table 3-1 MPMC register summary

Name	Offset	Type	Reset value	Description
MPMCControl	0x000	RW	**HRESETn** 0x1 **nPOR** 0x3	See *Control Register* on page 3-8
MPMCStatus	0x004	RO	**nPOR** 0x5	See on page 3-9*Status Register* on page 3-9
MPMCConfig	0x008	RW	**nPOR** 0x00-[a]	See *Configuration Register* on page 3-10
MPMCDynamicControl	0x020	RW	**nPOR** 0x006	See *Dynamic Memory Control Register* on page 3-11
MPMCDynamicRefresh	0x024	RW	**nPOR** 0x0	See *Dynamic Memory Refresh Timer Register* on page 3-14
MPMCDynamicReadConfig	0x028	RW	**nPOR** 0x0	See *Dynamic Memory Read Configuration Register* on page 3-15
MPMCDynamictRP	0x030	RW	**nPOR** 0x0F	See *Dynamic Memory Precharge Command Period Register* on page 3-16
MPMCDynamictRAS	0x034	RW	**nPOR** 0xF	See *Dynamic Memory Active To Precharge Command Period Register* on page 3-17
MPMCDynamictSREX	0x038	RW	**nPOR** 0xF	See *Dynamic Memory Self-refresh Exit Time Register* on page 3-18
MPMCDynamictAPR	0x03C	RW	**nPOR** 0xF	See *Dynamic Memory Last Data Out To Active Time Register* on page 3-19
MPMCDynamictDAL	0x040	RW	**nPOR** 0xF	See *Dynamic Memory Data-in To Active Command Time Register* on page 3-20
MPMCDynamictWR	0x044	RW	**nPOR** 0xF	See *Dynamic Memory Write Recovery Time Register* on page 3-21

Copyright © 2002-2006 ARM Limited. All rights reserved.

Table 3-1 MPMC register summary (continued)

Name	Offset	Type	Reset value	Description
MPMCDynamictRC	0x048	RW	**nPOR** 0x1F	See *Dynamic Memory Active To Active Command Period Register* on page 3-21
MPMCDynamictRFC	0x04C	RW	**nPOR** 0x1F	See *Dynamic Memory Auto-refresh Period Register* on page 3-22
MPMCDynamictXSR	0x050	RW	**nPOR** 0x1F	See *Dynamic Memory Exit Self-refresh Register* on page 3-23
MPMCDynamictRRD	0x054	RW	**nPOR** 0xF	See *Dynamic Memory Active Bank A to Active Bank B Time Register* on page 3-24
MPMCDynamictMRD	0x058	RW	**nPOR** 0xF	See *Dynamic Memory Load Mode Register To Active Command Time Register* on page 3-25
MPMCStaticExtendedWait	0x080	RW	**nPOR** 0x0	See *Static Memory Extended Wait Register* on page 3-25
MPMCDynamicConfig0	0x100	RW	**nPOR** 0x0	See *Dynamic Memory Configuration Registers 0-3* on page 3-26
MPMCDynamicRasCas0	0x104	RW	**nPOR** 0x303	See *Dynamic Memory RAS and CAS Delay Registers 0-3* on page 3-30
MPMCDynamicConfig1	0x120	RW	**nPOR** 0x0	See *Dynamic Memory Configuration Registers 0-3* on page 3-26
MPMCDynamicRasCas1	0x124	RW	**nPOR** 0x303	See *Dynamic Memory RAS and CAS Delay Registers 0-3* on page 3-30
MPMCDynamicConfig2	0x140	RW	**nPOR** 0x0	See *Dynamic Memory Configuration Registers 0-3* on page 3-26
MPMCDynamicRasCas2	0x144	RW	**nPOR** 0x303	See *Dynamic Memory RAS and CAS Delay Registers 0-3* on page 3-30
MPMCDynamicConfig3	0x160	RW	**nPOR** 0x0	See *Dynamic Memory Configuration Registers 0-3* on page 3-26
MPMCDynamicRasCas3	0x164	RW	**nPOR** 0x303	See *Dynamic Memory RAS and CAS Delay Registers 0-3* on page 3-30
MPMCStaticConfig0	0x200	RW	**nPOR** 0x-0[a]	See *Static Memory Configuration Registers 0-3* on page 3-31
MPMCStaticWaitWen0	0x204	RW	**nPOR** 0x0	See *Static Memory Write Enable Delay Registers 0-3* on page 3-34

Copyright © 2002-2006 ARM Limited. All rights reserved.

Table 3-1 MPMC register summary (continued)

Name	Offset	Type	Reset value	Description
MPMCStaticWaitOen0	0x208	RW	**nPOR** 0x0	See *Static Memory Output Enable Delay Registers 0-3* on page 3-34
MPMCStaticWaitRd0	0x20C	RW	**nPOR** 0x1F	See *Static Memory Read Delay Registers 0-3* on page 3-35
MPMCStaticWaitPage0	0x210	RW	**nPOR** 0x1F	See *Static Memory Page Mode Read Delay Registers 0-3* on page 3-36
MPMCStaticWaitWr0	0x214	RW	**nPOR** 0x1F	See *Static Memory Write Delay Registers 0-3* on page 3-37
MPMCStaticWaitTurn0	0x218	RW	**nPOR** 0xF	See *Static Memory Turn Round Delay Registers 0-3* on page 3-37
MPMCStaticConfig1	0x220	RW	**nPOR** 0x--[a]	See *Static Memory Configuration Registers 0-3* on page 3-31
MPMCStaticWaitWen1	0x224	RW	**nPOR** 0x0	See*Static Memory Write Enable Delay Registers 0-3* on page 3-34
MPMCStaticWaitOen1	0x228	RW	**nPOR** 0x0	See*Static Memory Output Enable Delay Registers 0-3* on page 3-34
MPMCStaticWaitRd1	0x22C	RW	**nPOR** 0x1F	See *Static Memory Read Delay Registers 0-3* on page 3-35
MPMCStaticWaitPage1	0x230	RW	**nPOR** 0x1F	See *Static Memory Page Mode Read Delay Registers 0-3* on page 3-36
MPMCStaticWaitWr1	0x234	RW	**nPOR** 0x1F	See on page 3-37*Static Memory Write Delay Registers 0-3* on page 3-37
MPMCStaticWaitTurn1	0x238	RW	**nPOR** 0xF	See *Static Memory Turn Round Delay Registers 0-3* on page 3-37
MPMCStaticConfig2	0x240	RW	**nPOR** 0x-0[a]	See *Static Memory Configuration Registers 0-3* on page 3-31
MPMCStaticWaitWen2	0x244	RW	**nPOR** 0x0	See *Static Memory Write Enable Delay Registers 0-3* on page 3-34
MPMCStaticWaitOen2	0x248	RW	**nPOR** 0x0	See *Static Memory Output Enable Delay Registers 0-3* on page 3-34
MPMCStaticWaitRd2	0x24C	RW	**nPOR** 0x1F	See *Static Memory Read Delay Registers 0-3* on page 3-35

Copyright © 2002-2006 ARM Limited. All rights reserved.

Table 3-1 MPMC register summary (continued)

Name	Offset	Type	Reset value	Description
MPMCStaticWaitPage2	0x250	RW	**nPOR** 0x1F	See *Static Memory Page Mode Read Delay Registers 0-3* on page 3-36
MPMCStaticWaitWr2	0x254	RW	**nPOR** 0x1F	See *Static Memory Write Delay Registers 0-3* on page 3-37
MPMCStaticWaitTurn2	0x258	RW	**nPOR** 0xF	See *Static Memory Turn Round Delay Registers 0-3* on page 3-37
MPMCStaticConfig3	0x260	RW	**nPOR** 0x-0[a]	See *Static Memory Configuration Registers 0-3* on page 3-31
MPMCStaticWaitWen3	0x264	RW	**nPOR** 0x0	See *Static Memory Write Enable Delay Registers 0-3* on page 3-34
MPMCStaticWaitOen3	0x268	RW	**nPOR** 0x0	See *Static Memory Output Enable Delay Registers 0-3* on page 3-34
MPMCStaticWaitRd3	0x26C	RW	**nPOR** 0x1F	See *Static Memory Read Delay Registers 0-3* on page 3-35
MPMCStaticWaitPage3	0x270	RW	**nPOR** 0x1F	See *Static Memory Page Mode Read Delay Registers 0-3* on page 3-36
MPMCStaticWaitWr3	0x274	RW	**nPOR** 0x1F	See *Static Memory Write Delay Registers 0-3* on page 3-37
MPMCStaticWaitTurn3	0x278	RW	**nPOR** 0xF	See *Static Memory Turn Round Delay Registers 0-3* on page 3-37
MPMCITCR	0xF00	RW	**HRESETn** 0x0 **nPOR** 0x0	See *Test Control Register* on page 4-5
MPMCITIP	0xF20	RW	-	See *Test Input Register* on page 4-5
MPMCITOP	0xF40	RW	-	See *Test Output Register* on page 4-8
MPMCPeriphID4	0xFD0	RO	**HRESETn** 0x33 **nPOR** 0x33	See *Conceptual Additional Peripheral Identification Registers* on page 3-38
MPMCPeriphID5	0xFD4	RO	**HRESETn** 0x0 **nPOR** 0x0	See *Conceptual Additional Peripheral Identification Registers* on page 3-38
MPMCPeriphID6	0xFD8	RO	**HRESETn** 0x0 **nPOR** 0x0	See *Conceptual Additional Peripheral Identification Registers* on page 3-38

Copyright © 2002-2006 ARM Limited. All rights reserved.

Table 3-1 MPMC register summary (continued)

Name	Offset	Type	Reset value	Description
MPMCPeriphID7	0xFDC	RO	**HRESETn** 0x0 **nPOR** 0x0	See *Conceptual Additional Peripheral Identification Registers* on page 3-38
MPMCPeriphID0	0xFE0	RO	**HRESETn** 0x72 **nPOR** 0x72	See *Peripheral Identification Registers* on page 3-40
MPMCPeriphID1	0xFE4	RO	**HRESETn** 0x11 **nPOR** 0x11	See *Peripheral Identification Registers* on page 3-40
MPMCPeriphID2	0xFE8	RO	**HRESETn** 0x14 **nPOR** 0x14	See *Peripheral Identification Registers* on page 3-40
MPMCPeriphID3	0xFEC	RO	**HRESETn** 0x07 **nPOR** 0x07	See *Peripheral Identification Registers* on page 3-40
MPMCPCellID0	0xFF0	RO	**HRESETn** 0x0D **nPOR** 0x0D	See *PrimeCell Identification Registers 0-3* on page 3-43
MPMCPCellID1	0xFF4	RO	**HRESETn** 0xF0 **nPOR** 0xF0	See *PrimeCell Identification Registers 0-3* on page 3-43
MPMCPCellID2	0xFF8	RO	**HRESETn** 0x05 **nPOR** 0x05	See *PrimeCell Identification Registers 0-3* on page 3-43
MPMCPCellID3	0xFFC	RO	**HRESETn** 0xB1 **nPOR** 0xB1	See *PrimeCell Identification Registers 0-3* on page 3-43

a. Tie-off dependent.

Copyright © 2002-2006 ARM Limited. All rights reserved.

3.3 Register descriptions

This section describes the MPMC registers.

3.3.1 Control Register

The MPMCControl Register is a three-bit, read/write register that controls the memory controller operation. You can alter the control bits during normal operation. You can access this register with zero wait states. Figure 3-1 shows the register bit assignments.

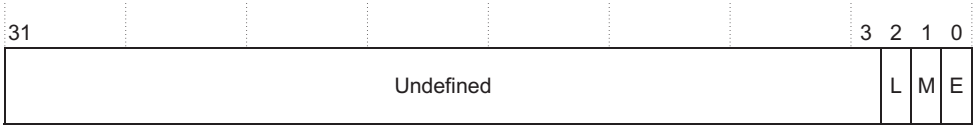

Figure 3-1 MPMCControl Register bit assignments

Table 3-2 lists the register bit assignments.

Table 3-2 MPMCControl Register bit assignments

Bits	Name	Function
[31:3]	-	Read undefined. Write as zero.
[2]	Low-power mode, L	Indicates normal, or low-power mode: 0 = normal mode (reset value on **nPOR**, and **HRESETn**) 1 = low-power mode. Entering low-power mode reduces memory controller power consumption. Dynamic memory is refreshed as necessary. The memory controller returns to normal functional mode by clearing the low-power mode bit (L), or by AHB, or power-on reset. This bit must only be modified when the MPMC is in idle state.[a]
[1]	Address mirror, M	Indicates normal or reset memory map: 0 = normal memory map 1 = reset memory map. Static memory chip select 1 is mirrored onto chip select 0 and chip select 4 (reset value on **nPOR**). On power-on reset, chip select 1 is mirrored to both chip select 0 and chip select 1 and chip select 4 memory areas. Clearing the M bit enables chip select 0 and chip select 4 memory to be accessed.

Copyright © 2002-2006 ARM Limited. All rights reserved.

Table 3-2 MPMCControl Register bit assignments (continued)

Bits	Name	Function
[0]	MPMC Enable, E	Indicates if the MPMC is enabled or disabled: 0 = disabled 1 = enabled (reset value on **nPOR** and **HRESETn**). Disabling the MPMC reduces power consumption. When the memory controller is disabled the memory is not refreshed. The memory controller is enabled by setting the enable bit, or by AHB, or power-on reset. This bit must only be modified when the MPMC is in idle state.[a]

a. The external memory cannot be accessed in low-power or disabled state. If a memory access is performed an error response is generated. The memory controller AHB register programming port can be accessed normally. You can program the MPMC registers in low-power and/or disabled state.

3.3.2 Status Register

The three-bit read-only MPMCStatus Register provides MPMC status information. This register is accessed with zero wait states.

Figure 3-2 shows the register bit assignments.

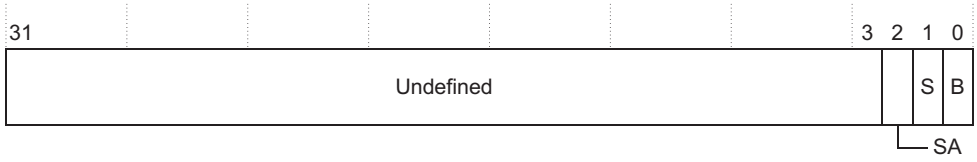

Figure 3-2 MPMCStatus Register bit assignments

Table 3-3 lists the register bit assignments.

Table 3-3 MPMCStatus Register bit assignments

Bits	Name	Function
[31:3]	-	Read undefined.

Copyright © 2002-2006 ARM Limited. All rights reserved.

Table 3-3 MPMCStatus Register bit assignments (continued)

Bits	Name	Function
[2]	Self-refresh acknowledge, **MPMCSREFACK**, SA	This bit indicates the operating mode of the MPMC: 0 = normal mode 1 = self-refresh mode (reset value on **nPOR**).
[1]	Write buffer status, S	This bit enables the MPMC to enter low-power mode or disabled mode cleanly: 0 = write buffers empty (reset value on **nPOR**) 1 = write buffers contain data.
[0]	Busy, B	This bit ensures that the memory controller enters the low-power or disabled mode cleanly by determining if the memory controller is busy or not: 0 = MPMC is idle (reset value on **HRESETn**) 1 = MPMC is busy performing memory transactions, commands, auto-refresh cycles, or is in self-refresh mode (reset value on **nPOR** and **HRESETn**).

3.3.3 Configuration Register

The two-bit, read/write, MPMCConfig Register configures the operation of the memory controller. It is recommended that you modify this register during system initialization, or when there are no current or outstanding transactions. You can ensure this by waiting until the MPMC is idle, and then entering low-power, or disabled mode. This register is accessed with one wait state.

Figure 3-3 shows the register bit assignments.

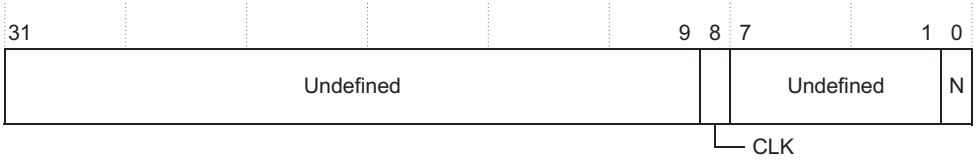

Figure 3-3 MPMCConfig Register bit assignments

Copyright © 2002-2006 ARM Limited. All rights reserved.

Table 3-4 lists the register bit assignments.

Table 3-4 MPMCConfig Register bit assignments

Bits	Name	Function
[31:9]	-	Read undefined. Write as zero.
[8]	Clock ratio, CLK	**HCLK**:**MPMCCLKOUT[3:0]** ratio: 0 = 1:1 (reset value on **nPOR**) 1 = 1:2.[a]
[7:1]	-	Read undefined. Write as zero.
[0]	Endian mode, N	Endian mode: 0 = little-endian mode 1 = big-endian mode. The **MPMCBIGENDIAN** signal determines the value of the endian bit on power-on reset, **nPOR**. Software can override this value. This field is unaffected by the AHB reset, **HRESETn**.[b]

a. While you can configure the MPMC in software to use a 1:2 clock ratio it is designed to be synthesized with a clock ratio of 1:1. It does not support a ratio of 1:2 **HCLK**:**MPMCCLK** for synthesis. If you require a ratio of 1:2 **HCLK**:**MPMCCLK**, you must use a clock ratio of 1:1 for synthesis and then you must program the MPMCConfig register to 1:2.

b. The value of the **MPMCBIGENDIAN** signal is reflected in this field. When programmed this register reflects the last value that is written into it. You must flush all data in the MPMC before switching between little-endian and big-endian modes.

3.3.4 Dynamic Memory Control Register

The nine-bit, read/write, MPMCDynamicControl Register controls dynamic memory operation. You can alter the control bits during normal operation. This register is accessed with zero wait states.

Figure 3-4 on page 3-12 shows the register bit assignments.

Copyright © 2002-2006 ARM Limited. All rights reserved.

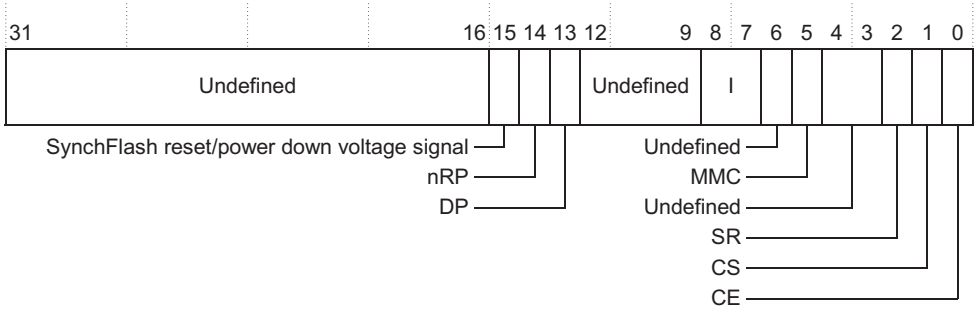

Figure 3-4 MPMCDynamicControl Register bit assignments

Table 3-5 lists the register bit assignments.

Table 3-5 MPMCDynamicControl Register bit assignments

Bits	Name	Function
[31:16]	-	Read undefined. Write as zero.
[15]	SyncFlash reset/power down voltage signal, **MPMCRPVHHOUT**	0 = normal voltage (reset value on **nPOR**) 1 = set **MPMCRPVHHOUT** high voltage. You can use this output externally in conjunction with the **nMPMCRPOUT** signal to indicate a high output voltage, see Table 3-6 on page 3-13.
[14]	SyncFlash reset/power down signal, **nMPMCRPOUT**, nRP	0 = **nMPMCRPOUT** signal LOW (reset value on **nPOR**) 1 = set **nMPMCRPOUT** signal HIGH.
[13]	Low-power SDRAM deep-sleep mode, DP,	0 = normal operation (reset value on **nPOR**) 1 = enter deep power down mode.
[12:9]	-	Read undefined. Write as zero.
[8:7]	SDRAM initialization, I	00 = issue SDRAM NORMAL operation command (reset value on **nPOR**) 01 = issue SDRAM MODE command 10 = issue SDRAM PALL (precharge all) command 11 = issue SDRAM NOP (no operation) command).
[6]	-	Read undefined. Write as zero.
[5]	Memory clock control, MMC	0 = **MPMCCLKOUT** enabled (reset value on **nPOR**) 1 = **MPMCCLKOUT** disabled. [a]
[4:3]	-	Read undefined. Write as zero.

Copyright © 2002-2006 ARM Limited. All rights reserved.

Table 3-5 MPMCDynamicControl Register bit assignments (continued)

Bits	Name	Function
[2]	Self-refresh request, **MPMCSREFREQ**, SR	0 = normal mode 1 = enter self-refresh mode (reset value on **nPOR**). By writing 1 to this bit self-refresh can be entered under software control. Writing 0 to this bit returns the MPMC to normal mode. The self-refresh acknowledge bit in the MPMCStatus Register must be polled to discover the current operating mode of the MPMC.[b]
[1]	Dynamic memory clock control, CS	0 = **MPMCCLKOUT** stops when all SDRAMs are idle and during self-refresh mode. 1 = **MPMCCLKOUT** runs continuously (reset value on **nPOR**). When clock control is LOW the output clock **MPMCCLKOUT** is stopped when there are no SDRAM transactions. The clock is also stopped during self-refresh mode.
[0]	Dynamic memory clock enable, CE	0 = clock enable of idle devices are deasserted to save power (reset value on **nPOR**) 1 = all clock enables are driven HIGH continuously.[c]

a. You can disable **MPMCCLKOUT** if there are no SDRAM memory transactions. When enabled you can use this field in conjunction with the dynamic memory clock control (CS) field.
b. The memory controller exits from power-on reset with the self-refresh bit on HIGH. To enter normal functional mode set this bit LOW. Writing to this register with a HIGH places this register into self-refresh mode. This functionality enables data to be stored over SDRAM self-refresh if the ASIC is powered down.
c. Clock enable must be HIGH during SDRAM initialization.

Table 3-6 shows the output voltage settings for different combinations of bits [15:14].

A block external to the MPMC can use the values of the **nMPMCRPOUT** and **MPMCRPVHHOUT** signals to generate the required voltage settings for the Micron SyncFlash reset signal. Some systems do not have an 8V output, or do not require the functionality to raise **nRP** to 8V. In these cases **nMPMCRPVHHOUT** can be ignored.

Table 3-6 Output voltage settings

nMPMCRPOUT	MPMCRPVHHOUT	Output
0	0	0V
0	1	0V
1	0	3V
1	1	8V

Copyright © 2002-2006 ARM Limited. All rights reserved.

3.3.5 Dynamic Memory Refresh Timer Register

The 11-bit, read/write, MPMCDynamicRefresh Register configures dynamic memory operation. It is recommended that you modify this register during system initialization, or when there are no current or outstanding transactions. You can ensure this by waiting until the MPMC is idle, and then entering low-power, or disabled mode. However, these control bits can, if necessary, be altered during normal operation. This register is accessed with one wait state.

Note

This register is used for all four dynamic memory chip selects. Therefore you must program the worse case value for all of the chip selects.

Figure 3-5 shows the register bit assignments.

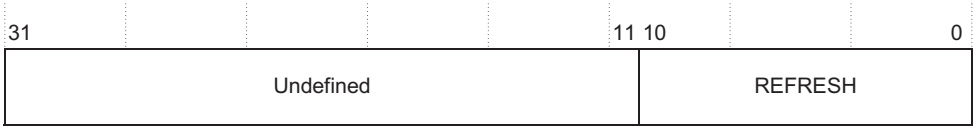

Figure 3-5 MPMCDynamicRefresh Register bit assignments

Table 3-7 lists the register bit assignments.

Table 3-7 MPMCDynamicRefresh Register bit assignments

Bits	Name	Function
[31:11]	-	Read undefined. Write as zero.
[10:0]	Refresh timer, REFRESH	`0x0` = refresh disabled (reset value on **nPOR**) `0x1` 1(x16) = 16 **HCLK** ticks between SDRAM refresh cycles `0x8` 8(x16) = 128 **HCLK** ticks between SDRAM refresh cycles `0x1-0x7FF` n(x16) = 16n **HCLK** ticks between SDRAM refresh cycles.

For example, for the refresh period of 16µs, and an **HCLK** frequency of 50MHz, you must program the following value into this register:

$$\frac{16 \times 10^{-6} \times 50 \times 10^{6}}{16} = 50 \text{ or } 0x32$$

Copyright © 2002-2006 ARM Limited. All rights reserved.

Note

- The refresh cycles are evenly distributed. However, there might be slight variations when the auto-refresh command is issued depending on the status of the memory controller.
- Unlike other SDRAM memory timing parameters the refresh period is programmed in the **HCLK** domain.

3.3.6 Dynamic Memory Read Configuration Register

The two-bit, read/write, MPMCDynamicReadConfig Register enables you to configure the dynamic memory read strategy. This register must only be modified during system initialization. This register is accessed with one wait state.

Note

This register is used for all four dynamic memory chip selects. Therefore the worse case value for all of the chip selects must be programmed.

Figure 3-6 shows the register bit assignments.

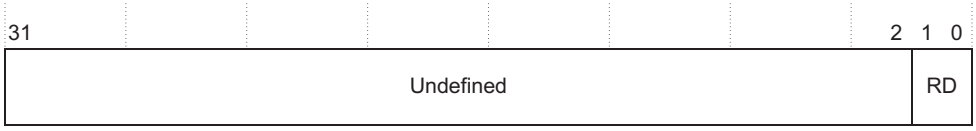

Figure 3-6 MPMCDynamicReadConfig Register bit assignments

Copyright © 2002-2006 ARM Limited. All rights reserved.

Table 3-8 lists the register bit assignments.

Table 3-8 MPMCDynamicReadConfig Register bit assignments

Bits	Name	Function
[31:2]	-	Read undefined. Write as zero.
[1:0]	Read data strategy, RD	00 = clock out delayed strategy, using **MPMCCLKOUT** (command not delayed, clock out delayed). Reset value on **nPOR**. 01 = command delayed strategy, using **MPMCCLKDELAY** (command delayed, clock out not delayed). 10 = command delayed strategy plus one clock cycle, using **MPMCCLKDELAY** (command delayed, clock out not delayed) 11 = command delayed strategy plus two clock cycles, using **MPMCCLKDELAY** (command delayed, clock out not delayed).

3.3.7 Dynamic Memory Precharge Command Period Register

The four-bit, read/write, MPMCDynamictRP Register enables you to program the precharge command period, t_{RP}. This register must only be modified during system initialization. This value is normally found in SDRAM data sheets as t_{RP}. This register is accessed with one wait state.

Note

This register is used for all four dynamic memory chip selects. Therefore the worse case value for all of the chip selects must be programmed.

Figure 3-7 shows the register bit assignments.

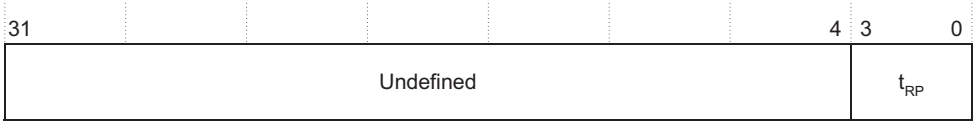

Figure 3-7 MPMCDynamictRP Register bit assignments

Copyright © 2002-2006 ARM Limited. All rights reserved.

Table 3-9 lists the register bit assignments.

Table 3-9 MPMCDynamictRP Register bit assignments

Bits	Name	Function
[31:4]	-	Read undefined. Write as zero.
[3:0]	Precharge command period, t_{RP}	0x0-0xE = n + 1 clock cycles[a] 0xF = 16 clock cycles (reset value on **nPOR**)

a. The delay is in **MPMCCLK** cycles.

3.3.8 Dynamic Memory Active To Precharge Command Period Register

The four-bit, read/write, MPMCDynamictRAS Register enables you to program the active to precharge command period, t_{RAS}. It is recommended that you modify this register during system initialization, or when there are no current or outstanding transactions. You can ensure this by waiting until the MPMC is idle, and then entering low-power, or disabled mode. This value is normally found in SDRAM data sheets as t_{RAS}. This register is accessed with one wait state.

Note

This register is used for all four dynamic memory chip selects. Therefore you must program the worse case value for all of the chip selects.

Figure 3-8 shows the register bit assignments.

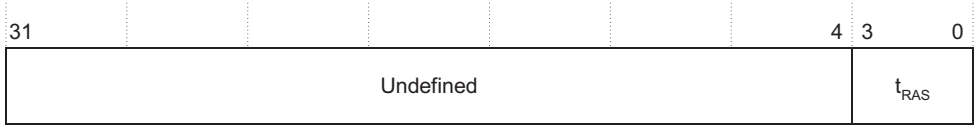

Figure 3-8 MPMCDynamictRAS Register bit assignments

Copyright © 2002-2006 ARM Limited. All rights reserved.

Table 3-10 lists the register bit assignments.

Table 3-10 MPMCDynamictRAS Register bit assignments

Bits	Name	Function
[31:4]	-	Read undefined. Write as zero.
[3:0]	Active to precharge command period, t_{RAS}	0x0-0xE = n + 1 clock cycles[a] 0xF = 16 clock cycles (reset value on **nPOR**)

a. The delay is in **MPMCCLK** cycles.

3.3.9 Dynamic Memory Self-refresh Exit Time Register

The four-bit, read/write, MPMCDynamictSREX Register enables you to program the self-refresh exit time, t_{SREX}. It is recommended that you modify this register during system initialization, or when there are no current or outstanding transactions. You can ensure this by waiting until the MPMC is idle, and then entering low-power, or disabled mode. This value is normally found in SDRAM data sheets as t_{SREX}, for devices without this parameter you use the same value as t_{XSR}. This register is accessed with one wait state.

Note

This register is used for all four dynamic memory chip selects. Therefore you must program the worse case value for all of the chip selects.

Figure 3-9 shows the register bit assignments.

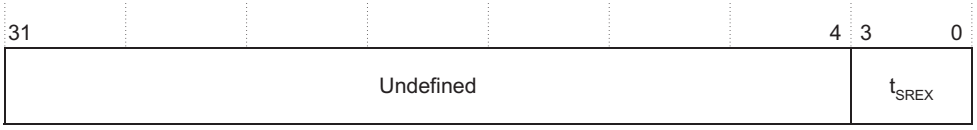

Figure 3-9 MPMCDynamictSREX Register bit assignments

Copyright © 2002-2006 ARM Limited. All rights reserved.

Table 3-11 lists the register bit assignments.

Table 3-11 MPMCDynamictSREX Register bit assignments

Bits	Name	Function
[31:4]	-	Read undefined. Write as zero.
[3:0]	Self-refresh exit time, t_{SREX}	`0x0-0xE` = n + 1 clock cycles[a] `0xF` = 16 clock cycles (reset value on **nPOR**)

a. The delay is in **MPMCCLK** cycles.

3.3.10 Dynamic Memory Last Data Out To Active Time Register

The four-bit, read/write, MPMCDynamictAPR Register enables you to program the last-data-out to active command time, t_{APR}. It is recommended that you modify this register during system initialization, or when there are no current or outstanding transactions. You can ensure this by waiting until the MPMC is idle, and then entering low-power, or disabled mode. This value is normally found in SDRAM data sheets as t_{APR}. This register is accessed with one wait state.

Note

This register is used for all four dynamic memory chip selects. Therefore the worse case value for all of the chip selects must be programmed.

Figure 3-10 shows the register bit assignments.

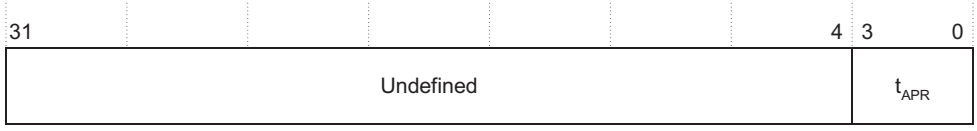

Figure 3-10 MPMCDynamictAPR Register bit assignments

Copyright © 2002-2006 ARM Limited. All rights reserved.

Table 3-12 lists the register bit assignments.

Table 3-12 MPMCDynamictAPR Register bit assignments

Bits	Name	Function
[31:4]	-	Read undefined. Write as zero.
[3:0]	Last-data-out to active command time, t_{APR}	0x0-0xE = n + 1 clock cycles[a] 0xF = 16 clock cycles (reset value on **nPOR**)

a. The delay is in **MPMCCLK** cycles.

3.3.11 Dynamic Memory Data-in To Active Command Time Register

The four-bit, read/write, MPMCDynamictDAL Register enables you to program the data-in to active command time, t_{DAL}. It is recommended that you modify this register during system initialization, or when there are no current or outstanding transactions. You can ensure this by waiting until the MPMC is idle, and then entering low-power, or disabled mode. This value is normally found in SDRAM data sheets as t_{DAL}, or t_{APW}. This register is accessed with one wait state.

Note

This register is used for all four dynamic memory chip selects. Therefore the worse case value for all of the chip selects must be programmed.

Figure 3-10 on page 3-19 shows the register bit assignments.

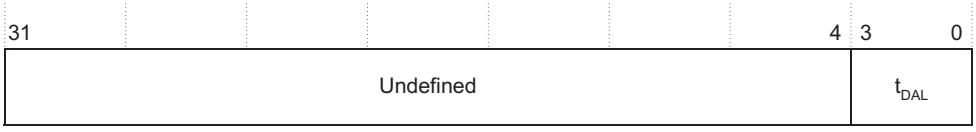

Figure 3-11 MPMCDynamictDAL Register bit assignments

Table 3-13 lists the register bit assignments.

Table 3-13 MPMCDynamictDAL Register bit assignments

Bits	Name	Function
[31:4]	-	Read undefined. Write as zero.
[3:0]	Data-in to active command, t_{DAL}	0x0-0xE = n clock cycles[a] 0xF = 15 clock cycles (reset value on **nPOR**)

Copyright © 2002-2006 ARM Limited. All rights reserved.

a. The delay is in **MPMCCLK** cycles.

3.3.12 Dynamic Memory Write Recovery Time Register

The four-bit, read/write, MPMCDynamictWR Register enables you to program the write recovery time, t_{WR}. It is recommended that you modify this register during system initialization, or when there are no current or outstanding transactions. You can ensure this by waiting until the MPMC is idle, and then entering low-power, or disabled mode. This value is normally found in SDRAM data sheets as t_{WR}, t_{DPL}, t_{RWL}, or t_{RDL}. This register is accessed with one wait state.

Note

This register is used for all four dynamic memory chip selects. Therefore the worse case value for all of the chip selects must be programmed.

Figure 3-12 shows the register bit assignments.

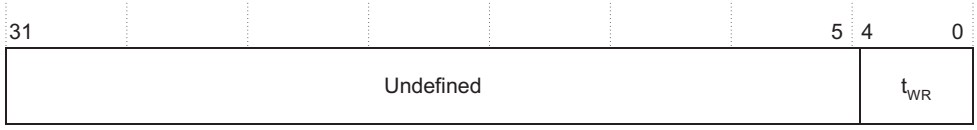

Figure 3-12 MPMCDynamictWR Register bit assignments

Table 3-14 lists the register bit assignments.

Table 3-14 MPMCDynamictWR Register bit assignments

Bits	Name	Function
[31:4]	-	Read undefined. Write as zero.
[3:0]	Write recovery time, t_{WR}	`0x0-0xE` = n + 1 clock cycles[a] `0xF` = 16 clock cycles (reset value on **nPOR**)

a. The delay is in **MPMCCLK** cycles.

3.3.13 Dynamic Memory Active To Active Command Period Register

The five-bit, read/write, MPMCDynamictRC Register enables you to program the active to active command period, t_{RC}. It is recommended that you modify this register during system initialization, or when there are no current or outstanding transactions.

Copyright © 2002-2006 ARM Limited. All rights reserved.

You can ensure this by waiting until the MPMC is idle, and then entering low-power, or disabled mode. This value is normally found in SDRAM data sheets as t_{RC}. This register is accessed with one wait state.

Note

This register is used for all four dynamic memory chip selects. Therefore the worse case value for all of the chip selects must be programmed.

Figure 3-13 shows the register bit assignments.

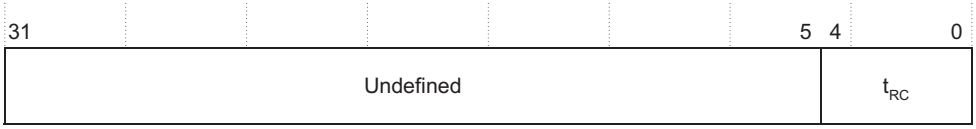

Figure 3-13 MPMCDynamictRC Register bit assignments

Table 3-15 lists the register bit assignments.

Table 3-15 MPMCDynamictRC Register bit assignments

Bits	Name	Function
[31:5]	-	Read undefined. Write as zero.
[4:0]	Active to active command period, t_{RC}	`0x0-0x1E` = n + 1 clock cycles[a] `0x1F` = 32 clock cycles (reset value on **nPOR**)

a. The delay is in **MPMCCLK** cycles.

3.3.14 Dynamic Memory Auto-refresh Period Register

The five-bit, read/write, MPMCDynamictRFC Register enables you to program the auto-refresh period, and auto-refresh to active command period, t_{RFC}. It is recommended that you modify this register during system initialization, or when there are no current or outstanding transactions. You can ensure this by waiting until the MPMC is idle, and then entering low-power, or disabled mode. This value is normally found in SDRAM data sheets as t_{RFC}, or sometimes as t_{RC}. This register is accessed with one wait state.

Note

This register is used for all four dynamic memory chip selects. Therefore the worse case value for all of the chip selects must be programmed.

Copyright © 2002-2006 ARM Limited. All rights reserved.

Figure 3-14 shows the register bit assignments.

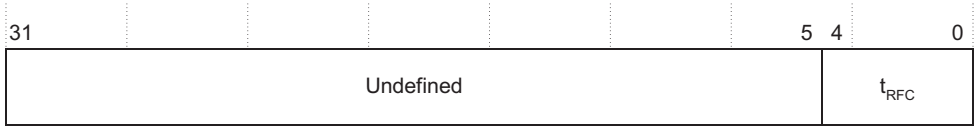

Figure 3-14 MPMCDynamictRFC Register bit assignments

Table 3-16 lists the register bit assignments.

Table 3-16 MPMCDynamictRFC Register bit assignments

Bits	Name	Function
[31:5]	-	Read undefined. Write as zero.
[4:0]	Auto-refresh period and auto-refresh to active command period, t_{RFC}	`0x0-0x1E` = n + 1 clock cycles[a] `0x1F` = 32 clock cycles (reset value on **nPOR**)

a. The delay is in **MPMCCLK** cycles.

3.3.15 Dynamic Memory Exit Self-refresh Register

The five-bit, read/write, MPMCDynamictXSR Register enables you to program the exit self-refresh to active command time, t_{XSR}. It is recommended that you modify this register during system initialization, or when there are no current or outstanding transactions. You can ensure this by waiting until the MPMC is idle, and then entering low-power, or disabled mode. This value is normally found in SDRAM data sheets as t_{XSR}. This register is accessed with one wait state.

Note

This register is used for all four dynamic memory chip selects. Therefore the worse case value for all of the chip selects must be programmed.

Figure 3-15 shows the register bit assignments.

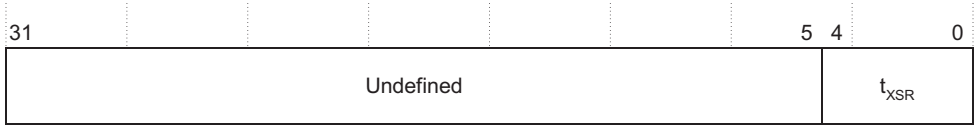

Figure 3-15 MPMCDynamictXSR Register bit assignments

Copyright © 2002-2006 ARM Limited. All rights reserved.

Table 3-17 lists the register bit assignments.

Table 3-17 MPMCDynamictXSR Register bit assignments

Bits	Name	Function
[31:5]	-	Read undefined. Write as zero.
[4:0]	Exit self-refresh to active command time, t_{XSR}	0x0-0x1E = n + 1 clock cycles[a] 0x1F = 32 clock cycles (reset value on **nPOR**)

a. The delay is in **MPMCCLK** cycles.

3.3.16 Dynamic Memory Active Bank A to Active Bank B Time Register

The four-bit, read/write, MPMCDynamictRRD Register enables you to program the active bank A to active bank B latency, t_{RRD}. It is recommended that you modify this register during system initialization, or when there are no current or outstanding transactions. You can ensure this by waiting until the MPMC is idle, and then entering low-power, or disabled mode. This value is normally found in SDRAM data sheets as t_{RRD}. This register is accessed with one wait state.

Note

This register is used for all four dynamic memory chip selects. Therefore the worse case value for all of the chip selects must be programmed.

Figure 3-16 shows the register bit assignments.

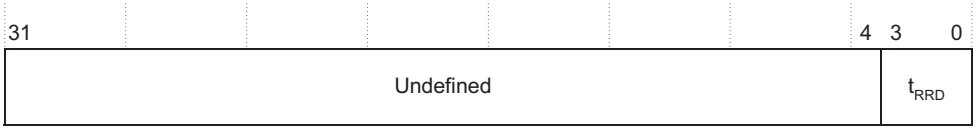

Figure 3-16 MPMCDynamictRRD Register bit assignments

Table 3-18 lists the register bit assignments.

Table 3-18 MPMCDynamictRRD Register bit assignments

Bits	Name	Function
[31:4]	-	Read undefined. Write as zero.
[3:0]	Active bank A to active bank B latency, t_{RRD}	0x0-0xE = n + 1 clock cycles[a] 0xF = 16 clock cycles (reset value on **nPOR**)

Copyright © 2002-2006 ARM Limited. All rights reserved.

a. The delay is in **MPMCCLK** cycles.

3.3.17 Dynamic Memory Load Mode Register To Active Command Time Register

The four-bit, read/write, MPMCDynamictMRD Register enables you to program the load mode register to active command time, t_{MRD}. It is recommended that you modify this register during system initialization, or when there are no current or outstanding transactions. You can ensure this by waiting until the MPMC is idle, and then entering low-power, or disabled mode. This value is normally found in SDRAM data sheets as t_{MRD}, or t_{RSA}. This register is accessed with one wait state.

Note

This register is used for all four dynamic memory chip selects. Therefore the worse case value for all of the chip selects must be programmed.

Figure 3-17 shows the register bit assignments.

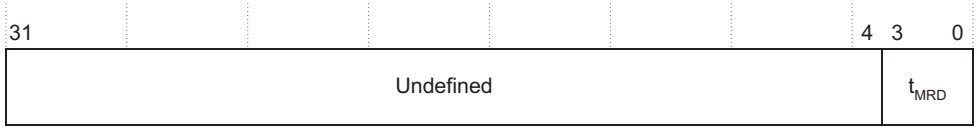

Figure 3-17 MPMCDynamictMRD Register bit assignments

Table 3-19 lists the register bit assignments.

Table 3-19 MPMCDynamictMRD Register bit assignments

Bits	Name	Function
[31:4]	-	Read undefined. Write as zero.
[3:0]	Load mode register to active command time, t_{MRD}	`0x0-0xE` = n + 1 clock cycles[a] `0xF` = 16 clock cycles (reset value on **nPOR**)

a. The delay is in **MPMCCLK** cycles.

3.3.18 Static Memory Extended Wait Register

The 10-bit, read/write, MPMCStaticExtendedWait Register times long static memory read and write transfers, that are longer than can be supported by the MPMCStaticWaitRd[n] or MPMCStaticWaitWr[n] Registers, when the EW bit of the MPMCStaticConfig Registers is enabled. There is only a single MPMCStaticExtendedWait Register. This is used by the relevant static memory chip

Copyright © 2002-2006 ARM Limited. All rights reserved.

select if the appropriate *ExtendedWait* (EW) bit in the MPMCStaticConfig Register is set. It is recommended that this register is modified during system initialization, or when there are no current or outstanding transactions. However, if necessary, you can alter the control bits during normal operation. This register is accessed with one wait state.

Figure 3-18 shows the register bit assignments.

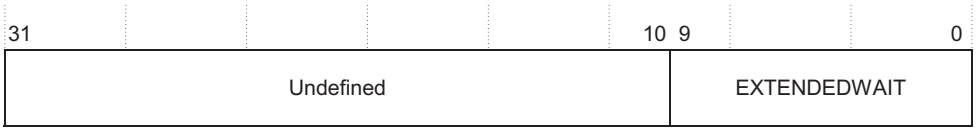

Figure 3-18 MPMCStaticExtendedWait Register bit assignments

Table 3-20 lists the register bit assignments.

Table 3-20 MPMCStaticExtendedWait Register bit assignments

Bits	Name	Function
[31:10]	-	Read undefined. Write as zero.
[9:0]	External wait time out, EXTENDEDWAIT	`0x0` = 16 clock cycles[a] (reset value on **nPOR**) `0x1-0x3FF` = (n+1) x16 clock cycles

a. The delay is in **HCLK** cycles.

For example, for a static memory read/write transfer time of 16µs, and an **MPMCCLK** frequency of 50MHz, the following value must be programmed into this register:

$$\frac{16 \times 10^{-6} \times 50 \times 10^{6}}{16} - 1 = 49$$

3.3.19 Dynamic Memory Configuration Registers 0-3

The 11-bit, read/write, MPMCDynamicConfig0-3 Registers enable you to program the configuration information for the relevant dynamic memory chip select. These registers is normally only modified during system initialization. These registers are accessed with one wait state.

Figure 3-19 on page 3-27 shows the register bit assignments.

 Copyright © 2002-2006 ARM Limited. All rights reserved.

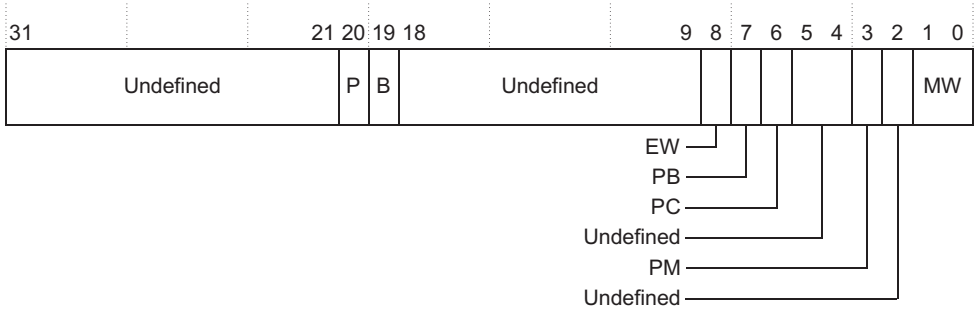

Figure 3-19 MPMCDynamicConfig0-3 Registers bit assignments

Table 3-21 lists the register bit assignments.

Table 3-21 MPMCDynamicConfig0-3 Registers bit assignments

Bits	Name	Function
[31:21]	-	Read undefined. Write as zero.
[20]	Write protect, P	0 = writes not protected (reset value on **nPOR**) 1 = write protected.
[19]	Buffer enable, B	0 = buffer disabled for accesses to this chip select (reset value on **nPOR**) 1 = buffer enabled for accesses to this chip select.[a]
[18:15]	-	Read undefined. Write as zero.
[14]	Address mapping, AM	See Table 3-22 on page 3-28. 0 = reset value on **nPOR**.
[13]	-	Read undefined. Write as zero.
[12:7]	Address mapping, AM	See Table 3-22 on page 3-28. 00000000 = reset value on **nPOR**.[b]
[6:5]	-	Read undefined. Write as zero.
[4:3]	Memory device, MD	00 = SDRAM (reset value on **nPOR**) 01 = low-power SDRAM 10 = Micron SyncFlash 11 = reserved.
[2:0]	-	Read undefined. Write as zero.

a. The buffers must be disabled during SDRAM and SyncFlash initialization. They must also be disabled when performing SyncFlash commands. The buffers must be enabled during normal operation.
b. The SDRAM column and row width and number of banks are computed automatically from the address mapping.

Copyright © 2002-2006 ARM Limited. All rights reserved.

Table 3-22 shows address mapping. Address mappings that are not shown in Table 3-22 are reserved.

Table 3-22 Address mapping

[14]	[12]	[11:9]	[8:7]	Description
16-bit external bus high-performance address mapping (Row, Bank, Column)				
0	0	000	00	16Mb (2Mx8), 2 banks, row length = 11, column length = 9
0	0	000	01	16Mb (1Mx16), 2 banks, row length = 11, column length = 8
0	0	001	00	64Mb (8Mx8), 4 banks, row length = 12, column length = 9
0	0	001	01	64Mb (4Mx16), 4 banks, row length = 12, column length = 8
0	0	010	00	128Mb (16Mx8), 4 banks, row length = 12, column length = 10
0	0	010	01	128Mb (8Mx16), 4 banks, row length = 12, column length = 9
0	0	011	00	256Mb (32Mx8), 4 banks, row length = 13, column length = 10
0	0	011	01	256Mb (16Mx16), 4 banks, row length = 13, column length = 9
0	0	100	00	512Mb (64Mx8), 4 banks, row length = 13, column length = 11
0	0	100	01	512Mb (32Mx16), 4 banks, row length = 13, column length = 10
16-bit external bus low-power SDRAM address mapping (Bank, Row, Column)				
0	1	000	00	16Mb (2Mx8), 2 banks, row length = 11, column length = 9
0	1	000	01	16Mb (1Mx16), 2 banks, row length = 11, column length = 8
0	1	001	00	64Mb (8Mx8), 4 banks, row length = 12, column length = 9
0	1	001	01	64Mb (4Mx16), 4 banks, row length = 12, column length = 8
0	1	010	00	128Mb (16Mx8), 4 banks, row length = 12, column length = 10
0	1	010	01	128Mb (8Mx16), 4 banks, row length = 12, column length = 9
0	1	011	00	256Mb (32Mx8), 4 banks, row length = 13, column length = 10
0	1	011	01	256Mb (16Mx16), 4 banks, row length = 13, column length = 9
0	1	100	00	512Mb (64Mx8), 4 banks, row length = 13, column length = 11
0	1	100	01	512Mb (32Mx16), 4 banks, row length = 13, column length = 10
32-bit external bus high-performance address mapping (Row, Bank, Column)				

Copyright © 2002-2006 ARM Limited. All rights reserved.

Table 3-22 Address mapping (continued)

[14]	[12]	[11:9]	[8:7]	Description
1	0	000	00	16Mb (2Mx8), 2 banks, row length = 11, column length = 9
1	0	000	01	16Mb (1Mx16), 2 banks, row length = 11, column length = 8
1	0	001	00	64Mb (8Mx8), 4 banks, row length = 12, column length = 9
1	0	001	01	64Mb (4Mx16), 4 banks, row length = 12, column length = 8
1	0	001	10	64Mb (2Mx32), 4 banks, row length = 11, column length = 8
1	0	010	00	128Mb (16Mx8), 4 banks, row length = 12, column length = 10
1	0	010	01	128Mb (8Mx16), 4 banks, row length = 12, column length = 9
1	0	010	10	128Mb (4Mx32), 4 banks, row length = 12, column length = 8
1	0	011	00	256Mb (32Mx8), 4 banks, row length = 13, column length = 10
1	0	011	01	256Mb (16Mx16), 4 banks, row length = 13, column length = 9
1	0	011	10	256Mb (8Mx32), 4 banks, row length = 13, column length = 8
1	0	100	00	512Mb (64Mx8), 4 banks, row length = 13, column length = 11
1	0	100	01	512Mb (32Mx16), 4 banks, row length = 13, column length = 10
32-bit external bus low-power SDRAM address mapping (Bank, Row, Column)				
1	1	000	00	16Mb (2Mx8), 2 banks, row length = 11, column length = 9
1	1	000	01	16Mb (1Mx16), 2 banks, row length = 11, column length = 8
1	1	001	00	64Mb (8Mx8), 4 banks, row length = 12, column length = 9
1	1	001	01	64Mb (4Mx16), 4 banks, row length = 12, column length = 8
1	1	001	10	64Mb (2Mx32), 4 banks, row length = 11, column length = 8
1	1	010	00	128Mb (16Mx8), 4 banks, row length = 12, column length = 10
1	1	010	01	128Mb (8Mx16), 4 banks, row length = 12, column length = 9
1	1	010	10	128Mb (4Mx32), 4 banks, row length = 12, column length = 8
1	1	011	00	256Mb (32Mx8), 4 banks, row length = 13, column length = 10
1	1	011	01	256Mb (16Mx16), 4 banks, row length = 13, column length = 9

Copyright © 2002-2006 ARM Limited. All rights reserved.

Table 3-22 Address mapping (continued)

[14]	[12]	[11:9]	[8:7]	Description
1	1	011	10	256Mb (8Mx32), 4 banks, row length = 13, column length = 8
1	1	100	00	512Mb (64Mx8), 4 banks, row length = 13, column length = 11
1	1	100	01	512Mb (32Mx16), 4 banks, row length = 13, column length = 10

You can connect a chip select to a single memory device. In this case the chip select data bus width is the same as the device width. Alternatively you can connect the chip select to a number of external devices. In this case the chip select data bus width is the sum of the memory device databus widths.

For example, for a chip select connected to:

- a 32-bit wide memory device, choose a 32-bit wide address mapping
- a 16-bit wide memory device, choose a 16-bit wide address mapping
- four x 8-bit wide memory devices, choose a 32-bit wide address mapping
- two x 8-bit wide memory devices, choose a 16-bit wide address mapping.

3.3.20 Dynamic Memory RAS and CAS Delay Registers 0-3

The four-bit, read/write, MPMCDynamicRasCas0-3 Registers enable you to program the RAS and CAS latencies for the relevant dynamic memory. It is recommended that you modify this register during system initialization, or when there are no current or outstanding transactions. You can ensure this by waiting until the MPMC is idle, and then entering low-power, or disabled mode. These registers are accessed with one wait state.

Note

The values programmed into these registers must be consistent with the values used to initialize the SDRAM memory device.

Figure 3-20 shows the register bit assignments.

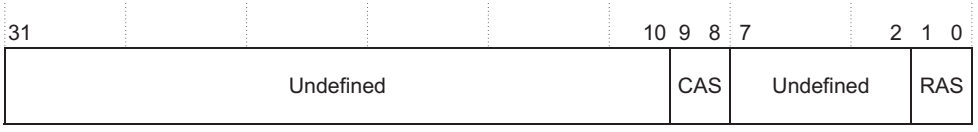

Figure 3-20 MPMCDynamicRasCas0-3 Registers bit assignments

Copyright © 2002-2006 ARM Limited. All rights reserved.

Table 3-23 lists the register bit assignments.

Table 3-23 MPMCDynamicRasCas0-3 Registers bit assignments

Bits	Name	Function
[31:10]	-	Read undefined. Write as zero.
[9:8]	CAS latency, CAS	00 = reserved 01 = one clock cycle[a] 10 = two clock cycles 11 = three clock cycles (reset value on **nPOR**)
[7:2]	-	Read undefined. Write as zero.
[1:0]	RAS latency (active to read/write delay), RAS	00 = reserved 01 = one clock cycle[a] 10 = two clock cycles 11 = three clock cycles (reset value on **nPOR**)

a. The RAS to CAS latency (RAS) and CAS latency (CAS) are both defined in **MPMCCLK** cycles.

3.3.21 Static Memory Configuration Registers 0-3

The eight-bit, read/write, MPMCStaticConfig0-3 Registers configure the static memory configuration. It is recommended that you modify this register during system initialization, or when there are no current or outstanding transactions. You can ensure this by waiting until the MPMC is idle, and then entering low-power, or disabled mode. These registers are accessed with one wait state.

Figure 3-21 shows the register bit assignments.

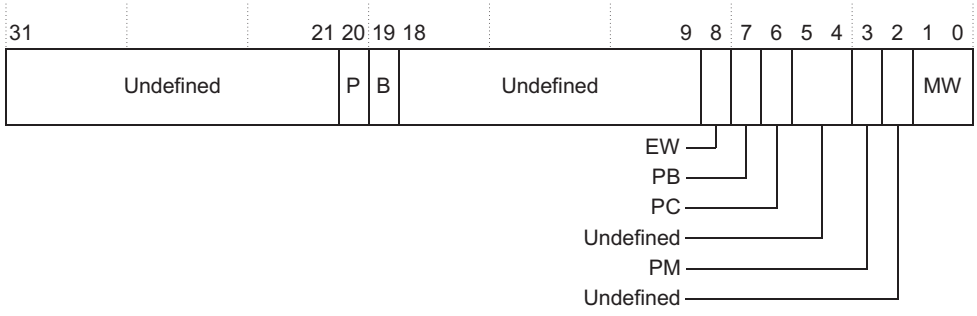

Figure 3-21 MPMCStaticConfig0-3 Registers bit assignments

Copyright © 2002-2006 ARM Limited. All rights reserved.

Table 3-24 lists the register bit assignments.

Table 3-24 MPMCStaticConfig0-3 Registers bit assignments

Bits	Name	Function
[31:21]	-	Read undefined. Write as zero.
[20]	Write protect, P	0 = writes not protected (reset value on **nPOR**) 1 = write protected.
[19]	Buffer enable, B	0 = write buffer disabled (reset value on **nPOR**) 1 = write buffer enabled.
[18:9]	-	Read undefined. Write as zero.
[8]	Extended wait, EW	0 = Extended wait disabled (reset value on **nPOR**) 1 = Extended wait enabled. Extended wait (EW) uses the MPMCStaticExtendedWait Register to time both the read and write transfers rather than the MPMCStaticWaitRd and MPMCStaticWaitWr Registers. This enables much longer transactions.[a]
[7]	Byte lane state, PB	0 = For reads all the bits in **nMPMCBLSOUT[3:0]** are HIGH. For writes the respective active bits in **nMPMCBLSOUT[3:0]** are LOW, reset value for chip select 0, 2, and 3 on **nPOR**. 1 = For reads the respective active bits in **nMPMCBLSOUT[3:0]** are LOW. For writes the respective active bits in **nMPMCBLSOUT[3:0]** are LOW. The **MPMCSTCS1PB** signal determines the value of the chip select 1 byte lane state field on power-on reset, **nPOR**. This value can be overridden by software. This field is unaffected by AHB reset, **HRESETn**. The byte lane state bit, PB, enables different types of memory to be connected. For byte-wide static memories the **nMPMCBLSOUT[3:0]** signal from the MPMC is usually connected to **nWE**, write enable. In this case, for reads, all the **nMPMCBLSOUT[3:0]** bits must be HIGH. This means that the byte lane state (PB) bit must be LOW. In other words, the PB bit influences the **WE** signal. When the PB bit is cleared, the **WE** signal is never active. When the PB bit is set, the **WE** signal is generated. 16-bit wide static memory devices usually have the **nMPMCBLSOUT[3:0]** signals connected to the **nUB** and **nLB**, upper byte and lower byte, signals in the static memory. In this case a write to a particular byte must assert the appropriate **nUB** or **nLB** signal LOW. For reads, all the **nUB** and **nLB** signals must be asserted LOW so that the bus is driven. In this case the byte lane state (PB) bit must be HIGH.[b]

 Copyright © 2002-2006 ARM Limited. All rights reserved.

Table 3-24 MPMCStaticConfig0-3 Registers bit assignments (continued)

Bits	Name	Function
[6]	Chip select polarity, PC	0 = active LOW chip select 1 = active HIGH chip select. The relevant **MPMCSTCSxPOL** signal determines the value of the chip select polarity on power-on reset, **nPOR**. Software can override this value. This field is unaffected by AHB reset, **HRESETn**.[c]
[5:4]	-	Read undefined. Write as zero.
[3]	Page mode, PM	0 = disabled (reset value on **nPOR**) 1 = async page mode enabled (page length four). In page mode the MPMC can burst up to four external accesses. Therefore devices with asynchronous page mode burst four or higher devices are supported. Asynchronous page mode burst two devices are not supported and must be accessed normally.
[2]	-	Read undefined. Write as zero.
[1:0]	Memory width, MW	00 = 8-bit (reset value for chip select 0, 2, and 3 on **nPOR**). 01 = 16-bit 10 = 32-bit 11 = reserved. The **MPMCSTCS1MW[1:0]** signal determines the value of the chip select 1 memory width field on power-on reset, **nPOR**. Software can override this value. This field is unaffected by AHB reset, **HRESETn**.[d]

a. Extended wait and page mode cannot be selected simultaneously.
b. For chip select 1 the value of the **MPMCSTCS1PB** signal is reflected in this field. When programmed this register reflects the last value that is written into it.
c. The value of the relevant **MPMCSTCSxPOL** signal is reflected in this field. When programmed this register reflects the last value that is written into it.
d. For chip select 1 the value of the **MPMCSTCS1MW[1:0]** signal is reflected in this field. When programmed this register reflects the last value that is written into it.

Note

Synchronous burst mode memory devices are not supported.

Copyright © 2002-2006 ARM Limited. All rights reserved.

3.3.22 Static Memory Write Enable Delay Registers 0-3

The four-bit, read/write, MPMCStaticWaitWen0-3 Registers enable you to program the delay from the chip select to the write enable. It is recommended that you modify this register during system initialization, or when there are no current or outstanding transactions. You can ensure this by waiting until the MPMC is idle, and then entering low-power, or disabled mode. These registers are accessed with one wait state.

Figure 3-22 shows the register bit assignments.

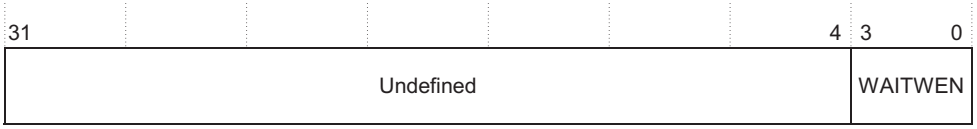

Figure 3-22 MPMCStaticWaitWen0-3 Registers bit assignments

Table 3-25 lists the register bit assignments.

Table 3-25 MPMCStaticWaitWen0-3 Registers bit assignments

Bits	Name	Function
[31:4]	-	Read undefined. Write as zero.
[3:0]	Wait write enable, WAITWEN	Delay from chip select assertion to write enable. 0000 = one **HCLK** cycle delay between assertion of chip select and write enable (reset value on **nPOR**) 0001-1111 = (n + 1) **HCLK** cycle delay[a].

a. The delay is (WAITWEN +1) x t_{HCLK}.

3.3.23 Static Memory Output Enable Delay Registers 0-3

The four-bit, read/write, MPMCStaticWaitOen0-3 Registers enable you to program the delay from the chip select or address change, whichever is later, to the output enable. It is recommended that you modify this register during system initialization, or when there are no current or outstanding transactions. You can ensure this by waiting until the MPMC is idle, and then entering low-power, or disabled mode. These registers are accessed with one wait state.

Figure 3-23 on page 3-35 shows the register bit assignments.

Copyright © 2002-2006 ARM Limited. All rights reserved.

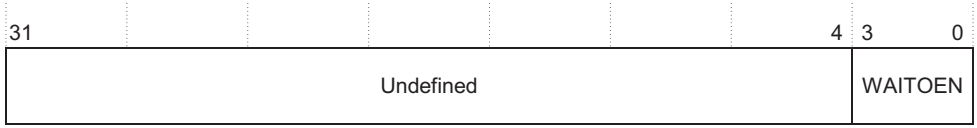

Figure 3-23 MPMCStaticWaitOen0-3 Registers bit assignments

Table 3-26 lists the register bit assignments.

Table 3-26 MPMCStaticWaitOen0-3 Registers bit assignments

Bits	Name	Function
[31:4]	-	Read undefined. Write as zero.
[3:0]	Wait output enable, WAITOEN	Delay from chip select assertion to output enable. 0000 = No delay (reset value on **nPOR**) 0001-1111= n cycle delay[a].

a. The delay is WAITOEN x t_{HCLK}.

3.3.24 Static Memory Read Delay Registers 0-3

The five-bit, read/write, MPMCStaticWaitRd0-3 Registers enable you to program the delay from the chip select to the read access. It is recommended that you modify this register during system initialization, or when there are no current or outstanding transactions. You can ensure this by waiting until the MPMC is idle, and then entering low-power, or disabled mode. It is not used if the extended wait bit is enabled in the MPMCStaticConfig0-3 Registers. These registers are accessed with one wait state.

Figure 3-24 shows the register bit assignments.

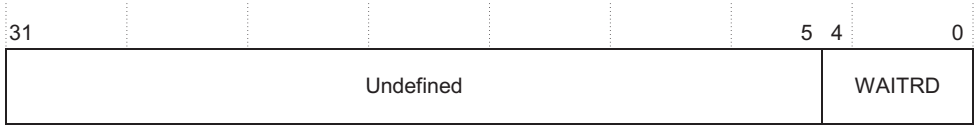

Figure 3-24 MPMCStaticWaitRd0-3 Registers bit assignments

Copyright © 2002-2006 ARM Limited. All rights reserved.

Table 3-27 lists the register bit assignments.

Table 3-27 MPMCStaticWaitRd0-3 Registers bit assignments

Bits	Name	Function
[31:5]	-	Reserved, read undefined, do not modify.
[4:0]	Nonpage mode read wait states or asynchronous page mode read first access wait state, WAITRD	Nonpage mode read or asynchronous page mode read, first read only: 00000-11110 = (n + 1) **HCLK** cycles for read accesses[a] 11111 = 32 **HCLK** cycles for read accesses (reset value on **nPOR**).

a. For nonsequential reads, the wait state time is (WAITRD + 1) x t_{HCLK}.

3.3.25 Static Memory Page Mode Read Delay Registers 0-3

The five-bit, read/write, MPMCStaticWaitPage0-3 Registers enable you to program the delay for asynchronous page mode sequential accesses. It is recommended that you modify this register during system initialization, or when there are no current or outstanding transactions. You can ensure this by waiting until the MPMC is idle, and then entering low-power, or disabled mode. This register is accessed with one wait state.

Figure 3-25 shows the register bit assignments.

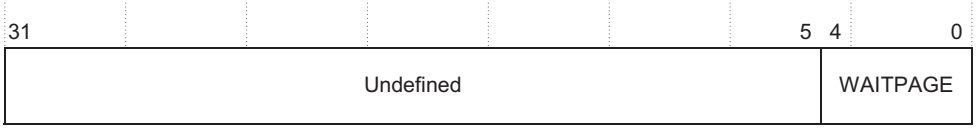

Figure 3-25 MPMCStaticWaitPage0-3 Registers bit assignments

Table 3-28 lists the register bit assignments.

Table 3-28 MPMCStaticWaitPage0-3 Registers bit assignments

Bits	Name	Function
[31:5]	-	Read undefined. Write as zero.
[4:0]	Asynchronous page mode read after the first read wait states, WAITPAGE	Number of wait states for asynchronous page mode read accesses after the first read: 00000-11110 = (n+ 1) **HCLK** cycle read access time[a] 11111 = 32 **HCLK** cycle read access time (reset value on **nPOR**).

a. For asynchronous page mode read for sequential reads, the wait state time for page mode accesses after the first read is (WAITPAGE + 1) x t_{HCLK}

Copyright © 2002-2006 ARM Limited. All rights reserved.

3.3.26 Static Memory Write Delay Registers 0-3

The five-bit, read/write, MPMCStaticWaitWr0-3 Registers enable you to program the delay from the chip select to the write access. It is recommended that you modify this register during system initialization, or when there are no current or outstanding transactions. You can ensure this by waiting until the MPMC is idle, and then entering low-power, or disabled mode.

These registers are not used if the extended wait (EW) bit is enabled in the MPMCStaticConfig Register. These registers are accessed with one wait state.

Figure 3-26 shows the register bit assignments.

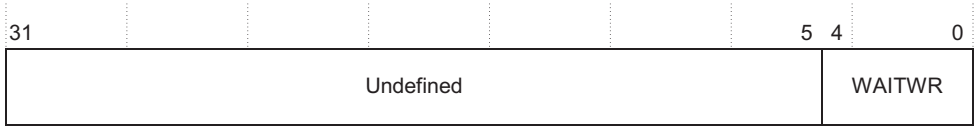

Figure 3-26 MPMCStaticWaitWr0-3 Registers bit assignments

Table 3-29 lists the register bit assignments.

Table 3-29 MPMCStaticWaitWr0-3 Registers bit assignments

Bits	Name	Function
[31:5]	-	Read undefined. Write as zero.
[4:0]	Write wait states, WAITWR	SRAM wait state time for write accesses after the first read: 00000-1110 = (n + 2) **HCLK** cycle write access time[a] 11111 = 33 **HCLK** cycle write access time (reset value on **nPOR**).

a. The wait state time for write accesses after the first read is WAITWR (n + 2) x t_{HCLK}

3.3.27 Static Memory Turn Round Delay Registers 0-3

The four-bit, read/write, MPMCStaticWaitTurn0-3 Registers enable you to program the number of bus turnaround cycles. It is recommended that you modify this register during system initialization, or when there are no current or outstanding transactions. You can ensure this by waiting until the MPMC is idle, and then entering low-power, or disabled mode. These registers are accessed with one wait state.

Figure 3-27 on page 3-38 shows the register bit assignments.

Copyright © 2002-2006 ARM Limited. All rights reserved.

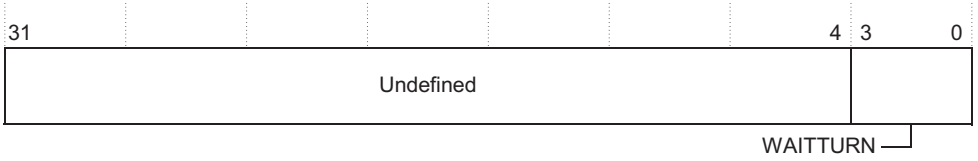

Figure 3-27 MPMCStaticWaitTurn0-3 Registers bit assignments

Table 3-30 lists the register bit assignments.

Table 3-30 MPMCStaticWaitTurn0-3 Registers bit assignments

Bits	Name	Function
[31:4]	-	Read undefined. Write as zero.
[3:0]	Bus turnaround cycles, WAITTURN	0000-1110 = (n + 1) **HCLK** turnaround cycles[a] 1111 = 16 **HCLK** turnaround cycles (reset value on **nPOR**).

a. Bus turnaround time is (WAITTURN + 1) x t_{HCLK}

To prevent bus contention on the external memory databus, the WAITTURN field controls the number of bus turnaround cycles added between static memory read and write accesses.

The WAITTURN field also controls the number of turnaround cycles between static memory and dynamic memory accesses.

3.3.28 Conceptual Additional Peripheral Identification Registers

The MPMCPeriphID4-7 Registers are four eight-bit read-only registers, that span address locations `0xFD0-0xFDC`. The registers can conceptually be treated as a single register that holds a 32-bit additional peripheral ID value.

Figure 3-28 shows the register bit assignments.

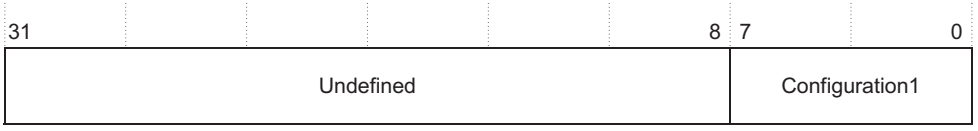

Figure 3-28 Conceptual MPMC Additional Peripheral ID Register bit assignments

Copyright © 2002-2006 ARM Limited. All rights reserved.

Table 3-31 lists the register bit assignments.

Table 3-31 Conceptual MPMC Additional Peripheral ID Register bit assignments

Bits	Name	Function
[31:8]	Reserved	Read undefined.
[7:0]	Configuration1	Additional peripheral configuration information

The configuration options are peripheral-specific. For MPMC, the four, eight-bit peripheral identification registers are described in the following subsections:

- *Additional Peripheral Identification Register 4*
- *Additional Peripheral Identification Registers 5-7*.

Additional Peripheral Identification Register 4

The MPMCPeriphID4 Register is hard-coded and the fields within the register determine the reset value. Table 3-32 shows the register bit assignments.

Table 3-32 MPMCPeriphID4 Register bit assignments

Bits	Name	Function
[31:8]	-	Read undefined.
[7:4]	Configuration	Number of buffers: 0000-1111 = n +1 buffers 0011 = four buffers, value for PL172.
[3:0]	Configuration	Number of AHB memory ports: 0000-1111 = n +1 memory ports 0011 = four memory ports, value for PL172.

Additional Peripheral Identification Registers 5-7

The MPMCPeriphID5-7 Registers are reserved.

Copyright © 2002-2006 ARM Limited. All rights reserved.

Table 3-33 shows the register bit assignments.

Table 3-33 MPMCPeriphID5-7 Registers bit assignments

Bits	Name	Function
[31:0]	-	Read undefined

3.3.29 Peripheral Identification Registers

The MPMCPeriphID0-3 Registers are four eight-bit read-only registers, that span address locations `0xFE0-0xFEC`. The registers can conceptually be treated as a single register that holds a 32-bit peripheral ID value.

Table 3-34 shows the register bit assignments.

Table 3-34 Conceptual MPMC Peripheral ID Register bit assignments

Bits	Name	Function
[31:24]	Configuration	Configuration options are peripheral-specific. See *Peripheral Identification Register 3* on page 3-43.
[23:20]	Revision	The peripheral revision number is revision dependent.
[19:12]	Designer	Designers ID number. This is `0x41` for ARM Limited.
[11:0]	Part number	Identifies the peripheral. The part number for PL172 is `0x172`.

Figure 3-29 on page 3-41 shows the correspondence between bits of the MPMCPeriphID0-3 Registers and the conceptual 32-bit MPMC Peripheral ID Register.

Copyright © 2002-2006 ARM Limited. All rights reserved.

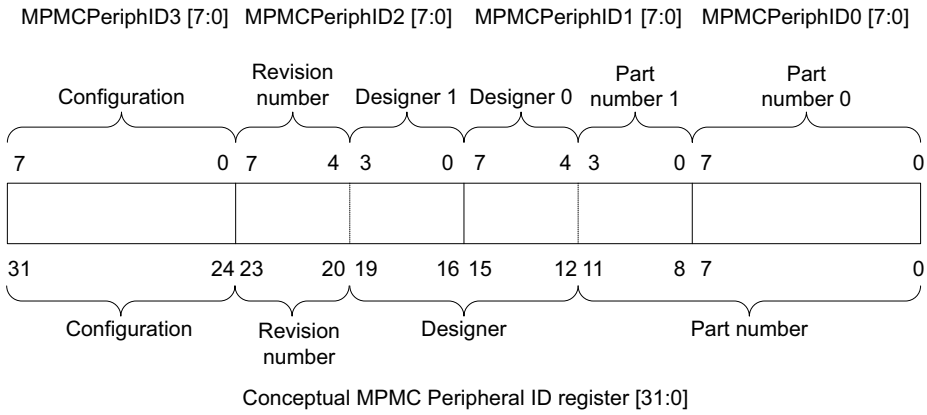

Figure 3-29 Peripheral Identification Register bit assignment

The four eight-bit peripheral identification registers are described in the following subsections:

- *Peripheral Identification Register 0*
- *Peripheral Identification Register 1* on page 3-42
- *Peripheral Identification Register 2* on page 3-42
- *Peripheral Identification Register 3* on page 3-43.

Peripheral Identification Register 0

The MPMCPeriphID0 Register is hard-coded and the fields within the register determine the reset value.

Table 3-35 shows the register bit assignments.

Table 3-35 MPMCPeriphID0 Register bit assignments

Bits	Name	Function
[31:8]	-	Read undefined
[7:0]	PartNumber0	These bits read back as `0x72`

Copyright © 2002-2006 ARM Limited. All rights reserved.

Peripheral Identification Register 1

The MPMCPeriphID1 Register is hard-coded and the fields within the register determine the reset value.

Table 3-36 shows the register bit assignments.

Table 3-36 MPMCPeriphID1 Register bit assignments

Bits	Name	Function
[31:8]	-	Read undefined
[7:4]	Designer0	These bits read back as `0x1`
[3:0]	PartNumber1	These bits read back as `0x1`

Peripheral Identification Register 2

The MPMCPeriphID2 Register is hard-coded and the fields within the register determine the reset value.

Table 3-37 shows the register bit assignments.

Table 3-37 MPMCPeriphID2 Register bit assignments

Bits	Name	Function
[31:8]	-	Read undefined.
[7:4]	Revision	These bits read back as the revision number, that can be between 0 and 15. [a]
[3:0]	Designer1	These bits read back as `0x4`.

a. 0 = PL172 Revision 1, 1 = PL172 Revision 2, 2 = PL172 Revision r2p3, 3 = PL172 Revision r2p4

Copyright © 2002-2006 ARM Limited. All rights reserved.

Peripheral Identification Register 3

The MPMCPeriphID3 Register is hard-coded and the fields within the register determine the reset value. Table 3-38 shows the register bit assignments.

Table 3-38 MPMCPeriphID3 Register bit assignments

Bits	Name	Function
[31:8]	-	Read undefined.
[5:3]	Configuration	Indicates the AHB master bus width: 000 = 32-bit wide 001 = 64-bit wide 010 = 128-bit wide 011 = 256-bit wide 100 = 512-bit wide 101 = 1024-bit wide 110-111 = reserved. For PL172 this field is set to 000.
[2]	Configuration	TIC interface: 0 = no 1 = yes For PL172 this field is set to 1.
[1]	Configuration	Data buffers: 0 = no 1 = yes For PL172 this field is set to 1.
[0]	Configuration	Static memory controller: 0 = no 1 = yes For PL172 this field is set to 1.

3.3.30 PrimeCell Identification Registers 0-3

The MPMCPCellID0-3 Registers are four eight-bit wide registers, that span address locations `0xFF0-0xFFC`. The registers can conceptually be treated as a single register that holds a 32-bit PrimeCell ID value. You can use this register for automatic BIOS configuration. The MPMCPCellID Register is set to `0xB105F00D`. This register is accessed with one wait state.

Figure 3-30 on page 3-44 shows the register bit assignments.

Copyright © 2002-2006 ARM Limited. All rights reserved.

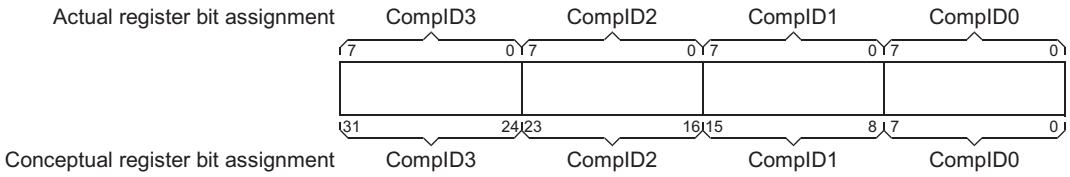

Figure 3-30 Conceptual PrimeCell ID Register bit assignments

Table 3-39 lists the register bit assignments.

Table 3-39 Conceptual PrimeCell ID Register bit assignments

PrimeCell ID register		**MPMCPCellID0-3 Register**		
Bits	**Reset value**	**Register**	**Bits**	**Function**
-	-	MPMCPCellID3	[31:8]	Read undefined
[31:24]	`0xB1`	MPMCPCellID3	[7:0]	These bits read back as `0xB1`
-	-	MPMCPCellID2	[31:8]	Read undefined
[23:16]	`0x05`	MPMCPCellID2	[7:0]	These bits read back as `0x05`
-	-	MPMCPCellID1	[31:8]	Read undefined
[15:8]	`0xF0`	MPMCPCellID1	[7:0]	These bits read back as `0xF0`
-	-	MPMCPCellID0	[31:8]	Read undefined
[7:0]	`0x0D`	MPMCPCellID0	[7:0]	These bits read back as `0x0D`

Copyright © 2002-2006 ARM Limited. All rights reserved.

Chapter 4
Programmer's Model for Test

This chapter describes the additional logic for integration testing. It contains the following sections:

Copyright © 2002-2006 ARM Limited. All rights reserved.

4.1 MPMC test harness overview

The additional logic for functional verification and integration vectors enables:

- capture of input signals to the block
- stimulation of the output signals.

The integration vectors provide a way of verifying that the MPMC is correctly wired into a system. This is done by separately testing three groups of signals:

AMBA signals

These are the AMBA bus signals.

Non-AMBA intra-chip signals

Non-AMBA intra-chip signals are on-chip signals that are not part of the AMBA bus, for example the interrupt request signals.

Primary inputs and outputs

The primary input and output signals are those that go off and on the chip.

4.1.1 AMBA test strategy

These signals are tested by performing various AMBA bus transactions.

4.1.2 Non-AMBA intra-chip integration test strategy

Many of these signals are difficult to control and observe. Therefore PrimeCells have integration test logic to make this task easier. These test features are enabled by programming the MultiPort Memory Controller Integration Test Control Register, MPMCITCR. The intra-chip input signal values can then be read using the MPMCITIP Register. You can control the intra-chip output signals using the MPMCITOP Register.

4.1.3 Primary I/O test strategy

The primary I/O signals are tested by performing a sequence of memory transactions.

 Copyright © 2002-2006 ARM Limited. All rights reserved.

4.2 Production test

The MPMC is designed to simplify:

- insertion of scan test cells
- use of *Automatic Test Pattern Generation* (ATPG).

Scan insertion and ATPG is the recommended method of manufacturing test.

Copyright © 2002-2006 ARM Limited. All rights reserved.

4.3 Summary of test registers

Table 4-1 shows how the MPMC test registers are memory-mapped.

Table 4-1 Test registers memory map

Name	Base offset	Type	Reset value	Description
MPMCITCR	0xF00	Read/write	0x0	*Test Control Register* on page 4-5
MPMCITIP	0xF20	Read/write	0x00	*Test Input Register* on page 4-5
MPMCITOP	0xF40	Read/write	0x0	*Test Output Register* on page 4-8

Copyright © 2002-2006 ARM Limited. All rights reserved.

4.4 Test register descriptions

This section describes the test registers.

4.4.1 Test Control Register

The MPMCITCR Register is a single-bit read/write control register. The T bit in this register controls the input test multiplexors. This register must only be used in test mode. This register is accessed with one wait state.

Figure 4-1 shows the register bit assignments.

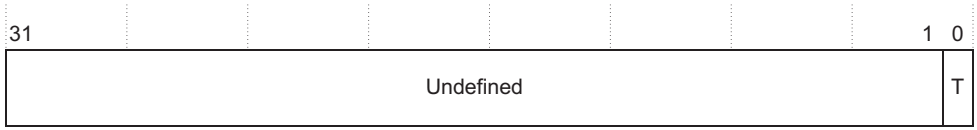

Figure 4-1 MPMCITCR Register bit assignments

Table 4-2 lists the register bit assignments.

Table 4-2 MPMCITCR Register bit assignments

Bits	Name	Description
[31:1]	-	Read undefined. Write as zero.
[0]	Test control register, T	0 = normal mode, reset value on **nPOR** or **HRESETn** to 1 = test mode.

4.4.2 Test Input Register

The MPMCITIP Register is a ten-bit read/write register. This register must only be used in test mode. This register is accessed with one wait state.

Figure 4-2 on page 4-6 shows the register bit assignments.

Copyright © 2002-2006 ARM Limited. All rights reserved.

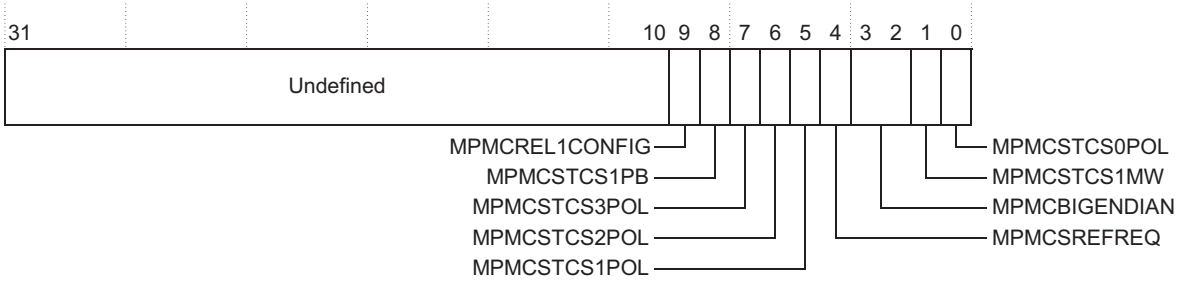

Figure 4-2 MPMCITIP Register bit assignments

Table 4-3 lists the register bit assignments.

Table 4-3 MPMCITIP Register bit assignments

Bits	Name	Description
[31:10]	-	Read undefined. Write as zero.
[9]	MPMCREL1CONFIG	Revision 1 configuration. Read: Read the value of this field if the MPMCITCR T bit is HIGH. Read the value of the **MPMCREL1CONFIG** input signal if the MPMCITCR T bit is LOW. Write: 0 = Clear this field if the MPMCITCR T bit is HIGH. 1 = Set this field if the MPMCITCR T bit is HIGH.
[8]	MPMCSTCS1PB	Polarity of byte lane for chip select1 signal **MPMCSTCS1PB**. Read: Read the value of this field if the MPMCITCR T bit is HIGH. Read the value of the **MPMCSTCS1PB** input signal if the MPMCITCR T bit is LOW. Write: 0 = Clear this field if the MPMCITCR T bit is HIGH. 1 = Set this field if the MPMCITCR T bit is HIGH.
[7]	MPMCSTCS3POL	Polarity of chip select3 signal **MPMCSTCS3POL**. Read: Read the value of this field if the MPMCITCR T bit is HIGH. Read the value of the **MPMCSTCS3POL** input signal if the MPMCITCR T bit is LOW. Write: 0 = Clear this field if the MPMCITCR T bit is HIGH. 1 = Set this field if the MPMCITCR T bit is HIGH.

Copyright © 2002-2006 ARM Limited. All rights reserved.

Table 4-3 MPMCITIP Register bit assignments (continued)

Bits	Name	Description
[6]	MPMCSTCS2POL	Polarity of chip select2 signal **MPMCSTCS2POL**. Read: Read the value of this field if the MPMCITCR T bit is HIGH. Read the value of the **MPMCSTCS2POL** input signal if the MPMCITCR T bit is LOW. Write: 0 = Clear this field if the MPMCITCR T bit is HIGH. 1 = Set this field if the MPMCITCR T bit is HIGH.
[5]	MPMCSTCS1POL	Polarity of chip select1 signal **MPMCSTCS1POL**. Read: Read the value of this field if the MPMCITCR T bit is HIGH. Read the value of the **MPMCSTCS1POL** input signal if the MPMCITCR T bit is LOW. Write: 0 = Clear this field if the MPMCITCR T bit is HIGH. 1 = Set this field if the MPMCITCR T bit is HIGH.
[4]	MPMCSTCS0POL	Polarity of chip select0 signal **MPMCSTCS0POL**. Read: Read the value of this field if the MPMCITCR T bit is HIGH. Read the value of the **MPMCSTCS0POL** input signal if the MPMCITCR T bit is LOW. Write: 0 = Clear this field if the MPMCITCR T bit is HIGH. 1 = Set this field if the MPMCITCR T bit is HIGH.

Copyright © 2002-2006 ARM Limited. All rights reserved.

Table 4-3 MPMCITIP Register bit assignments (continued)

Bits	Name	Description
[3:2]	MPMCSTCS1MW	Chip select one memory width **MPMCSTCS1MW**. Read: Read the value of this field if the MPMCITCR T bit is HIGH. Read the value of the **MPMCSTCS1MW** input signal if the MPMCITCR T bit is LOW. Write: 0 = Clear this field if the MPMCITCR T bit is HIGH. 1 = Set this field if the MPMCITCR T bit is HIGH.
[1]	MPMCBIGENDIAN	Big-endian enable signal **MPMCBIGENDIAN**. Read: Read the value of this field if the MPMCITCR T bit is HIGH. Read the value of the **MPMCBIGENDIAN** input signal if the MPMCITCR T bit is LOW. Write: 0 = Clear this field if the MPMCITCR T bit is HIGH. 1 = Set this field if the MPMCITCR T bit is HIGH.
[0]	MPMCSREFREQ (SR)	Read: Read the value of this field if the MPMCITCR T bit is HIGH. Read the value of the **MPMCSREFREQ** input signal if the MPMCITCR T bit is LOW. Write: 0 = Clear this field if the MPMCITCR T bit is HIGH. 1 = Set this field if the MPMCITCR T bit is HIGH.

4.4.3 Test Output Register

The MPMCITOP Register is a two-bit register. This register must only be used in test mode. This register is accessed with one wait state.

Figure 4-3 shows the register bit assignments.

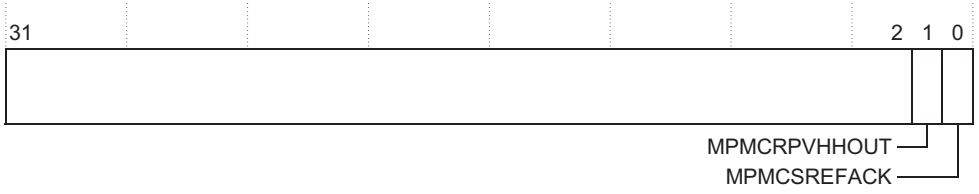

Figure 4-3 MPMCITOP Register bit assignments

Copyright © 2002-2006 ARM Limited. All rights reserved.

Table 4-4 lists the register bit assignments.

Table 4-4 MPMCITOP Register bit assignments

Bits	Name	Description
[31:2]	-	Read undefined. Write as zero.
[1]	MPMCRPVHHOUT	Signal **MPMCRPVHHOUT**. Read: Read the value of this field if the MPMCITCR T bit is HIGH. Read the value of the **MPMCRPVHHOUT** output signal if the MPMCITCR T bit is LOW. Write: 0 = Clear this field and the **MPMCRPVHHOUT** output signal if the MPMCITCR T bit is HIGH. 1 = Set this field and the **MPMCRPVHHOUT** output signal if the MPMCITCR T bit is HIGH.
[0]	MPMCSREFACK (SA)	Self-refresh acknowledge. Read: Read the value of this field if the MPMCITCR T bit is HIGH. Read the value of the **MPMCSREFACK** output signal if the MPMCITCR T bit is LOW. Write: 0 = Clear this field and the **MPMCSREFACK** output signal if the MPMCITCR T bit is HIGH. 1 = Set this field and the **MPMCSREFACK** output signal if the MPMCITCR T bit is HIGH.

Copyright © 2002-2006 ARM Limited. All rights reserved.

 Copyright © 2002-2006 ARM Limited. All rights reserved.

Appendix A
Signal Descriptions

This appendix describes the signals that interface with the MPMC controller block. It contains the following sections:

Copyright © 2002-2006 ARM Limited. All rights reserved.

A.1 AHB register signals

Table A-1 shows the AHB register signals.

Table A-1 AHB register signal descriptions

Name	Type	Source/ destination	Description
HADDRREG[11:2]	Input	AHB master layer	The AHB address bus.
HCLK	Input	Clock source	This clock times all bus transfers. All signal timings are related to the rising edge of **HCLK**.
HRDATAREG[20:0]	Output	AHB master layer	The read databus transfers data from the MPMC to the bus master during read operations.
HREADYINREG	Input	AHB slave layer	When HIGH, the **HREADYIN** signal indicates that a transfer has finished on the bus. You can drive this signal LOW to extend a transfer. An alternate slave generates this signal.
HREADYOUTREG	Output	AHB master layer and AHB slave layer	When HIGH, the **HREADYOUT** signal indicates that a transfer has finished on the bus. You can drive this signal LOW to extend a transfer.
HRESETn	Input	Reset controller	The bus reset signal is active LOW and resets the system and the bus.
HRESPREG[1:0]	Output	AHB master layer	The transfer response provides additional information on the status of a transfer. Four different responses are provided, OKAY, ERROR, RETRY, and SPLIT. The SDRAM Controller can respond with either the OKAY or ERROR responses. The ERROR response is generated when the transfer size is not 32 bits wide.
HSELMPMCREG	Input	AHB decoder	Slave select. MPMC controller register select signal. This signal indicates an access to the memory controllers control registers.
HSIZEREG[2:0]	Input	AHB master layer	Indicates the size of the transfer, that is typically byte (8-bit), halfword (16-bit), or word (32-bit). Only word (32-bit) transfers are permitted, (**HSIZE[2:0]** = 0b010) to access the MPMC registers. Transfer sizes other than 32 bits generate an ERROR response.

Copyright © 2002-2006 ARM Limited. All rights reserved.

Table A-1 AHB register signal descriptions (continued)

Name	Type	Source/ destination	Description
HTRANSREG	Input	AHB master layer	Transfer type. Indicates the type of the current transfer. This can be NONSEQUENTIAL, SEQUENTIAL, IDLE, or BUSY. Only **HTRANS[1]** is used. Indicating if a transfer is required or not.
HWDATAREG20TO19[20:19]	Input	AHB master layer	The write databus transfers data from the master to the bus slaves during write operations.
HWDATAREG15TO0[15:0]	Input	AHB master layer	The write databus transfers data from the master to the bus slaves during write operations.
HWRITEREG	Input	AHB master layer	Transfer direction. When HIGH this signal indicates a write transfer and when LOW a read transfer.

Copyright © 2002-2006 ARM Limited. All rights reserved.

A.2 AHB memory signals

Table A-2 shows the AHB memory signals. The signal names for each port can be found by substituting the port number, 0, 1, 2, or 3, for the symbol x. Disable unused ports by connecting their inputs to logic 0. The MPMC does not support RETRY or SPLIT transactions.

Table A-2 AHB memory signal descriptions

Name	Type	Source/ destination	Description
HADDRx[27:0]	Input	AHB master layer	The AHB address bus.
HBURSTx[2:0]	Input	AHB master layer	Burst type. Indicates if the transfer forms part of a burst. 4, 8, and 16 beat bursts are supported and the burst can be either incrementing or wrapping.
HMASTLOCKx	Input	AHB master layer	Locked transfers. When HIGH, this signal indicates that the master requires locked access to the SDRAM and no other master must be granted the bus until this signal is LOW.
HRDATAx[31:0]	Output	AHB master layer	The read databus transfers data from the MPMC to the bus master during read operations.
HREADYINx	Input	AHB slave layer	When HIGH, the **HREADYIN** signal indicates that a transfer has finished on the bus. You can drive this signal LOW to extend a transfer. An alternate slave generates this signal.
HREADYOUTx	Output	AHB master layer and AHB slave layer	When HIGH, the **HREADYOUT** signal indicates that a transfer has finished on the bus. You can drive this signal LOW to extend a transfer.
HRESPx[1:0]	Output	AHB master layer	The transfer response provides additional information on the status of a transfer. Four different responses are provided, OKAY, ERROR, RETRY and SPLIT. The SDRAM controller can respond with either the OKAY or ERROR responses. The ERROR response is generated when: • the transfer size is greater than permitted • the memory access is a write to a write protected region • the memory is accessed with the MCEnable bit disabled or the LowPowerMode bit is asserted.

Copyright © 2002-2006 ARM Limited. All rights reserved.

Table A-2 AHB memory signal descriptions (continued)

Name	Type	Source/ destination	Description
HSELMPMCxCS[7:0]	Input	AHB decoder	MPMC select signal. These signals indicate an access to memory. Specific chip select. **HSELMPMCxCS[0]** indicates the transfer is to chip select 0, **HSELMPMCxCS[1]** indicates the transfer is to chip select 1. Only one signal can go active at a time. The HSEL to chip select decoding is as follows: • **HSELMPMCxCS[0]** selects **nMPMCSTCSOUT[0]** • **HSELMPMCxCS[1]** selects **nMPMCSTCSOUT[1]** • **HSELMPMCxCS[2]** selects **nMPMCSTCSOUT[2]** • **HSELMPMCxCS[3]** selects **nMPMCSTCSOUT[3]** • **HSELMPMCxCS[4]** selects **nMPMCDYCSOUT[0]** • **HSELMPMCxCS[5]** selects **nMPMCDYCSOUT[1]** • **HSELMPMCxCS[6]** selects **nMPMCDYCSOUT[2]** • **HSELMPMCxCS[7]** selects **nMPMCDYCSOUT[3]**.
HSIZEx[2:0]	Input	AHB master layer	Indicates the size of the transfer. This is typically byte (8-bit), halfword (16-bit), or word (32-bit). Byte (8-bit), halfword (16-bit), and word (32-bit) transfers are permitted to access the external memory. Transfer sizes greater than 32 bits generate an ERROR response.
HTRANSx[1:0]	Input	AHB master layer	Transfer type. Indicates the type of the current transfer. This can be NONSEQUENTIAL, SEQUENTIAL, IDLE, or BUSY.
HWDATAx[31:0]	Input	AHB master layer	The write databus transfers data from the master to the bus slaves during write operations.
HWRITEx	Input	AHB master layer	Transfer direction. When HIGH this signal indicates a write transfer and when LOW a read transfer.
HSELMPMCxG	Input	AHB decoder	MPMC global select signal. This signal must go active with the **HSELMPMCxCS[7:0]** signals, and whenever the MPMC is accessed.

Copyright © 2002-2006 ARM Limited. All rights reserved.

A.3 Miscellaneous signals

The MPMC miscellaneous signals are described in:

- *Tie-off signals*
- *Clock signals* on page A-9
- *External Bus Interface (EBI) signals* on page A-9.

A.3.1 Tie-off signals

Table A-3 describes the tie-off signals.

Table A-3 Miscellaneous and clock signal descriptions

Name	Type	Source/ destination	Description
MPMCBIGENDIAN	Input	Reset controller	Endian select. If this bit is HIGH, big-endian mode is selected. Do not alter this signal during normal operation. Register an input signal on power-on reset, **nPOR**, in the reset controller to generate this signal. Alternatively, you can tie this signal to the appropriate value.
MPMCSREFACK	Output	Power management unit	Self-refresh acknowledgement.
MPMCSREFREQ	Input	Power management unit	Self-refresh request.
MPMCSTCS1PB	Input	Reset controller	Chip select 1 byte lane state selection. 0 = for reads all the bits in **nMPMCBLSOUT[3:0]** are HIGH. For writes the respective active bits in **nMPMCBLSOUT[3:0]** are LOW. 1 = for reads the respective active bits in **nMPMCBLSOUT[3:0]** are LOW. For writes the respective active bits in **nMPMCBLSOUT[3:0]** are LOW. Used for static memory devices. Do not alter this signal during normal operation. Register an input signal on power-on reset, **nPOR**, in the reset controller to generate this signal. Alternatively, you can tie this signal to the appropriate value.

Copyright © 2002-2006 ARM Limited. All rights reserved.

Table A-3 Miscellaneous and clock signal descriptions (continued)

Name	Type	Source/ destination	Description
MPMCSTCS1MW[1:0]	Input	Reset controller	Chip select 1 memory width selection: 00 = 8-bit wide memory 01 = 16-bit wide memory 10 = 32-bit wide memory 11 = reserved. Used for static memory devices. Do not alter this signal during normal operation. Register an input signal on power-on reset, **nPOR**, in the reset controller to generate this signal. Alternatively, you can tie this signal to the appropriate value.
MPMCSTCS0POL	Input	Reset controller	Chip select 0 polarity select. A HIGH value indicates CS active HIGH, a LOW value indicates CS active LOW. Used for static memory devices. Do not alter this signal during normal operation. Register an input signal on power-on reset, **nPOR**, in the reset controller to generate this signal. Alternatively, you can tie this signal to the appropriate value.
MPMCSTCS1POL	Input	Reset controller	Chip select 1 polarity select. A HIGH value indicates CS active HIGH, a LOW value indicates CS active LOW. Used for static memory devices. Do not alter this signal during normal operation. Register an input signal on power-on reset, **nPOR**, in the reset controller to generate this signal. Alternatively, you can tie this signal to the appropriate value.
MPMCSTCS2POL	Input	Reset controller	Chip select 2 polarity select. A HIGH value indicates CS active HIGH, a LOW value indicates CS active LOW. Used for static memory devices. Do not alter this signal during normal operation. Register an input signal on power-on reset, **nPOR**, in the reset controller to generate this signal. Alternatively, you can tie this signal to the appropriate value.

Copyright © 2002-2006 ARM Limited. All rights reserved.

Table A-3 Miscellaneous and clock signal descriptions (continued)

Name	Type	Source/ destination	Description
MPMCSTCS3POL	Input	Reset controller	Chip select 3 polarity select. A HIGH value indicates CS active HIGH, a LOW value indicates CS active LOW. Used for static memory devices. Do not alter this signal during normal operation. Register an input signal on power-on reset, **nPOR**, in the reset controller to generate this signal. Alternatively, you can tie this signal to the appropriate value.
MPMCREL1CONFIG	Input	Reset controller	PL172 revision 1 backwards compatibility tie-off. When tied HIGH the static memory addressing scheme used in PL172 Rel1v0 is used. When tied LOW the static memory addressing is right shifted. Used for static memory devices. Do not alter this signal during normal operation. Register an input signal on power-on reset, **nPOR**, in the reset controller to generate this signal. Alternatively, you can tie this signal to the appropriate value.
nPOR	Input	Reset controller	Power-on reset (active LOW).

Copyright © 2002-2006 ARM Limited. All rights reserved.

A.3.2 Test signals

Table A-4 shows the test signals.

Table A-4 Test signal descriptions

Name	Type	Source/ destination	Description
MPMCTESTREQA	Input	Pad	Test bus request A. This pin is used in TIC test mode. In test mode this pin is the test bus request A input signal and is required as a dedicated device pin. During normal system operation the **MPMCTESTREQA** signal requests entry into the test mode. During test **MPMCTESTREQA** is used, in combination with **MPMCTESTREQB**, to indicate the type of test vector that is applied in the following cycle.
MPMCTESTREQB	Input	Pad	Test bus request B. This pin is used in TIC test mode. During test **MPMCTESTREQB** is used, in combination with **MPMCTESTREQA**, to indicate the type of test vector that is applied in the following cycle. The test bus acknowledge signal gives external indication that the test bus has been granted and also indicates when a test access has completed. When **MPMCTESTACK** is LOW, the current test vector must be extended until **MPMCTESTACK** becomes HIGH.

A.3.3 Clock signals

Table A-5 shows the clock signals.

Table A-5 Clock signal descriptions

Name	Type	Source/ destination	Description
MPMCCLK	Input	Clock source	**MPMCCLK** times all external memory transfers. **MPMCCLK** must be synchronous to **HCLK**. **MPMCCLK** can operate at the same frequency as **HCLK** or twice the frequency of **HCLK**.
MPMCCLKDELAY	Input	Clock source	Delayed version of **MPMCCLK** used in command delayed mode to ensure that SDRAM set up and hold requirements are met.

A.3.4 External Bus Interface (EBI) signals

Table A-6 on page A-10 shows the EBI signals.

Copyright © 2002-2006 ARM Limited. All rights reserved.

Table A-6 EBI signal descriptions

Name	Type	Source/ destination	Description
MPMCEBIGNT	Input	EBI	The EBI grant signal goes active HIGH when the EBI grants the external memory bus.
MPMCEBIBACKOFF	Input	EBI	The EBI backoff signal goes active HIGH when the EBI wants to remove the memory controller from the memory bus so that another memory controller can be granted the memory bus.
MPMCEBIREQ	Output	EBI	The EBI request signal goes active HIGH when the memory controller requires the memory bus.

Copyright © 2002-2006 ARM Limited. All rights reserved.

A.4 Pad interface and control signals

Table A-7 shows the pad interface and control signals.

Table A-7 Pad interface and control signal descriptions

Name	Type	Source/ destination	Value on reset (nPOR)	Value during self-refresh	Description
MPMCADDROUT[27:0]	Output	Pad	0x0000000	Depends on static memory accesses	Address output. Used for both static and SDRAM devices. SDRAM memories use bits [14:0]. Static memories use bits [25:0].
MPMCCKEOUT[3:0]	Output	Pad	0xF	0x0	SDRAM clock enables. Used for SDRAM devices.
MPMCCLKOUT[3:0]	Output	Pad	Follows **MPMCCLK**	Follows **MPMCCLK**	SDRAM clocks. Used for SDRAM devices.
MPMCDATAIN[31:0]	Input	Pad			Read data from memory. Used for the static memory controller, the dynamic memory controller and the TIC.
MPMCDATAOUT[31:0]	Output	Pad	0x00000000	Depends on static memory accesses	Data output to memory. Used for the static memory controller, the dynamic memory controller and the TIC.
MPMCDQMOUT[3:0]	Output	Pad	0xF	0xF	Data mask output to SDRAMs. Used for SDRAM devices.
MPMCFBCLKIN[3:0]	Input	Pad	-	-	Feedback clocks. Used for SDRAM devices.
MPMCRPVHHOUT	Output	Pad	0	-	Voltage control for RP output signal.

Copyright © 2002-2006 ARM Limited. All rights reserved.

Table A-7 Pad interface and control signal descriptions (continued)

Name	Type	Source/ destination	Value on reset (nPOR)	Value during self-refresh	Description
MPMCTESTIN	Input	Pad	-	-	Test Mode. This places the MPMC into TIC test mode. Used for TIC test.
nMPMCBLSOUT[3:0]	Output	Pad	0xF	Depends on static memory accesses	Byte lane select, active LOW, for static memories. Used for static memory devices.
nMPMCCASOUT	Output	Pad	1	1	Column address strobe. Used for SDRAM devices.
nMPMCDATAOUTEN[3:0]	Output	Pad control	0xF	0xF	Tristate I/O pad output enable for the byte lanes of the external memory databus **MPMCDATA[31:0]**, active LOW. Enables the byte lanes [31:24], [23:16], [15:8], and [7:0] of the databus independently. Used for the static and dynamic memory controllers, and the TIC.
nMPMCDYCSOUT[3:0]	Output	Pad	0xF	0xF	SDRAM chip selects. Used for SDRAM devices.
nMPMCOEOUT	Output	Pad	1	Depends on static memory accesses	Output enable for static memories. Used for static memory devices.
nMPMCRASOUT	Output	Pad	1	1	Row address strobe. Used for SDRAM devices.

Copyright © 2002-2006 ARM Limited. All rights reserved.

Table A-7 Pad interface and control signal descriptions (continued)

Name	Type	Source/ destination	Value on reset (nPOR)	Value during self-refresh	Description
nMPMCRPOUT	Output	Pad	0	-	Reset power down to SyncFlash memory. Used for the dynamic memory controller.
nMPMCSTCSOUT[3:0]	Output	Pad	0xF	Depends on static memory accesses	Static memory chip selects. Default active LOW. Used for static memory devices.
nMPMCWEOUT	Output	Pad	1	Depends on static memory accesses	Write enable. Used for SDRAM and static memories. This signal is used as test acknowledge (**MPMCTESTACK**), during TIC test mode. The test bus acknowledge signal gives external indication that the test bus is granted and also indicates when a test access is completed. When **MPMCTESTACK** is LOW the current test vector must be extended until **MPMCTESTACK** becomes HIGH.

Copyright © 2002-2006 ARM Limited. All rights reserved.

A.5 Test Interface Controller (TIC) AHB signals

Table A-8 shows the TIC AHB signals.

Table A-8 TIC AHB signal descriptions

Name	Type	Source/ destination	Description
HADDRTIC[31:0]	Output	AHB slave layer	Address bus. The 32-bit AHB address bus.
HBUSREQTIC	Output	AHB arbiter	Bus request. A signal from the TIC to the bus arbiter that indicates that it requires the bus.
HBURSTTIC[2:0]	Output	AHB slave layer	Burst type. Indicates if the transfer forms part of a burst. The TIC always performs incrementing bursts of unspecified length.
HGRANTTIC	Input	AHB arbiter	Bus grant. This signal indicates that the TIC is currently the highest priority master. Ownership of the address and control signals changes at the end of a transfer when **HREADYINTIC** is HIGH, so the TIC gains access to the bus when both **HREADYINTIC** and **HGRANTTIC** are HIGH.
HLOCKTIC	Output	AHB arbiter	Locked transfers. When HIGH, this signal indicates that the TIC requires locked access to the bus. No other master must be granted the bus until this signal is LOW.
HPROT[3:0]	Output	AHB slave layer	The protection control signals indicate if the transfer is an opcode fetch or data access, in addition to whether the transfer is a supervisor mode access or a User mode access. These signals also indicate if the current access is cachable or bufferable.
HRDATATIC[31:0]	Input	AHB master layer	The read databus transfers data from the AHB slave to the TIC during read operations.
HREADYINTIC	Input	AHB slave layer	Transfer done. When HIGH, the **HREADYINTIC** signal indicates that a transfer has finished on the bus. You can drive this signal LOW to extend a transfer.
HRESPTIC[1:0]	Input	AHB slave layer	The transfer response provides additional information on the status of a transfer. Four different responses are provided, OKAY, ERROR, RETRY, and SPLIT.
HSIZETIC[2:0]	Output	AHB slave layer	Indicates the size of the transfer. This is typically byte (8-bit), halfword (16-bit), or word (32-bit). The TIC does not support larger transfer sizes.

Copyright © 2002-2006 ARM Limited. All rights reserved.

Table A-8 TIC AHB signal descriptions (continued)

Name	Type	Source/ destination	Description
HTRANSTIC[1:0]	Output	AHB slave layer	Transfer type. Indicates the type of the current transfer. This can be NONSEQUENTIAL, SEQUENTIAL, IDLE, or BUSY. The TIC does not use the BUSY transfer type.
HWDATATIC[31:0]	Output	AHB slave layer	Write databus. The write databus transfers data from the master to the bus slaves during write operations.
HWRITETIC	Output	AHB slave layer	Transfer direction. When HIGH, this signal indicates a write transfer and when LOW a read transfer.

Copyright © 2002-2006 ARM Limited. All rights reserved.

A.6 Scan test signals

Table A-9 shows the scan test signals.

Table A-9 Scan test signal descriptions

Name	Type	Source/ destination	Description
SCANENABLE	Input	Test controller	Scan enable
SCANINHCLK	Input	Test controller	Scan data input for **HCLK** scan chain
SCANINMPMCFBCLKIN0	Input	Test controller	Scan in for **MPMCFBCLKIN0** scan chain
SCANINMPMCFBCLKIN1	Input	Test controller	Scan in for **MPMCFBCLKIN1** scan chain
SCANINMPMCFBCLKIN2	Input	Test controller	Scan in for **MPMCFBCLKIN2** scan chain
SCANINMPMCFBCLKIN3	Input	Test controller	Scan in for **MPMCFBCLKIN3** scan chain
SCANINMPMCCLK	Input	Test controller	Scan in for **MPMCCLK** scan chain
SCANINCLKDELAY	Input	Test controller	Scan in for **MPMCCLKDELAY** scan chain
SCANOUTHCLK	Output	Test controller	Scan data output for **HCLK** scan chain
SCANOUTMPMCFBCLKIN0	Output	Test controller	Scan out for **MPMCFBCLKIN0** scan chain
SCANOUTMPMCFBCLKIN1	Output	Test controller	Scan out for **MPMCFBCLKIN1** scan chain
SCANOUTMPMCFBCLKIN2	Output	Test controller	Scan out for **MPMCFBCLKIN2** scan chain
SCANOUTMPMCFBCLKIN3	Output	Test controller	Scan out for **MPMCFBCLKIN3** scan chain
SCANOUTMPMCCLK	Output	Test controller	Scan out for **MPMCCLK** scan chain
SCANOUTCLKDELAY	Output	Test controller	Scan out for **MPMCCLKDELAY** scan chain

Copyright © 2002-2006 ARM Limited. All rights reserved.