ARM PrimeCell™ SDRAM Controller (PL170)

Technical Reference Manual

ARM DDI 0159D

ARM PrimeCell SDRAM Controller (PL170)
Technical Reference Manual

Release Information

Change history

Date	Issue	Change
15 October 1999	A	First release
22 December 1999	B	Minor changes on pages 1-3, 3-2, 3-4, 3-9, 3-10, A-2, A-5, A-6. Minor changes to Figures 2-1 and 2-2. Figure 2-3 added, with supporting text. Table 3-2 revised: new row added at end. Tables 3-8 to 3-11 revised: additional column and detail changes. Tables 3-12 and 3-13 added to show address mapping for 256M SDRAM.
18 April 2000	C	Minor changes on pages 1-4, 2-3 and A-6. Minor changes to Figure 2-3. Tables 3-12 and 3-13 added to show address mapping for 128M SDRAM. Tables 3-8 to 3-11 rearranged in SDRAM sequence. Table A-2 revised to include BIGENDIAN signal, with supporting note. Tables 3-8 to 3-15 revised. Tables 3-16 to 3-23 added.
29 June 2001	D	Configurations revised in Tables 3-13, 3-14, 3-18, 3-21, and 3-22. References to AP in Tables 3-8 to 3-23 changed to A to conform to configuration register.

Proprietary Notice

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`

Contents
ARM PrimeCell SDRAM Controller (PL170) Technical Reference Manual

Preface

This preface introduces the ARM PrimeCell SDRAM Controller (PL170) and its reference documentation. It contains the following sections:

About this document

This document is the technical reference manual for the ARM PrimeCell SDRAM Controller.

Intended audience

This document has been written for implementation engineers and architects, and provides a description of an optimal SDRAM controller architecture. The PrimeCell SDRAM Controller aims to help designers integrate high performance, low power, SDRAM into a design as quickly as possible.

Organization

This document is organized into the following chapters:

Chapter 1 ***Introduction***

Read this chapter for an introduction to the PrimeCell SDRAM Controller.

Chapter 2 ***Functional Overview***

Read this chapter for an overview of the PrimeCell SDRAM Controller in a unified memory system.

Chapter 3 ***Programmer's Model***

Read this chapter for a description of the registers and for details of system initialization.

Appendix A ***Signal Descriptions***

Read this appendix for a description of the PrimeCell SDRAM Controller signals.

Appendix B ***Command Descriptions***

Read this appendix for a description of the PrimeCell SDRAM Controller commands.

Typographical conventions

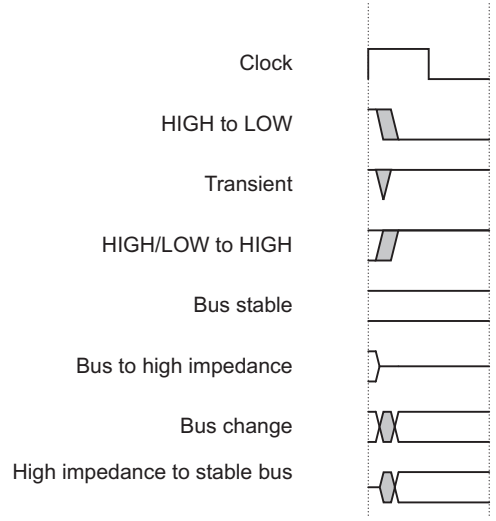
The following typographical conventions are used in this document:

bold Highlights ARM processor signal names, and interface elements such as menu names. Also used for terms in descriptive lists, where appropriate.

italic	Highlights special terminology, cross-references and citations.
`typewriter`	Denotes text that may be entered at the keyboard, such as commands, file names and program names, and source code.
`type`writer	Denotes a permitted abbreviation for a command or option. The underlined text may be entered instead of the full command or option name.
`typewriter italic`	Denotes arguments to commands or functions where the argument is to be replaced by a specific value.
`typewriter bold`	Denotes language keywords when used outside example code.

Timing diagram conventions

This manual contains one or more timing diagrams. *Key to timing diagram conventions* explains the components used in these diagrams. Any variations are clearly labeled when they occur. Therefore, no additional meaning should be attached unless specifically stated.



Key to timing diagram conventions

Shaded bus and signal areas are undefined, so the bus or signal can assume any value within the shaded area at that time. The actual level is unimportant and does not affect normal operation.

Further reading

This section lists publications by ARM Limited, and by third parties.

ARM periodically provides updates and corrections to its documentation. See `http://www.arm.com` for current errata sheets and addenda.

See also the ARM Frequently Asked Questions list at:
`http://www.arm.com/DevSupp/Sales+Support/faq.html`

ARM publications

AMBA Specification (Rev 2.0) (ARM IHI 00011)

ARM PrimeCell SDRAM (PL170) Integration Manual (PL170 INTM 0000).

ARM PrimeCell SDRAM (PL170) Design Manual (PL170 DDES 0000).

ARM PrimeCell EBI (PL170) Application Note (SC026 TRM 0000).

Feedback

ARM Limited welcomes feedback on both the ARM PrimeCell SDRAM Controller (PL170) and on the documentation.

Feedback on this document

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

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

General suggestions for additions and improvements are also welcome.

Feedback on the ARM PrimeCell SDRAM Controller

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

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

Chapter 1
Introduction

This chapter introduces the ARM PrimeCell SDRAM Controller (PL170) and contains the following section:

- *About the ARM PrimeCell SDRAM Controller (PL170)* on page 1-2.

1.1 About the ARM PrimeCell SDRAM Controller (PL170)

The PrimeCell SDRAM Controller is an *Advanced Microcontroller Bus Architecture* (AMBA) compliant, *System-on-a-Chip* (SoC) peripheral that is developed, tested and licensed by ARM.

The PrimeCell SDRAM Controller interfaces SDRAM to embedded SoC ASICs and ASSPs.

Refer to Figure 1-1 for a simplified block diagram of the PrimeCell SDRAM Controller.

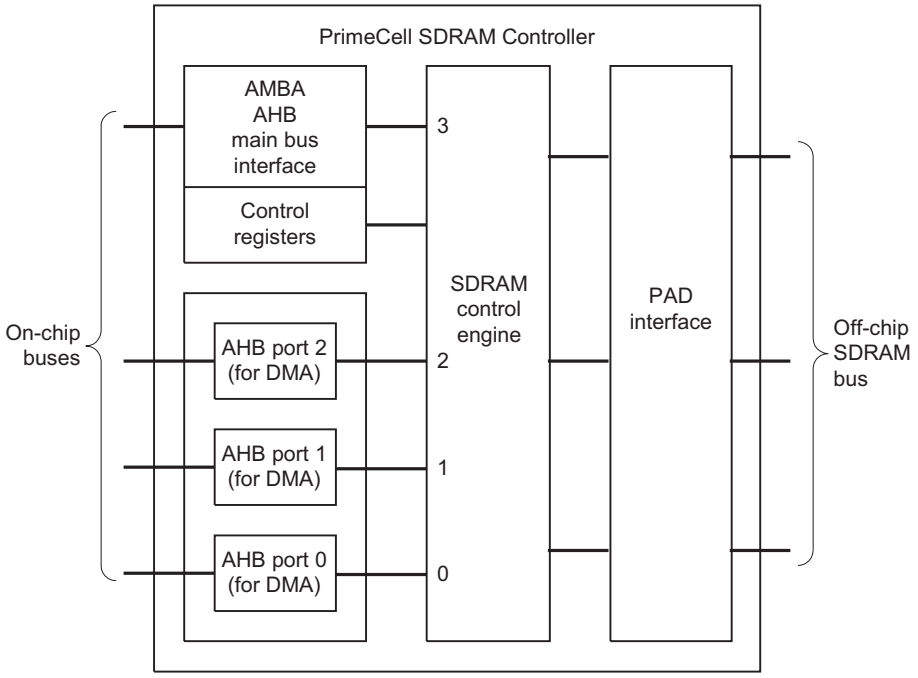


Figure 1-1 PrimeCell SDRAM Controller in a system

Note

Ports 0, 1 and 2 are optional slave AHB ports.

1.1.1 General information

shows the on-chip buses interfaced to the controller through the AMBA main AHB and optional slave AHB interfaces. The off-chip SDRAM bus is resynchronized to the PrimeCell SDRAM control engine through the pad interface. The main AHB interface also contains the control registers.

The four AHB interfaces can be altered to match the behavior of the PrimeCell SDRAM Controller to most applications. Bus interface variants can cover different buffering strategies depending on:

- whether the ARM processor is cached or not
- logic for different prefetch strategies
- off-chip data bus width conversion
- system-dependent control registers.

Abstracting out the pad interface permits the most appropriate internal to external bus resynchronization strategy to be adopted.

It is not envisaged that a user will need to modify the PrimeCell SDRAM control engine.

1.1.2 Features of the PrimeCell SDRAM Controller

The following features are provided by the PrimeCell SDRAM Controller:

- Compliance to the *AMBA Specification (Rev 2.0)* onwards for easy integration into SoC implementation.
- ARM BusTalk functional test environment used for validation.
- Compatible with ARM7, ARM9, and ARM10 processors.
- Optional AHB interface ports for system-specific, high-bandwidth, peripherals (e.g. *Direct Memory Access* (DMA) controllers).
- Optional data buffers are available in each AHB interface to improve system performance for example, when using uncached processors such as ARM7TDMI and ARM9TDMI.
- Four independently controlled chip selects.
- Data is transferred between the controller and the SDRAM in quad-word bursts.
- RTL can be compiled for a range of data bus widths. For example 32-bit AHB to 32-bit SDRAM, 32-bit AHB to 16-bit SDRAM.

- Three clock cycle latency from AMBA bus **HSELram** assertion to the issue of a SDRAM command.
- The controller engine supports the commands listed in Appendix B *Command Descriptions*.
- Provision to allow multiplexing of the external memory interface pins (address and data) with other memory controller signals (SRAM, ROM and so on). See the *ARM PrimeCell External Bus Interface (EBI) Application Note* for examples of how to multiplex memory interface signals.
- Two reset domains allow SDRAM contents to be preserved over a soft reset.
- Power saving modes dynamically control SDRAM **CKEOut[3:0]** and **CLKOut**.
- Controller supports 2K, 4K and 8K row address memory parts, i.e. typical 256MBit, 128MBit, 64MBit and 16MBit parts, with 8, 16 or 32 DQ bits per device.
- AHB ports support little- or big-endian byte order.

Chapter 2
Functional Overview

This chapter provides an overview of the ARM PrimeCell SDRAM Controller (PL170). It contains the following sections:

2.1 ARM PrimeCell SDRAM Controller (PL170) overview

Figure 2-1 shows the top-level hierarchy of the PrimeCell SDRAM Controller.

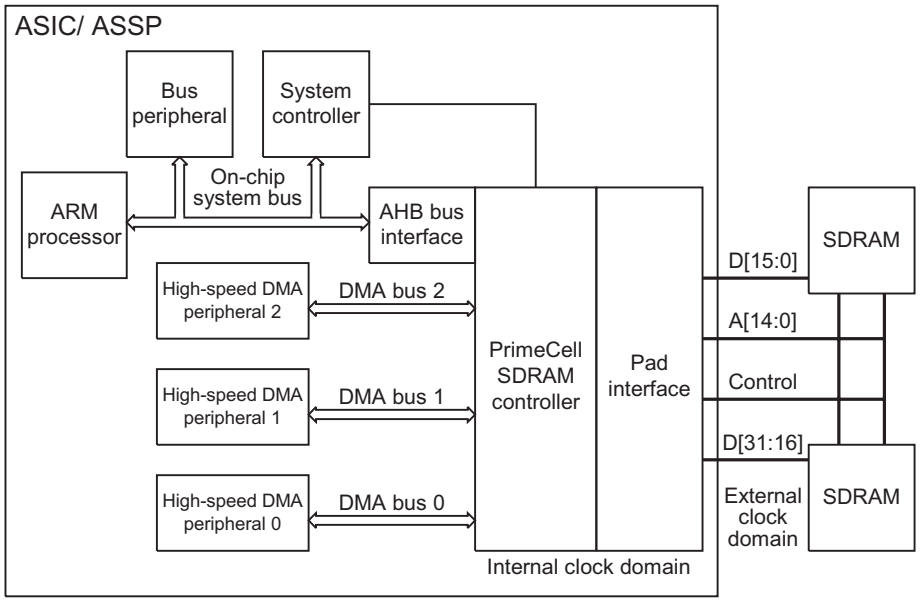


Figure 2-1 PrimeCell SDRAM Controller top level hierarchy

A minimum system comprises:

- *PrimeCell SDRAM control engine* on page 2-3
- *Main AHB interface* on page 2-3
- *Additional AHB ports* on page 2-5
- *Pad interface* on page 2-6.

Note

The additional AHB ports are optional, and can be included or omitted depending on the particular system requirements.

2.1.1 PrimeCell SDRAM control engine

The PrimeCell SDRAM control engine arbitrates between the access requests from the four AHB ports (connected to the main bus and DMA interfaces). As a result it generates a sequence of commands and issues these to the SDRAMs to transfer the requested data.

Arbitration

Arbitration is carried out so that port 0 has the highest priority, and port 3 the lowest. However, requests may get re-arbitrated depending on the ease of obtaining the data. For example, if ports 1 and 2 simultaneously request a data transfer and the port 1 request address is to a closed page (that is the requested page has to be opened), but port 2 address is for an already open page, the following sequence will occur:

1. ACT command to open the SDRAM row specified by the port 1 address.
2. Port 2 access.
3. Port 1 access.

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 port.

2.1.2 Main AHB interface

The main AHB interface is a slave on the main processor bus, it provides access to the SDRAM, and contains the control register block. For ARM processors which do not contain a write back cache (typically ARM7, and some ARM9s) a merging write buffer may optionally be included in the AHB interface. The PrimeCell SDRAM Controller does not support split transactions.

The bus interface produces read, write, refresh and mode register write requests to the SDRAM control engine, and software-supplied configuration information.

Data is transferred to and from the SDRAM as unbroken quad words. This is a convenient data packet size for ARM cache line fills, and buffered writes. For accesses smaller than a quad word, extra read data is discarded by the AHB interface. For writes, DQM pins (SDRAM data byte mask inputs) are used to force the SDRAMs to ignore invalid data. For access sizes larger than a quad word, multiple quad-word accesses are issued to the SDRAM control engine.

The INCR burst type does individual quad word transfers between the AHB interface logic and the controller engine. There are wait cycles between quad word transfers.

Non-aligned INCR4, INCR8 and INCR16 transfers issue an extra quad word to transfer tail-end data.

Control registers

The main bus interface has a sub-block containing a minimum of three control registers whose values are set by the processor, two configuration registers and the refresh timer. In addition to these registers, a write buffer time-out register may be added. The control registers are described in Chapter 3 *Programmer's Model*.

2.1.3 Optional features

The PrimeCell SDRAM Controller has the following optional features:

- *Merging write buffer*
- *Read buffer* on page 2-5.

Merging write buffer

A merging write buffer compacts writes of all widths into quad-word bursts, which can be efficiently transferred to the SDRAM. Simulation has shown ARM7, and uncached ARM9-based designs achieve approximately 15% improvements in data bandwidth by incorporating a write buffer in the PrimeCell SDRAM Controller.

Each AHB port of the SDRAM memory controller can include an 8-word merging write buffer. Operation of a write buffer is completely transparent to the programmer. Each write buffer is split into two halves. Each half holds a quad word, the size of the default SDRAM data burst length. The split into two quad words allows a new quad word to be buffered while the contents of the other quad-word buffer is transferred to memory. A quad-word buffer will merge noncontiguous word, halfword or byte writes to the same quad word address, using the same AHB port.

The buffers in each AHB port operate independently. The contents of a buffer will be transferred (flushed) to the SDRAM when:

- there is a write, to a SDRAM address outside the current merging quad word address
- there is a read, from the address of the merging quad word
- the write buffer timer reaches zero
- the write buffer is disabled.

The halves of the quad-word write buffer are exchanged when a write to a different quad word address starts (a write buffer miss). The buffer half previously merging data is flushed (its contents are written to SDRAM), while new data writes are merged in the new input buffer.

When a write back is triggered by a read from the memory address of the merging quad word, the write-back operation is completed before the requested data is read from memory. Write before read ensures the quad word returned is up-to-date (the buffer contents must be merged with the memory contents to ensure all the bytes of the quad word read from memory are up-to-date).

Each AHB port write buffer has an associated timer. The write buffer timer forces a write back (buffer flush) after a programmable delay. Any write to a buffer starts the associated timer by loading the time-out count into a down counter. The delay time is measured from the last write to the active half of a merging write buffer.

Note

A write buffer is not flushed by an access through another AHB port. As a result data coherency is not guaranteed. In most systems data coherency can be ensured by careful choice of the memory access sequence or by temporarily disabling the write buffers.

Read buffer

Each AHB port can include a read buffer. When present a read buffer holds the last quad word read through that AHB port. The read buffers in each AHB port work independently. Any read request to an address in the same quad word as the buffered data, uses the buffered data and will not cause an external memory access. A write, using the same AHB port, to an address in the same quad word as the buffered data will invalidate the buffered data, but only in the read buffer of that AHB port.

The read buffer will improve performance in systems without a CPU cache.

2.1.4 Additional AHB ports

The additional AHB ports are optional. They permit high-bandwidth peripherals direct access to the PrimeCell SDRAM Controller without data having to pass over the main system bus. The control engine prioritizes accesses and can hide the SDRAM ACT command (open a row).

With a single bus interface a data transfer to the PrimeCell SDRAM Controller can be halted whenever an access is requested to an address which is not available in an open page. With multiple bus interfaces, the halted time on one bus can be used to permit data transfer to another bus if its requested data is in an open page.

Each additional port has a simple AHB interface to allow easy connection of Primecell peripherals. An AHB interface does not support split transactions.

Any AHB port may include read and write buffers. These function in the same way as buffers in the main AHB interface port.

There are no coherency checks between any of the AHB ports. The buffer enable bits in configuration register 1 control all the instances of AHB port buffers. The write timeout register sets the timeout count for all the merging write buffers.

2.1.5 Pad interface

All data and control signals must be passed synchronously to the SDRAMs. Resynchronization from the ASIC clock domain to the external bus clock domain is achieved in this block. The external bus clock can either be fed back from the external clock (using a tristate pad), and used directly, or fed into a PLL so the phase of the external clock is fixed relative to the on-chip clock.

2.2 Overview of a Primecell SDRAM, ASIC/ASSP, unified memory system

Figure 2-2 shows the PrimeCell SDRAM Controller interfacing SDRAM chips to SoC ASIC peripherals.

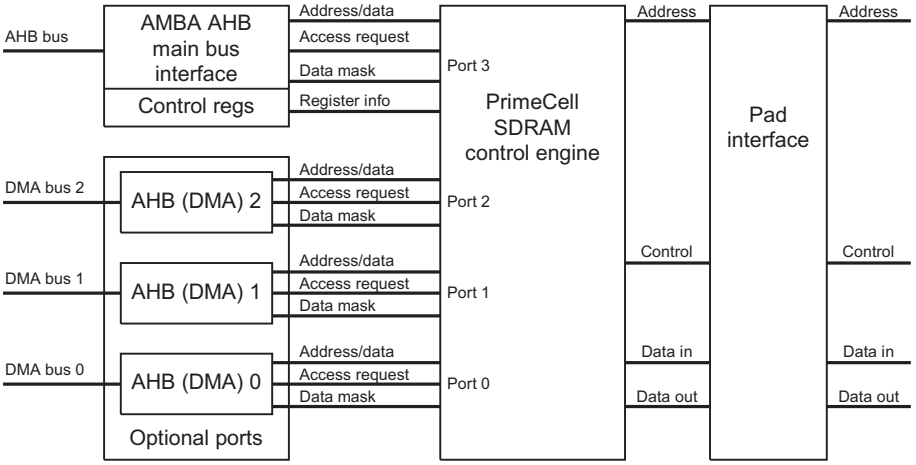


Figure 2-2 PrimeCell SDRAM Controller in a system

The system uses three types of bus:

- *External bus*
- *Internal bus* on page 2-8
- *DMA bus* on page 2-9 (optional).

2.2.1 External bus

The off-chip SDRAM bus (containing data, address and control signals) connects the ASIC or ASSP to the SDRAM.

Figure 2-3 on page 2-8 illustrates how the memory interface signals connect to a memory array for a 32 bit wide data bus. For a 16 bit wide data bus only Bytes 0 and 1 are used. **DQM[3:2]** and **DataI/O[31:16]** are left unconnected, or can be omitted from the external signals of the ASIC.

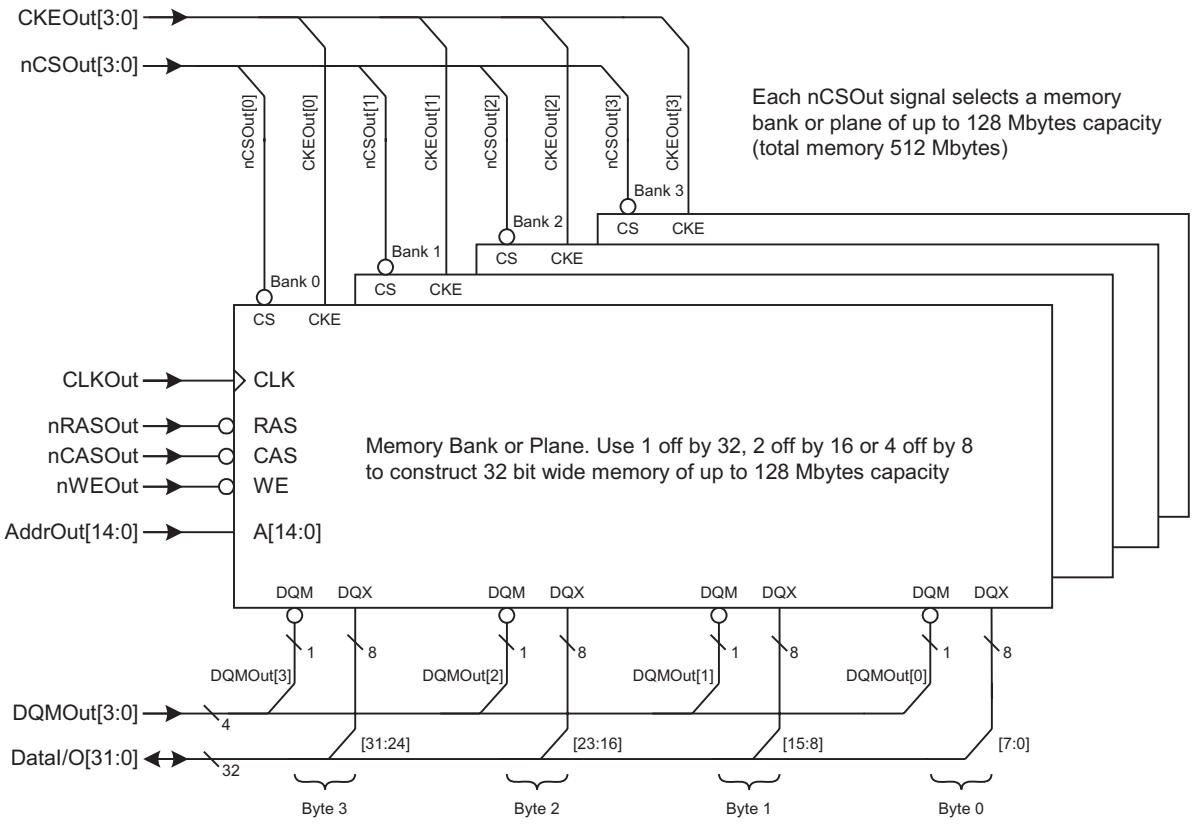


Figure 2-3 PrimeCell SDRAM Controller external bus

Note

Refer to *Address mapping* on page 3-11 for examples of how to connect **AddrOut[14:0]** to memory devices.

2.2.2 Internal bus

The on-chip bus enables communication between the on-chip peripherals. The PrimeCell SDRAM Controller appears as a standard slave on the on-chip bus and controls the SDRAM on the external bus.

2.2.3 DMA bus

To improve system performance, up to three DMA buses can also be interfaced to the PrimeCell SDRAM, using the optional additional AHB ports. The DMA channels permit multiple access requests to be presented to the controller at the same time. This enables it to hide many of the background operations (for example, bank activate and precharge), and so reduce the average system access latency.

Multi-port access

Multi-port access to the PrimeCell SDRAM Controller permits:

- higher memory bandwidth by hiding more of the nondata transfer operations
- potentially higher performance on the main AMBA bus by removing the heavy DMA traffic
- potentially less overall bus capacitance and therefore lower power consumption.

To permit maximum reuse flexibility the bus interface section has been separated from, and has a clean interface to, the PrimeCell SDRAM control engine.

Clock domains

Figure 2-1 on page 2-2 shows that there are two clock domains in a synchronous memory system, on-chip and off-chip. The off-chip clock domain will be phase-shifted with respect to the on-chip clock domain due to the pad delays. The on-to-off chip clock skew requires careful attention at clock speeds greater than 80MHz, so the pad interface has been separated from the main PrimeCell SDRAM control engine. The separation permits easy implementation of the most appropriate clock resynchronisation strategy taking into account interface speed and available pad library.

Maintaining memory during low-power sleep modes

In many systems, the contents of the memory system needs to be maintained during low-power sleep modes. To aid the design of this class of system, two features are provided in the PrimeCell SDRAM Controller:

- SDRAM refresh over soft reset
- a mechanism to place the SDRAMs into self-refresh mode.

The above features enable interaction with a system controller, which will typically be present to control the safe transition between power-up, reset, normal, and sleep modes. The associated control signals can be tied off if not required.

2.2.4 Example signal waveforms

This section contains figures which show the general sequence of signal transitions for write and read transfers. In these examples, the access is to an open SDRAM page.

Figure 2-4 on page 2-11 shows a PrimeCell SDRAM Controller write example (merging write buffer off).

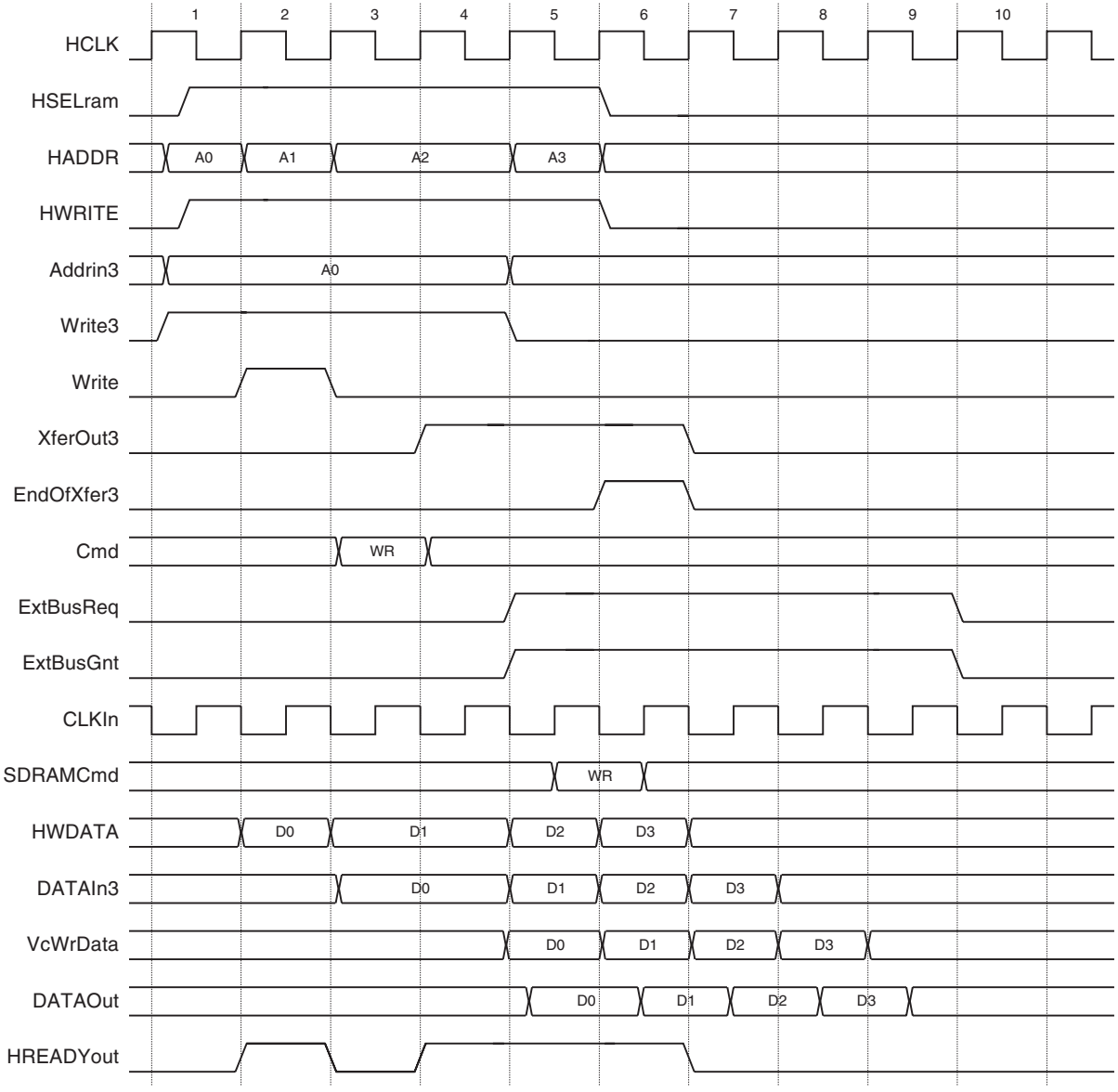


Figure 2-4 PrimeCell SDRAM Controller write example

Figure 2-5 on page 2-12 shows a second PrimeCell SDRAM Controller write example (delayed **ExtBusGnt** signal).

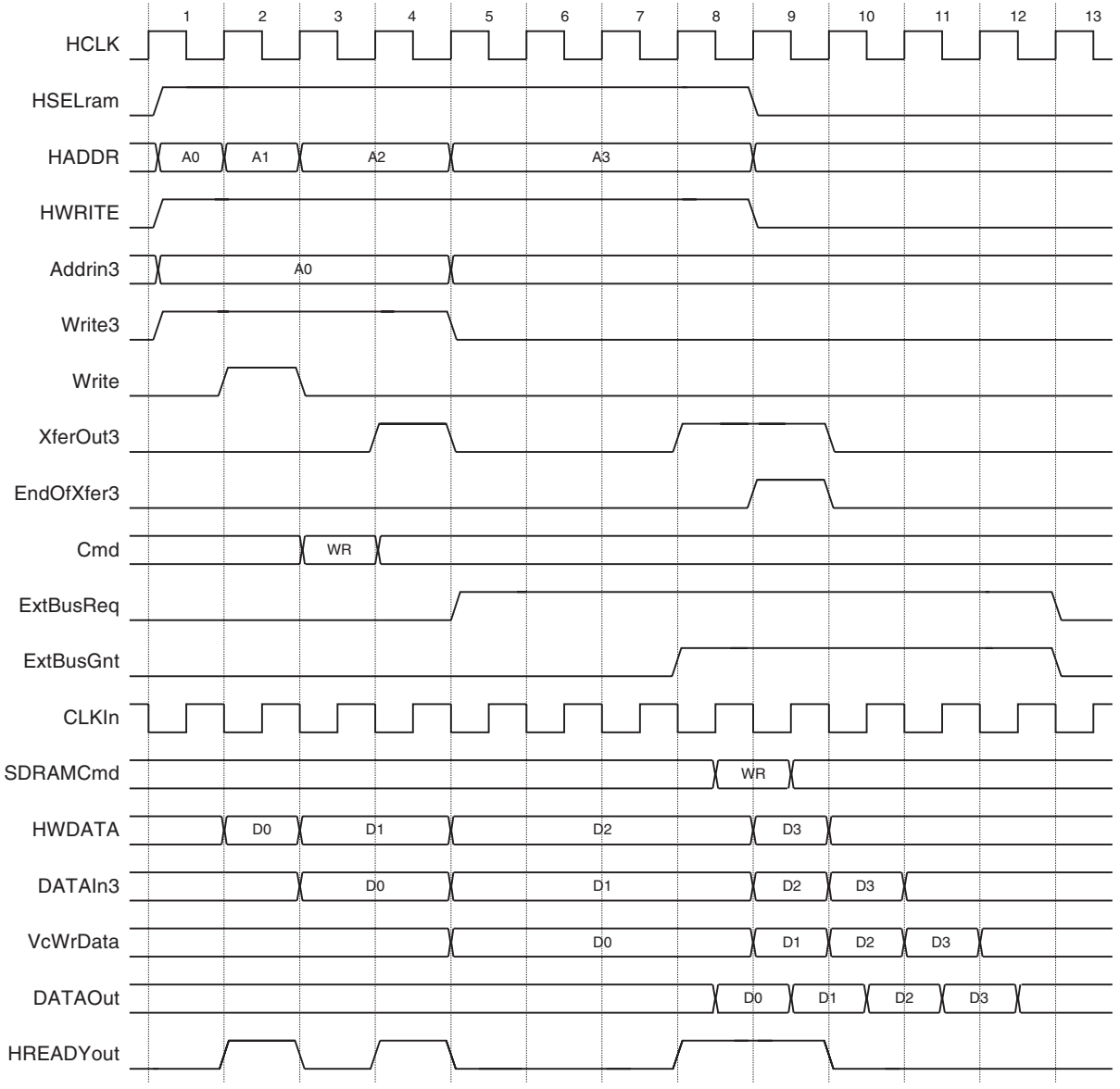


Figure 2-5 PrimeCell SDRAM Controller write example

Figure 2-6 on page 2-13 shows a PrimeCell SDRAM Controller read example.

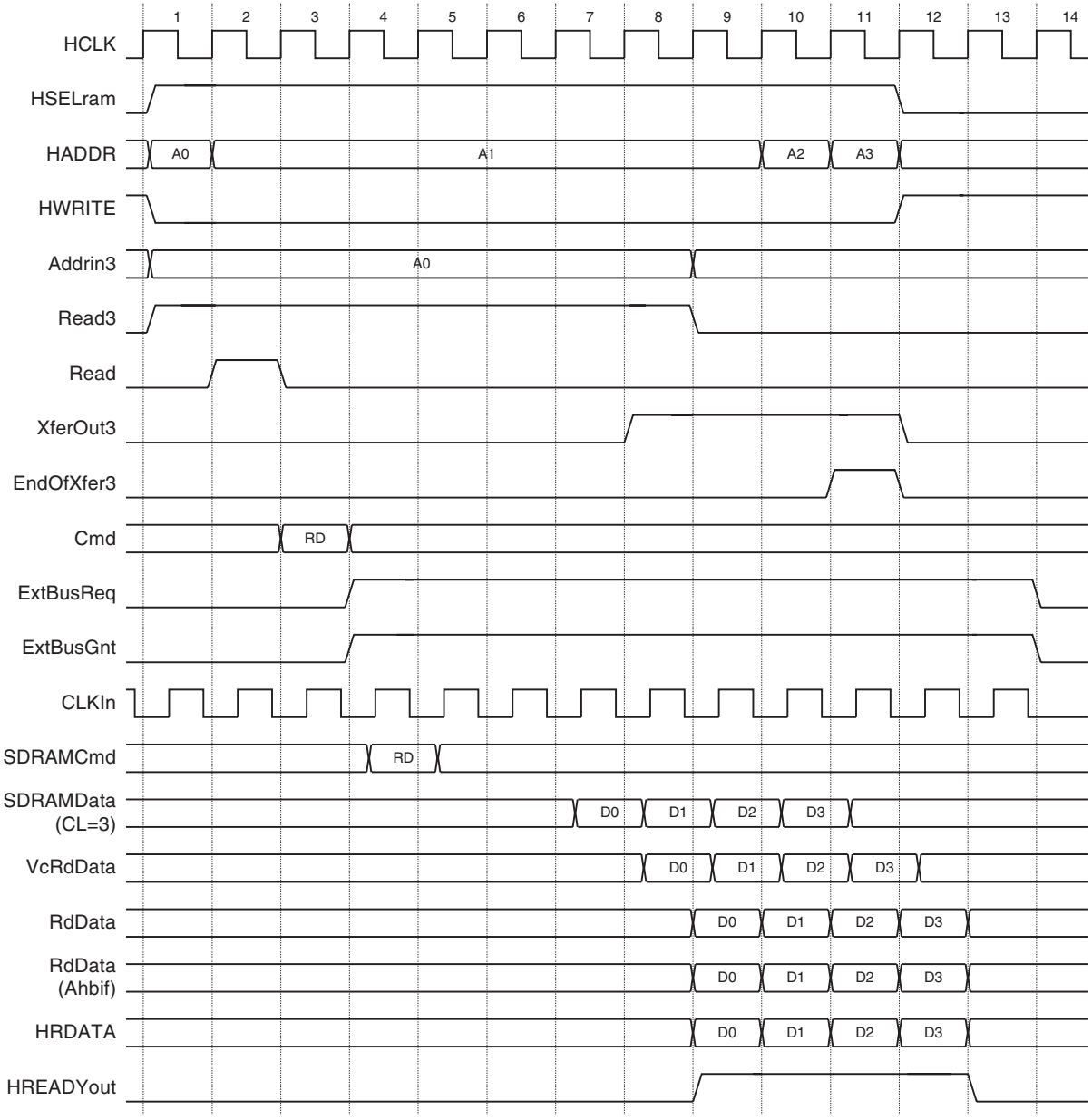


Figure 2-6 PrimeCell SDRAM Controller read example

Chapter 3
Programmer's Model

This chapter describes the registers of the PrimeCell SDRAM Controller (PL170), and provides details of system initialization. It contains the following sections:

3.1 About the programmer's model

The base address of the PrimeCell SDRAM Controller is fixed by the host ASIC AMBA address decoder. The PrimeCell SDRAM Controller registers are selected by **HSELreg3**. Address lines **HADDR3[3:2]** are used to select one of the registers when **HSELreg3** is asserted. The registers are word-aligned.

The main AHB port selects external memory when **HSELram3** is asserted. **HADDR3[28:0]** address the device bank, row, and column for memory transfers using the main AHB port.

3.2 Summary of PrimeCell SDRAM Controller registers

The PrimeCell SDRAM Controller registers are shown in Table 3-1.

Table 3-1 PrimeCell SDRAM register summary

Address	Type	Width	Reset value	Name	Description
SDRAM Base	Read/write	25	0x00F00000	Reg0[24:0]	Configuration register 0.
SDRAM Base + 0x04	Read/write	6	Note 1	Reg1[5:0]	Configuration register 1.
SDRAM Base + 0x08	Read/write	16	0x0080	Reg2[15:0]	Refresh timer register.
SDRAM Base + 0x0C	Read/write	16	0x0000	Reg3[15:0]	Write buffer time-out register, see note 2.

Note

1. Reset value 0x00 if no buffers
   Reset value of bit 2 is 1 if read buffer is present in design,
   Reset value of bit 3 is 1 if write buffer is present in design.
2. Register only present if write buffer(s) included in an AHB interface.

3.3 Register descriptions

The following PrimeCell SDRAM Controller registers are described in this section:

- *Configuration registers*
- *Refresh timer register* on page 3-8
- *Write buffer time-out register* on page 3-8.

3.3.1 Configuration registers

The configuration registers enable software to set a number of operating parameters for the SDRAM control engine. There are two configuration registers located in the PrimeCell SDRAM Controller bus interface:

- *Configuration register 0*
- *Configuration register 1* on page 3-7.

Configuration register 0

Configuration register 0 is a 32-bit wide read/write register. Table 3-2 shows the bit assignment for configuration register 0.

Table 3-2 Configuration register 0

Bits	Name	Type	Function
31:25, 16, 12, 8, 4, 0	-	-	Reserved for future expansion and should always be programmed with the binary value 0.
24	A	Read/write	AHB port auto pre-charge control for SDRAM accesses: 0 = No auto pre-charge 1 = Auto pre-charge (default value).
23, 22	R[1:0]	Read/write	RAS to CAS delay SDRAM mode: 00 Reserved 01 Reserved 10 RAS to CAS delay (reads) = 2 11 RAS to CAS delay (reads) = 3 (default value). For all writes and CAS latency 1 reads, add 1 clock cycle to these values.

Table 3-2 Configuration register 0 (continued)

Bits	Name	Type	Function
21, 20	C[1:0]	Read/write	CAS latency: 00 Reserved 01 CAS latency = 1 10 CAS latency = 2 11 CAS latency = 3 (default value).
19	X	Read/write	eXternal bus width: 0 = external bus width = 32 (default value) 1 = external bus width = 16.
18	C	Read/write	SDRAM clock enable (CKE) control: 0 = the clock enable of all IDLE devices are de-asserted to save power (default value) 1 = all clock enables are driven HIGH continuously.
17	E	Read/write	SDRAM clock control: E =1, **CLKOut** signal stops when all SDRAMs are idle. E = 0, **CLKOut** signal runs continuously (default value). The E bit should be set only when the dynamic CKE shutdown mode is active (E=1 and C=1 is not allowed).

Table 3-2 Configuration register 0 (continued)

Bits	Name	Type	Function
15, 11, 7, 3	B[3:0]	Read/write	Indicates whether the SDRAM attached to a chip select is a 2 or 4-bank device: 1 = 4-bank device. 0 = 2-bank device (default value). B[3] corresponds to nCSOut[3] B[2] corresponds to nCSOut[2] B[1] corresponds to nCSOut[1] B[0] corresponds to nCSOut[0].
14, 10, 6, 2	T[3:0]	Read/write	Sets the address multiplexing used for each chip select: 1 = x8 memory devices. 0 = x16 or x32 memory devices (default value). T[3] corresponds to nCSOut[3] T[2] corresponds to nCSOut[2] T[1] corresponds to nCSOut[1] T[0] corresponds to nCSOut[0].
13, 9, 5, 1	F[3:0]	Read/write	Sets the address multiplexing used for an active chip select, to work with 256M SDRAM. 1 = 256 MBit devices. 0 = not 256 MBit device (default value). F[3] corresponds to nCSOut[3]. F[2] corresponds to nCSOut[2]. F[1] corresponds to nCSOut[1]. F[0] corresponds to nCSOut[0].

Note

Software should not write to configuration register 0 when the SDRAM engine is busy. The SDRAM engine status, bit 5 in configuration register 1, can be used to check if the control engine is idle.

Configuration register 1

Configuration register 1 is a 32-bit wide read/write register. Table 3-3 shows the bit assignments for the configuration register 1.

Table 3-3 Configuration register 1

Bits	Name	Type	Function
31:6, 4	-	-	Reserved for future expansion and should always be programmed with the binary value 0.
5	B	Read	SDRAM engine status bit: 0 = SDRAM engine is idle. 1 = SDRAM engine is busy.
3	W	Read/write	Write buffer enable: 0 = Merging write buffer(s) disabled. 1 = Merging write buffer(s) enabled (default value). Disabling the write buffer(s) will flush any stored value(s) to the external memory.
2	R	Read/write	Read buffer enable: 0 = Read buffer(s) off. 1 = Read buffer(s) enabled.
1	M	Read/write	Control bit for memory device initialization.
0	I	Read/write	Control bit for memory device initialization.

Refer to Table 3-4 which shows the control bits for memory device initialization.

Table 3-4 Control bits for memory device initialization

I	M	Operation
1	1	Automatically issue NOP to the SDRAM.
1	0	Automatically issue a PALL to the SDRAM.
0	1	Enable SDRAM MODE command.
0	0	Normal operation (default value).

See *System initialization* on page 3-10 for an example of the use of I and M to initialize SDRAM after power on.

3.3.2 Refresh timer register

Refer to Table 3-5 which shows bit assignments for the refresh timer register.

Table 3-5 Refresh timer register

Bits	Name	Type	Function
15:0	-	Read/write	The refresh timer register is a 16-bit read/write register that is programmed with the number of **HCLK** ticks that should be counted between SDRAM refresh cycles.

For example, for the common refresh period of 16ms, and a **HCLK** frequency of 50MHz, the following value should be programmed into it:

$16 \times 10^{-6} \times 50 \times 10^{6} = 800$

The refresh timer is set to 128 by nPOR. To ensure a refresh interval of less than 16ms after reset, the minimum frequency of **HCLK** allowed is:

$128/(16 \times 10^{-6}) = 8MHz$

The refresh register should be written to as early as possible in the system start-up procedure, and in the first few cycles if the system clock is less than 8MHz.

3.3.3 Write buffer time-out register

This register is only present if write buffers are included in an AMBA AHB interface. Refer to Table 3-6 which shows bit assignments for the write buffer time-out register.

Table 3-6 Write buffer time-out register

Bits	Name	Type	Function
15:0	-	Read/write	The write buffer time-out register works with the optional merging write buffers. This 16-bit read/write register sets the delay time for a forced flush of a write buffer. This is useful to ensure, for instance, video display data is correctly updated.

A write to a merging write buffer loads the value in the timeout register into the time-out down counter of the buffer. When the time-out counter reaches 0 the merging write buffer content is written (flushed) to the external memory. The down counter is clocked by **HCLK**. Storing a value of 0 in the timeout register disables the write buffer timeout function.

3.4 System initialization

On power-on reset, software must initialize the PrimeCell SDRAM Controller and each of the SDRAMs connected to the controller. Check the SDRAM data sheet for the start up procedure, an example sequence is given below:

1. Wait 100µs to allow SDRAMs power and clocks to stabilize.
2. Set the I and M bits. This automatically issues a NOP to the SDRAMs.

   Note

   The M and I bits are in configuration register 1, see *Configuration registers* on page 3-4.

3. Wait 200µs.
4. Reset the M bit (I = 1, M = 0). This automatically issues a PRE-ALL to the SDRAMs.
5. Write 10 into the refresh timer register. This provides a refresh cycle every 10 clock cycles.
6. Wait for a time period equivalent to 80 clock cycles (8 refresh cycles).
7. Program the operational value into the refresh timer.
8. Select command write mode (I = 0, M = 1) and perform a read from each SDRAM connected to the controller. Address lines **HADDR[26:11]** encode the value output on the SDRAM address lines **AddrOut[14:0]**. See the SDRAM data sheet for the bit pattern programmed onto **AddrOut[14:0]**, for the required operational mode.
9. Program configuration register 0.

   Note

   Parameters programmed into the SDRAMs, such as burst length, RAS and CAS delays, must be consistent with the values written to configuration register 0. For X=0 (x32 data path), use incremental burst length 4. For X=1 (x16 data path), use incremental burst length 8.

10. Clear the M and I bits and set the other bits in configuration register 1 to their normal operational values.
11. The SDRAM is now ready for normal operation.

3.5 Address mapping

An AHB port accesses external SDRAM when its **HSELramx** signal is asserted. Table 3-7 shows the chip select address decoder truth table.

Table 3-7 Chip select address decoder truth table

HADDRx[28]	HADDRx[27]	nCSOut3	nCSOut2	nCSOut1	nCSOut0
0	0	1	1	1	0
0	1	1	1	0	1
1	0	1	0	1	1
1	1	0	1	1	1

Note

The symbol x represents the port numbers, 0, 1, 2, or 3.

Table 3-8 to Table 3-23 on page 3-14 give the AHB address bus to the SDRAM address AddrOut[14:0] mapping for various memory types. Memory type is selected by programming the T, B and F bits in configuration register 0. The mapping will only take effect if the define variable ADDRESS_MAPPING is set to 1.

Note

An * indicates the signal is not used.

** indicates the signal is fixed at logic 0.

*** indicates an internally-generated address.

The **HADDR** prefixes have been omitted from Table 3-8 to Table 3-23 on page 3-14.

Table 3-8 to Table 3-23 on page 3-14 show the address mapping for X=0 (32 bit external memory data width). Table 3-16 on page 3-13 to Table 3-23 on page 3-14 show the address mapping for X=1 (16 bit external memory data width)..

Table 3-8 16M SDRAM (1Mx16) T=0, B=0, F=0, X=0, use 11 as bank select

AddrOut	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Row	**	10*	**	10	11	21	20	19	18	17	16	15	14	13	12
Col	**	10*	**	10	A	25*	10*	9	8	7	6	5	4	3	2

Table 3-9 16M SDRAM (2Mx8) T=1, B=0, F=0, X=0, use 11 as bank select

AddrOut	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Row	**	11*	**	11	22	21	20	19	18	17	16	15	14	13	12
Col	**	11*	**	11	A	25*	10	9	8	7	6	5	4	3	2

Table 3-10 64M SDRAM (2Mx32, 4Mx16) T=0, B=1, F = 0, X=0, use 12 and 13 as bank selects

AddrOut	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Row	11*	10	11	23	22	21	20	19	18	17	16	15	14	13	12
Column	11*	10	11	23*	A	25*	24*	9	8	7	6	5	4	3	2

Table 3-11 64M SDRAM (8Mx8) T=1, B=1, F=0, X=0, use 12 and 13 as bank selects

AddrOut	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Row	11*	12	11	23	22	21	20	19	18	17	16	15	14	13	24
Col	11*	12	11	23*	A	25*	10	9	8	7	6	5	4	3	2

Table 3-12 128M SDRAM (16Mx8) T=1, B=1, F=0, X=0, use 12 and 13 as bank selects

AddrOut	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Row	11*	12	11	23	22	21	20	19	18	17	16	15	14	13	24
Col	11*	12	11	23*	A	25	10	9	8	7	6	5	4	3	2

Table 3-13 128M SDRAM (8Mx16, 4Mx32) T=0, B=1, F=0, X=0, use 12 and 13 as bank selects

AddrOut	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Row	11*	10	11	23	22	21	20	19	18	17	16	15	14	13	12
Col	11*	10	11	23*	A	25*	24	9	8	7	6	5	4	3	2

Table 3-14 256M SDRAM (16Mx16,) T=0, B=1, F=1, X=0, use 13 and 14 as bank selects

AddrOut	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Row	11	12	24	23	22	21	20	19	18	17	16	15	14	13	25
Col	11	12	24*	23*	A	26*	10	9	8	7	6	5	4	3	2

Table 3-15 256M SDRAM (32Mx8) T=1, B=1, F=1, X=0, use 13 and 14 as bank selects

AddrOut	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Row	11	12	24	23	22	21	20	19	18	17	16	15	14	13	25
Col	11	12	24*	23*	A	26	10	9	8	7	6	5	4	3	2

Table 3-16 16M SDRAM (1Mx16) T=0, B=0, F=0, X=1, use 11 as bank select

AddrOut	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Row	**	9*	**	9	11	10	20	19	18	17	16	15	14	13	12
Col	**	9*	**	9	A	24*	9*	8	7	6	5	4	3	2	**

Table 3-17 16M SDRAM (2Mx8) T=1, B=0, F=0, X=1, use 11 as bank select

AddrOut	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Row	**	10*	**	10	11	21	20	19	18	17	16	15	14	13	12
Col	**	10*	**	10	A	24*	9	8	7	6	5	4	3	2	**

Table 3-18 64M SDRAM (2Mx32, 4Mx16) T=0, B=1, F = 0, X=1, use 12 and 13 as bank selects

AddrOut	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Row	9*	10	9	22	11	21	20	19	18	17	16	15	14	13	12
Column	9*	10	9	22*	A	24*	23*	8	7	6	5	4	3	2	**

Table 3-19 64M SDRAM (8Mx8) T=1, B=1, F=0, X=1, use 12 and 13 as bank selects

AddrOut	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Row	11*	10	11	23	22	21	20	19	18	17	16	15	14	13	12
Col	11*	10	11	23*	A	24*	9	8	7	6	5	4	3	2	**

Table 3-20 128M SDRAM (16Mx8) T=1, B=1, F=0, X=1, use 12 and 13 as bank selects

AddrOut	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Row	11*	10	11	23	22	21	20	19	18	17	16	15	14	13	12
Col	11*	10	11	23*	A	24	9	8	7	6	5	4	3	2	**

Table 3-21 128M SDRAM (8Mx16) T=0, B=1, F=0, X=1, use 12 and 13 as bank selects

AddrOut	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Row	9*	10	9	22	11	21	20	19	18	17	16	15	14	13	12
Col	9*	10	9	22*	A	24*	23	8	7	6	5	4	3	2	***

Table 3-22 256M SDRAM (16Mx16) T=0, B=1, F=1, X=1, use 13 and 14 as bank selects

AddrOut	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Row	11	10	24	23	22	21	20	19	18	17	16	15	14	13	12
Col	11	10	24*	23*	A	25*	9	8	7	6	5	4	3	2	**

Table 3-23 256M SDRAM (32Mx8) T=1, B=1, F=1, X=1, use 13 and 14 as bank selects

AddrOut	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Row	11	10	24	23	22	21	20	19	18	17	16	15	14	13	12
Col	11	10	24*	23*	A	25	9	8	7	6	5	4	3	2	**

3.5.1 Remapping the AMBA address to the SDRAM address bus

The mapping presented in the previous section should satisfy the following two constraints:

- memory selected by each chip select should be contiguous
- the bank address should be changed on the page address for SDRAM.

Appendix A
Signal Descriptions

This appendix describes the signals that interface with the ARM PrimeCell SDRAM Controller (PL170). It contains the following sections:

A.1 On-chip signals

The on-chip signals are described in the following sections:

- *AMBA AHB signals*
- *Miscellaneous* on page A-4.

A.1.1 AMBA AHB signals

Refer to Table A-1 for an overview of the AMBA AHB signals.

Note

The PrimeCell SDRAM does not support split transactions.

All signals are prefixed with the letter H, ensuring that the AMBA AHB signals are differentiated from other similarly named signals in a system design.

Signal names for each DMA port can be found by substituting the port number, 0, 1 or 2, for the symbol x. Port 3 is the main AHB port and should be connected to the CPU bus.

Unused DMA ports are disabled by connecting their inputs to logic 0.

Table A-1 AMBA AHB signals

Name	Type	Source/ destination	Description
HCLK	Input	Clock source	This clock times all bus transfers. All signal timings are related to the rising edge of **HCLK**.
HRESETn	Input	Reset controller	The reset signal is active LOW and is used to reset the PrimeCell SDRAM Controller. This is the only active LOW signal.
HSELreg3	Input	Decoder	Selects the PrimeCell SDRAM Controller registers in the main AHB port (port 3). Address lines **HADDR[3:2]** are decoded to identify the target register.
HADDRx[28:0]	Input	Master	The AHB address bus.

Table A-1 AMBA AHB signals (continued)

Name	Type	Source/ destination	Description
HTRANSx[1:0]	Input	Master	Indicates the type of the current transfer, which can be NONSEQUENTIAL, SEQUENTIAL, IDLE or BUSY.
HWRITEx	Input	Master	When HIGH this signal indicates a write transfer and when LOW a read transfer.
HSIZEx[1:0]	Input	Master	Indicates the size of the transfer, which is typically byte (8-bit), halfword (16-bit) or word (32-bit).
HBURSTx[2:0]	Input	Master	Indicates if the transfer forms part of a burst. Four, eight and sixteen beat bursts are supported and the burst may be either incrementing or wrapping.
HWDATAx[31:0]	Input	Master	The write data bus is used to transfer data to the port during write operations. A minimum data bus width of 32 bits is recommended.
HSELramx	Input	Decoder	Selects a memory transfer.
HRDATAx[31:0]	Output	Slave	The read data bus is used to transfer data from the PrimeCell SDRAM Controller to the bus master or DMA device during read operations.
HREADYinx	Input	Slave	When HIGH the **HREADYinx** signal indicates that a transfer has finished on port x. This signal may be driven LOW by the bus master to extend a transfer.
HREADYoutx	Output	Slave	When HIGH the **HREADYoutx** signal indicates that a transfer has finished on port x. This signal may be driven LOW to extend a transfer.
HRESPx[1:0]	Output	Slave	The transfer response provides additional information on the status of a transfer. Only the OKAY response is returned by the PrimeCell SDRAM Controller.

A.1.2 Miscellaneous

Miscellaneous signals are shown in Table A-2.

Table A-2 Miscellaneous signals

Signal	Type	Source/ destination	Function
nPOR	Input	Reset controller	Power on reset (active low).
SREFReq	Input	Power manager	Self-refresh request.
SREFAck	Output	Power manager	Self-refresh acknowledgement.
BIGENDIAN	Input	System	Selects byte endian order.
SCANENABLE	Input	Test controller	Place holder for **SCANENABLE** signal.
SCANIN	Input	Test controller	Place holder for **SCANIN** signal.
SCANOUT	Output	Test controller	Place holder for **SCANOUT** signal.

Note

When **SREFReq** is asserted, the controller closes any open memory banks, then puts the memory into self-refresh mode. **SREFAck** indicates that the memory interface signals are set to command the self-refresh state. The system must ensure that the memory subsystem is idle before asserting **SREFReq**. De-asserting **SREFReq** will return the memory to normal operation. A burst of refresh cycles should be issued before the memory is accessed. Refer to the memory data sheet for refresh requirements.

When **BIGENDIAN** is asserted, all AHB ports operate with big-endian byte order. This is achieved by inverting the two LSB address lines of each AHB port. Internal operation is not affected. External memory always uses little-endian byte order.

A.2 Off-chip signals

The off-chip signals are shown in Table A-3.

Table A-3 Off-chip signals

Signal	Type	Source/ destination	Function
CLKOut	Output	Pad	SDRAM clock
CLKIn	Input	Pad	External SDRAM clock
CKEOut[3:0]	Output	Pad	SDRAM clock enables
nCSOut[3:0]	Output	Pad	SDRAM chip selects
AddrOut[14:0]	Output	Pad/ multiplexor	Address output for SDRAMs
nRASOut	Output	Pad	Row address strobe
nCASOut	Output	Pad	Column address strobe
nWEOut	Output	Pad	Write enable
DataIn[31:0]	Input	Pad/ multiplexor	Read data from the SDRAMs
DataOut[31:0]	Output	Pad/ multiplexor	Data output to SDRAMs
DQMOut[3:0]	Output	Pad	Data mask output to SDRAMs
DataEn	Output	Pad/ multiplexor	Data enable for write transfers
ExtBusReq	Output	Pad arbiter	External address/data bus request
ExtBusGnt	Input	Pad arbiter	External address/data bus grant
ExtCtlReq	Output	Pad arbiter	External control bus request
ExtCtlGnt	Input	Pad arbiter	External control bus grant

A.2.1 SDRAM memory interface signals

The SDRAM memory interface signals are arranged into three groups:

First group

The first group of signals connect directly to the external pads and are not influenced by any Req/Gnt control logic:

- **CKEOut[3:0]**
- **DQMOut[3:0]**
- **CLKOut.**

Second group

The second group are SDRAM control signals which may be multiplexed. When the PrimeCell SDRAM Controller requires access to these signals it asserts **ExtCtlReq**.

SDRAM commands which do not require an address will sample **ExtCtlGnt** and proceed when it is TRUE. Commands which also use the address bus will sample **ExtBusGnt** (**ExtCtlGnt** is ignored). These commands proceed when **ExtBusGnt** is TRUE. In this case, the arbiter for external pins must ensure that both control and address bus groups are available, and assert **ExtCtlGnt** and **ExtBusGnt** together. In most implementations **ExtCtlGnt** is connected to **ExtCtlReq**, as it is unlikely that any other interface will share pins with these signals.

- **nRASOut**
- **nCASOut**
- **nWEOut**
- **nCSOut[3:0]**.

Third group

Group three contains the memory data bus and address bus. When the PrimeCell SDRAM Controller requires access to data or address lines it asserts **ExtBusReq**. The controller waits for **ExtBusGnt** before proceeding.

- **AddrOut[14:0]**
- **DataOut/DataIn[31:0]**.

ExtCtlReq, ExtCtlGnt, ExtBusReq and **ExtBusGnt** signals are active HIGH and timed from **HCLK**. It is assumed that a memory interface arbiter, for multiplexed memory interface pins, will operate from **HCLK**. When a signal group has sole use of the external interface pins, the **relevant grant** signal should be tied HIGH or connected directly to the corresponding request signal. The controller only asserts a request when access is required to signal pins in the corresponding group. An arbiter should not require any other output signals from the PrimeCell SDRAM Controller.

Once asserted a grant signal should remain HIGH until the request is deasserted.

Appendix B
Command Descriptions

This appendix describes the commands that are supported by the ARM PrimeCell SDRAM Controller (PL170) engine. It contains the following section:

- *Commands* on page B-2.

B.1 Commands

The SDRAM contoller engine supports the commands shown in Table B-1.

Table B-1 Commands

Mnemonic	Operation
ACT	Opens an SDRAM row.
REF	CAS before RAS style refresh.
SREF	Self refresh.
PRE	Precharge, closes a bank.
RD	Read from an open row, row left open.
WR	Write to an open row, row left open.
RDA	Read followed by precharge.
WRA	Write followed by precharge.
MRS	Mode register set, programs SDRAM mode register.
NOP	No operation, used during the SDRAM initilization sequence.
PALL	Precharge all, used during the SDRAM initilization sequence.

Index

S

W

