

ARM PrimeCell™

Real Time Clock (PL031)

Technical Reference Manual



ARM PrimeCell™

Technical Reference Manual

Copyright © 2001 ARM Limited. All rights reserved.. All rights reserved.

Release Information

The following changes have been made to this document.

Change history

Date	Issue	Change
August 2001	A	First release
October 2001	B	Signal name nRtcRST changed to nRTCST . Additional interface reset information added.

Proprietary Notice

Words and logos marked with ® or ™ are registered trademarks or trademarks owned by ARM Limited, 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>

Contents

ARM PrimeCell Technical Reference Manual

Preface

About this document	x
Further reading	xii
Feedback	xiii

Chapter 1

Introduction

1.1 About the ARM PrimeCell Real Time Clock (PL031)	1-2
---	-----

Chapter 2

Functional Overview

2.1 ARM PrimeCell Real Time Clock (PL031) overview	2-2
2.2 PrimeCell RTC functional description	2-5
2.3 PrimeCell RTC operation	2-9

Chapter 3

Programmer's Model

3.1 About the programmer's model	3-2
3.2 Summary of PrimeCell RTC registers	3-3
3.3 General registers	3-4
3.4 Peripheral identification registers, RTCPeriphID0-3	3-8
3.5 PrimeCell identification registers, RTCCPCellID0-3	3-11
3.6 Interrupts	3-13

Chapter 4	Programmer's Model for Test	
4.1	PrimeCell RTC test harness overview	4-2
4.2	Scan testing	4-3
4.3	Test registers	4-4
4.4	Integration testing of block inputs	4-7
4.5	Integration testing of block outputs	4-8
4.6	Integration test summary	4-9
Appendix A	ARM PrimeCell Real Time Clock (PL031) Signal Descriptions	
A.1	AMBA APB signals	A-2
A.2	On-chip signals	A-3

List of Tables

ARM PrimeCell Technical Reference Manual

	Change history	ii
Table 3-1	PrimeCell RTC register summary	3-3
Table 3-2	RTCDR register	3-4
Table 3-3	RTCMR register	3-4
Table 3-4	RTCLR register	3-5
Table 3-5	RTCCR register	3-5
Table 3-6	RTCIMSC register	3-6
Table 3-7	RTCRIS register	3-6
Table 3-8	RTCMIS register	3-7
Table 3-9	RTCICR register	3-7
Table 3-10	Read-only registers	3-8
Table 3-11	RTCPeriphID0 register	3-9
Table 3-12	RTCPeriphID1 register	3-9
Table 3-13	RTCPeriphID2 register	3-9
Table 3-14	RTCPeriphID3 register	3-10
Table 3-15	RTCPCellID0 register	3-11
Table 3-16	RTCPCellID1 register	3-12
Table 3-17	RTCPCellID2 register	3-12
Table 3-18	RTCPCellID3 register	3-12
Table 4-1	Test registers memory map	4-4
Table 4-2	RTCITCR register	4-4
Table 4-3	RTCITOP register	4-5
Table 4-4	RTCTOFFSET register	4-5

List of Tables

Table 4-5	RTCTCOUNT register	4-6
Table 4-6	PrimeCell RTC integration test strategy	4-9
Table A-1	AMBA APB signal descriptions	A-2
Table A-2	On-chip signals	A-3

List of Figures

ARM PrimeCell Technical Reference Manual

	Key to timing diagram conventions	xi
Figure 1-1	PrimeCell RTC connection diagram	1-2
Figure 2-1	AMBA APB write access	2-3
Figure 2-2	AMBA APB read access	2-3
Figure 2-3	Interrupt generation	2-4
Figure 2-4	PrimeCell RTC block diagram	2-5
Figure 3-1	Peripheral identification register bit assignment	3-8
Figure 3-2	PrimeCell identification register bit assignment	3-11
Figure 4-1	PrimeCell RTC test harness	4-2
Figure 4-2	Output integration test harness, intra-chip outputs	4-8

Preface

This preface introduces the ARM PrimeCell Real Time Clock (PL031) and its reference documentation. It contains the following sections:

- *About this document* on page x
- *Further reading* on page xii
- *Feedback* on page xiii.

About this document

This document is the technical reference manual for the ARM PrimeCell Real Time Clock (PL031).

Intended audience

This document has been written for experienced hardware and software engineers who are implementing a basic alarm function or long time base counter using the ARM PrimeCell Real Time Clock (PL031).

Organization

This document is organized as follows:

Chapter 1 *Introduction*

Read this chapter for an introduction to the PrimeCell *Real Time Clock* (RTC) and its features.

Chapter 2 *Functional Overview*

Read this chapter for a description of the major functional blocks of the PrimeCell RTC.

Chapter 3 *Programmer's Model*

Read this chapter for a description of the PrimeCell RTC registers and programming details.

Chapter 4 *Programmer's Model for Test*

Read this chapter for an description of the logic in the PrimeCell RTC for functional verification and production testing.

Appendix A *ARM PrimeCell Real Time Clock (PL031) Signal Descriptions*

Read this appendix for details of the PrimeCell RTC signals.

Typographical conventions

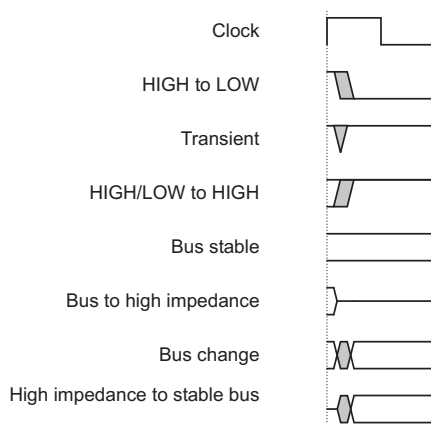
The following typographical conventions are used in this book:

<i>italic</i>	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.

<code>monospace</code>	Denotes text that can be entered at the keyboard, such as commands, file and program names, and source code.
<u><code>monospace</code></u>	Denotes a permitted abbreviation for a command or option. The underlined text can be entered instead of the full command or option name.
<i><code>monospace italic</code></i>	Denotes arguments to commands and functions where the argument is to be replaced by a specific value.
<code>monospace bold</code>	Denotes language keywords when used outside example code.

Timing diagram conventions

This manual contains one or more timing diagrams. The following key explains the components used in these diagrams.



Key to timing diagram conventions

Any variations are clearly labeled when they occur. Therefore, no additional meaning should be attached unless specifically stated.

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 from ARM Limited that provide additional information on developing code for the ARM family of processors.

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

This book contains information on the ARM PrimeCell Real Time Clock (PL031). Refer to the following books for related information:

- *AMBA Specification (Rev 2.0)* (ARM IHI 0011)
- *ARM PrimeCell RTC (PL031) Integration Manual* (PL031 INTM 0000)
- *ARM PrimeCell RTC (PL031) Design Manual* (PL031 DDES 0000).

Feedback

ARM Limited welcomes feedback on both the ARM PrimeCell Real Time Clock (PL031), and 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 Real Time Clock (PL031)

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 Real Time Clock (PL031). It contains the following sections:

- *About the ARM PrimeCell Real Time Clock (PL031)* on page 1-2.

1.1 About the ARM PrimeCell Real Time Clock (PL031)

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

The PrimeCell RTC is an AMBA slave module that connects to the *Advanced Peripheral Bus* (APB).

The PrimeCell RTC can be used to provide a basic alarm function or long time base counter. This is achieved by generating an interrupt signal after counting for a programmed number of cycles of a real-time clock input. Counting in one second intervals is achieved by use of a 1Hz clock input to the PrimeCell RTC. Figure 1-1 shows the connections to the PrimeCell RTC.

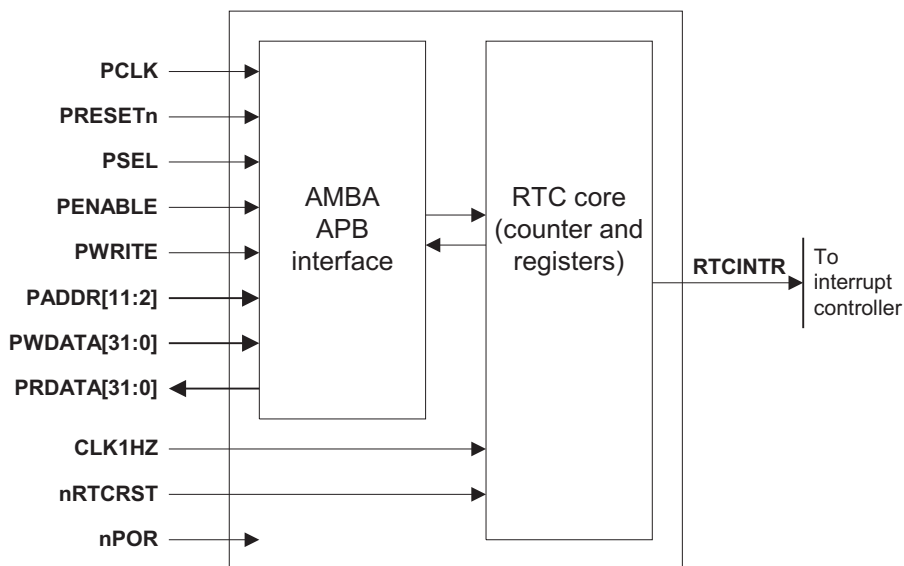


Figure 1-1 PrimeCell RTC connection diagram

1.1.1 Features of the PrimeCell RTC

The features of the PrimeCell RTC are:

- Compliance to the *AMBA Specification (Rev 2.0)* onwards for easy integration into SoC implementation.
- 32-bit up counter (free-running counter).
- Programmable 32-bit match compare register.
- Software maskable interrupt when counter and compare registers are identical.

Additional test registers and modes are implemented for functional verification and manufacturing test.

Chapter 2

Functional Overview

This chapter describes the major functional blocks of the ARM PrimeCell Real Time Clock (PL031). It contains the following sections:

- *ARM PrimeCell Real Time Clock (PL031) overview* on page 2-2
- *PrimeCell RTC functional description* on page 2-5
- *PrimeCell RTC operation* on page 2-9.

2.1 ARM PrimeCell Real Time Clock (PL031) overview

The ARM PrimeCell RTC (PL031) comprises:

- an AMBA APB interface
- a 32-bit counter
- a 32-bit match register
- a 32-bit comparator.

The CPU reads and writes data, and control and status information through the AMBA APB interface.

The 32-bit counter is incremented on successive rising edges of the input clock **CLK1HZ**. Counting in one second intervals is achieved by using a 1Hz clock signal for **CLK1HZ**. The counter is free-running and cannot be loaded. On reset, the counter:

- counts up from one
- reaches the maximum value, 0xFFFFFFFF
- wraps around to zero and continues incrementing.

RTC is loaded or updated by writing to the load register, **RTCLR**.

Reading the data register, **RTCDR**, gives the current value of the RTC.

The match register is programmed by writing to **RTCMR**. The counter and match values are compared in a comparator. When both values are equal, the interrupt **RTCINTR** is asserted HIGH. The CPU can use the interrupt to implement a basic time alarm function. The interrupt is cleared by writing any data value to the interrupt clear register **RTCICR**. The value in the match register can be read at any time.

The interrupt **RTCINTR** can be masked by writing to the interrupt match set or clear register, **RTCIMSC**. The raw status of the interrupt can be obtained by reading the **RTCRIS** register, and the masked version can be read from the **RTCMIS** register.

Synchronization logic is implemented to prevent propagation of metastable values when reading **RTCDR**. This ensures the stability of the data, even at the point that the counter is incrementing.

Figure 2-1 on page 2-3 shows an AMBA APB write access.

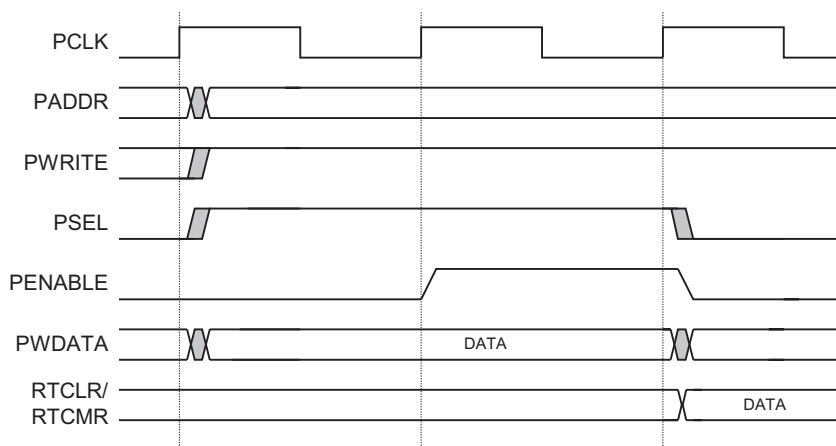


Figure 2-1 AMBA APB write access

Figure 2-2 shows an AMBA APB read access.

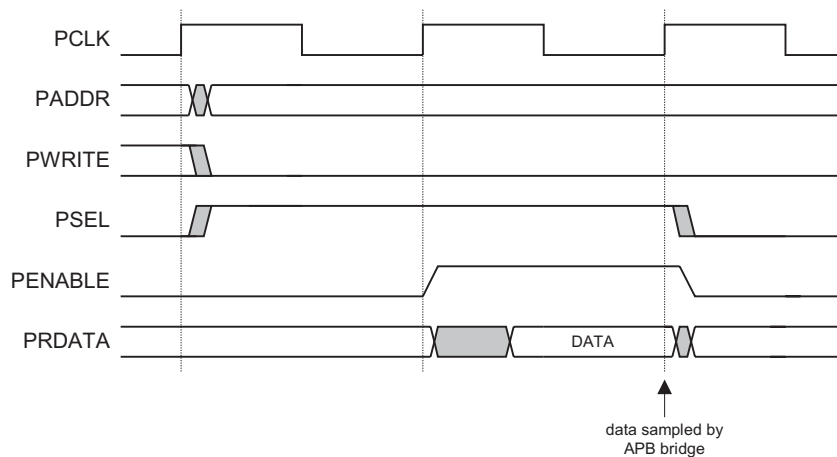


Figure 2-2 AMBA APB read access

Figure 2-3 on page 2-4 shows the interrupt generation when the current RTC value (RTCDR) value equals the match register value.

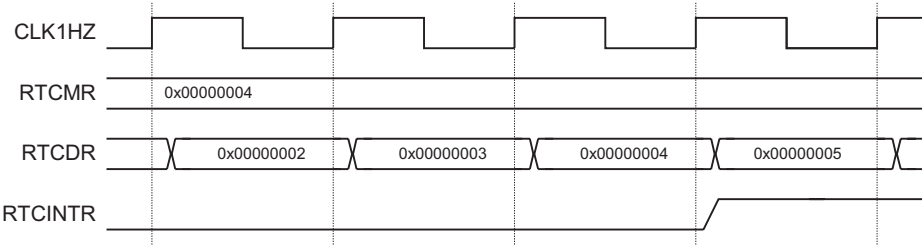


Figure 2-3 Interrupt generation

2.2 PrimeCell RTC functional description

Figure 2-4 shows the functionality of the PrimeCell RTC.

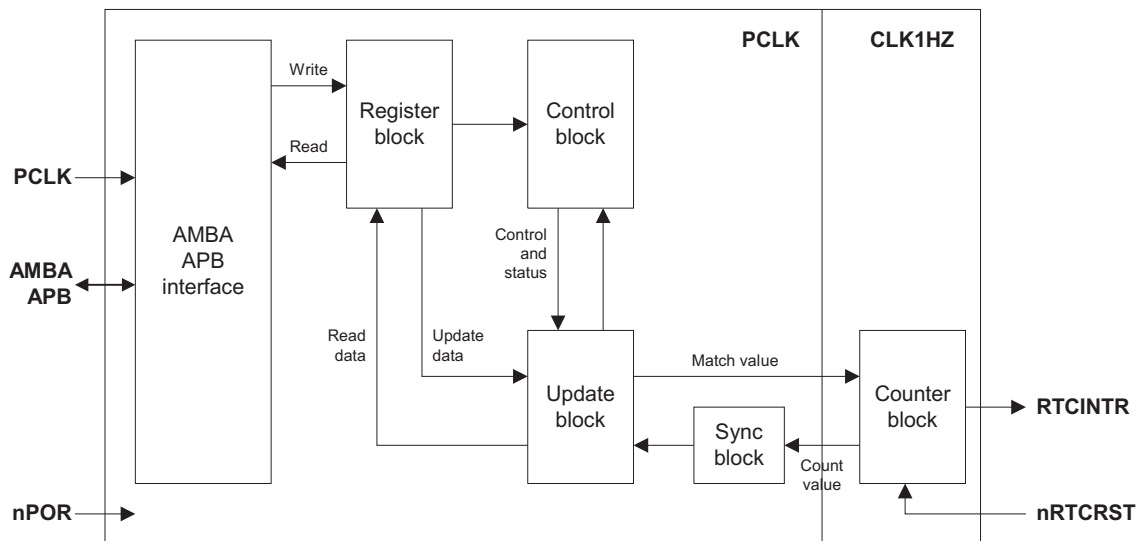


Figure 2-4 PrimeCell RTC block diagram

The functions of the PrimeCell RTC are described in the following sections:

- *AMBA APB interface* on page 2-6
- *Register block* on page 2-6
- *Control block* on page 2-6
- *Update block* on page 2-6
- *Synchronization block* on page 2-7
- *Counter block* on page 2-7
- *Test register and logic* on page 2-8.

2.2.1 AMBA APB interface

The AMBA APB interface generates read and write decodes for accesses to data, and control and status registers.

The AMBA APB is a local secondary bus which provides a low-power extension to the higher bandwidth *Advanced High-performance Bus* (AHB) or *Advanced System Bus* (ASB), within the AMBA system hierarchy. The AMBA APB groups narrow-bus peripherals to avoid loading the system bus and provides an interface using memory-mapped registers that are accessed under programmed control.

2.2.2 Register block

The register block stores data, written or to be read across the AMBA APB interface.

2.2.3 Control block

The control block is in the **PCLK** domain. Control and status signals from the control block are applied to the update block to control the generation of the RTC value and any updates to it.

A comparator is used to assert the interrupt **RTCINTR** when the current RTC value and match-compare register values are identical. Synchronization logic is also implemented in the control block because **PCLK** and **CLK1HZ** might be asynchronous.

2.2.4 Update block

The update block is used to calculate the update value of the RTC. The update block also generates an *equivalent match value* to be compared with the counter value in the **CLK1HZ** domain. It asserts an interrupt when the two values are equal.

There is a distinction between the RTC value and counter value:

- An update value (through the RTCLR register) is applied to the counter value in the update block. The resulting offset is applied to the current counter value to generate an updated RTC value.
- Reads from the data register (RTCDR) return the current value of the RTC alone and not that of the counter, as they are different.

Generally, an update to the absolute RTC value occurs after two rising clock edges of **PCLK**.

Also, an RTC enable bit is set when 1 is written to bit[0] of the RTC control register. When set, the RTC is started and subsequent writes have no effect. A read of bit[0] indicates the status of the RTC enable signal.

Note

The Offset register is zero on reset. It clocks through the offset value only when an update value is written to the RTC load register and holds that value until the next update is written.

2.2.5 Synchronization block

Because **PCLK** and **CLK1HZ** can be asynchronous, a synchronization block is included to implement synchronization logic that prevents the propagation of metastable values when there is transfer of data or control signals from one clock domain to another.

CLK1HZ to PCLK

The signals to be synchronized are:

Counter value This value in the Update block is used to calculate updates to the value of the RTC and to calculate the equivalent match value.

Raw and masked interrupts

These are synchronized to the **PCLK** domain registers:

RTCRIS Raw Interrupt Status register.

RTCMIS Masked Interrupt Status register.

2.2.6 Counter block

The counter is a free-running 32-bit counter that increments by one on each rising edge of **CLK1HZ**. The counter wraps from 0xFFFFFFFF to 0x00000000 on overflow and continues incrementing. The counter is free-running and cannot be loaded directly. It counts up from 0x00000001 on reset.

2.2.7 Test register and logic

There are test registers and logic implemented for functional block verification, and manufacturing and production testing using test vectors, that are applied through a *Test Interface Controller* (TIC) AMBA bus master block.

The test logic allows generation of a test clock enable signal, that propagates test vectors to the input signals of the block and captures values on the block outputs.

———— **Note** ————

Test registers must not be read or written to during normal use.

————

2.3 PrimeCell RTC operation

The operation of the PrimeCell RTC is described in the following sections:

- *Interface reset*
- *Clock signals*
- *PrimeCell RTC operation.*

2.3.1 Interface reset

The PrimeCell RTC requires three reset signals to reset the various parts, **nRTCRST**, **nPOR**, and **PRESETn**. **PRESETn** must be asserted LOW for a period long enough to reset the slowest block in the on-chip system, and then be taken HIGH again. The PrimeCell RTC requires **PRESETn** to be asserted LOW for at least one period of **PCLK**. **PRESETn** is used to reset most of the logic clocked by **PCLK**.

The interrupt output **RTCINTR** is LOW after reset.

Power on Reset (**nPOR**) resets the match register and offset register, and must therefore be deasserted synchronously to **PCLK**.

nRTCRST is used to reset the logic in the **CLK1HZ** domain, and must therefore be deasserted synchronously to **CLK1HZ**. In addition, **nRTCRST** must only be generated as a result of **nPOR**, and not a soft reset. Failure to do this results in the loss of the RTC value. **PRESETn** can be generated as a result of either **nPOR** or a soft reset.

The values of registers after reset are defined in Chapter 3 *Programmer's Model*.

2.3.2 Clock signals

The period of the clock signal **CLK1HZ** must be selected to determine the resolution of the RTC. For example, selecting a 1Hz clock signal produces a one second counter resolution.

There is a constraint on the ratio of clock frequencies for **PCLK** to **CLK1HZ**. The frequency of **PCLK** must be greater than three times the frequency of **CLK1HZ**.

$$F_{PCLK} > 3 \times F_{CLK1HZ}$$

2.3.3 PrimeCell RTC operation

After reset, values must be written to the load register **RTCLR** and match register **RTCMR**.

The counter increments by 1 on the rising edge of **CLK1HZ**.

To enable the interrupt, set the **RTCIMSC** register by writing a 1.

When the counter and match registers are identical, and the interrupt is not masked, the interrupt **RTCINTR** is asserted HIGH. The interrupt is cleared by writing 1 to the interrupt clear register **RTCICR**.

By using a 1Hz clock signal for **CLK1HZ**, the counter increments in one second intervals. This can be used to implement a real-time clock function in software as well as a basic alarm time function.

Chapter 3

Programmer's Model

This chapter describes the ARM PrimeCell Real Time Clock (PL031) registers and provides details required when programming the microcontroller. It contains the following sections:

- *About the programmer's model* on page 3-2
- *Summary of PrimeCell RTC registers* on page 3-3
- *General registers* on page 3-4
- *Peripheral identification registers, RTCPeriphID0-3* on page 3-8
- *PrimeCell identification registers, RTCPCellID0-3* on page 3-11
- *Interrupts* on page 3-13.

3.1 About the programmer's model

The base address of the PrimeCell RTC is not fixed, and might be different for any particular system implementation. The offset of any particular register from the base address is determined.

The following locations are reserved, and must not be used during normal operation:

- locations at offsets +0x20 to +0x7C and +0x94 to +0xFCC are reserved for possible future extensions
- locations at offsets +0x80 to +0x90 are reserved for test purposes
- locations at offsets +0xFD0 to +0xFDC are reserved for future identification registers.

3.2 Summary of PrimeCell RTC registers

The PrimeCell RTC registers are shown in Table 3-1.

Table 3-1 PrimeCell RTC register summary

Address	Type	Width	Reset value	Name	Description
RTC Base + 0x000	Read	32	0x00000000	RTCDR	Data register
RTC Base + 0x004	Read/write	32	0x00000000	RTCMR	Match register
RTC Base + 0x008	Read/write	32	0x00000000	RTCLR	Load register
RTC Base + 0x00C	Read/write	1	0x00000000	RTCCR	Control register
RTC Base + 0x010	Read/write	1	0x00000000	RTCIMSC	Interrupt mask set and clear register
RTC Base + 0x014	Read	1	0x00000000	RTCRIS	Raw interrupt status register
RTC Base + 0x018	Read	1	0x00000000	RTCMIS	Masked interrupt status register
RTC Base + 0x01C	Write	1	0x00000000	RTCICR	Interrupt clear register
RTC Base + 0x020–0x07C	-	-	-	-	Reserved
RTC Base + 0x080–090	-	-	-	-	Reserved for test purposes
RTC Base + 0x094–FCC	-	-	-	-	Reserved.
RTC Base + 0xFD0–FDC	-	-	-	-	Reserved for future ID expansion
RTC Base + 0xFE0	Read	8	0x31	RTCPeriphID0	Peripheral ID register bits [7:0]
RTC Base + 0xFE4	Read	8	0x10	RTCPeriphID1	Peripheral ID register bits [15:8]
RTC Base + 0xFE8	Read	8	0x*04 ^a	RTCPeriphID2	Peripheral ID register bits [23:16]
RTC Base + 0xFEC	Read	8	0x00	RTCPeriphID3	Peripheral ID register bits [31:24]
RTC Base + 0xFF0	Read	8	0D	RTCPCellID0	PrimeCell ID register bits [7:0]
RTC Base + 0xFF4	Read	8	F0	RTCPCellID1	PrimeCell ID register bits [15:8]
RTC Base + 0xFF8	Read	8	05	RTCPCellID2	PrimeCell ID register bits [23:16]
RTC Base + 0xFFC	Read	8	B1	RTCPCellID3	PrimeCell ID register bits [31:24]

a. * indicates the revision number (see *RTCPeriphID2* register on page 3-9).

3.3 General registers

The following registers are described in this section:

- *Data register, RTCDR*
- *Match register, RTCMR*
- *Load register, RTCLR* on page 3-5
- *Control register, RTCCR* on page 3-5
- *Interrupt mask set or clear register, RTCIMSC* on page 3-6
- *Raw interrupt status, RTCRIS* on page 3-6
- *Masked interrupt status, RTCMIS* on page 3-7
- *Interrupt clear register, RTCICR* on page 3-7.

The Peripheral identification registers are described in *Peripheral identification registers, RTCPeriphID0-3* on page 3-8.

The PrimeCell identification registers are described in *PrimeCell identification registers, RTCPCellID0-3* on page 3-11.

3.3.1 Data register, RTCDR

RTCDR is a 32-bit read data register. Reads from this register return the current value of the RTC. Table 3-2 shows the bit assignments for the RTCDR register.

Table 3-2 RTCDR register

Bits	Name	Type	Function
31:0	RTC data register	Read	Returns the current RTC value.

3.3.2 Match register, RTCMR

RTCMR is a 32-bit read/write match register. Writes to this register load the match register, and reads return the last written value. An equivalent match value is derived from this register. The derived value is compared with the counter value in the **CLK1HZ** domain to generate an interrupt. Table 3-3 shows the bit assignments for the RTCMR register.

Table 3-3 RTCMR register

Bits	Name	Type	Function
31:0	RTC match register	Read/write	Match register

3.3.3 Load register, RTCLR

RTCLR is a 32-bit read/write load register. Writes to this register load an update value into the RTC Update logic block where the updated value of the RTC is calculated. Reads return the last written value. Table 3-4 shows the bit assignments for the RTCLR register.

Table 3-4 RTCLR register

Bits	Name	Type	Function
31:0	RTC load register	Read/write	Load register

3.3.4 Control register, RTCCR

RTCCR is a 1-bit control register. When HIGH, the *counter enable* signal is asserted to enable the counter. Table 3-5 shows the bit assignments for the RTCCR register.

Table 3-5 RTCCR register

Bits	Name	Type	Function
31:1	-	Read/write	Reserved. Read unpredictable. Should be written as 0.
0	RTC start	Read/write	If set to 1, the RTC is enabled. Once it is enabled, any writes to this bit have no effect on the RTC until a system reset. A read returns the status of the RTC.

3.3.5 Interrupt mask set or clear register, RTCIMSC

RTCIMSC is a 1-bit read/write control register, and controls the masking of the interrupt generated by the RTC. Writing to bit position 0 sets or clears the mask. Reading this register returns the current value of the mask on the RTC interrupt (RTCINTR). Table 3-6 shows the bit assignments for the RTCIMSC register.

Table 3-6 RTCIMSC register

Bits	Name	Type	Function
31:1	Reserved	-	Reserved. Read as zero. Do Not Modify (DNM).
0	RTCIMSC	Read/write	Writing 1 sets the mask. Writing 0 clears the mask.

3.3.6 Raw interrupt status, RTCRIS

RTCRIS is read-only register. Reading this register gives the current raw status value of the corresponding interrupt prior to masking. A write has no effect. Table 3-7 shows the bit assignments for the RTCRIS register.

Table 3-7 RTCRIS register

Bits	Name	Type	Function
31:1	Reserved	-	Reserved. Read as zero. Do Not Modify (DNM).
0	RTCRIS	Read	Gives the raw interrupt state (prior to masking) of the RTCINTR interrupt.

3.3.7 Masked interrupt status, RTCMIS

RTCMIS is a 1-bit masked interrupt status register. It is a read-only register. Reading this register gives the current masked status value of the corresponding interrupt. A write has no effect. Table 3-8 shows the bit assignments for the RTCMIS register.

Table 3-8 RTCMIS register

Bits	Name	Type	Function
31:1	Reserved	-	Reserved. Read as zero. Do Not Modify (DNM).
0	RTCMIS	Read	Gives the masked interrupt status (after masking) of the RTCINTR interrupt.

3.3.8 Interrupt clear register, RTCICR

RTCICR is the interrupt clear register and is write-only. Writing 1 to bit position 0 clears the corresponding interrupt. Writing 0 has no effect. Table 3-9 shows the bit assignments for the RTCICR register.

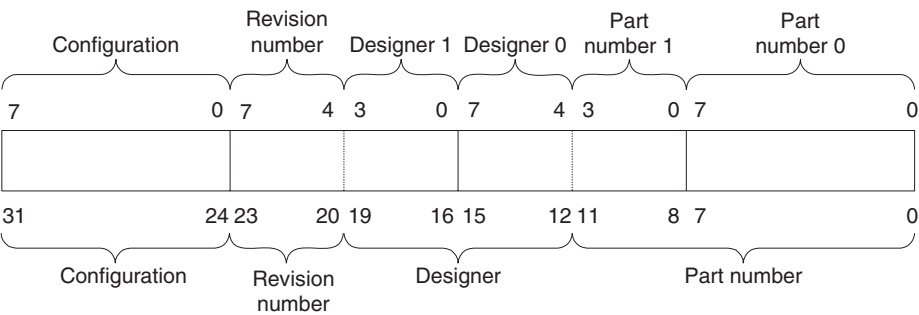
Table 3-9 RTCICR register

Bits	Name	Type	Function
31:1	Reserved	-	Reserved. Read as zero. Do Not Modify (DNM).
0	RTCICR	Write	Clears the RTCINTR interrupt. Writing 1 clears the interrupt. Writing 0 has no effect.

3.4 Peripheral identification registers, RTCPeriphID0-3

The RTCPeriphID0-3 registers are four, 8-bit registers that span address locations 0xFE0 to 0xFEC. The registers can conceptually be treated as a 32-bit register. Figure 3-1 shows the bit assignments for the RTCPeriphID0-3 registers.

Actual register bit assignment



Conceptual register bit assignment

Figure 3-1 Peripheral identification register bit assignment

The read-only registers provide the following options of the peripheral, as shown in Table 3-10.

Table 3-10 Read-only registers

Bits	Assignment	Description
11:0	Part number	Identifies the peripheral, using the three-digit product code 0x031.
19:12	Designer ID	Gives the designer identification. ARM Limited is 0x41 (ASCII A).
23:20	Revision	Is the revision number of the peripheral. The revision number starts from 0.
31:24	Configuration	Is the configuration option of the peripheral. The configuration value is 0.

3.4.1 RTCPeriphID0 register

The RTCPeriphID0 register is hard-coded. The fields in the register determine the reset value. Table 3-11 shows the bit assignments for the RTCPeriphID0 register.

Table 3-11 RTCPeriphID0 register

Bits	Name	Description
15:8	-	Reserved. Read undefined. Must read as zeros.
7:0	PartNumber0	These bits read back as 0x31.

3.4.2 RTCPeriphID1 register

The RTCPeriphID1 register is hard-coded. The fields in the register determine the reset value. Table 3-12 shows the bit assignments for the RTCPeriphID1 register.

Table 3-12 RTCPeriphID1 register

Bits	Name	Description
15:8	-	Reserved. Read undefined. Must read as zeros.
7:4	Designer0	These bits read back as 0x1.
3:0	PartNumber1	These bits read back as 0x0.

3.4.3 RTCPeriphID2 register

The RTCPeriphID2 register is hard-coded. The fields in the register determine the reset value. Table 3-13 shows the bit assignments for the RTCPeriphID2 register.

Table 3-13 RTCPeriphID2 register

Bits	Name	Description
15:8	-	Reserved. Read undefined. Must read as zeros.
7:4	Revision	These bits read back as the revision number. This can be between 0 and 15.
3:0	Designer1	These bits read back as 0x4.

3.4.4 RTCPeriphID3 register

The RTCPeriphID3 register is hard-coded. The fields in the register determine the reset value. Table 3-14 shows the bit assignments for the RTCPeriphID3 register.

Table 3-14 RTCPeriphID3 register

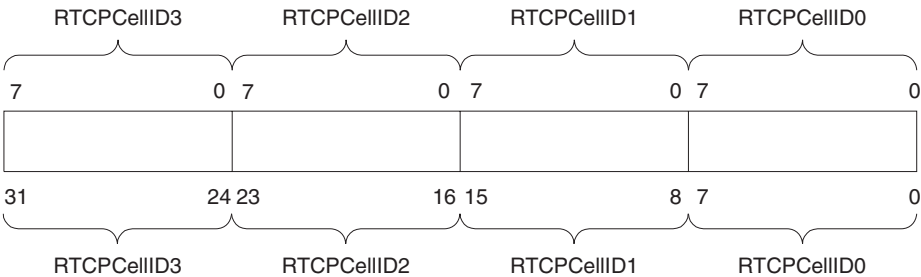
Bits	Name	Description
15:8	-	Reserved. Read undefined. Must read as zeros.
7:0	Configuration	These bits read back as 0x00

3.5 PrimeCell identification registers, RTCPCellID0-3

The RTCPCellID0-3 registers are four, 8-bit wide registers that span address locations 0xFF0 to 0xFFC. The registers can conceptually be treated as a 32-bit register, used as a standard cross-peripheral identification system. The RTCPCellID register is set to 0xB105F00D.

Figure 3-2 shows the bit assignments for the RTCPCellID0-3 registers.

Actual register bit assignment



Conceptual register bit assignment

Figure 3-2 PrimeCell identification register bit assignment

3.5.1 RTCPCellID0 register

The RTCPCellID0 register is hard-coded. The fields in the register determine the reset value. Table 3-15 shows the bit assignments for the RTCPCellID0 register.

Table 3-15 RTCPCellID0 register

Bits	Name	Description
15:8	-	Reserved. Read undefined. Must read as zeros.
7:0	RTCPCellID0	These bits read back as 0x0D.

3.5.2 RTCPCellID1 register

The RTCPCellID1 register is hard-coded. The fields in the register determine the reset value. Table 3-16 shows the bit assignments for the RTCPCellID1 register.

Table 3-16 RTCPCellID1 register

Bits	Name	Description
15:8	-	Reserved. Read undefined. Must read as zeros.
7:0	RTCPCellID1	These bits read back as 0xF0.

3.5.3 RTCPCellID2 register

The RTCPCellID2 register is hard-coded. The fields in the register determine the reset value. Table 3-17 shows the bit assignments for the RTCPCellID2 register.

Table 3-17 RTCPCellID2 register

Bits	Name	Description
15:8	-	Reserved. Read undefined. Must read as zeros.
7:0	RTCPCellID2	These bits read back as 0x05.

3.5.4 RTCPCellID3 register

The RTCPCellID3 register is hard-coded. The fields in the register determine the reset value. Table 3-18 shows the bit assignments for the RTCPCellID3 register.

Table 3-18 RTCPCellID3 register

Bits	Name	Description
15:8	-	Reserved. Read undefined. Must read as zeros.
7:0	RTCPCellID3	These bits read back as 0xB1.

3.6 Interrupts

A single, maskable, active HIGH interrupt **RTCINTR** is generated by the PrimeCell RTC when a match occurs between the counter and the equivalent match value:

- This interrupt is enabled or disabled by changing the mask bit in RTCIMSC. To enable the interrupt, set bit 0 HIGH.
- The status of the interrupt mask can be read from bit[0] of RTCMIS.
- Writing 1 to bit[0] of RTCICR clears the RTCINTR flag.
- The RTC interrupt, RTCINTR, is output through an output pin.

Chapter 4

Programmer's Model for Test

This chapter describes the additional logic for functional verification and production testing. It contains the following sections:

- *PrimeCell RTC test harness overview* on page 4-2
- *Scan testing* on page 4-3
- *Test registers* on page 4-4
- *Integration testing of block inputs* on page 4-7
- *Integration testing of block outputs* on page 4-8
- *Integration test summary* on page 4-9.

4.1 PrimeCell RTC test harness overview

The test harness provides integration vectors to enable:

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

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

AMBA signals

These are tested by checking the connections of all address and data bits.

Primary input and output signals

These are tested using a simple trickbox that demonstrates the correct connection of the input and output signals to external pads.

Intra-chip signals

The tests for these signals are system-specific and require the necessary tests to be written. Additional logic can be implemented to allow reads and writes to each intra-chip signal.

These test features are controlled by a test register. This allows testing of the PrimeCell RTC in isolation from the rest of the system using only transfers from the AMBA APB.

Off-chip integration test vectors are supplied via a 32-bit parallel *External Bus Interface* (EBI) and converted to internal AMBA bus transfers. The application of test vectors is controlled through the *Test Interface Controller* (TIC) AMBA bus master module.

Figure 4-1 shows a block diagram of the PrimeCell RTC test harness.

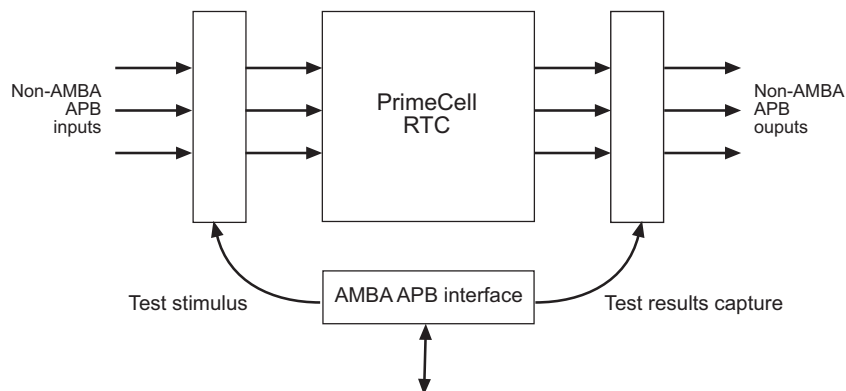


Figure 4-1 PrimeCell RTC test harness

4.2 Scan testing

The PrimeCell RTC has been designed to simplify the insertion of scan test cells and the use of *Automatic Test Pattern Generation* (ATPG) for an alternative method of manufacturing test.

The scan test pins are arranged to take account of the two clock domains, **PCLK** and **CLK1HZ**:

- **SCANENABLE**
- **SCANINPCLK** and **SCANOUTPCLK**
- **SCANINCLK1HZ** and **SCANOUTCLK1HZ**.

4.3 Test registers

The PrimeCell RTC test registers are memory-mapped as shown in Table 4-1.

Table 4-1 Test registers memory map

Address	Type	Width	Reset value	Name	Description
RTC Base + 0x80	Read/write	3	0x00000000	RTCITCR	Integration test control register
RTC Base + 0x84	Read/write	0	0x00000000	RTCITIP	Integration test input read or set register
RTC Base + 0x88	Read/write	1	0x00000000	RTCITOP	Integration test output read or set register
RTC Base + 0x8C	Read/write	32	0x00000000	RTCTOFFSET	Test offset register
RTC Base + 0x90	Read/write	32	0x00000000	RTCTCOUNT	Test count register

————— **Note** —————

Test registers must not be accessed during normal operation.

4.3.1 Integration test control register, RTCITCR

RTCITCR is the test control register. This general test register controls the operation of the PrimeCell RTC under test conditions. Table 4-2 shows the bit assignments for the RTCITCR register.

Table 4-2 RTCITCR register

Bits	Name	Description
31:3	-	Reserved. Unpredictable when read. Should be written as 0.
2	TESTOFFSET	Test offset enable. When this bit is set to 1, data can be written to and read from the offset register for test purposes. When this bit is set to 0, data cannot be written to or read from the offset register (normal operation). The reset value is 0.
1	TESTCOUNT	Test count enable. When this bit is set to 1, data can be written to and read from the counter register for test purposes. When this bit is set to 0, data cannot be written to or read from the counter register (normal operation). The reset value is 0.
0	ITEN	Integration test enable. When this bit is 1, the RTC is placed in integration test mode, otherwise it is in normal mode.

4.3.2 Integration test input read or set register, RTCITIP

RTCITIP is the integration test input read or set register. It is reserved for future use.

4.3.3 Integration test output read or set register, RTCITOP

RTCITOP is the integration test output read or set register. The primary outputs are read only and the intra-chip outputs are read/write. In integration test mode it allows outputs to be both written to and read from. Table 4-3 shows the bit assignments for the RTCITOP register.

Table 4-3 RTCITOP register

Bits	Name	Description
31:1	-	Reserved. Unpredictable when read.
0	RTCINTR	Intra-chip output. Writes specify the value to be driven on the RTCINTR line in the integration test mode. Reads return the value of the RTCINTR at the output of the test multiplexor.

4.3.4 Test offset register, RTCTOFFSET

RTCTOFFSET is the test offset register. It allows data to be written into the offset register for test purposes. Table 4-4 shows the bit assignments for the RTCTOFFSET register.

Table 4-4 RTCTOFFSET register

Bits	Name	Description
31:0	OFFSET	Read/write register allowing reads and writes to the offset register for test purposes.

4.3.5 Test count register, RTCTCOUNT

RTCTCOUNT is the test count register. It allows data to be written into the counter register for test purposes. Table 4-5 shows the bit assignments for the RTCTCOUNT register.

Table 4-5 RTCTCOUNT register

Bits	Name	Description
31:0	COUNT	Read/write register allowing reads and writes to the counter register for test purposes.

4.4 Integration testing of block inputs

There are no intra-chip inputs or block inputs to the PrimeCell RTC.

4.5 Integration testing of block outputs

This section describes the integration testing for the block intra-chip output. There are no primary outputs from the PrimeCell RTC.

4.5.1 Intra-chip outputs

Use this test for the **RTCINTR** output.

When you run integration tests with the PrimeCell RTC in a standalone test setup:

- Write a 1 to the ITEN bit in the integration test control register. This selects the test path from the RTCITOP register to the intra-chip output signal.
- Write a 1 and then a 0 to the RTCITOP register bit, and read the same register bit to verify that the value written is read out.

When you run integration tests with the PrimeCell RTC as part of an integrated system:

- Write a 1 to the ITEN bit in the integration control register. This selects the test path from the RTCITOP register bit to the intra-chip output signal.
- Write a 1 and then a 0 to the RTCITOP register bit to toggle the signal connections between the interrupt controller and the PrimeCell RTC. Read from the internal test registers of the interrupt controller to verify that the value written into the RTCITOP register bit is read out through the PrimeCell RTC.

Figure 4-2 shows the implementation details of the output integration test harness for intra-chip outputs.

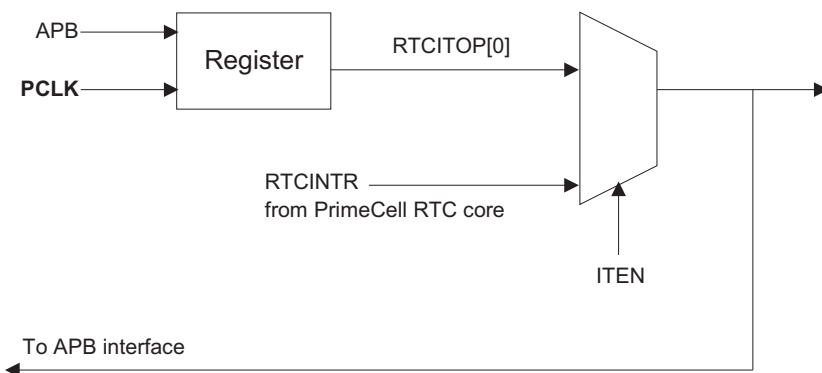


Figure 4-2 Output integration test harness, intra-chip outputs

4.6 Integration test summary

Table 4-6 summarizes the integration test strategy for all PrimeCell RTC pins.

Table 4-6 PrimeCell RTC integration test strategy

Name	Type	Source/destination	Test strategy
PRESETn	Input	Reset controller	Not tested using integration test vectors
PADDR	Input	APB	Register read/write
PCLK	Input	APB	Register read/write
PENABLE	Input	APB	Register read/write
PRDATA	Output	APB	Register read/write
PSEL	Input	APB	Register read/write
PWDATA	Input	APB	Register read/write
PWRITE	Input	APB	Register read/write
CLK1HZ	Input	Clock generator	Not tested using integration test vectors
nRTCRST	Input	Reset controller	Not tested using integration test vectors
nPOR	Input	Reset controller	Not tested using integration test vectors
RTCINTR	Output	Interrupt controller	Tested using RTCITOP register
SCANENABLE	Input	Test controller	Not tested using integration test vectors
SCANINPCLK	Input	Test controller	Not tested using integration test vectors
SCANINCLK1HZ	Input	Test controller	Not tested using integration test vectors
SCANOUTPCLK	Output	Test controller	Not tested using integration test vectors
SCANOUTCLK1HZ	Output	Test controller	Not tested using integration test vectors

Appendix A

ARM PrimeCell Real Time Clock (PL031) Signal Descriptions

This appendix describes the signals that interface with the ARM PrimeCell Real Time Clock (PL031). It contains the following sections:

- *AMBA APB signals* on page A-2
- *On-chip signals* on page A-3.

A.1 AMBA APB signals

The PrimeCell RTC module is connected to the AMBA APB as a bus slave. Table A-1 describes the AMBA APB signals that are used and produced.

Table A-1 AMBA APB signal descriptions

Name	Type	Source or destination	Description
PCLK	Input	Clock generator	AMBA APB clock, used to time all bus transfers.
PRESETn	Input	Reset controller	Bus reset signal (active LOW).
PSEL	Input	APB bridge	When HIGH, this signal indicates the RTC module has been selected by the AMBA APB bridge. This selection is a decode of the system address bus (ASB).
PENABLE	Input	APB bridge	AMBA APB enable signal. PENABLE is asserted HIGH for one cycle of PCLK to enable a bus transfer cycle.
PWRITE	Input	APB bridge	When HIGH, this signal indicates a write to a peripheral and when LOW, a read from a peripheral. This signal has the same timing as the peripheral address bus.
PADDR[11:2]	Input	APB bridge	Subset of AMBA APB bus.
PWDATA[31:0]	Input	APB bridge	Unidirectional AMBA APB write data bus.
PRDATA[31:0]	Output	APB bridge	Unidirectional AMBA APB read data bus.

A.2 On-chip signals

Table A-2 shows the non-AMBA signals from the block.

Table A-2 On-chip signals

Name	Type	Source or destination	Description
CLK1HZ	Input	Clock generator	1Hz clock input. This is the signal that clocks the counter during normal operation.
nRTCRST	Input	Reset controller	RTC reset signal (active LOW). It can be asserted asynchronously, but must be deasserted synchronously to CLK1HZ .
nPOR	Input	Reset controller	RTC power-on reset signal for RTCMR and offset registers. These registers must retain their value over the bus reset signal PRESETn .
RTCINTR	Output	Interrupt controller	Interrupt signal to the interrupt module. When HIGH, this signal indicates that a valid match has occurred between the counter value and the match register.
SCANENABLE	Input	Test controller	Place holder for scan enable input signal.
SCANINCLK	Input	Test controller	Place holder for scan data input signal (PCLK domain).
SCANOUTPCLK	Input	Test controller	Place holder for scan data output signal (PCLK domain).
SCANINCLK1HZ	Input	Test controller	Place holder for scan data input signal (CLK1HZ domain).
SCANOUTCLK1HZ	Input	Test controller	Place holder for scan data output signal (CLK1HZ domain).

Index

A

AMBA
 APB interface 2-6
 APB read access 2-3
 APB write access 2-2
AMBA APB signals A-2
AMBA signals 4-2

B

Base address 3-2
Block diagram 2-5
 test harness 4-2
Buses
 AMBA APB 2-6

C

Clock frequencies 2-9
Clock signals 2-9

Connection diagram 1-2
Control block 2-6
Control register 3-5
Counter block 2-7
Counter enable signal 3-5
Counter value 2-6

D

Data register 3-4

E

External Bus Interface (EBI) 4-2

I

Input clock 2-2
Input signals 4-2
Integration Test Input register 4-5

Integration Test Output register 4-5
Integration vectors 4-2
Interface reset 2-9
Interrupt clear register 3-7
Interrupt mask set or clear register 3-6
Interrupts
 RTCINTR 2-2, 2-6, 2-10, 3-13
Intra-chip signals 4-2

L

Load register 3-5

M

Masked interrupt status register 3-7
Match register 2-2, 3-4

N

Non-AMBA signals A-3
 nRTCRST input clock 2-2
 nRTCRST signal A-3

O

Offset register 2-7
 Output signals 4-2

P

PADDRH[7:6] signal A-2
 PCLK signal A-2
 PENABLE signal A-2
 Peripheral identification registers 3-8
 PRDATA[31:0] signal A-2
 PRESETn signal 2-9, A-2
 Primary I/O signals 4-2
 PrimeCell identification registers 3-11
 PrimeCell RTC
 block diagram 2-5
 components of 2-2
 features 1-3
 functional description 2-5
 operation 2-9
 overview 1-2
 testing 4-2
 PSEL signal A-2
 PWDATA[31:0] signal A-2
 PWRITE signal A-2

R

Raw and masked interrupts
 synchronizing 2-7
 Raw interrupt status register 3-6
 Register block 2-6
 Registers
 match 2-2
 offset 2-7
 peripheral identification 3-8
 PrimeCell identification 3-11
 RTCCR 3-5
 RTCDR 3-4

RTCICR 3-7
 RTCIMSC 3-6
 RTCLR 3-5
 RTCMIS 3-7
 RTCMR 3-4
 RTCPCellID0 3-11
 RTCPCellID1 3-12
 RTCPCellID2 3-12
 RTCPCellID3 3-12
 RTCPeriphID0 3-9
 RTCPeriphID1 3-9
 RTCPeriphID2 3-9
 RTCPeriphID3 3-10
 RTCRIS 3-6
 summary of 3-3
 test 4-2
 Reserved locations 3-2
 RTCCR register 3-5
 RTCDR register 3-4
 RTCICR register 3-7
 RTCIMSC register 3-6
 RTCINTR interrupt 3-13
 RTCINTR signal A-3
 RTCITCR test register 4-4
 RTCITIP test register 4-5
 RTCITOP test register 4-5
 RTCLR register 3-5
 RTCMIS register 3-7
 RTCMR register 3-4
 RTCPCellID0 register 3-11
 RTCPCellID1 register 3-12
 RTCPCellID2 register 3-12
 RTCPCellID3 register 3-12
 RTCPeriphID0 register 3-9
 RTCPeriphID1 register 3-9
 RTCPeriphID2 register 3-9
 RTCPeriphID3 register 3-10
 RTCRIS register 3-6
 RTCTCOUNT test register 4-6
 RTCTOFFSET test register 4-5

S

Scan test pins 4-3
 SCANENABLE 4-3
 SCANINCLK1HZ 4-3
 SCANINPCLK 4-3
 SCANOUTCLK1HZ 4-3

SCANOUTPCLK 4-3

Scan testing 4-3
 SCANENABLE pin 4-3
 SCANENABLE signal A-3
 SCANINCLK1HZ pin 4-3
 SCANINCLK1HZ signal A-3
 SCANINPCLK pin 4-3
 SCANINPCLK signal A-3
 SCANOUTCLK1HZ pin 4-3
 SCANOUTCLK1HZ signal A-3
 SCANOUTPCLK pin 4-3
 SCANOUTPCLK signal A-3

Signals

AMBA 4-2
 counter enable 3-5
 input 4-2
 intra-chip 4-2
 non-AMBA A-3
 nRTCRST A-3
 Output 4-2
 PADDRH[7:6] A-2
 PCLK A-2
 PENABLE A-2
 PRDATA[31:0] A-2
 PRESETn 2-9, A-2
 Primary I/O 4-2
 PSEL A-2
 PWDATA[31:0] A-2
 PWRITE A-2
 RTCINTR A-3
 SCANENABLE A-3
 SCANINCLK1HZ A-3
 SCANINPCLK A-3
 SCANOUTCLK1HZ A-3
 SCANOUTPCLK A-3
 test clock enable 2-8
 Synchronization block 2-7
 counter value 2-7
 raw and masked interrupts 2-7
 signals 2-7
 Synchronization logic 2-2, 2-6

T

Test clock enable signal 2-8
 Test Control register 4-4
 Test Count register 4-6
 Test harness 4-2

- block diagram 4-2
- Test Interface Controller (TIC) 2-8, 4-2
- Test logic 2-8
- Test Offset register 4-5
- Test registers 2-8, 4-2, 4-4
 - RTCITCR 4-4
 - RTCITIP 4-5
 - RTCITOP 4-5
 - RTCTCOUNT 4-6
 - RTCTOFFSET 4-5
- Testing the PrimeCell RTC 4-2

U

- Update block 2-6

