ARM PrimeCell PS2 Keyboard/Mouse Interface (PL050)

Technical Reference Manual

ARM®

ARM DDI 0143C

ARM PrimeCell PS2 Keyboard/Mouse Interface (PL050)
Technical Reference Manual

Release Information

The following changes have been made to this book.

Change History

Date	Issue	Confidentiality	Change
January 1999	A	Non-Confidential	First release
April 1999	B	Non-Confidential	Second release
May 1999	C	Non-Confidential	Third release

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.

Where the term ARM is used it means “ARM or any of its subsidiaries as appropriate”.

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 PS2 Keyboard/Mouse Interface (PL050) Technical Reference Manual

Chapter 4 Programmer's Model for Test

Appendix A Signal Descriptions

List of Tables
ARM PrimeCell PS2 Keyboard/Mouse Interface (PL050) Technical Reference Manual

List of Figures
ARM PrimeCell PS2 Keyboard/Mouse Interface (PL050) Technical Reference Manual

Preface

This preface introduces the *ARM PrimeCell PS2 Keyboard/Mouse Interface (PL050) Technical Reference Manual*. It contains the following sections:

- *About this manual* on page x
- *Feedback* on page xiv.

About this manual

This is the *Technical Reference Manual* (TRM) for the *ARM PrimeCell PS2 Keyboard/Mouse Interface (PL050)* (KMI).

Intended audience

This manual is written for system designers, system integrators, and programmers who are designing or programming a *System-on-Chip* (SoC) that uses the PrimeCell KMI.

Using this manual

This manual is organized into the following chapters:

Chapter 1 ***Introduction***

Read this chapter for an introduction to the PrimeCell KMI and its features.

Chapter 2 ***Functional Overview***

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

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

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

Chapter 4 ***Programmer's Model for Test***

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

Appendix A ***Signal Descriptions***

Read this appendix for details of the PrimeCell KMI signals.

Conventions

Conventions that this manual can use are described in:

- *Typographical* on page xi
- *Timing diagrams* on page xi
- *Signals* on page xii
- *Numbering* on page xiii.

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 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.

`monospace` 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.

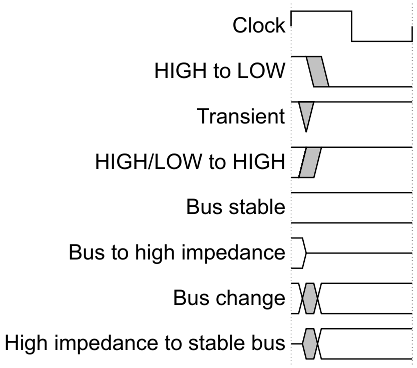
< and > Enclose replaceable terms for assembler syntax where they appear in code or code fragments. For example:

```
MRC p15, 0 <Rd>, <CRn>, <CRm>, <Opcode_2>
```

Timing diagrams

The figure named *Key to timing diagram conventions* on page xii explains the components used in timing diagrams. Variations, when they occur, have clear labels. You must not assume any timing information that is not explicit in the diagrams.

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.



Key to timing diagram conventions

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
- LOW for active-LOW signals.

Lower-case n At the start or end of a signal name denotes an active-LOW signal.

Prefix A Denotes global *Advanced eXtensible Interface* (AXI) signals.

Prefix AR Denotes AXI read address channel signals.

Prefix AW Denotes AXI write address channel signals.

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 P Denotes Advanced Peripheral Bus (APB) signals.

Prefix R Denotes AXI read data channel signals.

Prefix W Denotes AXI write data channel 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.

Additional reading

This section lists publications by ARM and by third parties.

See `http://infocenter.arm.com/help/index.jsp` for access to ARM documentation.

ARM publications

This manual contains information that is specific to the PrimeCell KMI. See the following documents for other relevant information:

- *AMBA Specification (Rev 2.0)* (ARM IHI 0011).
- *ARM PrimeCell PS2 Keyboard/Mouse Interface (PL050) Design Manual*(ARM DDES 0000).
- *ARM PrimeCell PS2 Keyboard/Mouse Interface (PL050) Integration Manual*(ARM INTM 0000).

Other publications

This section lists relevant documents published by third parties:

- *IBM PC-AT Technical Reference Manual.*

Feedback

ARM welcomes feedback on the PrimeCell KMI and its documentation.

Feedback on this product

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

- the product name
- a concise explanation.

Feedback on this manual

If you have any comments on this manual, send an e-mail to `errata@arm.com`. Give:

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

ARM also welcomes general suggestions for additions and improvements.

Chapter 1
Introduction

This chapter introduces the ARM PrimeCell *PS2* Keyboard/Mouse Interface (PL050) and contains the following sections:

- *About the ARM PrimeCell PS2 Keyboard/Mouse Interface (PL050)* on page 1-2
- *AMBA compatibility* on page 1-4.

1.1 About the ARM PrimeCell PS2 Keyboard/Mouse Interface (PL050)

The PrimeCell *PS2 Keyboard/Mouse Interface* (KMI) is an *Advanced Microcontroller Bus Architecture* (AMBA) compliant System-on-a-Chip peripheral that is developed, tested and licensed by ARM.

The PrimeCell KMI is an AMBA slave module, and connects to the *Advanced Peripheral Bus* (APB). The PrimeCell KMI can be used to implement a keyboard or mouse interface that is IBM PS2 or AT compatible.

Figure 1-1 illustrates the connections to the PrimeCell KMI.

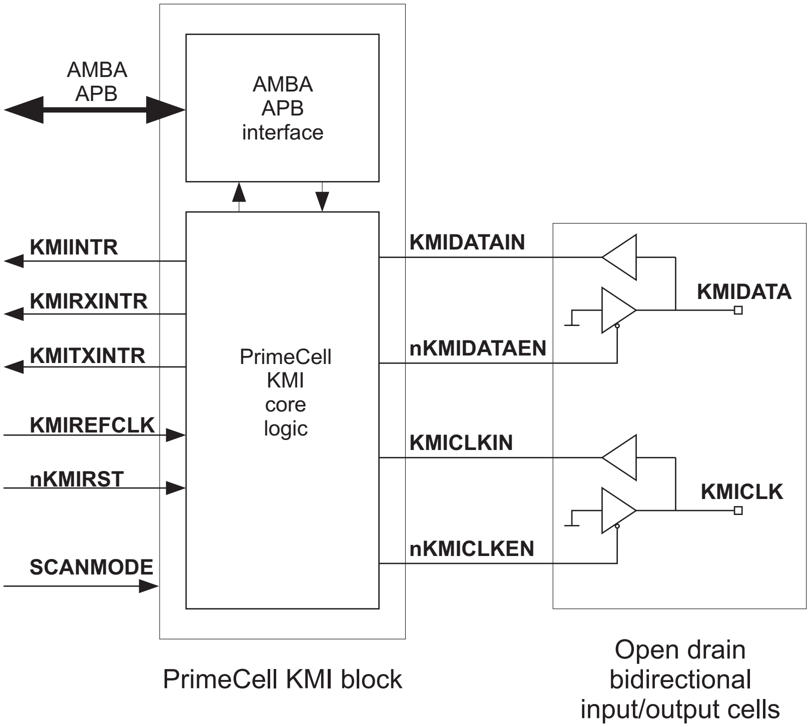


Figure 1-1 Connections to the PrimeCell KMI

1.1.1 Features of the PrimeCell KMI

The PrimeCell KMI has the following features:

- compliance to the *AMBA Specification (Rev 2.0)* onwards for easy integration into *System-on-a-Chip* (SoC) implementation
- IBM PS2 or AT-compatible keyboard or mouse interface
- half-duplex bidirectional synchronous serial interface using open-drain outputs for clock and data
- programmable 4-bit reference clock divider
- operation in polled or interrupt-driven mode
- separately maskable transmit and receive interrupts
- single combined interrupt output
- odd parity generation and checking
- register bits for override of keyboard clock and data lines.

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

1.2 AMBA compatibility

The PrimeCell KMI complies with the *AMBA Specification (Rev 2.0)* onwards. The fundamental differences from the *AMBA Specification Revision D* are:

- the timing of the strobe signal **PSTB** compared with the enable signal **PENABLE**
- the time at which read data is sampled
- a separate unidirectional read data bus **PRDATA**, and unidirectional write bus **PWDATA** (instead of the bidirectional data bus **PD**)
- the address bus is named **PADDR** (instead of **PA**).

This document assumes little-endian memory organization, where bytes of increasing significance are stored in increasing addresses in memory, and hence low-order bytes are transferred on the low-order bits of the data bus. Options for a big-endian system are described in the *ARM PrimeCell Keyboard/Mouse Interface (PL050) Integration Manual*.

Chapter 2
Functional Overview

This chapter describes the major functional blocks of the ARM PrimeCell *PS2* Keyboard/Mouse Interface (PL050) and contains the following sections:

2.1 ARM PrimeCell PS2 Keyboard/Mouse Interface (PL050) overview

The PrimeCell KMI provides a keyboard or mouse interface that is IBM PS2 or AT-compatible. The interface uses clock and data lines to implement a half-duplex bidirectional synchronous serial interface.

The PrimeCell KMI performs serial-to-parallel conversion on the data received from the keyboard/mouse peripheral, and parallel-to-serial conversion on the data transmitted to the peripheral. The CPU reads and writes data and control/status information via the AMBA APB interface.

In a typical configuration, **KMIDATAIN** and **KMICLKIN** inputs are connected to bidirectional input/output pads which have pull-up resistors for the pins KMIDATA and KMICLK. The outputs are high-impedance by default until the active LOW tristate enables **nKMIDATAEN** or **nKMICLKEN** are asserted LOW to pull the output pins LOW.

A reference clock **KMIREFCLK** is required to generate an internal 8MHz signal and a 4-bit divide allows division of the clock by 1 to 16.

After reset, the interface can be enabled to wait for one of two events:

- If data is written to the transmit register, a transmit sequence is initiated and the data is transmitted serially from the PrimeCell KMI.
- If the clock signal is pulled LOW by the keyboard/mouse peripheral, a receive sequence begins and data is clocked into the PrimeCell KMI.

The PrimeCell KMI can generate two individual maskable interrupts. These indicate one of the following situations:

- the transmit buffer is empty and another byte can be transmitted
- a byte has been received from the keyboard/mouse peripheral.

2.2 PrimeCell KMI functional description

The functions of the PrimeCell KMI are described in the following functional sections:

- *AMBA APB interface and register block* on page 2-4
- *Transmit block* on page 2-4
- *Receive block* on page 2-4
- *Controller block* on page 2-5
- *Timer/clock divider blocks* on page 2-5
- *Synchronization logic* on page 2-5
- *Test registers and logic* on page 2-5.

Figure 2-1 illustrates the PrimeCell KMI module functionality.

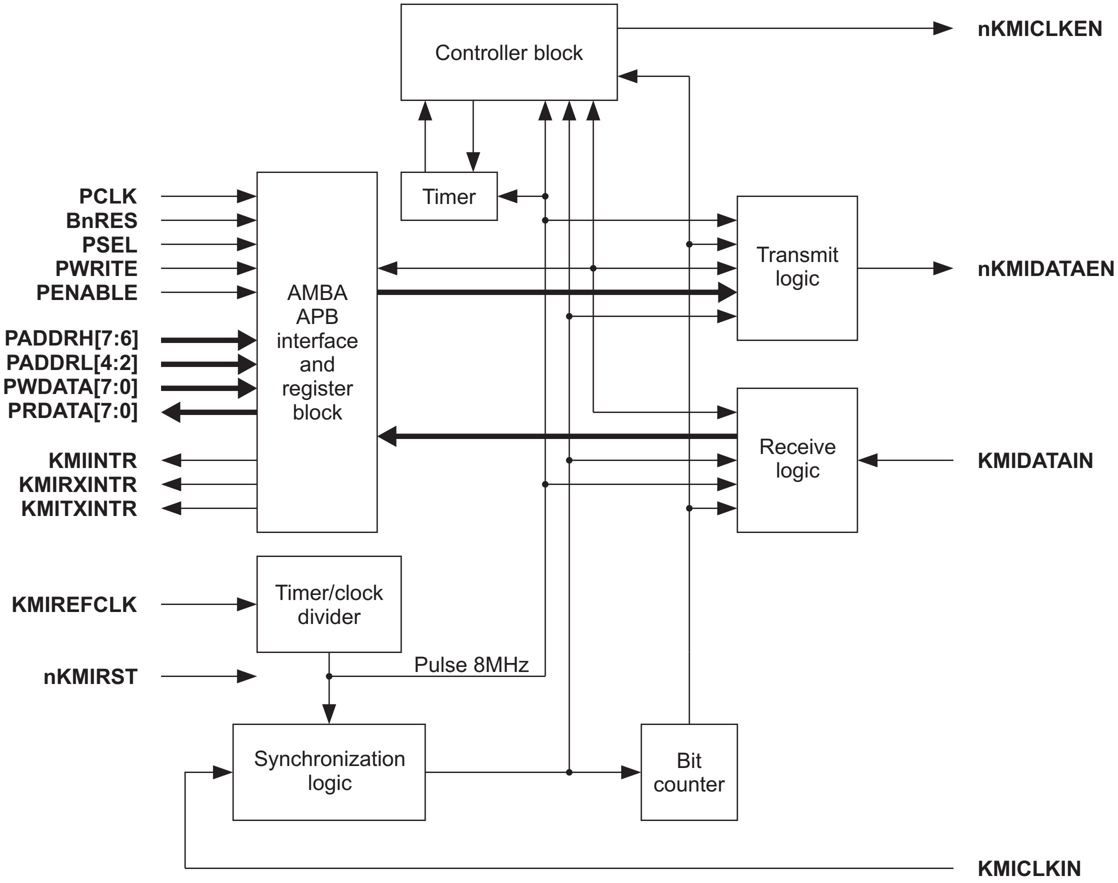


Figure 2-1 PrimeCell KMI block diagram

2.2.1 AMBA APB interface and register block

This block provides the interface for reading and writing to the PrimeCell KMI registers from the AMBA APB. All control register bits are synchronized to **KMIREFCLK** (the main clock for KMI logic) before being used in the PrimeCell KMI block.

The AMBA APB is a local secondary bus which provides a low-power extension to the higher bandwidth AMBA *Advanced High-performance Bus* (AHB), or AMBA *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 which are accessed under programmed control.

This block also generates individual transmit and receive interrupts. The transmit interrupt **KMITXINTR** is asserted HIGH to indicate that a byte can be written to the data register KMIDATA for transmission. The receive interrupt **KMIRXINTR** is asserted HIGH to indicate that a byte has been received and can be read from the data register KMIDATA. A combined interrupt **KMIINTR** is also asserted HIGH if either the transmit or receive interrupt is asserted.

The single combined interrupt may be used with a system interrupt controller that provides masking for the outputs of each peripheral. In this way, a global interrupt service routine would be able to read the entire set of sources from one wide register in the system interrupt controller. This is an attractive option where the time to read from the peripheral registers is significant compared to the CPU clock speed in a real-time system.

The peripheral supports both the above methods, since the overhead is small.

2.2.2 Transmit block

The transmit block converts the parallel transmit data into a serial bit stream with a rate dependent on the incoming PrimeCell KMI clock signal. This block performs odd parity generation, data framing and parallel-to-serial conversion. This block operates on **KMIREFCLK** with incoming PrimeCell KMI clock signal **KMICLKIN** providing the bit rate information. The data is shifted out on the falling edge of the **KMICLKIN** input.

2.2.3 Receive block

The receive block performs serial-to-parallel conversion on the serial data stream received on the **KMIDATAIN** input pin. The falling edge on the synchronized and sampled **KMICLKIN** input signal is used to sample the **KMIDATAIN** input line. The **KMIDATAIN** input is synchronized to the **KMIREFCLK** clock domain.

2.2.4 Controller block

The controller controls the overall operation of the PrimeCell KMI block. It controls the overall transmit and receive operation. If simultaneous request for transmission and reception occur, the transmit request is given priority.

2.2.5 Timer/clock divider blocks

These blocks generate the various timing signals required by the PrimeCell KMI logic.

2.2.6 Synchronization logic

Synchronization is provided for signals that cross clock domains, as described below:

- **KMICLKIN** input signal is synchronized to **KMIREFCLK** domain
- signals crossing from **PCLK** to **KMIREFCLK** domains
- signals crossing from **KMIREFCLK** to **PCLK** domains.

2.2.7 Test registers and logic

There are registers for functional block verification and manufacturing/production test using TICTalk vectors.

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

The test logic allows generation of a special test clock enable signal to propagate the test vectors applied to the input signal of the block and capture values at the block outputs.

2.3 PrimeCell KMI operation

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

- *Interface reset*
- *Clock signals*
- *Keyboard clock and data signals* on page 2-8
- *Keyboard/mouse data output* on page 2-9
- *Keyboard data input* on page 2-9
- *Timing requirements* on page 2-11.

2.3.1 Interface reset

The PrimeCell KMI is reset by the global reset signal **BnRES** and a block-specific reset signal **nKMIRST**. An external reset controller must use **BnRES** to assert **nKMIRST** asynchronously and negate it synchronously to **KMICLK**. **BnRES** should be asserted LOW for a period long enough to reset the slowest block in the on-chip system, and then taken HIGH again. The PrimeCell KMI requires **BnRES** to be asserted LOW for at least one period of **PCLK**.

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

2.3.2 Clock signals

The frequency of the clock signal **KMIREFCLK** should be selected to generate an internal nominal 8MHz clock signal.

The internal 8MHz clock signal is used for internal timing of signals, such as the keyboard request-to-send signal which is timed from 512 cycles of the internal 8MHz clock.

The 8MHz clock signal must have a minimum period of 117.2ns (that is, a maximum frequency of 8.533MHz) in order to ensure that the request-to-send command is greater than 60.0μs.

The 8MHz clock signal must have a maximum period of 166.7ns (that is, a minimum frequency of 6.0MHz) in order to ensure the correct sampling of keyboard clock signals as low as 1μs.

A nominal 8.0MHz internal clock signal (125ns period) will generate a typical request-to-send pulse width of 64μs.

Refer to Table 2-3 on page 2-11 for keyboard interface timing values.

The internal 8MHz clock is generated by dividing down from **KMIREFCLK** input clock. The divide value is programmed by writing to the KMICLKDIV register. A suitable **KMIREFCLK** frequency and divide value must be chosen to meet the constraints given below:

- $F_{\mathbf{KMIREFCLK}}$ <= (1 + KMICLKDIV) * 8.533MHz
- $F_{\mathbf{KMIREFCLK}}$ > (1 + KMICLKDIV) * 6.0MHz

An example table of divide values and **KMIREFCLK** frequency limits is shown in Table 2-1.

Table 2-1 Divide values and frequency limits

KMICLKDIV value	KMIREFCLK divided by	KMIREFCLK frequency (MHz)		
		Min	Typ	Max
0	1	6.0	8.0	8.533
1	2	12.0	16.0	17.067
2	3	18.0	24.0	25.600
3	4	24.0	32.0	34.133
4	5	30.0	40.0	42.667
5	6	36.0	48.0	51.200
6	7	42.0	56.0	59.733
7	8	48.0	64.0	68.267
8	9	54.0	72.0	76.800
9	10	60.0	80.0	85.333
10	11	66.0	88.0	93.867
11	12	72.0	96.0	102.400
12	13	78.0	104.0	110.930
13	14	84.0	112.0	119.470
14	15	90.0	120.0	128.000
15	16	96.0	128.0	136.530

2.3.3 Keyboard clock and data signals

The keyboard/mouse peripheral and PrimeCell KMI communicate using the clock and data signals. The sources of these signals are open-drain outputs with pull-up resistors, allowing either the peripheral or the PrimeCell KMI to force these lines LOW. When no communication is occurring, both clock and data are HIGH.

Data values are transmitted and received as a serial 11-bit data frame, which is shown in Figure 2-2.

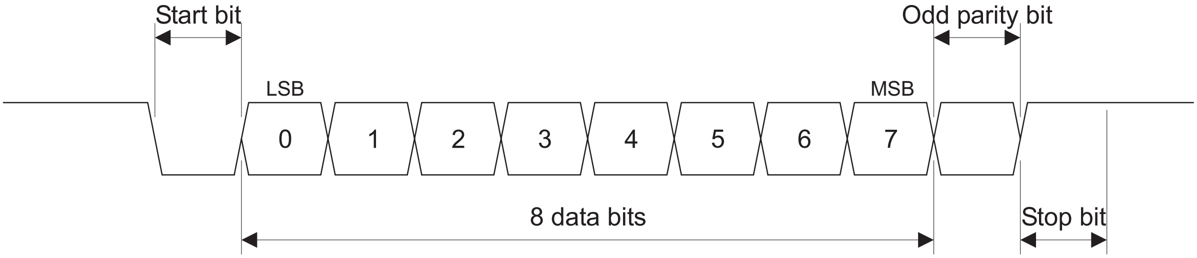


Figure 2-2 Serial data frame

The parity bit is used for error checking. An odd parity scheme is implemented such that:

- if an even number of data bits is set to 1 then the parity bit is asserted HIGH
- if an odd number of data bits is set to 1 is odd then the parity bit is LOW.

When the PrimeCell KMI transmits data to the keyboard/mouse peripheral, the PrimeCell KMI forces the data line to a LOW level, and allows the clock to go HIGH.

When the keyboard/mouse peripheral transmits or receives data from the PrimeCell KMI, it generates the clock signal to transfer the data. The PrimeCell KMI can prevent the keyboard/mouse peripheral from sending data by forcing the clock to a LOW level. At this time the data line can be either HIGH or LOW.

Line status indicated by keyboard clock and data signals is shown in Table 2-2.

Table 2-2 Line status

Clock	Data	Status
LOW	Don’t care	Inhibit status, data is stored in the keyboard.
HIGH	LOW	Request-to-send data from system to keyboard. Data is stored in the keyboard and the keyboard receives system data.
HIGH	HIGH	Ready status. Data can be sent from the keyboard to the system.

2.3.4 Keyboard/mouse data output

When the keyboard/mouse peripheral is ready to send data, it first checks for an inhibit or system-request-to-send status on the clock and data lines. If the clock line is LOW (inhibit status), data values are stored in the peripheral buffer. If the clock line is HIGH and the data line is LOW (request-to-send), data values are stored in the peripheral buffer and it receives system data.

If both the clock and data lines are HIGH, the keyboard/mouse peripheral transmits a LOW start bit, 8 data bits (least significant bit first), odd parity bit and the HIGH stop bit. Data will be valid before the falling edge, and beyond the rising edge of the clock.

During transmission the keyboard/mouse peripheral checks the clock for a HIGH level, at least every 60µs. If the PrimeCell KMI pulls the clock LOW from a HIGH level, after the keyboard/mouse starts sending data, a condition known as line contention occurs, and transmission is aborted. If line contention occurs before the rising edge of the tenth clock (parity bit), the keyboard/mouse returns the data line to a HIGH level. If line contention does not occur before the tenth clock, the keyboard/mouse completes the transmission.

After a transmission, the PrimeCell KMI can inhibit the keyboard/mouse until the system processes the input or until it requests a response to be sent.

2.3.5 Keyboard data input

When the system is ready to send data to the keyboard, it first checks if the keyboard is sending data. If the keyboard is sending but has not reached its tenth clock, the system can override the keyboard output by forcing the clock LOW. If the keyboard transmission has reached the tenth clock, the system must receive the transmission.

If the keyboard is not transmitting or if the system decides to override the keyboard output, the system forces the clock line LOW for more than 60µs while preparing to send. When the system is ready to send (data line will be LOW), it allows the clock to go to a HIGH level.

The keyboard checks the status of the clock line at intervals of no more than 8ms. If a request-to-send is detected, the keyboard counts eleven bits. After the tenth bit the keyboard forces the data line LOW, and counts one more bit (the stop bit). This is the line control which acknowledges to the system that the keyboard has received the data. Upon receipt of this signal the PrimeCell KMI returns a ready state, in which it can accept keyboard output.

If the keyboard data line is detected with a LOW level following the tenth bit (no stop bit), a framing error has occurred and the keyboard continues to count until the data line becomes HIGH. The keyboard then forces the data line LOW and sends a Resend.

Each system command or data transmission to the keyboard requires a response from the keyboard before the system can send its next output. The keyboard should respond within 16ms unless the system prevents the keyboard output. If the keyboard response is invalid or has a parity error, the system should send the command or data again.

Figure 2-3 and Figure 2-4 illustrate data sent from the keyboard and data sent to the keyboard respectively.

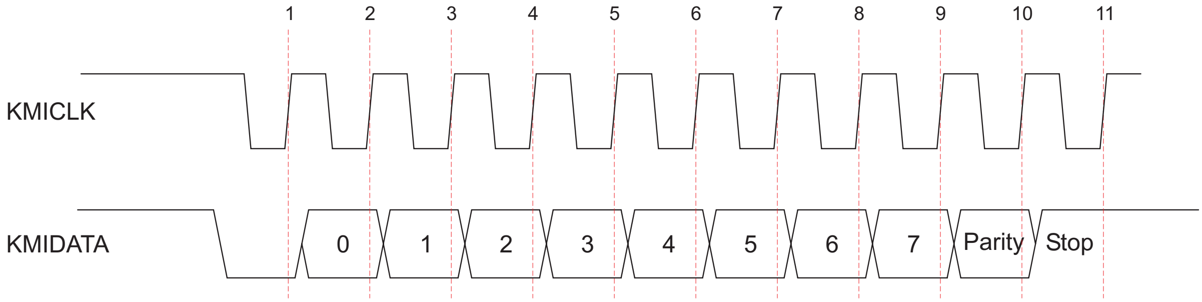


Figure 2-3 Data sent from the keyboard (KMI receive)

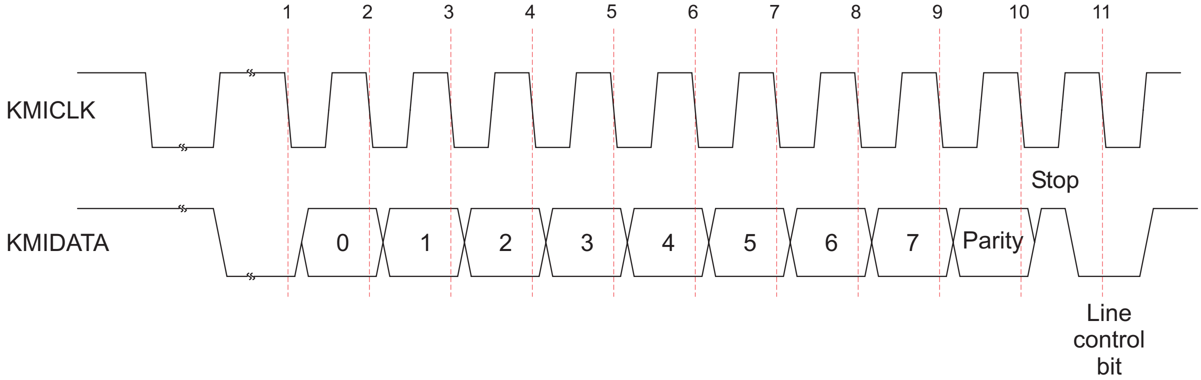


Figure 2-4 Data sent to the keyboard (KMI transmit)

2.3.6 Timing requirements

The timing requirements of the interface are shown in Figure 2-5 and Table 2-3.

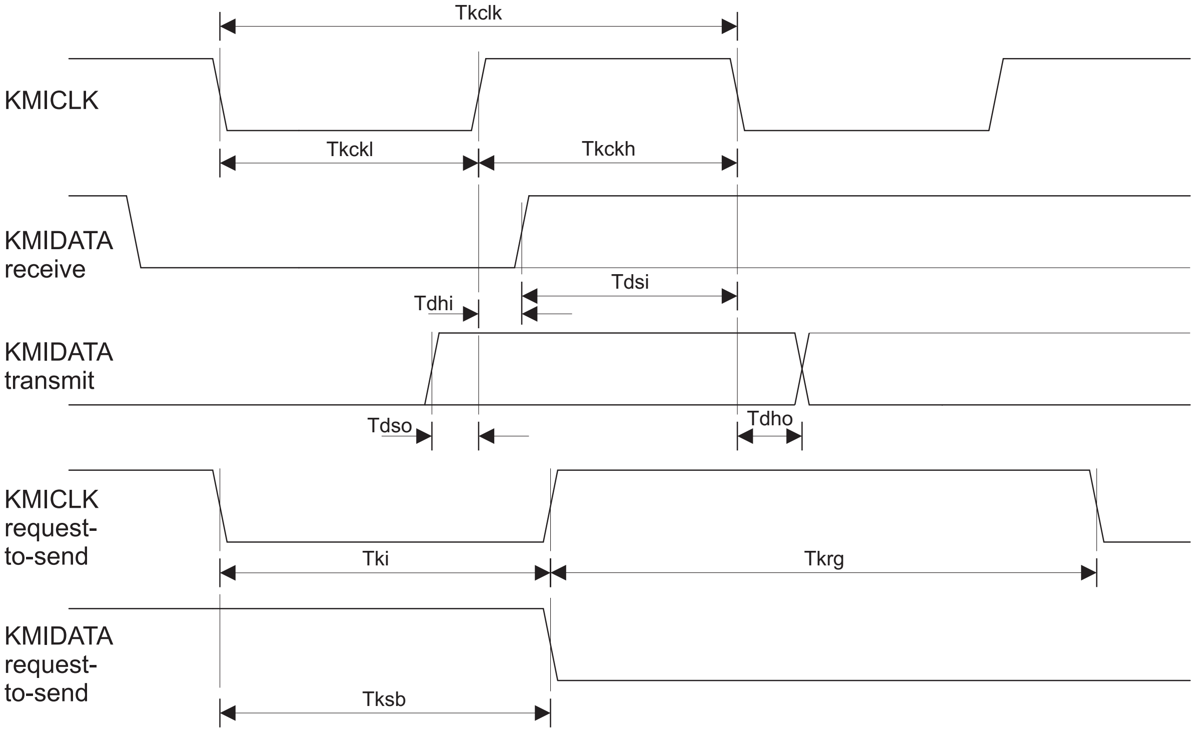


Figure 2-5 PrimeCell KMI timing and controller request to send protocol

Table 2-3 shows the nominal timings for the interface signals.

Table 2-3 Interface signals

Symbol	Parameters	Min	Typ	Max	Units	Notes
Tkclk	Keyboard clock period	1	-	100	µs	-
Tkckl	Keyboard clock LOW time	0.5	-	50	µs	-
Tkckh	Keyboard clock HIGH time	0.5	-	50	µs	-
Tdsi	Setup on **KMIDATA** to **KMICLK** falling for receive	1	-	Tkckh - 1 µs	µs	1

Table 2-3 Interface signals (continued)

Symbol	Parameters	Min	Typ	Max	Units	Notes
Tdhi	Hold on **KMIDATA** to **KMICLK** rising for receive	1	-	Tkckh - 1 µs	µs	1
Tdso	Setup on **KMIDATA** to **KMICLK** rising for transmit	Tkckl - 1 µs	-	Tkckl	-	1
Tdho	Hold on **KMIDATA** from **KMICLK** falling for transmit	0ns	-	1 µs	-	1
Tki	Time for which **KMICLK** is held low to request a send	60.0	64	85.3	µs	1, 2
Tkrg	**KMICLK** LOW from controller to **KMICLK** LOW from peripheral for request to send	1	-	-	µs	1
Tksb	**KMICLK** LOW to **KMIDATA** LOW hold time for request to send	1	-	-	µs	1, 3

Note 1 The **KMIDATA** and **KMICLK** signals in the diagrams and tables in this section relate to the respective external pad connections.

Note 2 Tki must be greater than 60.0 µs in order to guarantee that a keyboard will recognize a request-to-send command from the system. The request-to-send is timed by 512 cycles of the internal 8MHz clock, and this determines the maximum value of Tki.

Note 3 The **KMIDATA** will precede the **KMICLK** in this implementation, so the value for Tksb shown on the diagram above is negative, that is safe.

Chapter 3
Programmer's Model

This chapter describes the ARM PrimeCell *PS2* Keyboard/Mouse Interface (PL050) registers and provides details needed when programming the microcontroller. It contains the following sections:

3.1 About the programmer's model

The base address of the PrimeCell KMI is not fixed, and may be different for any particular system implementation. However, the offset of any particular register from the base address is fixed.

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

- locations at offsets `0x14`–`0x3c` and `0xa0`–`0xff` are reserved for possible future extensions
- locations at offsets +`0x40` through +`0x9c` are reserved for test purposes.

3.2 Summary of PrimeCell KMI registers

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

Table 3-1 PrimeCell KMI register summary

Address	Type	Width	Reset value	Name	Description
KMI Base + `0x00`	Read/ write	6	`0x00`	KMICR	Control register.
KMI Base + `0x04`	Read	7	`0x43`	KMISTAT	Status register.
KMI Base + `0x08`	Read/ write	8/ 8	`0x00`	KMIDATA	Received data (read)/ Data to be transmitted (write).
KMI Base + `0x0c`	Read/ write	4	`0x00`	KMICLKDIV	Clock divisor register.
KMI Base + `0x10`	Read	2	`0x00`	KMIIR	Interrupt status register.
KMI Base + `0x14–0x3c`	-	-	-	-	Reserved.
KMI Base + `0x40–9c`	-	-	-	-	Reserved (for test purposes).
KMI Base + `0xa0–ff`	-	-	-	-	Reserved.

3.3 Register descriptions

The following registers are described in this section:

- *KMICR: [6] (+ 0x00)*
- *KMISTAT: [7] (+ 0x04)* on page 3-5
- *KMIDATA: [8] (+ 0x08)* on page 3-6
- *KMICLKDIV: [4] (+ 0x0C)* on page 3-6
- *KMIIR: [2] (+ 0x10)* on page 3-7.

For each of the following register descriptions, the format of the title is:

Register name: [bit width] (Offset from Base).

3.3.1 KMICR: [6] (+ 0x00)

KMICR is the PrimeCell KMI control register. It contains five different bit fields that control various functions within the PrimeCell KMI. All bits are cleared to 0 on reset. Table 3-2 shows the bit assignments for the KMICR.

Table 3-2 KMICR register

Bits	Name	Type	Function
7:6	-	-	Reserved, read unpredictable, should be written as 0.
5	KMITYPE	Read/ write	0 = PS2/AT mode. 1 = No line control bit mode. The PrimeCell KMI defaults to PS2/AT mode. No line control bit mode is only used by keyboards that do not issue a line control bit to acknowledge data from the system.
4	KMIRXINTREn	Read/ write	Enable receiver interrupt. This bit field is used to enable the PrimeCell KMI receiver interrupt. If KMIRXINTREn = 1, the receiver interrupt is enabled.
3	KMITXINTREn	Read/ write	Enable transmitter interrupt. This bit field is used to enable the PrimeCell KMI transmitter interrupt. If KMITXINTREn = 1, the transfer interrupt is enabled.

Table 3-2 KMICR register (continued)

Bits	Name	Type	Function
2	KmiEn	Read/ write	The enable PrimeCell KMI bit field is used to enable the KMI. If KmiEn = 1, the KMI is enabled.
1	FKMID	Read/ write	The force KMI data LOW bit field is used to force the PrimeCell KMI data pad LOW regardless of the state of the KMI finite state machine (FSM). If FKMID = 1, the PrimeCell KMI data pad is forced LOW.
0	FKMIC	Read/ write	The force KMI clock LOW bit field is used to force the PrimeCell KMI clock pad LOW regardless of the state of the KMI FSM. If FKMIC = 1, the PrimeCell KMI clock pad is forced LOW.

3.3.2 KMISTAT: [7] (+ 0x04)

KMISTAT is the PrimeCell KMI status register. It is a 7-bit read-only register that indicates the status of the different lines in the PrimeCell KMI. Table 3-3 shows the bit assignments for the KMISTAT.

Table 3-3 KMISTAT register

Bits	Name	Type	Function
7	-	Read	Reserved, read unpredictable.
6	TXEMPTY	Read	This bit indicates that the transmit register is empty and ready to transmit. 0 = Transmit register full. 1 = Transmit register empty, ready to be written.
5	TXBUSY	Read	This bit indicates that the PrimeCell KMI is currently sending data. 0 = Idle. 1 = Currently sending data.
4	RXFULL	Read	This bit indicates that the receiver register is full and ready to be read. 0 = Receive register empty. 1 = Receive register full, ready to be read.
3	RXBUSY	Read	This bit indicates that the PrimeCell KMI is currently receiving data. 0 = Idle. 1 = Currently receiving data.

Table 3-3 KMISTAT register (continued)

Bits	Name	Type	Function
2	RXPARITY	Read	This bit reflects the parity bit for the last received data byte (odd parity).
1	KMIC	Read	This bit reflects the status of the **KMICLKIN** line after synchronizing and sampling.
0	KMID	Read	This bit reflects the status of the **KMIDATAIN** line after synchronizing.

3.3.3 KMIDATA: [8] (+ 0x08)

KMIDATA is the PrimeCell KMI transmit/receive data register. It is 8-bits wide. When KMIDATA is read, the data values received are accessed. When KMIDATA is written to, it is loaded into the transmit register and then serially shifted out onto the **nKMIDATAEN** pin. Table 3-4 shows the bit assignments for the KMIDATA register.

Table 3-4 KMIDATA register

Bits	Name	Type	Function
7:0	KMIDATA	Read/write	Receive/transmit register: Read - Receive register. Write - Transmit register.

3.3.4 KMICLKDIV: [4] (+ 0x0C)

KMICLKDIV is the PrimeCell KMI clock divisor register which is used to specify the division factor by which the input **KMIREFCLK** should be internally divided before further use. Based on the actual frequency of the clock signal in the system connected to the **KMIREFCLK** input, this register can be programmed with the appropriate divisor value in order to have an internal 8MHz signal.

The value programmed into this register should be between 0 and 15.

Table 3-5 shows the bit assignments for the KMICLKDIV register.

Table 3-5 KMICLKDIV register

Bits	Name	Type	Function
3:0	KMICLKDIV	Read/write	Clock divisor register. The divide ratio is given as: 1/(1 + KMICLKDIV) The internal nominal 8MHz clock signal will have a frequency F: where F**KMIREFCLK** = frequency of **KMIREFCLK** and KMICLKDIV = divider value written to KMICLKDIV register. For further details refer to Clock signals on page -6.

3.3.5 KMIIR: [2] (+ 0x10)

KMIIR is the PrimeCell KMI interrupt identification register. It is a read only register that contains interrupt status information for the KMI. Table 3-6 shows the bit assignments for the KMIIR.

Table 3-6 KMIIR register

Bits	Name	Type	Function
7:2	-	Read	Reserved, read unpredictable.
1	KMITXINTR	Read	This bit is set to 1 if the **KMITXINTR** transmit interrupt is asserted.
0	KMIRXINTR	Read	This bit is set to 1 if the **KMIRXINTR** receive interrupt is asserted.

3.4 Interrupts

Refer to *Interrupt generation logic* below for a functional description of the interrupt generation logic.

Two individual maskable active HIGH interrupts, **KMIRXINTR** and **KMITXINTR**, are generated by the PrimeCell KMI. Each of these interrupts may be enabled or disabled by changing the mask bits in KMICR. Setting the appropriate mask bit HIGH enables the corresponding interrupt. This allows for a system interrupt controller to provide the mask registers for each interrupt. In this way a global interrupt service routine would be able to read the entire set of sources from one wide register in the system interrupt controller. This is an attractive option where the time to read from the peripheral registers is significant compared to the CPU clock speed in a real time system.

The interrupts are also output as a combined single interrupt that is an OR function of the individual masked sources. **KMIINTR** is asserted if any of the two individual interrupts above are asserted and unmasked. This output can be connected to the system interrupt controller to provide another level of masking on a per-peripheral basis. This allows use of modular device drivers that always know where to find the interrupt source control register bits.

The status of the individual interrupt sources can be read from KMIIR.

3.4.1 Interrupt generation logic

The two maskable active HIGH interrupts, **KMITXINTR** and **KMIRXINTR**, and the combined single interrupt **KMIINTR** that are generated by the PrimeCell KMI, are described below:

KMITXINTR The transmit interrupt is set at completion of a transmit operation. A write to the KMIDATA register clears the **KMITXINTR**, provided that the interrupt was already set and is enabled.

KMIRXINTR On receiving a valid byte, the receive signal is asserted. Once set, it is cleared by a read of the KMIDATA register.

The individual masked interrupt outputs are also combined into a single output that is an OR function of the individual sources. This output can be connected to the system interrupt controller to provide another level of masking on an individual per-peripheral basis.

KMIINTR The combined KMI interrupt is asserted when any of the two individual interrupts above is asserted and the corresponding mask is enabled.

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 KMI test harness overview* on page 4-2
- *Scan testing* on page 4-4
- *Test registers* on page 4-5.

4.1 PrimeCell KMI test harness overview

The test block performs the following functions:

- generation of test clock enable (**ClkEn**), based on the contents of KMITCR
- generation of internal test resets (**nReset**, **nKMIRES**), based on the contents of KMITCR and the level on the **SCANMODE** pin
- write interface for the test register
- test input stimulus multiplexing for non-AMBA inputs (**KMICLKIN** and **KMIDATAIN**)
- capture of output signals (**nKMIDATAEN** and **nKMICLKEN**).

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

Off-chip 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 via the *Test Interface Controller* (TIC) AMBA bus master module.

During test the **KMIREFCLK** signal must be driven by the free-running **PCLK** clock signal so that the test vectors can be frequency-independent. This clock multiplexing must be performed externally from the PrimeCell KMI. Figure 4-1 shows the PrimeCell KMI test harness.

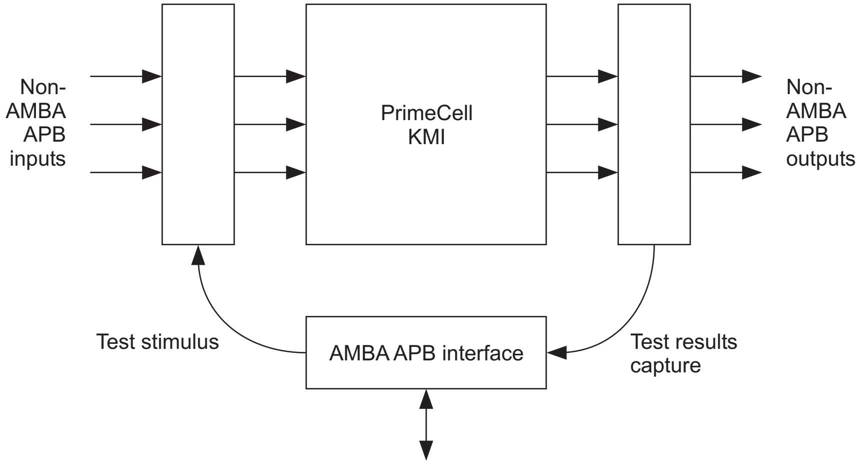


Figure 4-1

In the normal mode, the **ClkEn** signal is pulled high. If the TestModeEn bit is set, the device is said to be in the test mode. In the test mode, if the TestClkEn bit is not set, **ClkEn** is pulled HIGH. Otherwise, the internal test clock enable is selected, based on the RegClkMode bit value. If RegClkMode is set, a pulse on the **ClkEn** line is generated only on accesses to the KMITCER register address, that is, when the SelKMITCER input is asserted. If the RegClkMode bit is not set, a clock enable is generated on every access to the device. The **ClkEn** signal is generated with **PENABLE** and **PSEL** as gating terms.

When the **SCANMODE** pin is LOW, the test reset generated by ORing the external reset with the test reset is driven as the internal reset (**nReset** for **PCLK**-domain logic, and **nKMIRES** for **REFCLK**-domain logic). When **SCANMODE** is HIGH, the test reset does not have any effect, and the external resets **BnRES** and **nKMIRST** are driven on the internal reset lines. The test registers themselves are not affected by test reset.

When the TESTINPSEL bit in the KMITCR register is set, non-AMBA inputs such as **KMICLKIN** and **KMIDATAIN** are multiplexed with the corresponding bits in the KMITISR register.

4.2 Scan testing

This block 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.

During scan testing, the **SCANMODE** input must be driven HIGH to ensure that all internal data storage elements can be asynchronously reset. For normal use and application of manufacturing test vectors via the TIC, **SCANMODE** must be negated LOW.

4.3 Test registers

The PrimeCell KMI 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
KMI Base + 0x40–0x7c	Read/write	0	-	KMITCER	Test clock enable register.
KMI Base + 0x80	Read/write	5	0x00	KMITCR	Test control register.
KMI Base + 0x84	Read/write	4	0x00	KMITMR	Test mode register.
KMI Base + 0x88	Read/write	2	0x00	KMITISR	Test input stimulus register.
KMI Base + 0x8c	Read	3	0x03	KMITOCR	Test output capture register.
KMI Base + 0x90	Read	6	0x00	KMISTG1	Stage 1 timer register.
KMI Base + 0x94	Read	5	0x00	KMISTG2	Stage 2 timer register.
KMI Base + 0x98	Read	8	0x00	KMISTG3	Stage 3 timer register.
KMI Base + 0x9c	Read	4	0x06	KMISTATE	State register.

Each register shown in Table 4-1 is described below.

4.3.1 KMITCER [0] (+0x40–0x7c)

KMITCER is the PrimeCell KMI test clock enable register which is a 0-bit register. Table 4-2 shows the bit assignments for the KMITCER.

Table 4-2 KMITCER register

Bit	Name	Description
7:0	-	When in registered clock mode (refer to *Scan testing* on page 4-4), a test clock enable is produced only when this register is accessed (read or write).

KMITCER has a multiple word space in the register address map to allow for the generation of multiple test clock enable pulses.

4.3.2 KMITCR [5] (+0x80)

KMITCR is the PrimeCell KMI test control register. It is a general test register that controls operation of the PrimeCell KMI under test conditions. Table 4-3 shows the bit assignments for the KMITCR.

Table 4-3 KMITCR register

Bit	Name	Description
7:5	-	Reserved, read unpredictable, should be written as 0.
4	Test Input Select (TESTINPSEL)	By default, this bit is cleared to 0 for normal operation. This bit selects the source for the internal input signal for external non-AMBA inputs. When this bit is cleared to 0, the primary inputs are taken from the external pads (normal operation). When this bit is set to 1, the values programmed in KMITISR are used to drive the internal line.
3	Test Reset (TESTRST)	By default, this bit is cleared to 0 for normal operation when reset by **BnRES**. When this bit is set to 1, a reset is asserted throughout the module, EXCEPT for the test registers (this simulates reset by **BnRES** being asserted to 0).
2	Registered Clock Mode (REGCLK)	This bit selects the internal test clock mode: 0 = Strobe clock mode is selected, which generates a test clock enable on every APB access (read or write) to the block. Use of strobe clock mode allows testing with less test vectors when testing functions such as counters. The Test Clock Enable is generated from **PENABLE** ANDed with **PSEL**. 1 = Registered clock mode is selected, which only generates a test clock enable on an APB access to the KMITCER (KMI test clock enable register) location. This bit has no effect unless bit 0 and bit 1 are both set to 1. This bit is cleared to 0 by default on reset by **BnRES**.
1	Test Clock Enable (TESTCLKEN)	This bit selects the source of the test clock: 0 = The internal clock enable is continuously HIGH. 1 = The internal test clock enable is selected, so that the test clocks are enabled for only one period of the input clock per APB access. The internal clock enable mode depends on the setting of bit 2. This bit has no effect unless bit 0 is set to 1. This bit is cleared to 0 by default on reset by **BnRES**.
0	Test Mode Enable (TESTEN)	0 = Normal operating mode is selected. 1 = Test mode is selected. Bits 1 and 2 have no effect unless bit 0 is set to 1. This bit is cleared to 0 by default on reset by **BnRES**.

4.3.3 KMITMR [4] (+0x84)

KMITMR is the PrimeCell KMI test mode register which controls the specific test modes for the PrimeCell KMI.

If the S1NIB and S3NIB bits are 1, the first and third stage counters of the internal 17-bit timer that generates the 64µs and 16ms pulse are put in nibble mode. The counters count up as 00–11–22–33–44–55–66–77–88–99–AA–BB–CC–DD–EE–FF. If the STG1BYPASS and STG2BYPASS bits are 1, the enable for the second and third stage counters are bypassed and driven by the **Pulse8MHz** signal. These two features reduce the number of test vectors that are required to test the timer. Table 4-4 shows the bit assignments for the KMITMR.

Table 4-4 KMITMR register

Bit	Name	Description
7:4	-	Reserved, read unpredictable, should be written as 0.
3	S3NIB	0 = Normal operating mode is selected. 1 = Enable nibble mode for stage 3 of the timer.
2	S1NIB	0 = Normal operating mode is selected. 1 = Enable nibble mode for stage 1 of the timer.
1	STG2BYPASS	0 = Normal operating mode is selected. 1 = Allows the third stage of the timer to be clocked directly by the **Pulse8MHz** input.
0	STG1BYPASS	0 = Normal operating mode is selected. 1 = Allows the second stage of the timer to be clocked directly by the **Pulse8MHz** input.

4.3.4 KMITISR [2] (+0x88)

KMITISR is the PrimeCell KMI test input stimulus register. It provides test mode stimulus for the **KMIDATAIN** and the **KMICLKIN** inputs to the KMI. When the TESTINPSEL bit in the KMITCR register is 1, the values on the KMIDATAIN and the KMICLKIN bits in the KMITISR register are routed to the internal **KMIDATAIN** and **KMICLKIN** lines. Table 4-5 shows the bit assignments for the KMITISR.

Table 4-5 KMITISR register

Bit	Name	Description
7:2	-	Reserved, read unpredictable, should be written as 0.
1	KMIDATAIN	Test data input for the KMIDATAIN pin.
0	KMICLKIN	Test data input for the KMICLKIN pin.

4.3.5 KMITOCR [3] (+0x8c)

KMITOCR is the PrimeCell KMI test output capture register. It is a read-only register that provides observability for the primary outputs of the PrimeCell KMI. The **KMIINTR** interrupt is an OR of the two individual **KMITXINTR** and **KMIRXINTR** interrupts. The nKMIDATAEN and the nKMICLKEN bits provide visibility of the pad control signals. Table 4-6 shows the bit assignments for the KMITOCR.

Table 4-6 KMITOCR register

Bit	Name	Description
7:3	-	Reserved, read unpredictable.
2	KMIINTR	This bit returns the status of the **KMIINTR** output signal from the PrimeCell KMI. The **KMIINTR** interrupt is an OR of the two individual interrupts **KMIRXINTR** and **KMITXINTR**. 0 = **KMIINTR** is not asserted. 1 = **KMIINTR** is asserted.
1	nKMICLKEN	Read-only. This bit returns the status of the **nKMICLKEN** output signal from the PrimeCell KMI. The reset value of this output is 1 because the default value of this signal is 1. 0 = **nKMICLKEN** is asserted. 1 = **nKMICLKEN** is not asserted.
0	nKMIDATAEN	Read-only. This bit returns the status of the **nKMIDATAEN** output signal from the PrimeCell KMI. The reset value of this output is 1 because the default value of this signal is 1. 0 = **nKMIDATAEN** is asserted. 1 = **nKMIDATAEN** is not asserted.

4.3.6 KMISTG1 [6] (+0x90)

KMISTG1 is the PrimeCell KMI stage 1 timer register. It is a read-only register that provides observability of the stage 1 counter of the 17-bit timer. This counter stage is a 6-bit, free-running, up counter that operates on **REFCLK**. The counter reloads on overflow. Table 4-7 shows the bit assignments for the KMISTG1 register.

Table 4-7 KMISTG1 register

Bit	Name	Description
7:6	-	Reserved, read unpredictable.
5:0	KMISTG1	Read-only. These bits return the current count of the stage 1 counter of the 17-bit timer.

4.3.7 KMISTG2 [5] (+0x94)

KMISTG2 is the PrimeCell KMI stage 2 timer register. It is a read-only register that provides observability of the stage 2 counter of the 17-bit timer. This counter stage is a 3-bit, free-running, up counter that operates on **REFCLK**. The counter reloads on overflow. In addition, this register provides visibility of the 64µs and 16ms time-out pulses. Table 4-8 shows the bit assignments for the KMISTG2 register.

Table 4-8 KMISTG2 register

Bit	Name	Description
7:5	-	Reserved, read unpredictable.
4	Msec16	Read-only. Reflects the value of the 16ms time-out pulse.
3	Usec64	Read-only. Reflects the value of the 64µs pulse.
2:0	KMISTG2	Read-only. These bits return the current count of the stage 2 counter of the 17-bit timer.

4.3.8 KMISTG3 [8] (+0x98)

KMISTG3 is the PrimeCell KMI stage 3 timer register. It is a read-only register that provides observability of the stage 3 counter of the 17-bit timer. This counter stage is an 8-bit, free-running, up counter that operates on **REFCLK**. The counter reloads on reaching `0xf9` in normal mode. Table 4-9 shows the bit assignments for the KMISTG3 register.

Table 4-9 KMISTG3 register

Bit	Name	Description
7:0	KMISTG3	Read-only. These bits return the current count of the stage 3 counter of the 17-bit timer.

4.3.9 KMISTATE [4] (+0x9c)

KMISTATE is the PrimeCell KMI state register. It is a read-only register that provides observability of the internal states of the PrimeCell KMI controller state machine during testing. Table 4-10 shows the bit assignments for the KMISTATE register.

Table 4-10 KMISTATE register

Bit	Name	Description
7:4	-	Reserved, read unpredictable.
3:0	KMISTATE	Read-only. These bits return the current state of the KMI controller state machine.

Appendix A
Signal Descriptions

This appendix describes the signals that interface with the ARM PrimeCell PS2 Keyboard/Mouse Interface (PL050). It contains the following sections:

A.1 AMBA APB signals

The PrimeCell KMI module is connected to the AMBA APB as a bus slave. The AMBA APB signals are described in Table A-1.

Table A-1 AMBA APB signal descriptions

Name	Type	Source/ destination	Description
PCLK	Input	Clock generator	AMBA APB clock, used to time all bus transfers.
BnRES	Input	Reset controller	Bus reset signal (active LOW).
PSEL	Input	APB Bridge	When HIGH, this signal indicates that the PrimeCell KMI module has been selected by the AMBA APB bridge and that a data transfer is required.
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	AMBA APB transfer direction signal. When HIGH, this signal indicates a write to a peripheral and when LOW, a read from a peripheral.
PADDR[7:6] and **PADDR[4:2]**	Input	APB bridge	Subset of AMBA APB address bus, which is used by this peripheral for decoding its own register accesses.
PWDATA[7:0]	Input	APB bridge	Subset of unidirectional AMBA APB write data bus.
PRDATA[7:0]	Output	APB bridge	Subset of 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/ destination	Description
KMIREFCLK	Input	Clock generator	PrimeCell KMI reference clock.
nKMIRST	Input	Reset controller	PrimeCell KMI reset signal to the **KMIREFCLK** clock domain, and is active LOW. The reset controller must use **BnRES** to assert **nKMIRST** asynchronously, but negate it synchronously with **KMIREFCLK**.
KMITXINTR	Output	Interrupt controller	PrimeCell KMI transmit interrupt (active HIGH).
KMIRXINTR	Output	Interrupt controller	PrimeCell KMI receive interrupt (active HIGH).
KMIINTR	Output	Interrupt controller	PrimeCell KMI interrupt (active HIGH). A single, combined interrupt generated as an OR function of the two individually-maskable interrupts above.
SCANMODE	Input	Test controller	PrimeCell KMI scan test hold input. This signal must be asserted HIGH during scan testing to ensure that internal data storage elements can be reset asynchronously. **SCANMODE** must be negated LOW during normal use or when applying manufacturing test vectors via the TIC.

A.3 Signals to pads

Table A-3 describes the signals from the PrimeCell KMI to the input/output pads of the chip. It is the responsibility of the user to make proper use of the peripheral pins to meet the exact interface requirements.

Table A-3 Signals to pads

Name	Type	Source/ destination	Description
KMICLKIN	Input	Pad	PrimeCell KMI clock input.
KMIDATAIN	Input	Pad	PrimeCell KMI serial data input.

nKMIDATAEN	Output	Pad	PrimeCell KMI data pad enable (active LOW):
			LOW = Zero-driven.
			HIGH = Open drain high impedance.

nKMICLKEN	Output	Pad	PrimeCell KMI clock pad enable (active LOW):
			LOW = Zero-driven.
			HIGH = Open drain high impedance.