AMBA Keyboard/Mouse PS/2 Interface

Datasheet

ARM®

ARM DDI 0096B

AMBA Keyboard/Mouse PS/2 Interface
Datasheet

Release Information

Change history

Date	Issue	Confidentiality	Change
Nov 1996	A	Non-Confidential	First release.
Feb 1998	B	Non-Confidential	Second release.

Proprietary Notice

Confidentiality Status

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

Product Status

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

Web Address

`http://www.arm.com`

Contents
AMBA Keyboard/Mouse PS/2 Interface Datasheet

List of Tables
AMBA Keyboard/Mouse PS/2 Interface Datasheet

List of Figures
AMBA Keyboard/Mouse PS/2 Interface Datasheet

Preface

This preface introduces the *AMBA Keyboard/Mouse PS/2 Interface Datasheet*. It contains the following sections:

About this Datasheet

This is the *Datasheet* (DS) for the *AMBA Keyboard/Mouse PS/2 Interface*.

Intended audience

This Datasheet is written for system designers, system integrators, and programmers who are designing or programming a *System-on-Chip* (SoC).

Using this Datasheet

This Datasheet is organized into the following chapters:

Chapter 1 *Introduction*

Read this chapter for a high-level view of the Keyboard/Mouse PS/2 Interface and a description of its features.

Chapter 2 *Signal Descriptions*

Read this chapter for a description of the APB and internal signals.

Chapter 3 *Functional Description*

Read this chapter for an overview of the Keyboard/Mouse PS/2 Interface operation.

Chapter 4 *Programmer's Model*

Read this chapter for a summary of the Keyboard/Mouse PS/2 Interface registers, descriptions and interrupts.

Appendix A *Test Harness*

Read this appendix for a description of the test harness and test register descriptions.

Conventions

Conventions that this Datasheet can use are described in:

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

Typographical

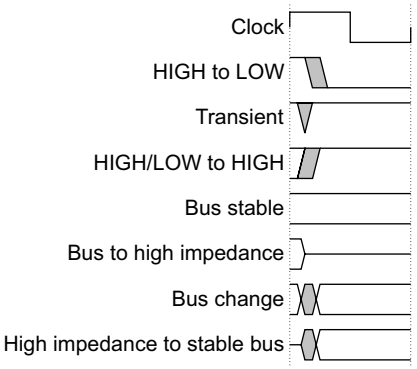
The typographical conventions are:

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.
`mono`space	Denotes a permitted abbreviation for a command or option. You can enter the underlined text instead of the full command or option name.
`monospace italic`	Denotes arguments to monospace text where the argument is to be replaced by a specific value.
`monospace bold`	Denotes language keywords when used outside example code.
< and >	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.

Further reading

This section lists publications by ARM and by third parties.

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

ARM publications

This Datasheet contains information that is specific to the Keyboard/Mouse PS/2 Interface. See the following document for other relevant information:

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

Feedback

ARM welcomes feedback on the Keyboard/Mouse PS/2 Interface 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 Datasheet

If you have any comments on this Datasheet, 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 describes an overview of the AMBA Keyboard/Mouse PS/2 Interface Datasheet.

1.1 Overview

The Keyboard/Mouse Interface is an AMBA slave module that connects to the *Advanced Peripheral Bus (APB)*. For more information about AMBA, please refer to the *AMBA Specification* (ARM IHI 0001).

1.1.1 Features

The AMBA Keyboard/Mouse Interface has the following features:

- can be used as a keyboard or mouse interface
- simple two pin serial communication link
- programmable interface enable and disable
- register driven keyboard clock and data override
- single transmit and receive status register
- separate transmit and receive interrupts
- can be used in polled or interrupt driven systems
- hardware derived odd parity and high stop bit for transmitted words
- parity bit of received words available for checking purposes.

Additional test registers and modes are implemented to provide efficient testing.

Chapter 2
Signal Descriptions

The chapter contain the following sections:

2.1 APB Signals

The Keyboard and Mouse Interface peripheral is connected to the APB bus as a slave. Table 2-1: APB signal descriptions describes the APB signals used.

Table 2-1 APB signal descriptions

Name	Type	Source/Destination	Description
BnRES	In	Reset Controller	System Reset Clock Initiates a cold reset.
PA[5:2]	In	APB Bus	APB Address Internal register map address source.
PD[7:0]	BiDir	APB Bus	APB Data Driven by APB bridge during writes, driven by the peripheral during reads.
PSEL	In	APB Bus	APB Slave Select When HIGH, this signal indicates that this peripheral is currently selected for access.
PSTB	In	APB Bus	APB Strobe Times all APB bus transfers.
PWRITE	In	APB Bus	APB Write/Read Defines the access type. Write when HIGH, read when LOW.

2.2 Internal Signals

Table 2-2 lists the internal signal descriptions.

Table 2-2 Internal signal descriptions

Name	Type	Source/Destination	Description
KbClkIn	In	KCLK Pad	Keyboard Clock.
KbDatIn	In	KDATA Pad	Keyboard Data.
KbdRXint	Out	Interrupt Controller	Keyboard Receive Interrupt. Indicates that data is available to be read by the receive buffer.
KbdTXint	Out	Interrupt Controller	Keyboard Transmit Interrupt. Indicates that the keyboard interface buffer is empty.
nKbClkEn	Out	KCLK Pad	Keyboard Clock Pad Enable.
nKbDataEn	Out	KDATA Pad	Keyboard Data Pad Enable.
RefClk	In	REFCLK Pad	Reference Clock. Free running clock.

Chapter 3
Functional Description

This chapter contains the following sections:

3.1 Block diagram

Figure 3-1 shows the Keyboard/Mouse Interface block diagram.

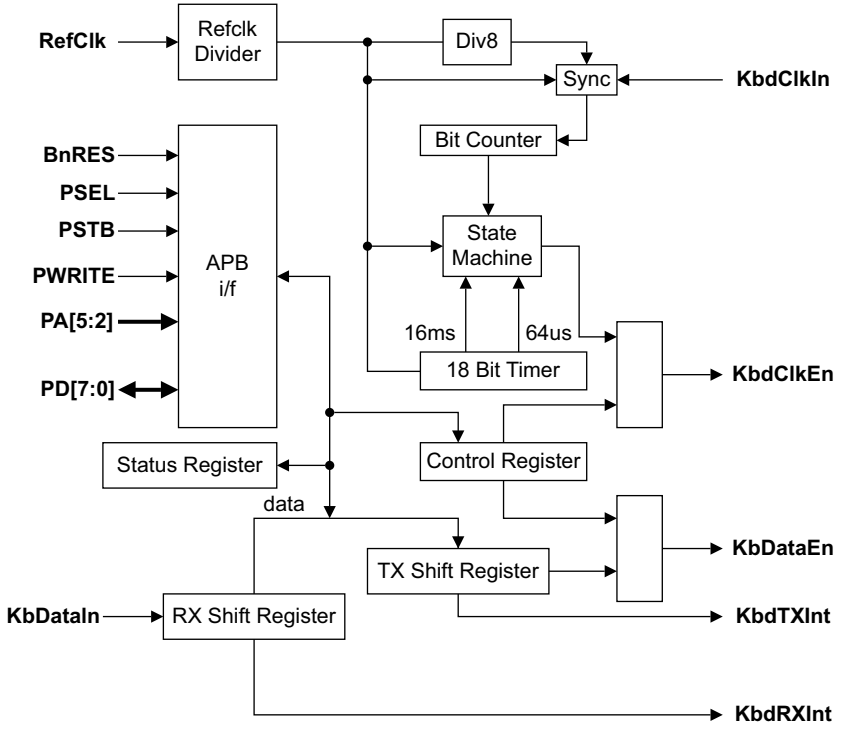


Figure 3-1 Keyboard/Mouse Interface block diagram

3.2 Overview of the Keyboard/Mouse Interface Operation

The Keyboard interface generates two interrupts:

KbdTXlnt indicates that the transmit buffer is empty and that another byte can be transmitted.

KbdRXInt indicates that a byte has been received by the interface.

The keyboard interface is held in reset until the enable bit in the KBDCR control register is set. The interface can be controlled on the basis of the interrupts generated, or by polling the status flags in the control register. The TX interrupt is generated when the transmit buffer has been emptied and the interface is ready to be programmed with another character for transmission. The RX interrupt is set when a complete character has been received in the receive buffer, and the byte is ready to be read from the register. The received data parity bit, RXP, is available in the control register. Odd parity is used. The keyboard and mouse interface state machines are clocked at a frequency of 8MHz, derived from the **RefClk** input. A frequency just less than 8MHz should also allow the block to operate properly.

The **KbClkIn** signal is sampled by a internal 1MHz clock, which is derived from the 8MHz internal clock, to reduce the effects of noise and metastability.

In a typical configuration, **KbdClkIn** and **KbdDataIn** are connected to open drain I/O pads with pull-up resistors, for example KCLK and KDATA at the peripheral interface. These are controlled by their respective **KbdClkEn** and **KbdDataEn** signals.

The external clock signal is always driven by the keyboard device, except when the host system wants to prevent transmission by the keyboard.

When the interface has been enabled, it awaits one of two possible events:

- If data is written to the transmit register, the **KbdTXInt** interrupt is cleared LOW and the byte is subsequently transmitted. The transmit sequence is initiated by pulling the external clock pad (KCLK) LOW, then returning it to HIGH impedance, followed by pulling the external data pad (KDATA) LOW. The serial byte, parity and stop bits are then transferred to the keyboard.
- If the clock signal from the keyboard goes LOW it signals the start of a transmission by the keyboard device. This clocks the incoming byte (terminated by parity and stop bits) into the receive register, setting **KbdRXInt** HIGH on completion. Further transmissions by the keyboard device cannot occur until the received data is read, thereby clearing the **KbdRXInt** interrupt.

If simultaneous transmission and reception occur, the keyboard device is inhibited from sending its data until transmission is complete.

If the transmit register is written during reception, the transmission from the interface is delayed until reception from the keyboard device has completed.

If the transmit register is written while the receive register is full, the transmission from the interface occurs and it is possible for the receive register to be read during this transmission.

Transmission and reception timeouts of 64μs and 16ms are applied respectively, though no timeout interrupts are generated.

The timing requirements of the interface are shown in Figure 3-2: Keyboard/mouse controller receive protocol.

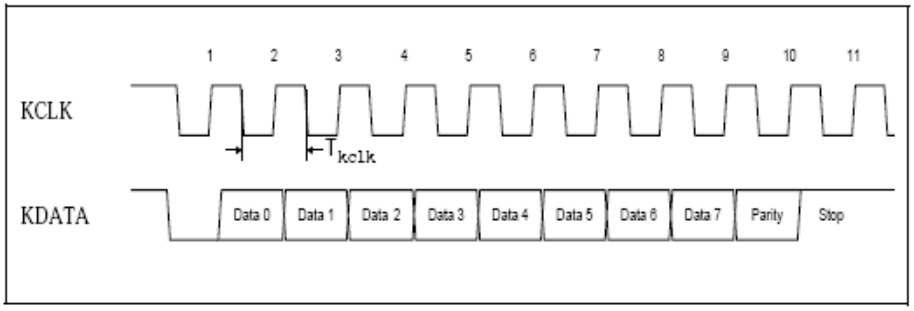


Figure 3-2 Keyboard/mouse controller receive protocol

The timing requirements of the interface are shown in Figure 3-3: Keyboard/mouse timing and controller request to send protocol.

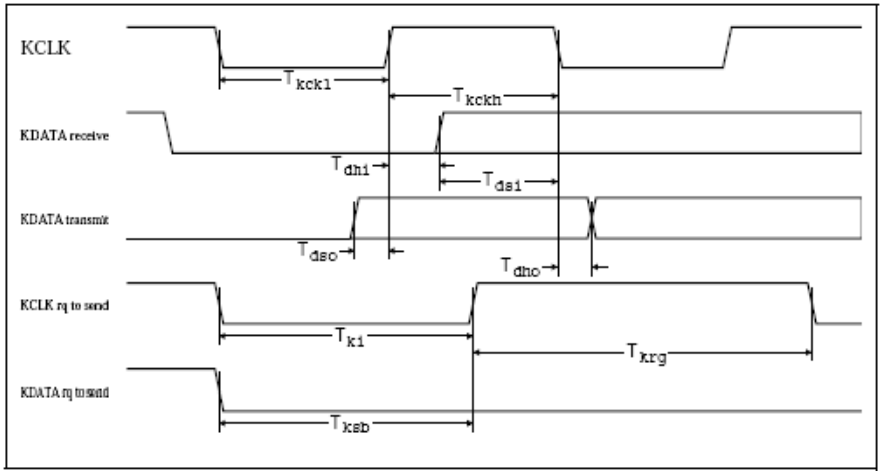


Figure 3-3 Keyboard/mouse timing and controller request to send protocol

Table 3-1: Keyboard/mouse interface timings shows the normal timings for the interface signals.

Table 3-1 Keyboard/mouse interface timings

Symbol	Parameters	Minimum	Type	Maximum	Units	Notes
Tkclk	Keyboard clock period	1		100	µs	
TkckI	Keyboard clock LOW time	0.5		50	µs	
Tkckh	Keyboard clock HIGH time	0.5		50	µs	
Tdsi	Setup on **KDATA** to **KCLK** falling for Receive	1		Tkckh - 1µs	µs	1
Tdhi	Hold on **KDATA** from **KCLK** rising for Receive	1		Tkckh - 1µs	µs	1
Tdso	Setup on **KDATA** to **KCLK** rising for Transmit	Tkckl - 1µs		Tkckl		1
Tdho	Hold on **KDATA** from **KCLK** falling for Transmit	0ns		1µs		1
Tki	Time for which **KCLK** is held LOW to request a send	63.5	64	64.5	µs	1
Tkrg	**KCLK** LOW from controller to **KCLK** LOW from peripheral for request to send	1			µs	1
Tksb	**KCLK** LOW to **KDATA** LOW hold time for request to send	1			µs	1, 2

Note

- The **KDATA** and **KCLK** signals in the diagrams and tables in this section relate to the respective external pad connections.
- The **KDATA** will proceed the **KCLK** in this implementation, so the value for Tksb shown on the diagram above is negative, that is, safe.

Chapter 4
Programmer's Model

This chapter contains the following sections:

4.1 Introduction

The base address of the Keyboard/Mouse Interface is not fixed and depends on the particular system implementation. However, the offset of any register from the base address is always fixed.

Note

The locations at offsets `+0x20` through `+0x2C` are reserved for test purposes and should not be used during normal operation.

4.2 Summary of Keyboard/Mouse Interface Registers

Table 4-1 lists the keyboard/mouse interface registers.

Table 4-1 Keyboard/mouse interface registers

Offset	Type	Width	Reset value	Name	Description
0x00	R/W	4	0x0	KBDCR	Keyboard Control Register
0x04	R/W	8	b'1000XXX	KBDSTAT	Keyboard Status Register
0x08	R/W	8	0x00	KBDDATA	Keyboard Data transfer Register
0x0C	R/W	4	0x0	KBDCLKDIV	Keyboard reference clock divisor value to provide the 8MHz internal clock

4.3 Register descriptions

The following registers are provided at offsets from a base address decoded to drive the **PSEL** input:

KBDCR (MSECR)

Control register

KBDSTAT (MSESTAT)

Status register

KBDDATA (MSEDATA)

Transmit/receive data register

KBDCLKDIV (MSECLKDIV)

Clock division register

There are additional test registers described in Appendix A *Test Harness*.

4.3.1 KBDCR [4] (+0x00)

This is a control register that provides direct access to the **KCLK** and **KDATA** outputs, and a bit to enable the interface. A write access to this location updates the control signals. A read returns the three values written. Table 4-2: KBDCR register details shows the bit details for the register.

Table 4-2 KBDCR register details

Bit	Name	Function
3	ENA	Keyboard enable bit. 0 = disable 1 = enable
2	N/A	Reserved - Reads as zero.
1	FKD	Force KDATA pad LOW 1 = Force the pad LOW regardless of the state of the Keyboard FSM.
0	FKC	1 = Force KCLK pad LOW regardless of the state of the Keyboard FSM.

4.3.2 KBDSTAT [8] (+0x04)

This read-only register provides status information such as the busy state of the FSM, parity of last received data, and so on. Table 4-3: KBDSTAT register details shows the bit details of the register.

Table 4-3 KBDSTAT register details

Bit	Name	Function
7	TXE	TX register. empty: 0 = not ready 1 = enable, ready to transmit
6	TXB	TX Busy: 1 = currently sending data
5	RXF	RX Full: 1 = Rx. register full, ready to be read
4	RXB	RX Busy: 1 = currently receiving data
3	ENA	Enable bit: 1 = Function enable
2	RXP	Parity bit indication for last received data byte (odd parity)
1	KBD	Value on KDATA pin (after synchronizing)
0	KBC	Value on KCLK pin (after synchronizing and sampling by Div8 clock)

4.3.3 KBDDATA [8] (+0x08)

The data register is used to write bytes to be transmitted across the serial link and to read bytes received. Table 4-4: KBDDATA register details, shows the bit details of the register.

Table 4-4 KBDDATA register details

Bit	Name	Function
7:0	KBDAT	TX/RX data

4.3.4 KBDCLKDIV [4] (+0x0C)

The KBDCLKDIV register is used to define the divide ration required to generate an internal 8MHz keyboard interface clock from the **RefClk** input clock.

Table 4-5 KBDCLKDIV Register bit function

Bits	Name	Function
3:0	DivVal	Divide Value. The **RefClk** division ration is given by the equation divide ratio = DivVal + 1

4.4 Interrupts

Two interrupt signals, **KbdTXInt** and **KbdRXInt**, are available as internal output signals from the peripheral.

The **KbdTXInt** interrupt is asserted HIGH when the keyboard interface transmit buffer is empty, in other words it is ready to be programmed with another character for transmission.

The **KbdRXInt** interrupt is asserted HIGH when a complete character has been received and transferred to the receive buffer, and is available for reading.

Appendix A
Test Harness

This appendix describes the following sections.

A.1 Test interface

Extra registers are provided inside the Keyboard/Mouse interface for test purposes only, and should not be accessed during normal mode of operation. They are memory mapped as follows:

Table A-1 Summary of Keyboard/Mouse interface test registers

Offset	Type	Width	Reset Value	Name	Description
0x20	W	8	0x00	KBDTEST1	Keyboard Test Register 1 Test control register.
0x24	W	3	0x0	KBDTEST2	Keyboard Test Register 2 Primary I/O control register.
0x28	R	8	b'0000XX00	KBDTEST3	Keyboard Test Register 3 Internal signal monitor register.
0x2C	R	8	0x00	KBDTEST4	Keyboard Test Register 4 State machine status register.

A.2 Test Register Descriptions

A.2.1 KBDTEST1 [7] (+0x20)

This write-only register provides control over the clocks source to the block for test purposes.

Table A-2 KBDTEST1 register details

Bit	Name	Function
7	CD	Clear the Divide by 8 counter Writing a one to this location generates a pulse to reset the divide by 8 counter.
6	RC1	Register Div8 Clock (1MHz) Used in test mode as a source for the 1MHz synchronizing clock signal, by successive write to the location.
4:3	C(1:0)	Clock Select Lines **X0** Normal operation. The input 8MHz is selected and the Div8 (1MHz) clock is generated from the 8MHz clock. **0** **PSEL** ANDed with **PSTB** is driven as the internal 8MHz clock. This means that when this clock source is selected, every access to the block generates a positive clock pulse internally. The Div8 clock is generated from this internal 8MHz clock in this mode, ie. the pulsed clock. **11** The MC and RC1 bits are used as the clock sources for the 8MHz and 1MHz clocks respectively. To generate an internal clock transition, program these bits with the last written values inverted (in other words, write 0 and then 1 creates a LOW to HIGH transition and vice versa).
2	T2	When 1, the output of the timer prescaler is connected to KBDCR bit[2] (RXP) allowing its LOW time of 16µs +/- 7 µs to be checked.
1	T1	When set to 1, this bit allows the timeout section of the timer (the last 8 bits) to be clocked at 8 times the normal prescaler output rate.
0	T0	This bit allows the middle section of the timer (for timing 64µs) to be clocked by the 8MHz input directly, rather than from the prescaler output.

A.2.2 KBDTEST2 [4] (+0x24)

This write-only test register provides control of the clock and other signals. Table A-3: KBDTEST2 register details shows the bit details of the register.

Table A-3 KBDTEST2 register details

Bit	Name	Function
3	TICBnRES	When HIGH, this will cause all of the peripherals normal mode logic/registers to be reset, in other words, EXCLUDING the test registers.
2	RKC	Registered Keyboard Clock bit. Used to drive the Keyboard Clock path internally.
1	RKD	Registered Keyboard Data bit. Used in test mode to drive the Keyboard Data path internally.
0	SEL	Mux Select line for use in test mode. When set to 1, this bit selects the RKC and RKD bits for use internally. Otherwise the normal block inputs are used.

A.2.3 KBDTEST3 [8] (+0x28)

This test register is read-only and provides visibility of the pad control signal and other internal signals. Table A-4: KBDTEST3 register details shows the bit details of the register.

Table A-4 KBDTEST3 register details

Bit	Name	Function
7	ms_16	Reflects the value from the end of timeout chain.
6	µs_64	Reflects the value from the middle of timeout chain.
5	µs_16	Reflects the value from the first stage of timeout chain.
4	DIV8	Reflects the value at the output of the divide by 8 counter for the Div8 clock.
3	DIn	Reflects the value of the internal **KbDataIn** signal.

Table A-4 KBDTEST3 register details (continued)

Bit	Name	Function
2	CIn	Reflects the value of the internal **KbClkIn** signal before sampling by the Div8 (1MHz) clock.
1	KD	Reflects the value of the **nKbDataEn** output signal.
0	KC	Reflects the value of the **nKbClkEn** output signal.

A.2.4 KBDTEST4 [8] (+0x2C)

This test register is read-only, and provides visibility of the internal state machine and the bit counter decode signals. Table A-5: KBDTEST4 register details shows the bit details of the register.

Table A-5 KBDTEST4 register details

Bit	Name	Function
7	BC12	Bit Counter = 12 - Input to state machine.
6	BC11	Bit Counter = 11 - Input to state machine.
5	TRES	Reset Timer - State bit 5.
4	CLKOE	Clock Output Enable - State bit 4.
3	CRES	Reset bit Counter - State bit 3.
2	RXB	Receive Busy - State bit 2.
1	TXB	Transmit Busy - State bit 1.
0	SRX	Set Receive interrupt - State bit 0.