ARM PrimeCell™ Smart Card Interface (PL130)

Technical Reference Manual

ARM®

ARM DDI 0148B

ARM PrimeCell Smart Card Interface (PL130)
Technical Reference Manual

Release Information

Change history

Date	Issue	Confidentiality	Change
January 1996	A	Non-Confidential	First release.
March 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
ARM PrimeCell Smart Card Interface (PL130) Technical Reference Manual

List of Tables
ARM PrimeCell Smart Card Interface (PL130) Technical Reference Manual

List of Figures
ARM PrimeCell Smart Card Interface (PL130) Technical Reference Manual

Preface

This preface introduces the *ARM PrimeCell Smart Card Interface (PL130) Technical Reference Manual*. It contains the following sections:

About this Technical Reference Manual

This is the *Technical Reference Manual* (TRM) for the *ARM PrimeCell Smart Card Interface*.

Intended audience

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

Using this Technical Reference Manual

This Technical Reference Manual is organized into the following chapters:

Chapter 1 *Introduction*

Read this chapter for a preamble on the ARM PrimeCell Smart Card Interface (SCI) and its features.

Chapter 2 *Functional Overview*

Read this chapter for a description of the block diagram and functionality of the PrimeCell SCI.

Chapter 3 *Programmer's Model*

Read this chapter for a description of the block diagram and functionality of the PrimeCell SCI.

Chapter 4 *Programmer's Model for Test*

Read this chapter for a description of the test registers and signals of the PrimeCell SCI.

Appendix A *ARM PrimeCell Smart Card Interface (PL130) Signal Descriptions*

Read this chapter for a description of the AMBA APB signals, on-chip signals and signals to pads.

Conventions

Conventions that this Technical Reference Manual can use are described in:

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

Typographical

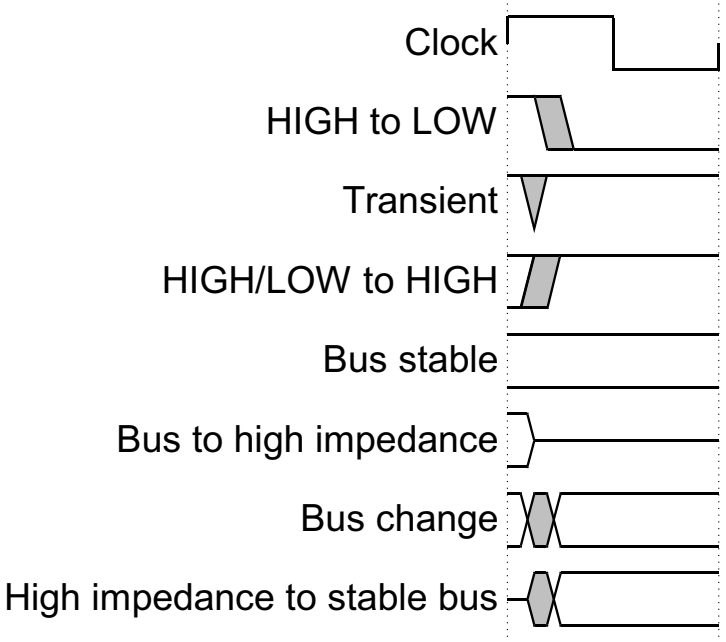
The typographical conventions are:

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.
`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

See the following documents for other relevant information:

- *AMBA® Specification (Rev 2.0)* (ARM IHI 0011).
- *ARM PrimeCell Smart Card (PL130) Integration Manual* (PL130 INTM 0000)
- *ARM PrimeCell Smart Card (PL130) Design Manual* (PL130 DDES 0000).

Other publications

EMV '96 Integrated Circuit Card, Specifications for Payment Systems (Version 3.0 June 1996) hereby referred to as *EMV Standard*.
Part I: Electromechanical characteristics, logical interface and transmission protocols. Published by Europay International S.A., Mastercard International Inc, and Visa International Association.

ISO/IEC 7816-3: Information technology - Identification cards - Integrated circuit(s) card(s) with contacts.

Part 3: Electronic signals and transmission protocols.

Feedback

ARM Limited welcomes feedback on both the ARM PrimeCell Smart Card Interface (PL130), and the 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.

General suggestions for additions and improvements are also welcome.

Feedback on this Technical Reference Manual

If you have any comments on this Technical Reference 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 Smart Card Interface (PL130) and contains the following sections:.

1.1 About the ARM PrimeCell Smart Card Interface (PL130)

The PrimeCell *Smart Card Interface* (SCI) is an *Advanced Microcontroller Bus Architecture* (AMBA) compliant System-on-a-Chip peripheral that is developed, tested and licensed by ARM. Refer to Figure 1-1 on page 1-3 to for a block diagram of the SCI.

The PrimeCell SCI is an AMBA slave module and connects to the AMBA *Advanced Peripheral Bus* (APB), and interfaces to an external Smart Card reader. The SCI can autonomously control data transfer to and from the smart card. Transmit and receive data FIFOs are provided to reduce the required interaction between the CPU core or an AMBA *Advanced High-performance Bus* (AHB) master and the peripheral.

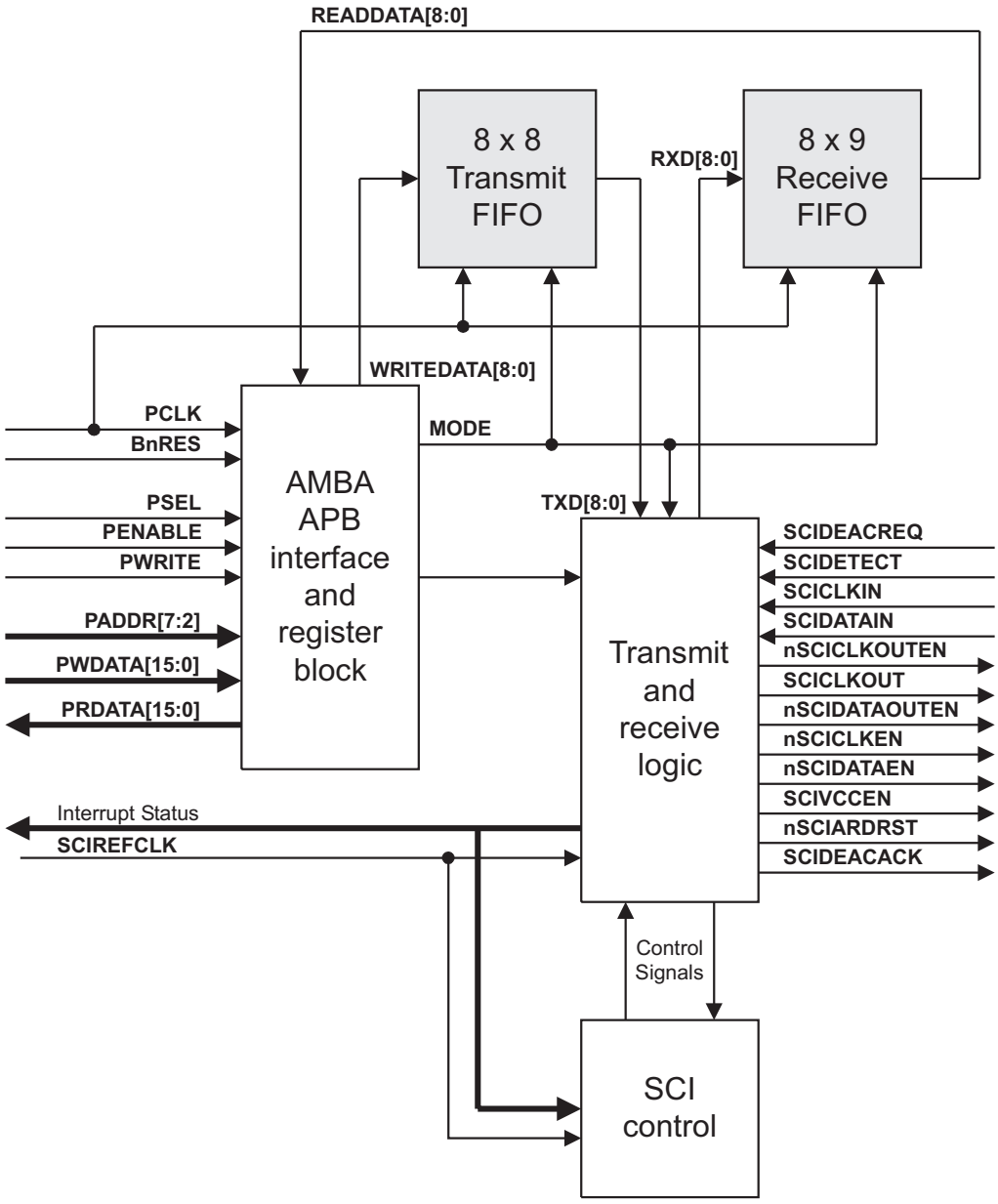


Figure 1-1 PrimeCell SCI block diagram and PADS connections

The features of the PrimeCell SCI are covered under the following headings:

- *Features of the PrimeCell SCI*
- *Programmable parameters*.

1.1.1 Features of the PrimeCell SCI

The following features are provided by the PrimeCell SCI:

- compliance to the *AMBA Specification (Rev 2.0)* onwards for easy integration into *System-on-a-Chip* (SoC) implementation
- supports asynchronous T0 and T1 transmission protocols
- supports clock rate conversion factor F = 372, with bit rate adjustment factors D = 1, 2 or 4 supported
- eight character deep buffered TX and RX paths
- direct interrupts for TX and RX FIFO level monitoring
- interrupt status register
- hardware initiated card deactivation sequence on detection of card removal
- software initiated card deactivation sequence on transaction complete
- limited support for synchronous smart cards via registered input/output.

1.1.2 Programmable parameters

The following key parameters are programmable:

- Smart Card clock frequency
- communication baud rate
- protocol convention
- card activation time
- card deactivation time
- check for maximum time for first character of *Answer To Reset* (ATR) reception
- check for maximum duration of ATR character stream
- check for maximum time for receipt of first character of data stream
- check for maximum time allowed between characters
- character guard time
- clock guard time
- transmit character retry
- receive character retry

- transmit FIFO tide level
- receive FIFO tide level.

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

1.2 AMBA compatibility

The PrimeCell SCI 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 data 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. This block can also be used in a system with a big-endian memory organization, and several methods of achieving this are described in the *PrimeCell Smart Card Interface (PL130) Integration Manual.*

Chapter 2
Functional Overview

This chapter describes the major functional blocks of the ARM PrimeCell Smart Card Interface (PL130).

2.1 ARM PrimeCell Smart Card Interface (PL130) overview

The PrimeCell *Smart Card Interface* (SCI) conforms to Part 1 of the *Integrated Circuit Specification for Payment Systems Electromechanical Characteristics, Logical Interface, and Transmission Protocols* (Version 3.0 June 1996). This standard is published jointly by Europay International S.A., Mastercard International Incorporated, and Visa International Service Association and is subsequently referred to as the *EMV Standard*. This standard refers to, and is based upon, the *ISO 7816* series of standards. The user is expected to be familiar with both the *EMV Standard* and *ISO 7816-3*.

The PrimeCell SCI performs:

- serial to parallel conversion on data received
- parallel to serial conversion on data transmitted to an external smart card.

The host CPU reads and writes data and control information via the AMBA APB interface. The transmit and receive paths are buffered with internal FIFO memories allowing up to 8-bytes to be stored independently in both transmit and receive modes.

The PrimeCell SCI includes a programmable baud rate generator and, in conjunction with a secondary value counter, provides programmable *elementary time units* (etus).

This peripheral has been designed such that it allows close monitoring of all stages of a typical card session via mask enabled interrupts. The interrupt architecture allows a choice of:

- a polled approach by examining the interrupt status register on assertion of a single common interrupt
- the interrupt sources can be fed direct to the interrupt controller for immediate identification.

The transmit and receive FIFO interrupts are asserted and de-asserted automatically depending on their programmed trigger threshold levels.

Parity errors are automatically checked by hardware on received data.

Interpretation of the received data stream is always performed by the user's application software.

Card deactivation is initiated automatically via hardware on card removal, but it is also possible to deactivate the card via software by writing to the respective control register. Also a second deactivation request input is available for direct control via an alternative hardware source.

Although some parameter values are currently fixed by the *EMV Specification*, the design has taken into account that these may be changed in the future. The design strategy means that the user should be able to incorporate future changes to the specification with minimal effort.

A brief summary of the individual blocks of the design is given in the following sections.

2.2 PrimeCell SCI functional description

A diagrammatic view of the PrimeCell SCI is shown in Figure 2-1 on page 2-5.

Note

For clarity test logic is not shown.

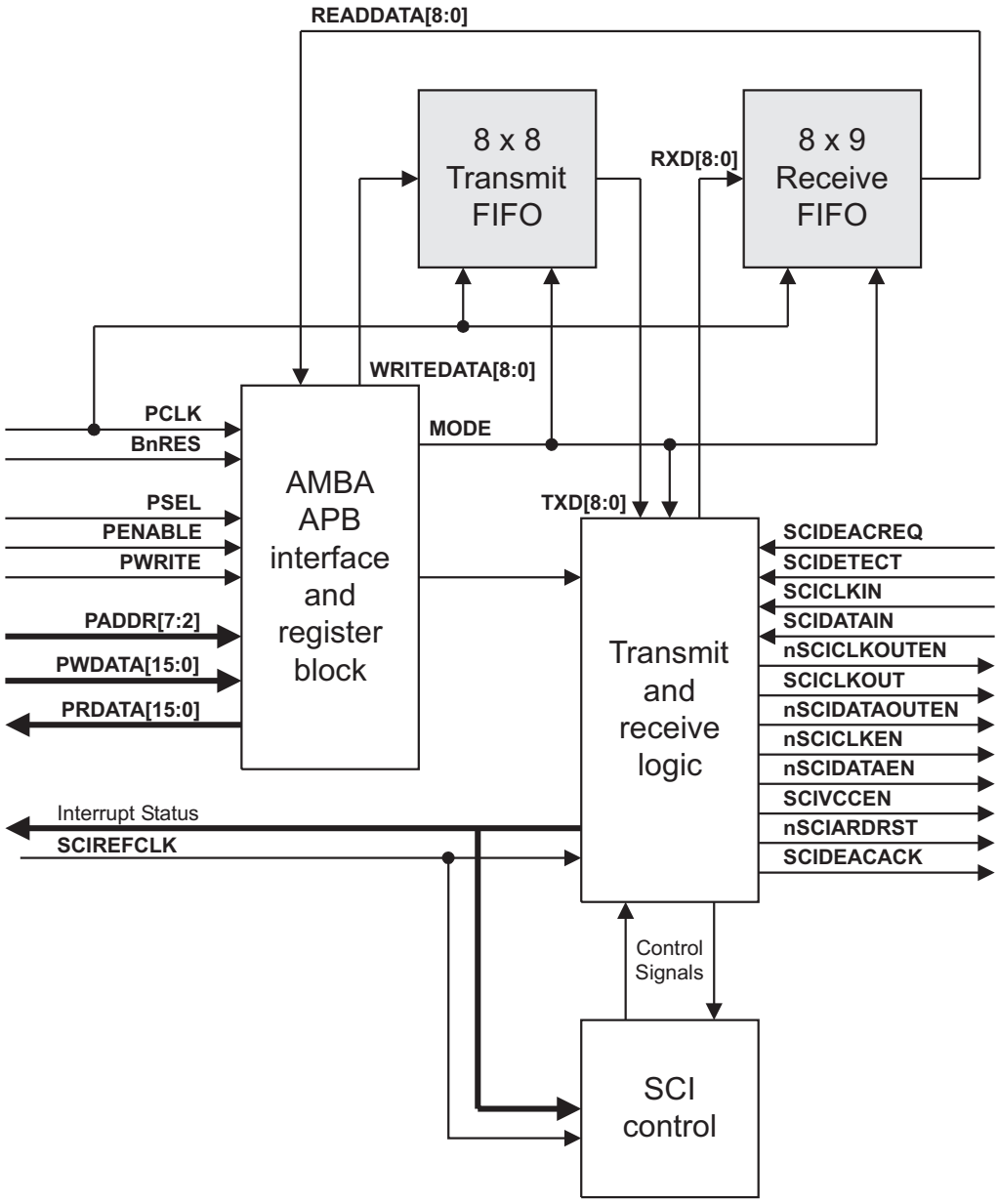


Figure 2-1 PrimeCell SCI block diagram

2.2.1 AMBA APB interface

The AMBA APB interface generates read and write decodes for accesses to the status/control registers and transmit/receive FIFO memories.

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.

2.2.2 Register block

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

2.2.3 Transmit and receive logic

This block implements the main functionality of the PrimeCell SCI with the control information being fed from the PrimeCell SCI control block. It contains all the counters used for timing the various stages of transactions during a card session. Also, it drives all the card control signals with the exception of data.

In this implementation some counters are multi-purpose, that is, they are loaded with values required by a particular transaction stage, thereby minimizing the logic required to perform the overall function.

The SCIBAUDCNT and SCIVALUECNT counters provide the means of performing the bit rate conversion and bit rate adjustment. Together they define the *elementary time unit* (etu) period.

The card clock, **SCICLKOUT**, is derived from the output of the SCICLKICCCNT divide counter.

The SCIACTTIME counter is multi-purpose and is used for six different operations:

- debounce timing
- activation timing
- *answer to reset* (ATR) start time
- ATR duration time
- deactivation time
- warm reset timing.

The SCIDATATIME counter is a multi-purpose counter that is used to measure:

- block guard time

- character guard time
- time between characters
- time between characters and blocks.

The SCIACTTIME and SCIDATATIME counters are controlled by the SCI Control block.

The transmit and receive block controls the sequencing of the power, clock, reset and data line during activation and deactivation processes.

With the exception of the transmit and receive FIFO interrupts, this block generates all the remaining interrupts that provide a means of monitoring the individual stages of a card session.

2.2.4 SCI control logic

This block controls the PrimeCell SCI using the timeouts and other control signals received from the transmit and receive logic block.

2.2.5 Transmit FIFO

The transmit FIFO is an 8-deep 8-bit wide circular buffer. Reads and writes to the transmit FIFO result in an access to buffer locations pointed to by a read and write pointer respectively. The transmit FIFO is used to buffer data that is subsequently sent to the card.

The Transmit Tide Mark Interrupt **SCITXTIDEINTR** is generated from within this block and is asserted when the level falls below the programmed value.

If there is an unsuccessful transmission when using the T0 protocol, the PrimeCell SCI asserts the SCI Transmit Error Interrupt (**SCITXERRINTR**) and sets bit [4], the SCI Transmit Error Interrupt Status (SCITXERRIS) bit, of the SCIIIR register. The PrimeCell SCI will stop transmitting and before any further characters can be transmitted, the error condition must be cleared by flushing the transmit FIFO.

2.2.6 Receive FIFO

The receive FIFO is an 8-deep 9-bit wide circular buffer. Reads and writes to the receive FIFO result in an access to buffer locations pointed to by a read and write pointer respectively. Data is stored in the least significant eight bits, and the ninth bit is set when a parity error is detected on the received data.

The **SCIRXTIDEINTR** interrupt is generated from within this block and is asserted when the level rises above the programmed value.

An SCI Receive Timeout Interrupt (**SCIRTOUTINTR**) is asserted and bit [9], the SCI Receive Timeout Status (SCIRTOUTIS), of the SCIIIR register is set under the following condition:

The receive FIFO contains at least one character, and no characters have been read for a time corresponding to the value programmed in the SCIRXTIME register.

2.2.7 Test registers and logic

There are registers and logic 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 SCI operation

The PrimeCell SCI provides a communication medium for the data transactions between a Smart Card and hardware/software that effectively form a Smart Card Reader.

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

- *Interface reset*
- *Clock signals*
- *Response to an ideal card session* on page 2-10
- *Warm reset sequence* on page 2-16
- *Response to a non-ideal card session* on page 2-16
- *Data transfer* on page 2-19
- *Character framing* on page 2-20
- *EMV character timing for T=0 (character protocol)* on page 2-22
- *Transmit* on page 2-23
- *Receive* on page 2-24
- *Block time and time between barriers* on page 2-24
- *Parity error* on page 2-24
- *RXREAD interrupt* on page 2-25.

2.3.1 Interface reset

The PrimeCell SCI is reset by the global reset signal **BnRES** and a block-specific reset signal **nSCIRST**. An external reset controller must use **BnRES** to assert **nSCIRST** asynchronously and negate it synchronously to **SCIREFCLK**. **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 SCI requires **BnRES** to be asserted for at least one period of **PCLK** and **nSCIRST** to be asserted for at least one period of **SCIREFCLK**.

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

2.3.2 Clock signals

The PrimeCell SCI has two input clock signals, **PCLK** and **SCIREFCLK**.

The **SCIREFCLK** frequency value should be selected in accordance with the chosen method of operation. For further details refer to

It is allowable to drive **SCIREFCLK** using the **PCLK** signal when the frequency value is suitable for the chosen method of operation, but separate clock signals may be required due to other system constraints.

However, there is a recommended constraint on the ratio of frequencies for **PCLK** and **SCIREFCLK**.

$\mathbf{F_{SCIREFCLK}} <= 2 \times \mathbf{F_{PCLK}}$

The frequency of **SCIREFCLK** must be less than twice the frequency of **PCLK**. If **SCIREFCLK** is more than twice the frequency of **PCLK** then transmission rates may not be fast enough to comply with the specification.

There is no restraint if the frequency of **SCIREFCLK** is less than the frequency of **PCLK**.

2.3.3 Response to an ideal card session

This section gives a brief overview of an ideal transaction sequence, and how it is monitored. Following this is a section that describes the typical errors that may occur during a non-ideal transaction and the ensuing actions.

Notification of errors is provided to the host by the PrimeCell SCI via a choice of:

- 12 direct interrupts
- a single ORed version, the SCI Interrupt signal **SCIINTR**, coupled with subsequent reading of the SCI Interrupt Identification Register status register SCIIIR.

Note

In the following descriptions, setting of any of the 12 individual inputs implies that the **SCIINTR** signal is also consequently set.

Stages of a card session

A card session comprises the following stages:

- *PrimeCell SCI reset and initial configuration* on page 2-11
- *Stages of a card session*
- *Contact activation and cold reset sequence* on page 2-13
- *Answer To Reset sequence* on page 2-13
- *Execution of a transaction* on page 2-14
- *Contact deactivation sequence and card removal* on page 2-14.

PrimeCell SCI reset and initial configuration

The PrimeCell SCI is reset by the asynchronous LOW application of **BnRES** and **nSCIRST**. These signals are then synchronously removed with reference to their respective clocks. The PrimeCell SCI is then configured by writing the initial values listed in Table 2-1 in preparation for a card being inserted into a Smart Card Reader and subsequent communication.

Note

The SCIDTIME register controls the timing of the card deactivation sequence and must be initialized before the activation sequence takes place. This is mandatory. Failure to carry out this action may result in damage to the card if it is removed prematurely.

Table 2-1 Initial configuration values

Register	Value	Comments
SCICR0	0x0	The control register 0 is set to direct convention, even parity, receive/transmit handshaking disabled.
SCICR1	0x0	Receive mode, timeouts initially disabled.
SCIIER	0xfff	Enable all interrupts.
SCISTABLE	0x64	For a 48MHz (21ns) reference clock, the stable (debounce) time is in terms of multiples of 1.38ms (0xffff x 21ns). An initial value of 136ms is proposed.
SCIATIME	0xafc8	The SCIATIME register must be programmed to between 40000 and 45000 (0xafc8) smart card clock cycles to satisfy the minimum cold and warm reset **nSCICARDRST** LOW time requirements.
SCIDTIME	0x2710	The SCIDTIME is in terms of reference clock periods. It times the three stages of the deactivation sequence. The total time of the deactivation sequence must not take longer than 1ms to complete. An initial value of 10000 (0x2710) periods is suggested which is equivalent to an SCIDTIME of approximately 0.21ms for a 48MHz (21ns) reference clock. This gives a total deactivation time of approximately 0.65ms.
SCIATRSTIME	0x9c40	The SCIATRSTIME is in terms of smart card clock cycles. After de-assertion of the reset **nSCICARDRST** signal, the start of the ATR sequence must occur within 40000 (0x9c40) smart card cycles.

Table 2-1 Initial configuration values (continued)

Register	Value	Comments
SCIATRDTIME	0x4b00	The SCIATRDTIME is in terms of *elementary time units* (etus). The complete ATR character sequence must be received within 19200 (0x4b00) etus.
SCICHTIME	0x2580	The SCICHTIME is in terms of etus and is the maximum interval between the leading edges of two consecutive characters. In the case of the ATR sequence this is 9600 (0x2580) etus. It is also applicable to characters in the transaction data stream and is T = 0, or T = 1 mode dependent.
SCIRXTIDE	0x0	This is the receive FIFO tide level. Although a value of zero is proposed, any value between 0x0 and 0x7 can be used. With a value of zero, the **SCIRXTIDEINTR** interrupt is generated when the initial TS character is loaded into the receive FIFO.
SCICLKICC	0x17	The SCICLKICC value is used to divide down the reference clock to provide the smart card clock. The final smart card frequency should be within the range 1-5MHz. For a 48MHz reference clock, SCICLKICC should be programmed with a value of 23 (0x17) to provide an initial 1MHz Smart Card clock frequency.
SCIBAUD	0x174	For a 48MHz reference clock, 1MHz smart card clock, and an SCI value of 10 (0xA), then SCIBAUD should be programmed to 178A (0x6f8)
SCIVALUE	0x10	An SCIVALUE value of 10 (0xa) is proposed as the sample value.

Smart card insertion and detection

A Smart Card is inserted into the Smart Card Reader.

The Smart Card Reader signals to the PrimeCell SCI that a card has been detected by setting the **SCIDETECT** signal high. In response to this, the PrimeCell SCI starts its *debounce* timer.

The Smart Card must remain in the interface for the debounce period, which is initially defined by the value written to the SCISTABLE register.

On expiry of the debounce period, the PrimeCell SCI notifies the host that a card has been successfully inserted by asserting the SCI Card In Interrupt signal **SCICARDININTR** and setting bit [0], the SCICARDINIS bit, of the SCIIIR status register.

At this point there are no clocks or power applied to the Smart Card and the input/output signals are held LOW by the interface. These signals are then applied in a controlled manner by the activation sequence.

Contact activation and cold reset sequence

The host now signals the PrimeCell SCI that it can activate the card, that is power up the inserted card in an ordered manner. The PrimeCell SCI activation sequence has been divided into three equal phases, timed by the value programmed in the SCI activation time register SCIATIME.

The PrimeCell SCI activation sequence includes the *cold* reset sequence which must be in effect for between 40000 and 45000 Smart Card clock cycles. This is programmed using the SCIATIME register.

The PrimeCell SCI performs the following activation sequence:

1. Assert **nSCICARDRST** low.
2. Wait for SCIATIME Smart Card clock cycles.
3. Enable VCC, configure **SMDATA** signal as high impedance.
4. Wait for SCIATIME Smart Card clock cycles.
5. Enable **SCICLKOUT** clock.
6. Wait for SCIATIME Smart Card clock cycles.
7. De-assert **nSCICARDRST** high.

The host initiates the activation sequence by writing a 1 to bit 0 (STARTUP) of the SCI control register 2 (SCICR2).

The PrimeCell SCI notifies the host that the activation sequence is complete by asserting the SCI card up interrupt signal **SCICARDUPINTR** and setting bit [2], SCICARDUPIS, within the SCIIIR.

The ATR on the input/output line from the Smart Card will begin between 400 and 40000 cycles from reset de-assertion.

Answer To Reset sequence

The ATR sequence contains information about the card requirements for subsequent data transactions. The first character within the ATR stream is called the TS character and contains the convention information (direct or inverse format) on how the remaining ATR and future data transaction characters will be interpreted.

On reception of the first character, the SCI RXTIDE interrupt **SCIRXTIDEINTR** signal is asserted and bit [10], the SCI RXTIDE interrupt status bit SCIRXTIDEIS is set within the SCIIIR status register.

The PrimeCell SCI should read this character, establish the respective required convention and then, if necessary, reprogram the SCI control register 0 (SCICR0). It is also recommended that the RXFIFO RXTIDE level be programmed to a higher value.

These processes should be completed prior to the start of reception of the next character.

In brief, the ATR sequence includes configuration values for:

- the clock frequency
- baud rate
- guard times
- protocol type.

The remainder of the ATR sequence is received, read via the RXFIFO in the selected convention, interpreted by the host software and the PrimeCell SCI programmed accordingly with the extracted values.

Execution of a transaction

After the interface has been configured by extracting the parameters from the ATR stream, host to card communication, and vice versa, can now proceed.

The direction of flow is controlled via the value written to bit [2], the MODE bit, in the SCI control register 1 (SCICR1).

The host is always in control of how many characters it sends to the card and how many characters it expects to be returned by the card.

The character streams must meet the timing requirements of either the T0 or T1 protocol. These are covered in TBD *EMV character timing for T=0 (character protocol)* on page 2-21 and TBD *EMV character timing for T=1 (block protocol)* on page 2-22.

Contact deactivation sequence and card removal

The final step in a typical card session is contact deactivation where the signals and power are removed in a defined sequence. Contact deactivation takes precedence over all other operations in order that the card is not electrically damaged.

The deactivation sequence can be initiated by software by writing a 1 to bit [1], the FINISH bit, of the SCICR2 register. Also, there are two hardware signals which can be used:

- On detection of a cards removal at any time in a session, signified by the **SMIDETECT** signal being low, then contact deactivation will be initiated. The **SMIDECTECT** signal usually originates from a Smart Card reader.
- In addition, it is possible to initiate contact deactivation by asserting the signal **SCIDEREQ** which may be fed from an alternative source other than a Smart Card reader.

The PrimeCell SCI deactivation sequence has been divided into three equal phases, timed by the value programmed in the SCI Deactivation Time (SCIDTIME) register and is in terms of reference clock **SCIREFCLK** periods.

The PrimeCell SCI deactivation sequence must complete within 1 millisecond. This means that the SCIDTIME register needs to be programmed with a third of the total time. For example, for an **SCIREFCLK** frequency of 48MHz (period 21ns) an SCIDTIME value of 10000 would equate to a total deactivation time of around 0.65ms.

The PrimeCell SCI performs the following deactivation sequence:

1. Assert **nSCICARDRST** low.
2. Wait for SCIATIME reference clock (**SCIREFCLK**) cycles.
3. Drive SCICLKOUT low.
4. Wait for SCIATIME reference clock (**SCIREFCLK**) cycles.
5. Drive nSCIDATAEN high.
6. Wait for SCIATIME reference clock (**SCIREFCLK**) cycles.
7. Drive VCC low.

On completion of the deactivation sequence, the PrimeCell SCI asserts the SCI Card Down Interrupt signal **SCICARDDNINTR** and sets bit [3], the SCI Card Down Interrupt Status bit SCICARDDNIS, of the SCIIIR status register.

The card may then be safely removed if desired, but it may remain and be re-activated for another transaction if so required by the host.

On recognition of the card being removed, that is, **SCIDETECT** transitioning from HIGH to LOW, the PrimeCell SCI asserts the SCI Card Out Interrupt **SCICARDOUTINTR** and sets bit [1], the SCI Card Out Interrupt Status bit (SCICARDOUTIS) of the SCIIIR register.

2.3.4 Warm reset sequence

This section describes an ideal card session However, if the ATR sequence is found to be in error, as described in *Response to an ideal card session*, then the PrimeCell SCI will initiate a *warm* reset sequence in an attempt to restart reception of the ATR stream:

1. Assert **nSCICARDRST** LOW.
2. Maintain VCC and clock stable
3. Put the PrimeCell SCI into reception mode
4. Wait for SCIATIME Smart Card clock cycles
5. De-assert **nSCICARDRST** HIGH.

The host initiates the activation sequence by writing a 1 to bit 0 (STARTUP) of the SCICR2 control register.

The ATR on the input/output line from the Smart Card will begin between 400 and 40000 cycles from reset de-assertion.

If the start bit of the ATR stream is not received within this time, then the PrimeCell SCI will automatically initiate the deactivation sequence without the need for software intervention.

2.3.5 Response to a non-ideal card session

The PrimeCell SCI has to resolve non-ideal transactions such as:

- removal of the card before a transaction has completed
- timing/parity errors that may occur during the data flow.

The PrimeCell SCI monitors each transaction stage via interrupt and status generation, allowing software to respond accordingly. For more detailed information on the individual and final shared interrupts, refer to Chapter 3 *Programmer's Model*.

Notification of errors is provided to the host by the PrimeCell SCI via a choice of either twelve direct interrupts or their single ORed version, the SCI Interrupt Signal **SCIINTR**, coupled with subsequent reading of the SCI Interrupt Identification Register SCIIIR.

Note

In the following descriptions, setting of any of the twelve individual inputs implies that the **SCIINTR** signal is also consequently set.

Card removed at any time between activation and deactivation

The PrimeCell SCI must ensure that no electrical damage is caused to the card if it is removed whilst still powered up, that is, it must immediately be deactivated in a defined sequence.

Card deactivation takes precedence over any other operation and can be initiated by software or hardware. The card must be powered down in less than 1 millisecond. See *Contact deactivation sequence and card removal* on page 2-14 for details of this sequence.

On recognition of the card being removed, that is, **SCIDETECT** transitioning from HIGH to LOW, the PrimeCell SCI asserts the SCI Card Out Interrupt **SCICARDOUTINTR** and sets bit [1], the SCI Card Out Interrupt Status bit (SCICARDOUTIS) of the SCIIIR register.

Card inserted, debounce time not met, then card removed

This is very similar to the above, though not as critical, as power will not been applied to the card. The card is only activated on successful completion of the debounce period. As above the card must immediately be deactivated using the controlled deactivation sequence.

On recognition of the card being removed, that is **SCIDETECT** transitioning from HIGH to LOW, the PrimeCell SCI asserts the SCI Card Out Interrupt **SCICARDOUTINTR** and sets bit [1], the SCI Card Out Interrupt Status bit (SCICARDOUTIS) of the SCIIIR register.

Card inserted, debounce time not met, no ATR start bit received within the specified time

The SCIATRSTIME register value is programmed with the value 40000. This represents the maximum number of Smart Card clock cycles after de-assertion of the **SCICARDRST** signal during which the start bit of the card's ATR sequence should be received.

If the ATR start bit is not received before this maximum number of Smart Card clock cycles has expired, then the PrimeCell SCI asserts the ATR Start Timeout Interrupt **SCIARTSTOUTINTR** and sets bit 5 of the SCIIIR register, the SCI Card Out Interrupt Status bit (SCIATRSTOTIS).

The PrimeCell SCI will automatically initiate the deactivation sequence without the need for software intervention.

ATR start bit received, but time between two successive characters exceeds the specified time.

If the ATR start bit is received within the specified time, then the time between leading edges of the ATR characters is checked to be less than the specified maximum limit.

This value is programmed into the SCI Character Time register (SCICHTIME).

During the ATR, the delay between the leading edges of any two consecutive characters from the card shall be a minimum of 12, but not more than 9600 etus. See Table 2-1 on page 2-10 for more details of the definition of the initial etu values for ATR reception.

If the SCICHTIME value is exceeded at any time during reception of the ATR data stream, then the SCI Character TimeOut Interrupt signal (**SCICHTOUTINTR**) is asserted and bit 8, the SCI Character TimeOut Interrupt status bit (SCICHTOUTIS), of the SCIIIR register is set.

The host software will respond to the interrupt and initiate a warm reset sequence by writing a 1 to the WRESET bit of the SCICR2 control register.

ATR start received, but the duration of the total ATR data stream exceeded the specified time

The card shall transmit all the characters to be returned during an ATR within 19200 etus. This time is measured between the leading edge of the start bit of the first character (TS) and 12 etus after the leading edge of the start bit of the last character.

The maximum value, which is currently fixed at 19200 by the *EMV Specification*, is programmed into the SCI ATR Duration Time register (SCIATRDTIME). If the SCIATRDTIME time is not met, then the SCI ATR Duration TimeOut Interrupt signal **SCIATRDTOUTINTR** is asserted and bit 6, the SCI ATR TimeOut Status bit (SCIATRDTOUTIS) of the SCIIIR register is set.

The host software will respond to the interrupt and initiate a warm reset sequence by writing a 1 to the WRESET bit of the SCICR2 control register.

ATR received, but parity errors are found within the received data

If, during the reception of the ATR stream, an error such as parity failure is recognized by the PrimeCell SCI or the associated software, then the PrimeCell SCI will initiate a warm reset by writing a 1 to the WRESET bit of the SCICR2 control register.

Data transaction in progress, time for block arrival exceeded

If the maximum delay from start leading edges of the last character from the PrimeCell SCI, that gave the right to send to the card, and the first character sent by the card exceeds a specified time, then the PrimeCell SCI will assert the SCI Block TimeOut Interrupt (**SCIBLKTOUTINTR**) and set bit 7, the SCI Block TimeOut Interrupt Status (SCIBLKTOUTIS) bit, of the SCIIIR register.

The block timeout value is programmed by writing to the SCIBLKTIME register.

2.3.6 Data transfer

Data rates

The duration of a bit within a character is termed the *elementary time unit* (etu). The etu is set by programming the SCIBAUD and SCIVALUE registers.

Value X BAUD rate clock

The value in the SCIBAUD register is used to define a clock which is a multiple of the baud rate. This is known as the Value X BAUD rate clock. The Value X BAUD rate clock is generated by dividing the reference clock by 1 + SCIBAUD. The SCIVALUE register defines the number of Value X BAUD rate clock periods which make up an etu.The etu is programmable and has different values dependent upon the stage of card processing.

During the ATR, the bit duration is known as the initial etu and is given by the following equation:

$$\text{Initial etu} = \frac{372}{f} \text{ seconds}$$

Figure 2-2 Initial etu equation

where f is the Smart Card clock frequency in Hertz.

Following the ATR (and establishment of the global parameters F and D), the bit duration is known as the current etu, and is given by the following equation:

$$\text{current etu} = \frac{F}{f} \times \frac{1}{D} \text{ seconds}$$

Figure 2-3 Current etu equation

where F and D are the clock rate conversion and bit rate adjustment parameters returned by the card, and f is the clock frequency applied to the Smart Card.

The etu is set by programming the SCIBAUD and SCIVALUE registers.

The SCIVALUE defines the number of baud rate clock periods that define the etu.

Therefore:

$$1\ \text{etu} = \frac{(1 + \text{SCIBAUD})}{(\text{Reference Clock})} \times \text{SCIVALUE}$$

Figure 2-4 1 etu

Thus the following equation must always be satisfied:

$$\frac{(1 + \text{SCIBAUD})}{(\text{Reference Clock})} \times \text{SCIVALUE} = \frac{F}{f} \times \frac{1}{D}$$

Figure 2-5 Satisfied equation

See ISO 7816-3 for the possible values of F and D that can be returned by the card.

Note

The EMV standard specifies that f must be in the range 1–5 MHz. ISO 7816-3 merely specifies a lower bound of 1 MHz.

2.3.7 Character framing

Figure 2-6 on page 2-21 shows the structure of a character. Each character consists of:

- a start bit
- eight data bits
- a parity bit
- a number of bits making up the *guard time* before the start bit of the next character.

The *guard time* is the delay between the trailing edge of the parity bit of a character, and the leading edge of the start bit of the next character. The guard time for characters transmitted by the interface is controlled by the SCICHGUARD register.

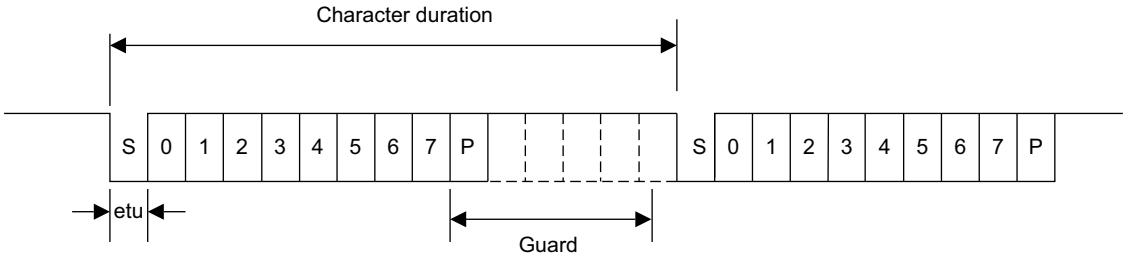


Figure 2-6 Character structure

Figure 2-7 shows how the interface interprets characters transmitted from the card. At time TS, the card pulls the bidirectional input/output line LOW to begin the start bit.

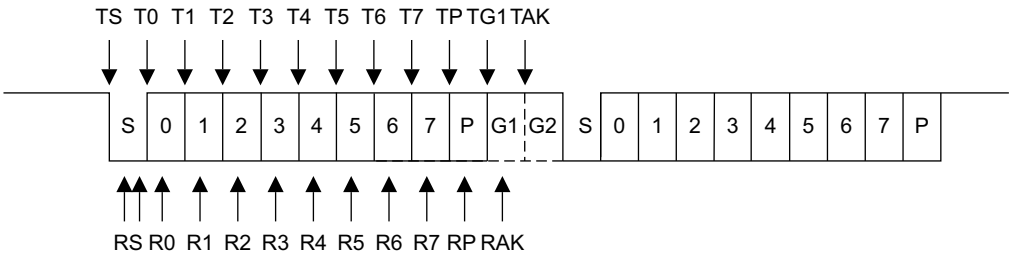


Figure 2-7 Character timing

The interface detects the leading edge of the start bit after four reference clock cycles. The input/output line is sampled at RS, approximately half an etu after the leading edge of the start bit. If the input/output line is not LOW, the start bit is deemed to be invalid and is ignored by the interface.

The first data bit is sampled at 1 etu after RS, and subsequent data bits (including the parity bit) are sampled at 1 etu intervals. The T0 protocol defines a mechanism whereby the receiver can request retransmission of a character by pulling the input/output line LOW during the guard time following the character. When a retransmission is requested, the interface starts to pull the input/output line LOW at RAK.

The transmitter (the card in this case) samples the input/output line at TAK. The interface holds the input/output line for a total of 2 etus. This is shown in Figure 2-8 on page 2-22.

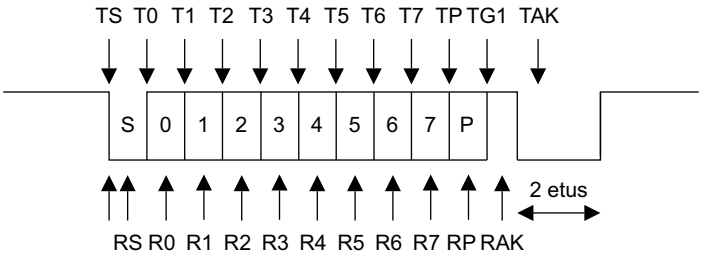


Figure 2-8 To transmit request

2.3.8 EMV character timing for T=0 (character protocol)

The minimum interval between the leading edges of the start bits of two consecutive characters sent by the interface to the Smart Card is between 12 and 266 etus as indicated by the value of TC1 returned within the ATR stream.

The minimum interval between the leading edges of the start bits of two consecutive characters sent by the Smart Card to the interface is 12 etus.

The maximum interval between the start leading edge of any character sent by the Smart Card and the start leading edge of the previous character sent either by the Smart Card or the interface (the *work waiting time*) will not exceed 960 x D x WI = 9600 etus. (The bit rate conversion factor, D, will have a default value of 1. WI will not have a default value of 10 as TC2, which would contain the value, is not returned in the ATR.)

The minimum interval between the leading edges of the start bits of two consecutive characters sent in opposite directions will be 16 etus.

Note

The minimum interval between the leading edges of the start bits of two characters sent by interface to the Smart Card is always governed by the value of TC1. It may be less than minimum interval of 16 etus allowed between two characters sent in opposite directions.

2.3.9 EMV character timing for T=1 (block protocol)

The minimum interval between the leading edges of the start bits of two consecutive characters, sent by the interface to the Smart Card will be between 11 and 266 etus as indicated by the value of TC1 returned within the ATR stream.

The minimum interval between the leading edges of the start bits of two consecutive characters sent by the Smart Card to the interface will be 11 etus.

The maximum interval between the leading edges of the start bits of to consecutive characters in the same block the *Character Waiting Time* (CWT) will not exceed (2*CWI + 11) etus. The *Character Waiting Integer* (CWI) will have a value of 0 to 5, resulting in the CWT having the range 12 to 43 etus.

The maximum interval between the leading edge of the start bit of the last character that gave the right to send to the Smart Card and the leading edge of the first character sent by the Smart Card the *Block Waiting Time* (BWT) will not exceed ((2*BWI x 960) + 11) etus. The *Block Waiting Integer* (BWI) will have a value in the range 0 to 4, resulting in the BWT having the range 971 to 15371 etus.

The minimum interval between the leading edges of the start bits of two consecutive characters sent in opposite directions the *Block Guard Time* (BGT) will be 22 etus.

2.3.10 Transmit

Characters that are to be sent to the card are first written into the SCIDATA FIFO and then automatically transmitted to the card at timed intervals. Direction of communication is controlled by the MODE bit of the SCICR1 register. Changing from transmit to receive does not take place until the last character stored in the transmit FIFO has been sent. This enables the MODE bit to be written immediately after writing the last character in a block. This is necessary because the card can respond to the transmission almost immediately (minimum turnaround time measured from start bit to start bit is 16 etus for T0 and 22 etus for T1).

If character-transmit handshaking is enabled (mandatory for T0), the input/output line is sampled at 1 etu after the parity bit. If the card indicates that it did not receive the character correctly, the character is retransmitted a maximum of TXRETRY times (set using the SCIRETRY register) before the transmission is aborted and a **SCITXERRINTR** interrupt generated. The interface waits for four etus after an error is detected before the character is retransmitted. If a character fails to be transmitted and a **SCITXERRINTR** interrupt is generated, the transmit/receive interlock mechanism must be reset by flushing the transmit FIFO before any subsequent transmit or receive operation.

The interval between successive characters sent by the interface is governed by the SCICHGUARD register, which defines the character guard time. SCICHGUARD is only used to control the transmission of characters and is not used by the receive hardware.

The minimum interval between the last character sent by the card and the next character sent by the interface is governed by the SCIBLKGUARD register, which defines the block guard time.

When the number of characters held in the transmit FIFO falls below the level defined in the SCITXTIDE register, a **SCITXTIDEINTR** interrupt is generated. The number of characters held in the transmit FIFO can be determined by reading the SCITXCOUNT register. Writing to the SCITXCOUNT register flushes the transmit FIFO.

2.3.11 Receive

Characters are read from the interface by reading the SCIDATA register. When read, SCIDATA has 9 bits:

- eight data bits
- one parity status bit.

See *SCIDATA: [9] (+ 0x00)* on page 3-6 for more information.

Before characters can be read from the interface, the MODE bit of the SCICR1 register must be set to 0. An interlock mechanism ensures that if the MODE bit is set to receive and there are still characters remaining in the transmit FIFO, these are transmitted before any characters are read from the card.

To switch from receive mode to transmit mode, write a 1 to the MODE bit within the SCICR1 control register during or after reception of the final byte within the incoming datastream.

2.3.12 Block time and time between barriers

Two registers define an upper limit on the time waited for a character to be transmitted by the card:

SCIBLKTIME defines the timeout limit for the first character in a block

SCICHTIME defines the maximum allowed time between characters (excluding the first character in a block).

The SCICHTIME counter is linked to the SCIBLKTIME counter and not start until the SCIMBLKTIME counter has stopped (That is, the first character in the block has arrived, or if the block timer has been disabled). The transmit/receive interlock mechanism prevents the SCICHTIME and SCIBLKTIME counters from running while characters are still present in the transmit FIFO.

2.3.13 Parity error

If character-transmit handshaking is enabled (RXNAK = 1) and the interface detects a parity error, it signals this to the card by pulling the input/output line down for 2 etus at 10.5 etus after the leading edge of the start bit.

The maximum number of times the interface attempts to receive a character is governed by the SCIRETRY register. If, after RXRETRY further attempts, the character has not be successfully received, a parity error for the character is flagged. When this character is read, bit 8 of SCIDATA is set to 1 to indicate that a parity error has occurred.

2.3.14 RXREAD interrupt

An RXREAD interrupt can be caused either:

- by the receive FIFO exceeding its tide mark (defined by SCITIDE)
- by a read access timeout (defined by SCIRXTIME).

The read access timeout limit can be reprogrammed dynamically, which provides a way of enabling the receive timeout mechanism only at the end of blocks. This is achieved by initially setting the receive timeout threshold to a high enough value to guarantee that an **RXREAD** interrupt can only have been caused by a tide mark condition (excluding an error condition where a card stops transmitting part way through a block). When the end of the block is reached, RXTIME is reprogrammed with the correct value to process the trailing characters in the block.

Chapter 3
Programmer's Model

This chapter describes the ARM PrimeCell Smart Card Interface (PL130) registers and provides details needed when programming the microcontroller. It contains the following:

3.1 About the programmer's model

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

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

- locations at offsets 0x70 through 0x7c are reserved for possible future extensions
- locations at offsets 0x80 through 0xff are reserved for test purposes.

3.2 Summary of PrimeCell SCI registers

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

Table 3-1 PrimeCell SCI register summary

Address	Type	Width	Reset value	Name	Description
SCI Base + 0x00	Read Write	9	0x--	SCIDATA SCIDATA	Data register.
SCI Base + 0x04	Read Write	6	0x00	SCICR0 SCICR0	Control register 0.
SCI Base + 0x08	Read Write	6	0x00	SCICR1 SCICR1	Control register 1.
SCI Base + 0x0c	Write	3	0x0	SCICR2	Control register 2.
SCI Base + 0x10	Read Write	12	0x000	SCIIER SCIIER	Interrupt enable register.
SCI Base + 0x14	Read Write	6	0x00	SCIRETRY SCIRETRY	Retry limit register.
SCI Base + 0x18	Read Write	8	0x00	SCITIDE SCITIDE	FIFO tide mark register.
SCI Base + 0x1c	Read Write	5/0	0x00	SCITXCOUNT SCITXCOUNTCLR	Transmit FIFO count register. Transmit FIFO count clear register.
SCI Base + 0x20	Read Write	5/0	0x00	SCIRXCOUNT SCIRXCOUNTCLR	Receive FIFO count register. Receive FIFO count clear register.
SCI Base + 0x24	Read	4	0xa	SCIFR	Flag register.
SCI Base + 0x28	Read Write	16	0x0000	SCIRXTIME SCIRXTIME	Receive read time-out register.
SCI Base + 0x2c	Read Write	11	0x000	SCIISTAT SCIISTAT	Smart Card status register.
SCI Base + 0x30	Read Write	16	0x0000	SCISTABLE SCISTABLE	De-bounce timer.

Table 3-1 PrimeCell SCI register summary (continued)

Address	Type	Width	Reset value	Name	Description
SCI Base + 0x34	Read Write	16	0x0000	SCIATIME SCIATIME	Activation event time.
SCI Base + 0x38	Read Write	16	0x0000	SCIDTIME SCIDTIME	Deactivation event time.
SCI Base + 0x3c	Read Write	16	0x0000	SCIATRSTIME SCIATRSTIME	Time to start of ATR reception.
SCI Base + 0x40	Read Write	16	0x0000	SCIATRDTIME SCIATRDTIME	Maximum duration of the ATD character stream.
SCI Base + 0x44	Read Write	16	0x0000	SCIBLKTIME SCIBLKTIME	Receive time-out between blocks.
SCI Base + 0x48	Read Write	16	0x0000	SCICHTIME SCICHTIME	Character to character time-out.
SCI Base + 0x4c	Read Write	8	0x00	SCICLKICC SCICLKICC	External Smart Card clock frequency.
SCI Base + 0x50	Read Write	16	0x0000	SCIBAUD SCIBAUD	Baud rate clock.
SCI Base + 0x54	Read Write	8	0x00	SCIVALUE SCIVALUE	SCIBAUD cycles.
SCI Base + 0x58	Read Write	8	0x00	SCICHGUARD SCICHGUARD	Character to character extra guard time.
SCI Base + 0x5c	Read Write	8	0x00	SCIBLKGUARD SCIBLKGUARD	Block guard time.
SCI Base + 0x60	Read Write	2	0x0	SCISYNCCR SCISYNCCR	Asynchronous/synchronous multiplexing control.
SCI Base + 0x64	Read Write	2	0x0	SCISYNCDATA SCISYNCDATA	Synchronous Smart Card data.
SCI Base + 0x68	Read	2	0x0	SCIRAWSTAT	Raw input/output and clock status.

Table 3-1 PrimeCell SCI register summary (continued)

Address	Type	Width	Reset value	Name	Description
SCI Base + 0x6c	Read Write	12	0x00a	SCIIIR SCIICR	Interrupt identification register/interrupt clear register.
SCI Base + 0x70 to 0x7c	-	-	-	-	Reserved.
SCI Base + 0x80 to 0xff	-	-	-	-	Reserved for test purposes.

3.3 Register descriptions

The following registers are described in this section:

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

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

3.3.1 SCIDATA: [9] (+ 0x00)

SCIDATA is the data register and is used for both transmitting and receiving characters. The write data is transmitted through nSCIDATAOUTEN. The read data is received through SCIDATAIN. Table 3-2 shows bit assignments for SCIDATA.

Note

Software should write to this register only after setting the MODE bit to 1. Writes to this register with the MODE but set to 0 will be ignored by the hardware.

Table 3-2 SCIDATA register read/write bits

Bits	Name	Type	Function
8	PARITY	Read	Parity error flag. Set to 1 if a parity error was detected when receiving the character corresponding to bits 7 to 0.
7:0	DATA	Read/Write	Eight data bits. These correspond to the character being read or written.

3.3.2 SCICR0: [6] (+ 0x04)

SCICR0 is control register 0. It configures the convention for interpreting characters, controls parity convention and enables the handshake mechanism to indicate that a parity error has occurred. The initial character returned by the card in the ATR sequence determines the convention.

There are two conventions:

Inverse A LOW state on the input/output line is interpreted as logic one and the *Most Significant Bit* (MSB) of the data byte is the first bit after the start bit.

Direct A LOW state on the input/output line is interpreted as logic zero and the *Least Significant Bit* (LSB) of the data byte is the first bit after the start bit.

Separate bits are used to control the logic sense and the bit ordering. This allows for nonstandard conventions to be configured.

The register bits (0 to 1) should be set to 00 before reading the initial (TS) character from the *Answer To Reset* (ATR) stream. The TS character determines the convention that the remainder of the ATR stream has been encoded with.

Inverse convention is configured by writing 11 to SCICR0[1:0] after reading the TS character and before reading any subsequent characters in the ATR.

The register bits 2 and 4 control the parity convention used (odd or even parity) and bits 3 and 5 are used to enable the handshaking mechanism. The handshaking mechanism is initiated by the receiver pulling down the input/output line whenever a parity error has occurred, and is ended by a character retry. Separate controls exist for the transmit and receive paths. The maximum number of attempts made to either transmit or receive a character is specified in the SCIRETRY register.

Character retry is not applied during ATR reception, hence this handshake should be programmed initially with the value 0x0.

Bits 2 to 5 should be set to 0xa if the T=0 protocol is requested by the contents of the initial (TS) character of the ATR stream. Table 3-3 shows bit assignments for SCICRO.

Table 3-3 SCICR0 register read/write bits

Bits	Name	Type	Function
5	RXNAK	Read/write	Enables character receipt handshaking. If TXNAK = 0, the SCI pulls the input/output line LOW if it detects a parity error. If TXNAK = 1, the SCI does not pull the input/output line LOW if it detects a parity error.
4	RXPARITY	Read/write	Receive parity setting. 0 = Even parity. 1 = Odd parity.
3	TXNAK	Read/write	Enables character transmission handshaking. If TXNAK = 0, the SCI does not check to see if the receiver has pulled the input/output line LOW to indicate a parity error. If TXNAK = 1, the SCI checks, after each character has been transmitted, to see if the receiver has pulled the input/output line LOW to indicate a parity error.

Table 3-3 SCICR0 register read/write bits (continued)

Bits	Name	Type	Function
2	TXPARITY	Read/write	Transmit parity setting. 0 = Even parity. 1 = Odd parity.
1	ORDER	Read/write	Specifies ordering of the data bits. 0 = LOW interpreted as logic 0, lsb is the first bit after the start bit (direct convention). 1 = LOW interpreted as logic 1, msb is the first bit after the start bit (inverse convention).
0	SENSE	Read/write	Inverts sense of input/output line for data and parity bits. 0 = Direct convention. 1 = Inverse convention.

3.3.3 SCICR1: [6] (+0x08)

SCICR1 is control register 1. Table 3-4 shows bit assignments for SCICR1.

Table 3-4 SCICR1 register read/write bits

Bits	Name	Type	Function
5	EXDBNCE	Read/write	External debounce. 0 = Use the whole of the internal debounce timer. 1 = Bypass the non programmable section of the internal debounce timer.
4	BGTEN	Read/write	Block guard timer enable. 0 = Disable. 1 = Enable.
3	CLKZ1	Read/write	SCICLK output configuration. 0 = SCICLK configured as buffer output. 1 = SCICLK configured as pulled down (open drain).

Table 3-4 SCICR1 register read/write bits (continued)

Bits	Name	Type	Function
2	MODE	Read/write	Interface direction of communication control. 0 = Receive. 1 = Transmit.
1	BLKEN	Read/write	Block timeout enable. 0 = Disable. 1 = Enable.
0	ATRDEN	Read/write	ATR duration timeout enable. 0 = Disable. 1 = Enable.

CLKZ1 Used to configure the **SCICLK** output pad. If an external pull-up resistor is connected to the Smart Card clock signal, as is the case for synchronous card systems (where both the terminal and the card can pull the clock line LOW), the **SCICLK** output should be configured as pull-down only.

EXDBNCE Used to bypass the non programmable portion of the debounce timer, allowing a zero-debounce time by setting the programmable portion of the timer to 0.

3.3.4 SCICR2: [3] (+0x0c)

SCICR2 is control register 2 and is a write-only register used to initiate activation, deactivation and *warm* reset events. If a write occurs during the deactivation sequence, it is ignored. At any other time, a 1 written to the FINISH bit immediately starts the deactivation sequence. Table 3-5 on page 3-11 shows bit assignments for SCICR2.

Note

Writes to this register should be performed only during the appropriate phase of the card session. Writes performed in a card phase are not internally latched and stored for use in subsequent phases.

The software can write to the STARTUP bit only after a valid card has been found to be present. Writes when the card is not present will be ignored by the hardware.

The software can write to the WRESET bit only after the activation sequence is over.

The software can write to the FINISH bit only after a valid card has been found to be present in the system.

Table 3-5 SCICR2 register write bits

Bits	Name	Type	Function
2	WRESET	Write	Writing a 1 to this bit initiates a *warm* reset.
1	FINISH	Write	Writing a 1 to this bit deactivates the card.
0	STARTUP	Write	Writing a 1 to this bit starts the activation of the card.

3.3.5 SCIIER: [12] (+0x10)

SCIIER is the interrupt enable register and contains twelve bits that are used to enable the twelve interrupts. Table 3-6 shows bit assignments for SCIIER.

Note

The interrupt status bits visible in the SCIIIR register are free from masking. This is to allow polled mode of operation in systems which do not use an interrupt controller.

Software can read the interrupt status bits through the SCIIIR even if their corresponding mask bits are set.

Clearing the mask bits will not clear the interrupt status. However the pin-level interrupts will be cleared. The status bits are ANDed with the mask bits to create the pin-level interrupts.

Table 3-6 SCIIER register read/write bits

Bits	Name	Type	Function
11	SCITXTIDEIE	Read/write	Interrupt enable for SCITXTIDEINTR.
10	SCIRXTIDEIE	Read/write	Interrupt enable for SCIRXTIDEINTR.
9	SCIRTOUTIE	Read/write	Interrupt enable for SCIRTOUTINTR.
8	SCICHTOUTIE	Read/write	Interrupt enable for SCICHTOUTINTR.
7	SCIBLKTOUTIE	Read/write	Interrupt enable for SCIBLKTOUTINTR.
6	SCIATRDTOUTIE	Read/write	Interrupt enable for SCIATRDTOUTINTR.

Table 3-6 SCIIER register read/write bits (continued)

Bits	Name	Type	Function
5	SCIATRSTOUTIE	Read/write	Interrupt enable for SCIATRSTOUTINTR
4	SCITXERRIE	Read/write	Interrupt enable for SCITXERRINTR.
3	SCICARDDNIE	Read/write	Interrupt enable for SCICARDDNINTR.
2	SCICARDUPIE	Read/write	Interrupt enable for SCICARDUPINTR.
1	SCICARDOUTIE	Read/write	Interrupt enable for SCICARDOUTINTR.
0	SCICARDINIE	Read/write	Interrupt enable for SCICARDININTR.

3.3.6 SCIRETRY: [6] (+0x14)

SCIRETRY is the retry limit register and is used to configure the number of transmit and receive retries that are allowed.

If character transmit handshaking is enabled (TXNAK = 1), the TXRETRY field of this register contains a 3-bit value (0 to 2). This specifies the maximum number of attempts that can be made to retransmit a character that has been incorrectly received by the card, before aborting the transmission and generating a TXERR interrupt.

For normal T0 operation, the TXRETRY field should be set to 011 for a maximum of three retries. A value of 000 (no retries) causes a TXERR interrupt to occur as soon as an error is detected.

For T1 operation, character-based handshaking should be turned off (TXNAK = 0), in which case TXRETRY is not used.

The RXRETRY field of this register contains a 3-bit value (3 to 5). This specifies the maximum number of times the interface will request retransmission of a character after detection of a parity error.

For normal T0 operation, the RXRETRY field should be set to 011 for a maximum of three retries. A value of 000 (no retries) writes the received character to the receive FIFO with no request for retransmission in the event of a parity error. This has the same effect as setting RXNAK to 0.

If the character has not been successfully received after RXRETRY attempts, the parity error flag for that character is set. This appears as bit 8 of the SCIDATA register when it is read. Table 3-7 shows bit assignments for SCIRETRY.

Table 3-7 SCIRETRY register read/write bits

Bits	Name	Type	Function
5:3	RXRETRY	Read/write	Specifies the maximum number of retries to receive when a parity error has occurred in reception.
2:0	TXRETRY	Read/write	Specifies the maximum number of times that a character will be retransmitted following the detection of a parity error.

3.3.7 SCITIDE: [8] (+0x18)

SCITIDE is the FIFO tide mark register and is used to set the trigger points for the **TXTIDE** and **RXTIDE** interrupts.

The RXTIDE field of this register (7:4) contains a trigger point for the receive FIFO. When the number of characters in the receive FIFO exceeds RXTIDE, an **RXTIDE** interrupt is generated. Setting RXTIDE to 0 causes an interrupt as soon as the receive FIFO is non-empty. A value of 8 or more prevents any interrupt from occurring. An **RXTIDE** interrupt can only occur when the MODE bit of the SCICR1 register is set to 0 (receive).

The TXTIDE field of this register (3:0) contains the trigger point for the **TXTIDE** interrupt. When the number of characters in the transmit FIFO falls below this threshold, a **TXTIDE** interrupt is generated. Setting these bits to 0 prevents **TXTIDE** interrupts from being generated. Only values between 0 and 8 (inclusive) are valid.

Note

Writes to the TXFIFO register are allowed only if the MODE bit is set for transmission. The **TXTIDE** interrupt, however, is not qualified with the MODE bit. This allows the software to be notified of the current fill level of the transmit FIFO even after the data direction has shifted from transmit to receive. However, the actual act of filling the transmit FIFO should be done based on higher level software considerations and not purely on the fact that the transmit FIFO has room for more data. By not qualifying the **TXTIDE** interrupt with the MODE bit, system performance may be increased since the interrupt is being raised as early as possible.

A **TXTIDE** interrupt can only occur if the MODE bit of the SCICR1 register is set to 1 (transmit).

A character is not removed from the transmit FIFO until it has been successfully transmitted. Table 3-8 shows bit assignments for SCITIDE.

Table 3-8 SCITIDE register read/write bits

Bits	Name	Type	Function
7:4	RXTIDE	Read/write	Trigger point for **SCIRXTIDEINTR**.
3:0	TXTIDE	Read/write	Trigger point for **SCITXTIDEINTR**.

3.3.8 SCITXCOUNT/SCITXCOUNTCLR: [5/0] (+0x1c)

SCITXCOUNT/SCITXCOUNTCLR is the transmit FIFO count/transmit FIFO count clear register. It returns the number of characters (including any character currently being transmitted) in the transmit FIFO when read, and flushes the transmit FIFO when written (with any value).

If there is an unsuccessful transmission when using the T0 protocol, the interface generates a **TXERR** interrupt and stops transmitting. Before any further characters can be transmitted or received, the error condition must be cleared by flushing the transmit FIFO. Table 3-9 shows bit assignments for SCITXCOUNT/SCITXCOUNTCLR.

Table 3-9 SCITXCOUNT/SCITXCOUNTCLR register read/write bits

Bits	Name	Type	Function
4:0	TXCOUNT	Read	Transmit FIFO count.
0	TXCOUNTCLR	Write	Transmit FIFO count clear.

3.3.9 SCIRXCOUNT/SCIRXCOUNTCLR: [5/0] (+0x20)

SCIRXCOUNT/SCIRXCOUNTCLR is the receive FIFO count/receive FIFO count clear register. It returns the number of characters in the receive FIFO when read, and flushes the receive FIFO when written (with any value). Table 3-10 shows bit assignments for SCIRXCOUNT/SCIRXCOUNTCLR.

Table 3-10 SCIRXCOUNT/SCIRXCOUNTCLR register read/write bits

Bits	Name	Type	Function
4:0	RXCOUNT	Read	Receive FIFO count.
0	RXCOUNTCLR	Write	Receive FIFO count clear.

3.3.10 SCIFR: [4] (+0x24)

SCIFR is the flag register and is a read-only register which contains four bits that indicate the status of the transmit and receive FIFOs. Table 3-11 shows bit assignments for SCIFR.

Table 3-11 SCIFR register read bits

Bits	Name	Type	Function
3	RXFE	Read	RXFIFO empty status.
2	RXFF	Read	RXFIFO full status.
1	TXFE	Read	TXFIFO empty status.
0	TXFF	Read	TXFIFO full status.

3.3.11 SCIRXTIME: [16] (+0x28)

SCIRXTIME is the receive read timeout register and stores the receive read timeout value.

An **SCIRTOUT** interrupt is triggered if the receive FIFO contains at least one character, and no characters have been read for a time corresponding to SCIRXTIME Smart Card clock cycles.

Writing a value of zero is illegal. Table 3-12 shows bit assignments for SCIRXTIME.

Table 3-12 SCIRXTIME register read/write bits

Bits	Name	Type	Function
15:0	RXTIME	Read/write	Receive read timeout value.

Table 3-13 lists the range and resolution of the timeout value for various Smart Card clock frequencies.

Table 3-13 RXTIME ranges and resolutions

Frequency	Range (approx)	Resolution
10 MHz	0-6 ms	0.1 µs
5 MHz	0-13 ms	0.2 µs
1 MHz	0-65 ms	1 µs
500 kHz	0-131 ms	2 µs

3.3.12 SCIISTAT: [10] (+0x2c)

SCIISTAT is the ICC status register and provides direct access to Smart Card signals. It is only required if a non-EMV-compliant configuration is used. The status register is updated automatically during activation, deactivation and *warm* reset events. Table 3-14 on page 3-17 shows bit assignments for SCIISTAT.

Note

The SCI does not have a separate bit to distinguish between EMV and non-EMV compliant cards. It is the responsibility of the software to follow certain sequences in either case to ensure correct and consistent behavior. The software should not write to the STARTUP bit in case it is dealing with a non-EMV compliant card. It should set the STARTUP bit in the case of EMV compliant cards.

If the STARTUP bit has been written to – indicating an EMV compliant card – the software should not write to the Smart Card Status register.

If the STARTUP bit has not been written to – indicating a non-EMV compliant card – the activation sequence should be performed by explicit writes to the relevant bits in this register.

Deactivation is done ONLY through internal hardware in both EMV and non-EMV configurations.

Software should not write to this register during card validation via the hardware debounce mechanism.

Writes to this register are ignored during deactivation.

Table 3-14 SCIISTAT register read/write bits

Bits	Name	Type	Function
9	CARDPRESENT	Read	1 if Smart Card is present.
8	nSCIDATAEN	Read	Tristate control for external off-chip buffer for data.
7	nSCIDATAOUTEN	Read	Tristate output buffer control for data.
6	SCICLKOUT	Read	Smart Card clock output.
5	nSCICLKEN	Read	Tristate control for external off-chip buffer for clock.
4	nSCICLKOUTEN	Read	Tristate output buffer control for clock.
3	DATAEN	Read/write	Enable Smart Card data. 0 forces the Smart Card data LOW.
2	CLKEN	Read/write	Enable Smart Card clock. 0 forces the Smart Card clock LOW.
1	CRESET	Read/write	Controls Smart Card reset signal.
0	POWER	Read/write	Controls Smart Card VCC.

3.3.13 SCISTABLE: [16] (+0x30)

SCISTABLE is the debounce timer register and determines how long the **SCIDETECT** signal must hold a stable HIGH value, before the interface registers the insertion of a card. This is notified by setting the **CARDININTR** interrupt. Table 3-15 shows bit assignments for SCISTABLE.

Table 3-15 SCISTABLE register read/write bits

Bits	Name	Type	Function
15:0	STABLE	Read/write	Stores the debounce time.

The debounce timer is constructed as a 16-bit counter feeding a programmable 8-bit counter, the former being loaded with 0xffff and the latter with the value contained in the SCISTABLE register. When enabled, the 16-bit counter decrements the 8-bit counter value until it reaches zero.

The logic is configured to provide a debounce time of (SCISTABLE + 1) multiples of the 16-bit full count.

For a 48MHz reference clock, this gives a programmable debounce time in the range 1.38ms to 350.28ms in 1.38ms steps.

The 16-bit counter can be bypassed for test purposes by setting the EXDBNCE bit HIGH. The reference clock then feeds the 8-bit SCISTABLE counter, with the interrupt status and signals being set when this counter reaches zero.

If the debounce period is satisfied, an **SCICARDININTR** interrupt is set.

Note

After completion of the card deactivation sequence, the **SCICARDININTR** will be reset.

A falling edge on **SCIDETECT** and a rising edge on **SCIDEACREQ** immediately resets **SCICARDININTR.**

3.3.14 SCIATIME: [16] (+0x34)

SCIATIME is the activation event time register and defines the duration, in Smart Card clock cycles, of the time taken for each of the three stages of the card activation process. The programmed value must satisfy the minimum **nSCICARDRST** LOW time of 40000 cycles, and should be sufficient time for the interface power to stabilize. Table 3-16 shows bit assignments for SCIATIME.

Table 3-16 SCIATIME register read/write bits

Bits	Name	Type	Function
15:0	ATIME	Read/write	Stores the total time for the three stages of the card activation process time.

3.3.15 SCIDTIME: [16] (+0x38)

SCIDTIME is the deactivation event time register and defines the duration, in reference clock cycles, of the time taken for each of the three stages of the card deactivation process. The deactivation sequence can be initiated by software, by writing a 1 to the FINISH bit of the SCICR2, or by hardware, on detection of card removal, by the SCIDETECT signal going LOW, or on assertion of SCIDEREQ from the PMU. Table 3-17 shows bit assignments for SCIDTIME.

Table 3-17 SCIDTIME register read/write bits

Bits	Name	Type	Function
15:0	DTIME	Read/write	Stores the total time for the three stages of the card deactivation process time.

3.3.16 SCIATRSTIME: [16] (+0x3c)

SCIATRSTIME is the time to start of ATR reception register. It defines the receive timeout threshold from the deassertion of **SCICARDRST** to the start of the first character in the ATR. This is specified in Smart Card clock cycles and its initial value is 40000 Smart card clock cycles.

The time in seconds, from the deassertion of **SCICARDRST** to the start of the first character in the ATR, will vary with Smart Card clock frequency.

On timeout, this timer will set an interrupt, **SCIATRSTOUTINTR**. Table 3-18 shows bit assignments for SCIATRSTIME.

Table 3-18 SCIATRSTIME register read/write bits

Bits	Name	Type	Function
15:0	ATRSTIME	Read/write	ATR reception start timeout threshold from the deassertion of nSCICARDRST.

3.3.17 SCIATRDTIME: [16] (+0x40)

SCIATRDTIME is the maximum duration of the ATR character stream register. It defines the receive timeout threshold for the duration of the ATR sequence, from the start bit of the first ATR character until the end of the ATR block. This is specified in *elementary time units* (etus) and should be programmed to 19200 etus, as per the current specifications. On timeout, this timer will set an **SCIATRDTOUTINTR** interrupt.

In normal operation this interrupt will be disabled, after the full ATR block has been received, by setting the ATRDEN bit in SCICR1 to 0 using a software routine.
Table 3-19 shows bit assignments for SCIATRSTIME.

Table 3-19 SCIATRSTIME register read/write bits

Bits	Name	Type	Function
15:0	ATRDTIME	Read/write	ATR reception duration timeout threshold from the start bit of the first ATR character.

3.3.18 SCIBLKTIME: [16] (+0x44)

SCIBLKTIME is the receive timeout between blocks register. It is used to configure the maximum delay from the leading edge of the last character that gave the right to send to the card, and the first character to be sent by the card. SCIBLKTIME applies to both T0 and T1 protocols:

- For T0, the SCIBLKTIME is effectively offset by 12 etus internally, so the value to be programmed into the SCIBLKTIME register should be required timeout period in etus minus 12.

- For T1, the SCIBLKTIME is effectively offset by 11 etus internally, so the value to be programmed into the SCIBLKTIME register should be required timeout period in etus minus 11.

Failure to meet the SCIBLKTIME value results in the setting of the internal SCIBLKTOUTINTR interrupt bit in the SCIIR register, and the external **SCIBLKTOUTINTR** interrupt signal. Table 3-20 shows bit assignments for SCIBLKTIME.

Note

The SCIBLKTIME parameter is not applicable to the ATR reception. the reception of the first character of the ATR stream must not exceed the value programmed in the SCIATRSTIME register, which is in terms of Smart Card clock cycles. The SCIATRSTIME value is defined as 40000 Smart Card clock cycles.

Table 3-20 SCIBLKTIME register read/write bits

Bits	Name	Type	Function
15:0	BLKTIME	Read/write	Defines the time for block timeout.

3.3.19 SCICHTIME: [16] (+0x48)

SCICHTIME is the character to character timeout register. It defines the maximum time in etus between the leading edge of two consecutive characters for both T0 (character) and T1 (block) protocols. It is also in force during reception of the ATR.

For T0, the time between characters is termed the Work Waiting Time.

For T1, the time between characters is termed the Character Waiting Time (CWT), and the respective characters must reside in the same block.

Note

For T0, the SCICHTIME is effectively offset by 12 etus internally, so the value to be programmed into the SCICHTIME register should be required timeout period in etus minus 12.

For T1, the SCICHTIME is effectively offset by 11 etus internally, so the value to be programmed into the SCICHTIME register should be required timeout period in etus minus 11.

Failure to meet the SCICHTIME results in setting the interrupt **SCICHTOUTINTR**. Table 3-21 shows bit assignments for SCICHTIME.

Table 3-21 SCICHTIME register read/write bits

Bits	Name	Type	Function
15:0	SCICHTIME	Read/write	Defines the time for character timeout.

3.3.20 SCICLKICC: [8] (+0x4c)

SCICLKICC is the external Smart Card clock frequency register and contains the divisor used to generate the Smart Card clock frequency.

The Smart Card clock frequency (F) is generated by dividing the reference clock by (SCICLKICC + 1) and then dividing again by 2.

$$F = \frac{(\text{refclock})}{(\text{SCICLKICC} + 1)} \times 2$$

Figure 3-1 Smart card frequency

If SCICLKICC is set to 0, F = $^{\text{refclock}}/_{2}$.

If SCICLKICC is set to 1, F = $^{\text{refclock}}/_{4}$.

SCICLKICC is an 8-bit register that must be programmed with a value between 0 and 255 before **SCICLK** is enabled. Table 3-22 shows bit assignments for SCICLKICC.

Table 3-22 SCICLKICC register read/write bits

Bits	Name	Type	Function
7:0	CLKICC	Read/write	Defines the Smart Card clock frequency.

3.3.21 SCIBAUD: [16] (+0x50)

SCIBAUD is the baud rate clock register. It defines the divide value used to generate a Value X BAUD rate clock from the reference clock, where Value is set in the SCIVALUE register. SCIBAUD is a 16-bit register that must be programmed with a value between 0x1 and 0xffff.

The frequency of the Value X BAUD rate clock is equal to the frequency of the reference clock divided by (SCIBAUD + 1). Table 3-23 shows bit assignments for SCIBAUD.

Table 3-23 SCIBAUD register read/write bits

Bits	Name	Type	Function
7:0	BAUD	Read/write	The divide value used to define the baud rate clock frequency.

3.3.22 SCIVALUE: [8] (+0x54)

SCIVALUE is the baud cycles register and defines the number of SCIBAUD cycles per etu. This register is 8-bits wide and may be programmed with any value between 5 and 255. Table 3-24 shows bit assignments for SCIVALUE.

Table 3-24 SCIVALUE register read/write bits

Bits	Name	Type	Function
7:0	Value	Read/write	Defines the number of SCIBAUD cycles per etu.

3.3.23 SCICHGUARD: [8] (+0x58)

SCICHGUARD is the character to character extra guard time register. It defines the extra guard time that will be added to the minimum duration between leading edges of the start bits of two consecutive characters, for subsequent communication from the interface to the Smart Card.

The SCICHGUARD value is derived from the TC1 value that is extracted from the ATR character stream. Table 3-25 shows bit assignments for SCICHGUARD.

Table 3-25 SCICHGUARD register read/write bits

Bits	Name	Type	Function
7:0	SCICHGUARD	Read/write	Defines the minimum duration between the leading edges of the start bits of two consecutive characters for subsequent communication from the interface to the Smart Card.

The TC1 value can be between 0 and 255. The software must read the TC1 value and program the SCICHGUARD register as described in Table 3-26 to provide the resultant guard time in etus.

Table 3-26 SCIVALUE register read/write bits

ATR TC1 value	SCICHGUARD value		Resultant guard time (etus)	
	T0	T1	T0	T1
0 =<TC1<255	TC1	TC1 + 1	TC1 + 12	TC1 + 11
255	0	0	12	11

A TC1 value of 255 indicates that the minimum delay between the start leading edges of two consecutive characters will be 12 etus if T=0, or 11 etus if T=1 is to be used.

Note

The TXNAK bit value in the SCICR0 register is used by the interface hardware to determine whether T0 or T1 protocol is in operation.

3.3.24 SCIBLKGUARD: [8] (+0x5c)

SCIBLKGUARD is the block guard time register. Table 3-27 on page 3-25 shows bit assignments for SCIBLKGUARD. It defines the minimum time in etus between the leading edges of two consecutive characters sent in opposite directions.

For T0, the SCIBLKGUARD minimum time shall be 16 work etus.

For T1, the SCIBLKGUARD minimum time shall be 22 work etus.

Note

For T0, SCIBLKGUARD is effectively offset by 12 etus internally, so the value to be programmed into the SCIBLKGUARD register should be required time period in etus minus 12.

For T1, the SCIBLKGUARD is effectively offset by 11 etus internally, so the value to be programmed into the SCIBLKGUARD register should be required time period in etus minus 11.

Table 3-27 SCIBLKGUARD register read/write bits

Bits	Name	Type	Function
7:0	BLOCKGUARD	Read/write	Defines the minimum time in etus between the leading edges of two consecutive characters sent in opposite directions.

3.3.25 SCISYNCCR: [2] (+0x60)

SCISYNCCR is the asynchronous/synchronous multiplexing register and is used to select the sources of the Smart Card clock and the input/output lines.

The Smart Card can operate in either asynchronous mode, where the clock is derived from the reference clock and data is driven directly by the interface or the card, or synchronous mode, where the system processor provides the clock and data by writing appropriate values to the SCISYNCDATA register. Table 3-28 shows bit assignments for SCISYNCCR.

Table 3-28 SCISYNCCR register read/write bits

Bits	Name	Type	Function
1	SELCLK	Read/write	Selects the source of the Smart Card clock. 0 = The Smart Card clock is derived from the reference clock. 1 = SCISYNCDATA register bit 1 (**WCLK**) drives the SCICLK Smart Card clock.
0	SELDATA	Read/write	Selects the signal used to drive the input/output line. 0 = FIFO data drives the SCIDATA input/output line. 1 = SCISYNCDATA register bit 0 (**WDATA**) drives the SCIDATA input/output line.

3.3.26 SCISYNCDATA: [4] (+0x64)

SCISYNCDATA is the synchronous Smart Card data register and contains the source of alternate values to be used to drive the Smart Card input/output and clock signal. Table 3-29 shows bit assignments for SCISYNCDATA.

Table 3-29 SCISYNCDATA register read/write bits

Bits	Name	Type	Function
3	WCLKEN	Read/write	If **SELCLK** = 0 and **WCLKEN** = 0, the **SCICLKEN** line is forced LOW.
2	WDATAEN	Read/write	If **SELDATA** = 0 and **WDATAEN** = 0, the **SCIDATAEN** line is forced LOW.
1	WCLK	Read/write	If **SELCLK** = 1, the Smart Card clock is driven with **WCLK**.
0	WDATA	Read/write	If **SELDATA** = 1 and **WDATA** = 0, the input/output line is forced LOW.

3.3.27 SCIRAWSTAT: [2] (+0x68)

SCIRAWSTAT is the raw input/output and clock status register and provides read access to the raw status of the Smart Card input/output and clock signals. Table 3-30 shows bit assignments for SCIRAWSTAT.

Table 3-30 SCIRAWSTAT register read bits

Bits	Name	Type	Function
1	RCLK	Read	Raw value of the clock.
0	RDATA	Read	Raw value of the input/output line.

Note

In non-EMV mode of operation, the incoming bit stream from the card should be read from the SCIRAWSTAT register. The received data will not be available in the receive FIFO.

3.3.28 SCIIIR/SCIICR: [12] (+0x6c)

SCIIIR/SCIICR is the interrupt identification/interrupt clear register and provides a means of identifying and clearing the twelve individual interrupts.

There are twelve possible interrupt sources which are accessed directly by the interrupt controller. These twelve sources are ORed together and presented as a single interrupt source to the interrupt controller.

Reading the SCIIIR provides a means of identifying the specific interrupt source. Reading the SCIIIR does not modify its contents. For SCIIIR[8:0], writing a 1 to the respective bit position clears the interrupt. Writing a 0 to any position has no effect. SCIIIR[11:10] are dynamically modified by operations on the respective FIFOs. SCIIIR[9] resets whenever a valid read on the SCIDATA register occurs.

The interrupt status bits that are readable using this register are not masked by their respective enable bits. Table 3-31 shows bit assignments for SCIIIR/SCIICR.

Table 3-31 SCIIIR/SCIICR register read/write bits

Bits	Name	Type	Function
11	SCITXTIDEINTR	Read	Transmit FIFO tide mark reached interrupt.
10	SCIRXTIDEINTR	Read	Receive FIFO tide mark reached interrupt.
9	SCIRTOUTINTR	Read	Receive FIFO read timeout interrupt.
8	SCICHTOUTINTR SCICHTOUTINTRCLR	Read Write	Between character timeout interrupt. 1 = Clear interrupt.
7	SCIBLKTOUTINTR SCIBLKTOUTINTRCLR	Read Write	Between block timeout interrupt. 1 = Clear interrupt.
6	SCIATRDTOUTINTR SCIATRDTOUTINTRCLR	Read Write	ATR reception duration timeout interrupt. 1 = Clear interrupt.
5	SCIATRSTOUTINTR SCIATRSTOUTINTRCLR	Read Write	ATR reception start timeout interrupt. 1 = Clear interrupt.
4	SCITXERRINTR SCITXERRINTRCLR	Read Write	Character transmission error interrupt. 1 = Clear interrupt.
3	SCICARDDNINTR SCICARDDNINTRCLR	Read Write	Smart Card powered down interrupt. 1 = Clear interrupt.

Table 3-31 SCIIIR/SCIICR register read/write bits (continued)

Bits	Name	Type	Function
2	SCICARDUPINTR SCICARDUPINTRCLR	Read Write	Smart Card powered up interrupt. 1 = Clear interrupt.
1	SCICARDOUTINTR SCICARDOUTINTRCLR	Read Write	Smart Card removed interrupt. 1 = Clear interrupt.
0	SCICARDINTR SCICARDINTRCLR	Read Write	Smart Card inserted interrupt. 1 = Clear interrupt.

Chapter 4
Programmer's Model for Test

The ARM PrimeCell Smart Card Interface (PL010) contains additional logic for functional verification and production testing:

4.1 PrimeCell SCI test harness overview

The additional logic for functional verification and production testing allows:

- stimulation of input signals to the block
- generation of a special test clock enable signal to propagate test vectors.

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.

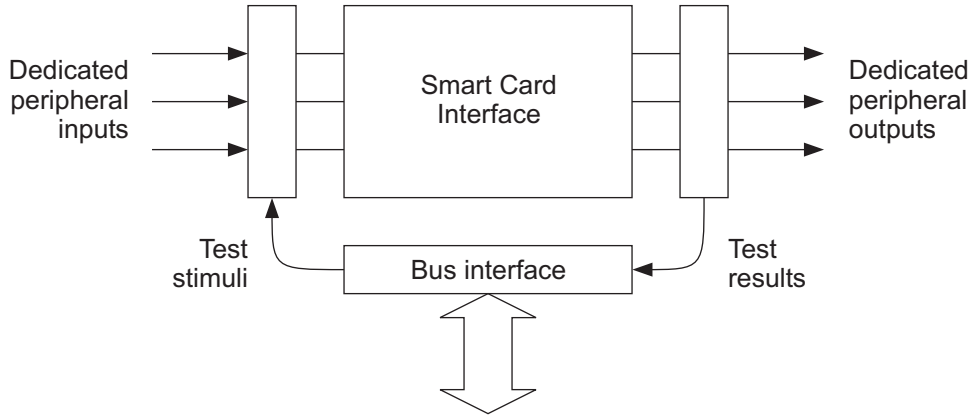


Figure 4-1 PrimeCell SCI test harness

4.2 Scan testing

The SCI peripheral 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 SCI test registers are memory-mapped as follows shown in Table 4-1.

Table 4-1 Test registers memory map

Address	Type	Width	Reset Value	Name	Description
SCI Base + 0x80	Read/write	5	0x00	SCITCR	Test control register.
SCI Base + 0x84	Read/write	6	0x00	SCITMR	Test mode register.
SCI Base + 0x88	Read/write	4	0x0	SCITISR	Test input stimulus register.
SCI Base + 0x8c	Read	2	0x0	SCITOCR	Test output capture register.
SCI Base + 0x90	Read	16	0x0000	SCIDATATIME	Data timer read back register.
SCI Base + 0x94	Read	16	0x0000	SCIBAUDCNT	Baud counter read back register.
SCI Base + 0x98	Read	16	0x0000	SCIVALUECNT	Value counter read back register.
SCI Base + 0x9c	Read	16	0x0000	SCIRTSTBPRECNT	Receive read timeout counter/stable counter prescaler read back.
SCI Base + 0xa0	Read	16	0x0000	SCIACTTIME	Activation timer read back.
SCI Base + 0xa4	Read	16	0x0000	SCICLKICCCNT	Smart clock generating counter read back.
SCI Base + 0xa8	Read	4	0x0	SCISTATE	Control state machine status.
SCI Base + 0xac to 0xbc			--		Reserved (for test purposes).
SCI Base + 0xc0 to 0xfc	Read/write	0	0x0	SCITCER	Test clock enable register.

4.3.1 SCITCR [16] (+0x80)

SCITCR is the test control register and specifies the clocking scheme to be used in test mode. It does this by controlling the clocking mode and the input pin multiplexing during test mode.

In the clock enable test mode, all clock inputs to the PrimeCell SCI are driven by the same signal. Test mode is said to have been entered when the TESTEN bit is set to 1. In test mode, if the TESTCLKEN bit is set to 1, *test clocking* is enabled. When test clocking is enabled, if the REGCLK bit is 0, a pulse is generated on the **TESTCLKEN**

signal on every access to the SCI. When test clocking is enabled, if the REGCLK bit is 1, a pulse is generated on the **TESTCLKEN** signal only when the Test Clock Enable register (SCITCER) is accessed.

The TESTRST bit in the register allows the flip-flops in the design to be asynchronously reset in test mode. This helps to avoid cross propagation in simulations and resets the internal registers during production level testing.

The TESTINSPEL bit selects the source for the internal input signal for external non-AMBA inputs. Table 4-2 shows bit assignments for SCITCR.

Table 4-2 SCITCR register read/write bits

Bit	Name	Type	Description
15:5			Reserved, read unpredictable, should be written as 0. All bits are cleared to 0 on reset by **BnRES.**
4	TESTINPSEL	Read/write	Test Input Select. By default, this bit is cleared to 0 for normal operation. This bit selects the source for the primary 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 value programmed in SCITISR is used to determine the data bit used to drive the receive data path for external non-AMBA inputs while in test mode.
3	TESTRST	Read/write	Test Reset. 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).

Table 4-2 SCITCR register read/write bits (continued)

Bit	Name	Type	Description
2	REGCLK	Read/write	Registered Clock Mode. 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 SCITCER 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	TESTCLKEN	Read/write	Test Clock Enable. This bit selects the source of the test clock: 0 = The internal clock enable is forced continuously HIGH. 1 = The internal test clock enable is selected, so that 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	TESTEN	Read/write	Test Mode Enable. 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**.

4.3.2 SCITMR [6] (+0x84)

SCITMR is the test mode register and controls the specific test modes for the SCI. These test modes improve controllability so that a high fault coverage can be achieved.

If NIB bits of the counters are set to 1, the counters count up as 0000 - 1111 - 2222... dddd - eeee - ffff. This feature reduces the number of test vectors required to test the counter.

All the bits are read as 0 after reset. shows bit assignments for SCITMR.

Table 4-3 SCITMR register read/write

Bit	Name	Type	Description
5	CLKICCNIBCNT	Read/write	Place SCICLKICCCNT in nibble mode.
4	ACTNIBCNT	Read/write	Place SCIACTTIME in nibble mode.
3	RTSTBPRENIBCNT	Read/write	Place SCIRTSTBPRECNT in nibble mode.
2	VALUENIBCNT	Read/write	Place SCIVALUECNT in nibble mode.
1	BAUDNIBCNT	Read/write	Place SCIBAUDCNT in nibble mode.
0	DATANIBCNT	Read/write	Place SCIDATATIME in nibble mode.

4.3.3 SCITISR:[4] (+0x88)

SCITISR is the test input stimulus register and provides direct control of the SCI non-AMBA primary inputs. Table 4-4 shows bit assignments for SCITISR.

Table 4-4 SCITISR register read/writebits

Bit	Name	Type	Description
3	SCIDEACREQ	Read/write	Programmable test stimulus to primary input **SCIDEACREQ**.
2	SCIDETECT	Read/write	Programmable test stimulus to primary input **SCIDETECT.**
1	SCIDATAIN	Read/write	Programmable test stimulus to primary input **SCIDATAIN**.
0	SCICLKIN	Read/write	Programmable test stimulus to primary input **SCICLKIN**.

4.3.4 SCITOCR:[2](+0x8c)

SCITOCR is the test output capture register and is a read-only register that provides observation of the non-AMBA outputs of the SCI. Table 4-5 shows bit assignments for SCITOCR.

Table 4-5 SCITOCR register read bits

Bit	Name	Type	Description
13	SCIDEACACK	Read	Test observation for primary output of deactivation acknowledgement to the PMU. This bit is cleared to 0 by default on reset by BnRES.
12	SCIINTR	Read	Test observation for OR of interrupts source. This bit is cleared to 0 by default on reset by BnRES.
11	SCITXTIDEINTR	Read	Test observation for SCITXTIDEINTR
10	SCIRXTIDEINTR	Read	Test observation for SCIRXTIDEINTR
9	SCIRTOUTINTR	Read	Test observation for SCIRTOUTINTR
8	SCICHTOUTINTR	Read	Test observation for SCICHTOUTINTR
7	SCIBLKTOUTINTR	Read	Test observation for SCIBLKTOUTINTR
6	SCIATRDTOUTINTR	Read	Test observation for SCIATRDTOUTINTR
5	SCITRSTOUTINTR	Read	Test observation for SCITRSTOUTINTR
4	SCITXERRINTR	Read	Test observation for SCITXERRINTR
3	SCICARDDNINTR	Read	Test observation for SCICARDDNINTR
2	SCICARDUPINTR	Read	Test observation for SCICARDUPINTR
1	SCICARDOUTINTR	Read	Test observation for SCICARDOUTINTR
0	SCICARDININTR	Read	Test observation for SCICARDININTR

4.3.5 SCIDATATIME: [16] (+0x90)

SCIDATATIME is the data timer read back register and returns the current state of the data timer. shows bit assignments for SCIDATATIME.

The counter is used for various activities during data transfer. They are:

- block timeout
- character timeout
- block guard time

- character guard time.

Table 4-6 SCIDATATIME register read bits

Bit	Name	Type	Description
[15:0]	DATATIME	Read	Status of DATA timer.

4.3.6 SCIBAUDCNT:[16](+0x94)

SCIBAUDCNT is the baud counter read back register and returns the current state of the baud counter. Table 4-7 shows bit assignments for SCIBAUDCNT.

Table 4-7 SCIBAUDCNT register read bits

Bit	Name	Type	Description
[15:0]	BAUDCNT	Read	Status of baud counter.

4.3.7 SCIVALUECNT:[16](+0x98)

SCIVALUECNT is the value counter read back register and returns the current state of the value counter. Table 4-8 shows bit assignments for SCIVALUECNT.

Table 4-8 SCIVALUECNT

Bit	Name	Type	Description
[15:0]	VALUECNT	Read	Status of Value counter.

4.3.8 SCIRTSTBPRECNT:[16](+0x9c)

SCIIRTSTBPRECNT is the receive read timeout counter/stable counter prescaler read back register. It returns the current state of the receive read timeout counter/stable counter prescaler. shows bit assignments for SCIRTSTBPRECNT.

Table 4-9 SCIRTSTBPRECNT register read bits

Bit	Name	Type	Description
[15:0]	RTSTBRECNT	Read	Status of receive read timeout counter/stable counter prescaler.

4.3.9 SCIACTTIME:[16](+0xa0)

SCIACTTIME is the activation timer read back register and returns the current state of the activation timer. shows bit assignments for SCIACTTIME.

This counter is used for various activities during activation/deactivation. They are:

- debounce time
- activation sequence time
- ATR start timeout
- ATR duration timeout
- deactivation sequence time.

Table 4-10 SCIACTTIME register read bits

Bit	Name	Type	Description
[15:0]	ACTTIME	Read	Status of receive read timeout counter/stable counter prescaler.

4.3.10 SCICLKICCCNT:[16](+0xa4)

SCICLKICCCNT is the Smart Card clock generating counter read back register and returns the current state of the SCICLKICC counter. Table 4-11 shows bit assignments for SCICLKICCCNT.

Table 4-11 SCICLKICCCNT register read bits

Bit	Name	Type	Description
[15:0]	CLKICCCNT	Read	Status of SCICLKICC counter.

4.3.11 SCISTATE:[4](+0xa8)

SCISTATE is the control state machine status register and returns the current state of the control state machine. Table 4-12 shows bit assignments for SCISTATE.

Table 4-12 SCISTATE register read bits

Bit	Name	Type	Description
[15:0]	SCISTATE	Read	Status of control state machine states.

4.3.12 SCITCER[0](+0xc0 to 0xfc)

SCITCER is the test clock enable register and is a virtual register that is selected for address offsets from 0xc0 to 0xfc. In registered clock mode, a test clock enable is produced each time this register is accessed. Reads return zeros and write data is ignored. Table 4-13 shows bit assignments for SCITCER.

Table 4-13 SCITCER register read/write bits

Bit	Name	Type	Description
[0]	SCITCER	Read	When in registered clock mode (refer to *SCITCR [16] (+0x80)* on page 4-4), a test clock enable is produced only when this register is accessed (read or write). A read will return zeros. Write data is ignored.

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

Appendix A
ARM PrimeCell Smart Card Interface (PL130) Signal Descriptions

This appendix describes the signals that interface with the ARM PrimeCell Smart Card Interface (PL130). It contains the following:

A.1 AMBA APB signals

The PrimeCell SCI is connected to the AMBA APB bus as a bus slave. With the exception of the **BnRES** signal, the APB signals have a **P** prefix and are active HIGH. Active LOW signals contain a lower case **n.** The AMBA APB signals are described in Table A-1.

Table A-1 AMBA APB signal descriptions

Name	Type	Source/ destination	Description
BnRES	Input	Reset controller	Bus reset signal, active LOW.
PADDR[7:2]	Input	APB bridge	Subset of AMBA APB address bus.
PCLK	Input	Clock generator	AMBA APB clock, used to time all bus transfers.
PENABLE	Input	APB bridge	AMBA APB enable signal. **PENABLE** is asserted HIGH for one cycle of **PCLK** to enable a bus transfer.
PRDATA [15:0]	Output	APB bridge	Subset of unidirectional AMBA APB read data bus.
PSEL	Input	APB bridge	PrimeCell SCI select signal from decoder. When HIGH this signal indicates the slave device is selected by the APB bridge, and that a data transfer is required.
PWDATA [15:0]	Input	APB bridge	Subset of unidirectional AMBA APB write data bus.
PWRITE	Input	APB bridge	AMBA APB transfer direction signal, indicates a write access when HIGH, read access when LOW.

A.2 On-chip signals

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

Table A-2 On-chip signals

Name	Type	Source/ destination	Description
SCIREFCLK	Input	Clock generator	PrimeCell SCI reference clock
nSCIRST	Input	Reset controller	PrimeCell SCI reset signal to **SCIREFCLK** clock domain, active LOW. The reset controller must use **BnRES** to assert **nSCIRST** asynchronously but negate it synchronously with **SCIREFCLK**.
SCICARDININTR	Output	Interrupt controller	PrimeCell SCI card in interrupt (active HIGH).
SCICARDOUTINTR	Output	Interrupt controller	PrimeCell SCI card out interrupt (active HIGH).
SCICARDUPINTR	Output	Interrupt controller	PrimeCell SCI card powered up interrupt (active HIGH).
SCICARDDNINTR	Output	Interrupt controller	PrimeCell SCI card powered down interrupt (active HIGH).
SCITXERRINTR	Output	Interrupt controller	PrimeCell SCI character transmission error interrupt (active HIGH).
SCIATRSTOUTINTR	Output	Interrupt controller	PrimeCell SCI ATR start timeout interrupt (active HIGH).
SCIATRSDOUTINTR	Output	Interrupt controller	PrimeCell SCI ATR duration timeout interrupt (active HIGH).
SCIBLKTOUTINTR	Output	Interrupt controller	PrimeCell SCI block timeout interrupt between blocks (active HIGH).
SCICHTOUTINTR	Output	Interrupt controller	PrimeCell SCI character timeout interrupt between characters (active HIGH).
SCIRTOUTINTR	Output	Interrupt controller	PrimeCell SCI receive FIFO read timeout interrupt (active HIGH)
SCIRXTIDEINTR	Output	Interrupt controller	PrimeCell SCI receive FIFO tide mark reached interrupt (active HIGH).

Table A-2 On-chip signals (continued)

Name	Type	Source/ destination	Description
SCITXTIDEINTR	Output	Interrupt controller	PrimeCell SCI transmit FIFO tide mark reached interrupt (active HIGH).
SCIINTR	Output	Interrupt controller	PrimeCell SCI interrupt (active HIGH). A single combined interrupt generated as an OR function of the twelve individually maskable interrupts above.
SCANMODE	Input	Test controller	PrimeCell SCI scan test hold input. This signal must be asserted HIGH during scan testing to ensure that internal data storage elements can be asynchronously reset. **SCANMODE** must be negated LOW during normal use or when applying manufacturing test vectors via the *Test Interface Controller* (TIC).

A.3 Signals to pads

Table A-3 describes the signals from the PrimeCell SCI to 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
SCICLKIN	Input	PAD	PrimeCell SCI clock input.
SCIDATAIN	Input	PAD	PrimeCell SCI serial data input.
nSCICLKOUTEN	Output	PAD	Tristate output buffer control (active LOW).
SCICLKOUT	Output	PAD	Clock output.
nSCIDATAOUTEN	Output	PAD	Data output enable (typically drives an open-drain configuration, active LOW).
nSCICLKEN	Output	PAD	Tristate control for external off-chip buffer (active LOW).
nSCIDATAEN	Output	PAD	Tristate control for external off-chip buffer (active LOW).
SCIVCCEN	Output	PAD	Supply voltage controls (active HIGH).
nSCICARDRST	Output	PAD	Reset to card (active LOW).
SCIDETECT	Input	PAD	Card detects signal from card interface device (active HIGH).
SCIDEACREQ	Input	PMU	Card deactivation request signal from PMU (active HIGH).
SCIDEACACK	Output	PMU	Card deactivation acknowledgement to PMU (active HIGH).

Index

R

S

T

V

W

