StrongARM® SA-1101 Microprocessor Companion Chip

Developer's Manual

PRELIMINARY

January 1999

Notice: This document contains preliminary information on new products in production. The specifications are subject to change without notice. Verify with your local Intel sales office that you have the latest technical reference manual before finalizing a design.

Order Number: 278170-002

Information in this document is provided in connection with Intel products. No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted by this document. Except as provided in Intel's Terms and Conditions of Sale for such products, Intel assumes no liability whatsoever, and Intel disclaims any express or implied warranty, relating to sale and/or use of Intel products including liability or warranties relating to fitness for a particular purpose, merchantability, or infringement of any patent, copyright or other intellectual property right. Intel products are not intended for use in medical, life saving, or life sustaining applications.

Intel may make changes to specifications and product descriptions at any time, without notice.

Designers must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." Intel reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them.

The SA-1101 may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Contact your local Intel sales office or your distributor to obtain the latest specifications and before placing your product order.

Copies of documents which have an ordering number and are referenced in this document, or other Intel literature may be obtained by calling 1-800-548-4725 or by visiting Intel's website at http://www.intel.com.

Copyright © Intel Corporation, 1999

*Third-party brands and names are the property of their respective owners.

ARM and the ARM Powered logo are trademarks and StrongARM is a registered trademark of ARM Limited.

Contents

1 Introduction 1-1

1.1 System Block Diagram 1-1
1.2 Signal Description 1-2

2 Functional Description 2-1

2.1 Functional Blocks 2-1
2.1.1 SA-1100 Memory Interface and Shared Memory Controller 2-2
2.1.2 Video Subsystem 2-2
2.1.2.1 Unified Display Memory Mode 2-3
2.1.2.2 Dedicated Display Memory Mode (Snoop) 2-4
2.1.3 Peripherals 2-5
2.2 Clock Generation and Distribution 2-5
2.3 Reset 2-6
2.4 Modes of Operations 2-7
2.4.1 Normal Mode 2-7
2.4.2 Doze Mode 2-7
2.4.3 Sleep Mode 2-9
2.5 Test 2-10

3 Interface and Shared Memory Controller 3-1

3.1 Signal Description 3-1
3.1.1 SA-1100 Memory Interface 3-1
3.1.2 AMBA System Bus Interface (ASB) 3-2
3.1.3 Update FIFO 3-3
3.1.4 VGA Controller 3-3
3.1.5 Test Interface Controller (TIC) 3-4
3.1.6 System Interface 3-4
3.2 Functional Description 3-4
3.2.1 ASB and APB Register Access 3-4
3.2.2 USB Data Transfer 3-5
3.2.3 Unified Display Memory Mode 3-5
3.2.4 SMC Arbitration 3-5
3.2.5 DRAM Address Generation 3-5
3.2.6 DRAM Control Signal Generation 3-6
3.2.7 Dedicated Display Memory Mode – Snoop Operation 3-6
3.3 Programmer's Model 3-8
3.3.1 Control Register (SKCR) 3-8
3.3.2 Shared Memory Controller Register (SMCR) 3-8
3.3.3 Snoop Register (SNPR) 3-9
3.3.4 Memory Map 3-9
3.4 AC Characteristics 3-10
3.4.1 Shared DRAM 3-10
3.4.2 Register Reads 3-11
3.4.3 Register Writes 3-11

4 Video Memory Controller 4-1

4.1 Signal Description 4-1

4.1.1 External Interface to DRAM 4-1
4.1.1.1 AMBA Peripheral Bus Interface (APB) 4-2
4.1.1.2 Update FIFO 4-2
4.1.1.3 VGA Controller 4-3
4.2 Functional Description 4-3
4.2.1 Data Transfer Arbitration 4-3
4.2.2 FIFO Emptying 4-4
4.2.3 Memory Refresh 4-4
4.2.4 Burst Transfers 4-5
4.2.5 Sleep 4-5
4.2.6 Reset 4-5
4.3 Programmer's Model 4-6
4.3.1 Configuration Register (VMCCR) 4-6
4.3.2 Memory Map 4-7
4.4 AC Characteristics 4-7
4.4.1 Video DRAM Read and Write 4-7
4.4.2 Memory Refresh 4-8
4.4.3 Self-Refresh 4-9
4.5 Test 4-9

5 Update FIFO 5-1

5.1 Signal Description 5-1
5.1.1 AMBA Peripheral Bus 5-1
5.1.2 Video Memory Controller Interface 5-2
5.1.3 Shared Memory Controller Interface 5-2
5.1.4 Update FIFO Interface 5-2
5.2 Programmer's Model 5-3
5.2.1 Update FIFO Control Register (UFCR) 5-3
5.2.2 Update FIFO Status Register (UFSR) 5-3
5.2.3 Memory Map 5-3
5.3 Test 5-4
5.3.1 FIFO Level Register 5-4
5.3.2 FIFO Data Register 5-4
5.4 Memory Map 5-4

6 Arbiter 6-1

6.1 Block Diagram 6-1
6.2 Signal Description 6-2
6.3 Functional Description 6-3
6.3.1 Request/Grant 6-3
6.3.2 Reset 6-3
6.4 Programmer's Model 6-3
6.5 Test 6-3

7 Decoder 7-1

7.1 Signal Description 7-2
7.2 Functional Description 7-3
7.2.1 Slave Response 7-4
7.2.2 Decode Control State Machine 7-4
7.2.3 Address Decoder 7-5
7.2.4 Reset 7-5

7.2.5 Programmer's Model 7-5
7.3 Test 7-5

8 Test Interface Controller 8-1

8.1 Block Diagram 8-1
8.2 Signal Description 8-2
8.3 Functional Description 8-3
8.3.1 Test Mode 8-4
8.3.2 Vectors 8-4

9 ASB-APB Bridge 9-1

9.1 Block Diagram 9-1
9.2 Signal Description 9-2
9.3 Functional Description 9-6

10 System Controller 10-1

10.1 Functional Description 10-1
10.1.1 PWM Outputs 10-1
10.1.2 Video Clock Jitter Compensation 10-1
10.2 Programmer's Model 10-1
10.2.1 Power Control Register (SKPCR) 10-2
10.2.2 Clock Divider Register (SKCDR) 10-3
10.2.3 DAC Data Registers (DACDR1 and DACDR2) 10-4
10.2.4 Memory Map 10-4

11 IEEE 1284 Parallel Port Interface 11-1

11.1 Signal Descriptions 11-2
11.1.1 IEEE Port I/O: Compatibility Mode 11-2
11.1.2 IEEE Port I/O: Nibble Mode 11-3
11.1.3 IEEE Port I/O: Byte Mode 11-4
11.1.4 IEEE Port I/O: Extended Capabilities Port (ECP) Mode 11-5
11.1.5 IEEE Port I/O: Enhanced Paralled Port (EPP) Mode 11-6
11.1.6 Interrupt I/O Signals 11-7
11.1.7 Miscellaneous Signals 11-7
11.2 Functional Description 11-8
11.2.1 Block Diagram 11-8
11.2.2 Functionality 11-8
11.2.3 Communication Sequence 11-9
11.2.3.1 Negotiation Sequence 11-9
11.2.3.2 Termination 11-10
11.2.4 Compatibility Mode 11-10
11.2.5 Nibble Mode 11-11
11.2.6 Byte Mode 11-11
11.2.7 ECP Mode 11-11
11.2.8 EPP Mode 11-12
11.2.9 FIFO Transfers 11-12
11.3 Programmer's Model 11-13
11.3.1 IEEE Config Register 11-14
11.3.2 Control Register 11-15
11.3.3 Data Register 11-15
11.3.4 Addr Register 11-16

11.3.5 Status Register 11-16
11.3.6 IntStatus Register 11-17
11.3.7 FifoLevels Register 11-18
11.3.8 InitTime Register 11-19
11.3.9 TimerStatus Register 11-19
11.3.10 FIFO Reset Register 11-19
11.3.11 ReloadValue Register 11-20
11.4 Test Registers 11-20
11.4.1 TestControl Register 11-21
11.4.2 TestDataIn Register 11-21
11.4.3 TestDataInEn Register 11-21
11.4.4 TestCtrlIn Register 11-22
11.4.5 TestCtrlInEn Register 11-22
11.4.6 TestDataStat Location 11-22

12 VGA Controller 12-1

12.1 Video System Block Diagram 12-1
12.2 Functional Description 12-2
12.2.1 VGA Data Path 12-2
12.2.2 VGA Control Logic 12-2
12.3 Programmer's Model 12-3
12.3.1 Video Control Register (VideoControl) 12-3
12.3.2 VGATiming 0 Register (VgaTiming0) 12-5
12.3.3 VGATiming 1 Register (VgaTiming1) 12-6
12.3.4 VGA Timing 2 Register (VgaTiming2) 12-7
12.3.5 VGA Timing 3 Register (VgaTiming3) 12-8
12.3.6 VGA Border Color Register (VgaBorder) 12-9
12.3.7 VGADMA Base Address Register (VgaDBAR) 12-9
12.3.8 VGADMA Channel Current Address Register (VgaDCAR) 12-9
12.3.9 VGA Status Register (VgaStatus) 12-10
12.3.10 VGA Interrupt Mask Register (VgaInterruptMask) 12-10
12.3.11 Vga Interrupt Register (VgaInterrupt) 12-11
12.3.12 VGA Palette Registers (VgaPalette) 12-11
12.3.13 DAC Control Register (DacControl) 12-11
12.3.14 Register Map Locations 12-12
12.4 Test 12-13
12.4.1 VGA Controller Test Register 12-13
12.4.2 Memory Map 12-13
12.5 Characteristics 12-14
12.6 Application 12-15
12.6.1 IREF Resistor 12-15

13 USB Host Interface Controller 13-1

13.1 Functional Description 13-1
13.1.1 USB Reset 13-1
13.1.2 USB Suspend 13-1
13.1.3 Power Management 13-2
13.1.3.1USB Clock Stopping 13-2
13.1.3.2Port Power Enable 13-2
13.1.3.3Port Resume Interrupt 13-3
13.1.4 Power-Management Routines 13-3

13.1.4.1Initial Port Power-Down Sequence....13-3
13.1.4.2Low-Power Mode in Suspend State Sequence....13-3
13.1.4.3Low-Power Mode after Device Disconnect Sequence....13-4
13.2 Programmer's Model....13-4
13.2.1 Status Register....13-4
13.2.2 Reset Register....13-5
13.3 Register Memory Map....13-6
13.4 Test Interface....13-7
13.4.1 Transfer Address Register (USTAR)....13-7
13.4.2 Write Data FIFO Register (USWER)....13-7
13.4.3 Read Data FIFO Regsiter (USRFR)....13-7
13.4.4 Next Data FIFO Register (USNFR)....13-7
13.4.5 Test Command Setup Register (USTCSR)....13-7
13.4.6 Status Register (Test) (USSR)....13-8
13.4.7 Test Memory Map....13-8

14 Interrupt Controller....14-1

14.1 Functional Description....14-1
14.1.1 Logic Diagrams....14-1
14.1.2 Interrupt Sources....14-3
14.1.3 Interrupts....14-4
14.1.4 Wake-Up Interrupts....14-4
14.1.5 Multiple Interrupts....14-4
14.2 Programmer's Model....14-5
14.2.1 Registers....14-5
14.2.1.1Source Bit Positions....14-5
14.2.2 INTTEST0 – INTTEST1....14-6
14.2.3 INTENABLE0 – INTENABLE1....14-6
14.2.4 INTPOL0 – INTPOL1....14-6
14.2.5 INTTSTSEL....14-6
14.2.6 INTSTATCLR0 – INTSTATCLR1....14-6
14.2.7 INTSET0 – INTSET1....14-6
14.2.8 Memory Map....14-7
14.3 Characteristics....14-7
14.4 Application....14-7

15 PS/2 Trackpad and Mouse Interfaces....15-1

15.1 Registers....15-1
15.1.1 Control Register (KBDCR)....15-1
15.1.2 Status Register (KBDSTAT)....15-2
15.1.3 Transmit/Receive Data Register (KBDDATA)....15-2
15.1.4 Clock Division Register (KBDCLKDIV)....15-2
15.1.5 Clock Precount Register (KBDPRECNT)[8]....15-3
15.1.6 Register Memory Map....15-3
15.2 Functional Description....15-4
15.3 Test Interface....15-6
15.3.1 KBDTEST1 (MSETEST1)....15-7
15.3.2 KBDTEST2 (MSETEST2)....15-8
15.3.3 KBDTEST3 (MSETEST3)....15-8
15.3.4 KBDTEST4 (MSETEST4)....15-9
15.4 Application....15-9

16 General-Purpose I/O Interface 16-1
16.1 Functional Description 16-1
16.1.1 Sleep 16-1
16.2 Programmer's Model 16-1
16.2.1 Port A Data Write Register (PADWR) 16-1
16.2.2 Port B Data Write Register (PBDWR) 16-2
16.2.3 Port A Data Read Register (PADRR) 16-2
16.2.4 Port B Data Read Register (PBDRR) 16-2
16.2.5 Port A Data Direction Register (PADDR) 16-2
16.2.6 Port B Data Direction Register (PBDDR) 16-3
16.2.7 Port A Sleep State Register (PASSR) 16-3
16.2.8 Port B Sleep State Register (PBSSR) 16-3
16.2.9 Memory Map 16-3
16.3 Test 16-4
16.4 Application 16-4

17 Keypad Interface 17-1
17.1 Programmer's Model 17-1
17.1.1 Port X Data Write Register (PXDWR) 17-1
17.1.2 Port X Data Read Register (PXDRR) 17-1
17.1.3 Port Y Data Write Register (PYDWR) 17-1
17.1.4 Port Y Data Read Register (PYDRR) 17-2
17.1.5 Register Memory Map 17-2
17.2 Test 17-2
17.2.1 Application 17-2

18 PCMCIA Interface 18-1
18.1 PCMCIA Interface Block Diagram 18-2
18.2 Functional Description 18-3
18.2.1 Buffer Control 18-3
18.2.2 Voltage Control 18-3
18.2.3 Reset Signals 18-3
18.2.4 Control Signals 18-4
18.2.5 Other Signals 18-4
18.2.6 Interrupts 18-4
18.3 Logic Diagram 18-5
18.4 Programmer's Model 18-6
18.4.1 Status Register (PCSR) 18-6
18.4.2 Control Register (PCCR) 18-7
18.4.3 Sleep State Register (PCSSR) 18-8
18.4.4 Memory Map 18-9

19 Memory Map 19-1

20 Pin and Padout 20-1
20.1 Test Pin Multiplexing 20-8

21 Package and Power 21-1
21.1 Package 21-1
21.2 Power Dissipation 21-4
21.2.1 Power Supply Voltages 21-4

21.2.2 Absolute Maximum Ratings 21-5
21.2.3 DC Operating Conditions 21-5

Figures

1-1 SA-1100/SA-1101 Example System 1-1
2-1 SA-1101 Block Diagram 2-1
2-2 Unified Display Memeory Mode Data Flow 2-3
2-3 Dedicated Display Memory Mode Data Flow 2-4
2-4 SA-1101 Reset Sequence 2-7
3-1 Snoop Addressing 3-7
3-2 Burst of Three DRAM Access 3-10
3-3 Register Reads 3-11
3-4 Register Writes 3-11
4-1 Burst of Four DRAM Access 4-7
4-2 CAS Before RAS Refresh 4-8
4-3 Self-Refresh 4-9
6-1 Arbiter Block Diagram 6-1
7-1 Decoder Block Diagram 7-1
7-2 Decoder with Decode Cycles 7-3
7-3 Decoder State Machine with Decode Cycles 7-5
8-1 Test Interface Controller Block Diagram 8-1
8-2 Read Vectors and Turn Around 8-5
8-3 TIC Vectors and AMBA Transfer 8-5
8-4 Vectors and Waited Transfers 8-6
9-1 Block Diagram of the ASB-APB Bridge 9-1
9-2 APB Write Cycle 9-4
9-3 APB Read Cycle 9-5
9-4 State Machine for APB Controller 9-6
11-1 IEEE 1284 Port Interface Block Diagram 11-8
11-2 IEEE Config Register Bit Coding 11-14
11-3 Control Register Bit Coding 11-15
11-4 Data Register Bit Coding 11-15
11-5 Addr Register Bit Coding 11-16
11-6 Status Register Bit Coding 11-16
11-7 IntStatus Register Bit Coding 11-17
11-8 FifoLevels Register Bit Coding 11-18
11-9 InitTime Register Bit Coding 11-19
11-10 TimerStatus Register BIt Coding 11-19
11-11 ReloadValue Register Bit Coding 11-20
12-1 Video System Block Diagram 12-1
14-1 Interrupt and Wake-Up Generation 14-1
14-2 Interrupt and Wake-Up Combination 14-2
14-3 Interrupt Source Positions in Register Set 0 14-5
14-4 Interrupt Source Positions in Register Set 1 14-5
15-1 Keyboard/Mouse Controller Receive Protocol 15-4
15-2 Timing and Controller Request to Send Protocol 15-5
17-1 Keypad Example 17-3
18-1 PCMCIA Interface Block Diagram 18-2
18-2 PCMCIA Interface Logic Diagram 18-5

21-1 SA-1101 256 mBGA Mechanical Drawing 21-1

Tables

1-1 Signal Descriptions 1-2
2-1 Clock Frequencies 2-6
2-2 Test Clock Frequencies 2-11
3-1 SA-1100 Memory Interface Signals 3-1
3-2 ASB Interface Signals 3-2
3-3 Update FIFO Interface Signals 3-3
3-4 VGA Controller Interface Signals 3-3
3-5 Test Interface Controller Signals 3-4
3-6 System Interface Signals 3-4
3-7 DRAM Control Signals 3-6
3-8 Snoop Register (SNPR) 3-9
3-9 SMC Memory Map 3-9
3-10 Shared Memory Timing Parameters 3-10
3-11 Register Read and Write Timing Parameters 3-12
4-1 External Interface to DRAM Signals 4-1
4-2 APB Bus Interface Signals 4-2
4-3 Update FIFO Interface Signals 4-2
4-4 VGA Controller Interface Signals 4-3
4-5 Request Arbiter Priorities 4-4
4-6 SleepState Bit Encoding 4-6
4-7 VMC Memory Map 4-7
4-8 Video Memory Timing Parameters 4-8
4-9 Test Memory Map for the VMC 4-9
5-1 APB Interface Signals 5-1
5-2 VMC Interface Signals 5-2
5-3 SMC Interface Signals 5-2
5-4 Update FIFO Interface Signals 5-2
5-5 UFCR and UFSR Memory Map 5-3
5-6 Test Memory Map for the Update FIFO 5-4
6-1 Bus Master Priorities 6-3
7-1 Decoder with Decode Cycles Response Combinations 7-4
8-1 Signal Descriptions 8-2
8-2 Basic TIC Operation 8-3
8-3 TIC Vectors 8-4
9-1 Signal Descriptions 9-2
9-2 APB Signals 9-3
9-3 Address Decoding Block Signals 9-3
10-1 System Controller - Read/Write Locations 10-4
11-1 IEEE Port I/O Signals in Compatibility Mode 11-2
11-2 IEEE Port I/O Signals in Nibble Mode 11-3
11-3 IEEE Port I/O Signals in Byte Mode 11-4
11-4 IEEE Port I/O Signals in Extended Capabilities Port (ECP) Mode 11-5
11-5 IEEE Port I/O Signals in Enhanced Parallel Port (EPP) Mode 11-6
11-6 Interrupt Signals 11-7
11-7 Miscellaneous Signals 11-7
11-8 Outcomes of the Negotiation Sequence 11-9

11-9 Types of Termination....11-10
11-10 IEEE 1284 Parallel Port Interface Register Summary....11-13
11-11 Test Registers Summary....11-20
12-1 VGA Controller Register Map Locations....12-12
12-2 VGA Controller Test Registers....12-13
12-3 DAC DC Characteristics....12-14
12-4 DAC AC Characteristics....12-14
13-1 USB Host Interface Register Memory Map....13-6
13-2 Test Memory Map for the USB....13-8
14-1 Interrupt Sources....14-3
14-2 Interrupt Registers....14-7
14-3 Interrupt Characteristics....14-7
15-1 Trackpad/Mouse Register Memory Map....15-3
15-2 Keyboard/Mouse Interface Timings....15-5
16-1 GPIO Register Memory Map....16-3
17-1 Keypad Register Memory Map....17-2
18-1 Buffer Control Signals....18-3
18-2 Interrupt Signals....18-4
18-3 S0 Control Line Sleep State....18-8
18-4 S1 Control Line Sleep State....18-9
18-5 PCMCIA Interface Register Memory Map....18-9
19-1 SA-1101 Memory Map....19-1
20-1 Pin Descriptions....20-1
20-2 External Pins....20-8
21-1 256 Mini-BGA Dimensional Attributes....21-2
21-2 SA-1101 Pinout – 256 Mini-Ball Grid Array....21-3
21-3 SA-1101 Power Consumption at 3.3 V VDD....21-4
21-4 SA-1101 Power Supply Voltages....21-4
21-5 Absolute Maximum Ratings....21-5
21-6 SA-1101 DC Operating Conditions....21-5

Introduction 1

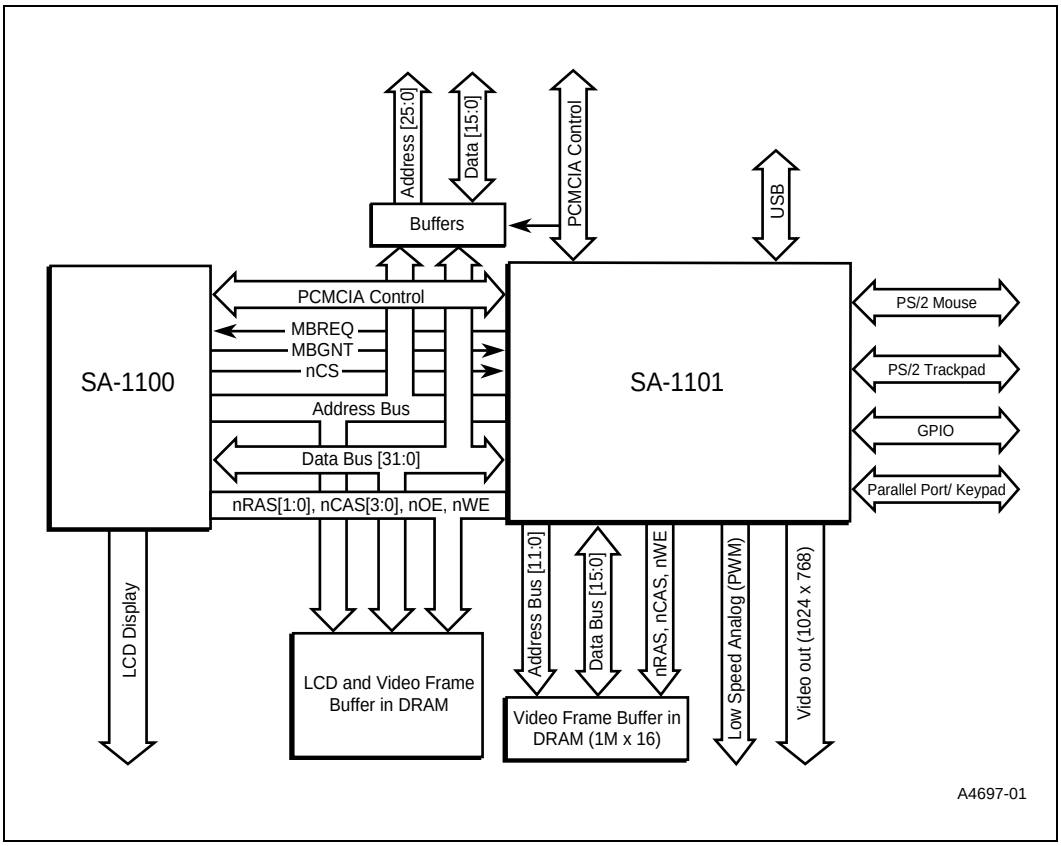
The StrongARM® SA-1101 (SA-1101) device is a companion chip to the StrongARM® SA-1100 Microprocessor (SA-1100), adding a variety of functions appropriate for use in a handheld computer system.

1.1 System Block Diagram

The SA-1101 brings a new level of integration to small systems, providing a variety of I/O functions that enable complete systems to be built with very little “glue” logic. In addition to a complete video subsystem, the SA-1101 includes a USB host controller, extensive support for PCMCIA, two PS/2 ports and keypad support, a full-function parallel port, and other I/O capabilities.

Figure 1-1 shows how the SA-1100 and the SA-1101 can be used in a handheld computing device that displays on both an LCD (from the SA-1100) and a video CRT (from the SA-1101). The two display devices can have separate DRAM frame buffers as shown, or they can share common DRAM storage using SA-1100 main memory. With separate memories, the LCD frame buffer is controlled by the SA-1100 and the video frame buffer is controlled by the SA-1101.

Figure 1-1. SA-1100/SA-1101 Example System



1.2 Signal Description

Table 1-1 describes the signals.

Table 1-1. Signal Descriptions (Sheet 1 of 4)

Name	Type	Description
SA-1100 Processor Interface		
A[21:10]	I/O	Shared 12-bit memory address bus. This multiplexed bus conveys the address for memory accesses to DRAM. The same address signals are used to access SA-1101 on-chip registers.
D[31:0]	I/O	Shared 32-bit memory data bus.
nOE	I/O	Shared memory output enable. This signal is connected to DRAM output enable pins, enabling them to drive data onto the data bus during read cycles.
nWE	I/O	Shared DRAM write enable. This signal is connected to DRAM write enables, to perform writes. In conjunction with CAS[3:0], it is used for byte writes.
nRAS[1:0]	I/O	DRAM RAS. These signals are connected to the two DRAM row address strobe (RAS) pins that address up to the two physical banks of DRAM.
nCAS[3:0]	I/O	DRAM CAS. These signals are connected to the DRAM column address strobe (CAS) pins, one for each byte of the 4-byte data width, enabling byte writes (in conjunction with nWE).
MBREQ	O	The request signal. This signal indicates that the SA-1101 needs to take the SA-1100 memory bus for reading video frame buffer data or for performing DMA into SA-1100 memory. Should be connected to the SA-1100 GPIO[22[.
MBGNT	I	The grant signal from the SA-1100. This signal indicates that it has given up the bus for use by the SA-1101. Should be connected to the SA-1100 GPIO[22[.
nCS	I	Chip select for the SA-1101 register access.
nPOE	I	PCMCIA output enable. This PCMCIA signal is input, gated within the SA-1101, and used to perform reads from memory and attribute space.
nPWE	I	PCMCIA write enable. This signal is input, gated within the SA-1101, and is used to perform writes to memory and to attribute space.
nPIOW	I	PCMCIA I/O write. This signal is input, gated within the SA-1101, and is used to perform write transactions to the PCMCIA I/O space.
nPIOR	I	PCMCIA I/O read. This signal is input, gated within the SA-1101, and is used to perform read transactions from the PCMCIA I/O space.
nPCE[2:1]	I	PCMCIA card enable. These signals are input, gated within the SA-1101, and used to select a PCMCIA card. Bit 2 enables the high byte lane and bit 1 enables the low byte lane.
nIOIS16	O	I/O is 16 bit. This signal is an output and is an acknowledge from the PCMCIA card that the current address is a valid 16-bit wide I/O address.
nPWAIT	O	PCMCIA wait. This signal is an output and is driven low by the PCMCIA card to lengthen the transfers from the SA-1100.
PSKTSEL	I	PCMCIA socket select. This signal is an input and is used to route control, address and data signals to one of the PCMCIA sockets. When PSKTSEL is low, socket zero is selected. When PSKTSEL is high, socket one is selected. This signal has the same timing as address.

Table 1-1. Signal Descriptions (Sheet 2 of 4)

Name	Type	Description
PCMCIA Interface		
S0_nCD[2:1]	I	Socket 0 card detect.
S0_READY_nIREQ	I	Socket 0 ready/busy.
S0_nCE[2:1]	O	Socket 0 card byte enables.
S0_nOE	O	Socket 0 output enable.
S0_nWE	O	Socket 0 write enable.
S0_nIORD	O	Socket 0 IOR.
S0_nIOWR	O	Socket 0 IOW.
S0_nWAIT	I	Socket 0 wait.
S0_nIOIS16	I	Socket 0 IO_16 signal.
S0_RESET	O	Socket 0 reset signal.
S0_BVD1_nSTSCHG	I	Socket 0 VDD voltage sense signals/card status changed.
S0_BVD2_nSPKR	I	Socket 0 VDD voltage sense signals/audio digital speaker.
S0_nVS[2:1]	I	Socket 0 VSS voltage sense signals.
S0_VPP0	O	Voltage control.
S0_VPP1	O	Voltage control.
S0_VCC0	O	Voltage control.
S0_VCC1	O	Voltage control.
S1_nCD[2:1]	I	Socket 1 card detect.
S1_READY_nIREQ	I	Socket 1 ready/busy.
S1_nCE[2:1]	O	Socket 1 card byte enables.
S1_nOE	O	Socket 1 output enable.
S1_nWE	O	Socket 1 write enable.
S1_nIORD	O	Socket 1 IOR.
S1_nIOWR	O	Socket 1 IOW.
S1_nWAIT	I	Socket 1 wait.
S1_nIOIS16	I	Socket 1 IO_16 signal.
S1_RESET	O	Socket 1 reset signal.
S1_BVD1_nSTSCHG	I	Socket 1 VDD voltage sense signals/card status changed.
S1_BVD2_nSPKR	I	Socket 1 VDD voltage sense signals/audio digital speaker.
S1_nVS[2:1]	I	Socket 1 VSS voltage sense signals.
S1_VPP0	O	Voltage control.
S1_VPP1	O	Voltage control.
S1_VCC0	O	Voltage control.
S1_VCC1	O	Voltage control.
PCM_ST[6:0]	O	PCMCIA steering logic signals.

Table 1-1. Signal Descriptions (Sheet 3 of 4)

Name	Type	Description
Video Interface		
BLUE	O	Analog blue out.
RED	O	Analog red out.
DAC_VSS	P/G	Analog ground.
GREEN	O	Analog green out.
IREF	I	Analog current reference.
DAC_VDD	P/G	Analog supply voltage.
HSYNC	O	Horizontal sync out.
VSYNC	O	Vertical sync out.
USB Interface		
USB_PLUS	I/O	USB transceiver plus.
USB_MINUS	I/O	USB transceiver minus.
Parallel Port Interface		
PPBUFEN_KPY8	I/O	IEEE 1284: data direction control/keypad Y8.
PPBUSY_KPY9	I/O	IEEE 1284: busy signal/keypad Y9.
PPDATA[7:0]_KPX[7:0]	I/O	IEEE 1284: bidirectional data/keypad X[7:0].
PPPERROR_KPY6	I/O	IEEE 1284: Perror/keypad Y6.
PPSELECT_KPY5	I/O	IEEE 1284: select signal/keypad Y5.
PPnACK_KPY4	I/O	IEEE 1284: nACK signal/keypad Y4.
PPnAUTOFD_KPY3	I/O	IEEE 1284: nAutoFd signal/keypad Y3.
PPnFAULT_KPY7	I/O	IEEE 1284: nFault signal/keypad Y7.
PPnINIT_KPY2	I/O	IEEE 1284: nInit signal/keypad Y2.
PPnSELECTIN_KPY1	I/O	IEEE 1284: nSELECTin signal/keypad Y1.
PPnSTROBE_ KPY0	I/O	IEEE 1284: nStrobe signal /keypad Y0.
GPB[5:0]_KPY [15:10]	I/O	General-purpose input/output or keypad.
PS/2 Trackpad Interface		
TPCLK	I/O	PS/2 track pointer clock.
TPDATA	I/O	PS/2 track pointer data.
PS/2 Mouse Interface		
MSCLK	I/O	PS/2 mouse clock.
MSDATA	I/O	PS/2 mouse data.
Brightness Contrast		
PWM1	O	PWM output 1.
PWM2	O	PWM output 2.

Table 1-1. Signal Descriptions (Sheet 4 of 4)

Name	Type	Description
General-Purpose I/O Signals		
GPA[7:0]	I/O	General-purpose input /output.
GPB[6]	I/O	General-purpose input /output.
Video Frame Buffer		
FB_A[11:0]	O	Address output to dedicated video frame buffer.
FB_D[15:0]	I/O	Data in/out for dedicated video frame buffer.
nFB_LCAS	O	Column address strobe for lower byte of video memory.
nFB_UCAS	O	Column address strobe for upper byte of video memory.
nFB_WE	O	Write enable for dedicated video frame buffer.
nFB_RAS	O	Row address strobe for dedicated video frame buffer.
Miscellaneous Signals		
INT	O	Interrupt out.
nRESET	I	SA-1101 reset.
nTEST	I	Not test is a signal used for off-board production testing of the device. It MUST be tied HIGH for normal operation. If it is driven low at any time, then internal damage to the device may occur.
CLK	I	Master clock in. Should be connected to SA-1100 GIPIO[27].
VDD_FLT	I	Problem on VDD.
BAT_FLT	I	Battery fault or sleep.
Power and Ground		
PLL_VDD	P/G	Power for PLL.
PLL_VSS	P/G	GND for PLL.
VDD	P/G	Positive supply for the core. Three pins are allocated to this supply.
VDDX	P/G	Positive supply for the I/O pins. Twenty-four pins are allocated for VDDX.
VSS	P/G	Ground supply for the core. Four pins are allocated to VSS.
VSSX	P/G	Ground supply for the I/O pins. Twenty-five pins are allocated to VSSX.

Note: The signal types defined in Table 1-1 are as follows:

I = Input only
O = Output onlt
I/O - Input/output
P/G= Power/ground

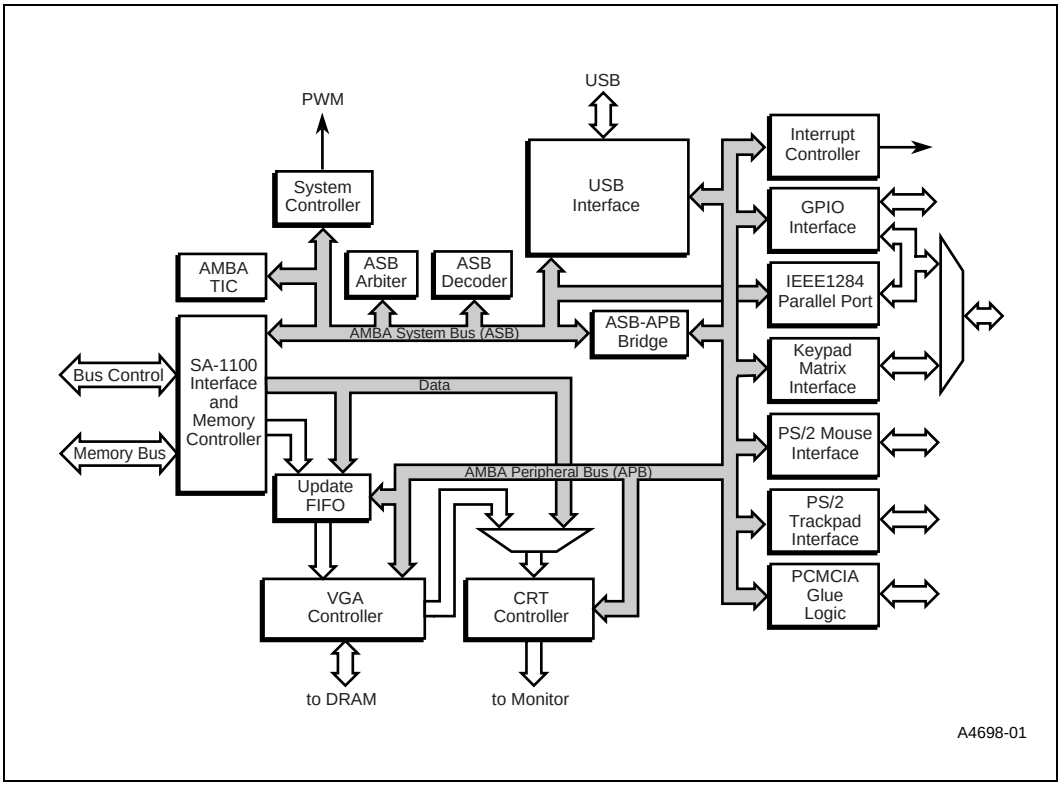
Functional Description 2

This chapter provides a functional overview of the StrongARM® SA-1101 microprocessor companion chip (SA-1101).

2.1 Functional Blocks

Figure 2-1 shows the functional blocks of the SA-1101.

Figure 2-1. SA-1101 Block Diagram



2.1.1 SA-1100 Memory Interface and Shared Memory Controller

The SA-1101 is designed around the ARM®microcontroller bus architecture. The StrongARM® SA-1100 microprocessor (SA-1100) Memory Interface and Shared Memory Controller (SMC) interfaces to the SA-1100's memory bus. Functionally the SMC operates in three different modes.

- For reads and writes to its registers
  The SA-1101 looks like SRAM, using an SA-1100 Flash ROM access type. These are 32-bit accesses with no byte or 16-bit write capability.
- For access to the main memory
  The SA-1101 takes over the shared DRAM interface. It enables its DRAM controller and drives SA-1100 DRAM with the appropriate control signals. This mode is used for video and LCD display data in unified display memory mode and for USB data transfers. See Section 2.1.2.1, "Unified Display Memory Mode" on page 2-3
- For snooping updates to the video frame buffer in dedicated display memory mode
  The SA-1101 captures DRAM writes on the SA-1100's memory bus. See Section 2.1.2.2, "Dedicated Display Memory Mode (Snoop)" on page 2-4.

2.1.2 Video Subsystem

The SA-1101 video subsystem provides the capability to drive an RGB monitor from the SA-1100. The image displayed on the SA-1100's LCD display and the SA-1101's video output can be different both in terms of content and resolution. As a result, two separate areas of memory are required for frame buffers:

- The LCD display controlled by the SA-1100
- The video display controlled by the SA-1101

For the operation of its video subsystem, the SA-1101 can operate in either of two modes: unified display memory mode and dedicated display memory mode. These modes, distinguished by the location of the video frame buffer memory, involve the interoperation of the SMC, the update FIFO, the Video Memory Controller (VMC), and the VGA controller.

2.1.2.1 Unified Display Memory Mode

In unified display memory mode, both video display data and LCD display data are stored in frame buffers within SA-1100 main memory. Because both buffers are in the SA-1100's main memory, the SA-1101 periodically accesses SA-1100 main memory to get video refresh data for its video display using the membus request (MBREQ) and membus grant (MBGNT) pins.

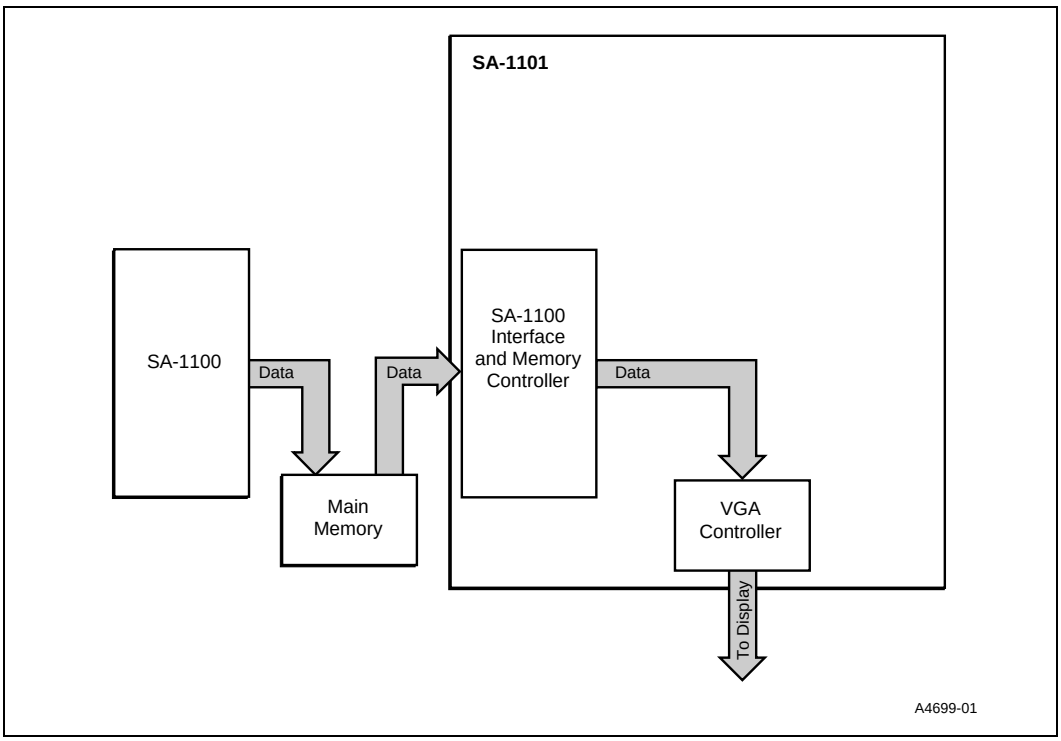
The SA-1101 asserts the MBREQ when it needs to access the shared memory. The SA-1100 completes any current memory operation before it grants the request by asserting MBGNT and giving the SA-1101 control of the SA-1100 main memory. (Other processor interface functions remain under SA-1100 control, synchronized to SA-1100 clocks, and continue operating independently.)

The SA-1101 assumes control of the main memory and does a burst read, filling a FIFO (the Video FIFO) internal to the SA-1101. The VGA controller unloads the output end of the FIFO at a slower rate, leaving time (between read bursts for video refresh) for the LCD controller and the SA-1100 CPU to access memory themselves. When the read has finished, the SA-1101 returns control of shared DRAM to the SA-1100. The SA-1101's Shared Memory Controller can read data at high bandwidth from the SA-1100 memory to meet the needs of medium-resolution video displays (640x480). Higher resolutions will typically require dedicated display memory mode. See Section 2.1.2.2, "Dedicated Display Memory Mode (Snoop)" on page 2-4

The duration of the SA-1101's possession of SA-1100 memory is strictly limited to avoid affecting the bandwidth of SA-1100 processes, including cache load/store accesses and LCD display refreshes. A register value in the SA-1101sets the length of possession.

Figure 2-2 shows the unified display memory mode data flow.

Figure 2-2. Unified Display Memeory Mode Data Flow



2.1.2.2 Dedicated Display Memory Mode (Snoop)

In dedicated display memory mode, the SA-1101 has its own DRAM attached to a separate memory interface. Data for the video display is kept in this memory (dedicated frame buffer). Therefore, the high-bandwidth accesses required to refresh the video display are restricted to this memory interface and do not interrupt SA-1100 memory operations. However the SA-1100 still needs to update the video display memory.

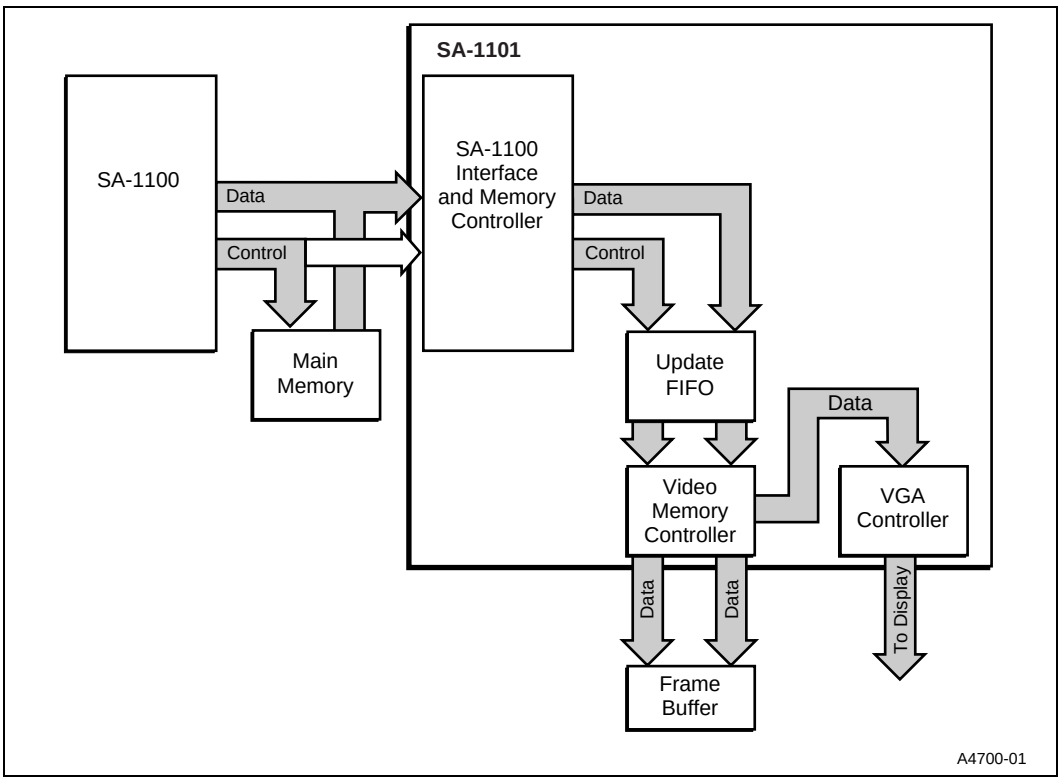
The SA-1100 writes into a specified region of memory as its frame buffer (that may or may not represent physical memory). The SA-1101 captures all write cycles going to this range of addresses and transfers the data to its own local dedicated frame buffer. Once there, it is read out repeatedly to refresh the video display. This is called “snoop” mode of operation.

Any attempted read by the SA-1100 to the region specified for the SA-1101’s frame buffer returns data from the SA-1100 memory only; the SA-1101 frame buffer does not respond to regular DRAM reads to that space. If, on the other hand, there is no physical memory present in that address range in SA-1100 memory, there will be no response to read attempts to those addresses – the SA-1100 will be addressing non-existent memory.

A 256Kx16 or 1Mx16 DRAM can be connected directly to the SA-1101. The SA-1101’s Video Memory Controller (VMC) provides a 16-bit datapath and address/control signals to this attached video frame buffer. For video display resolutions of 640x480 and 800x600 with up to 8bpp (bits per pixel), a 256Kx16 DRAM has sufficient capacity. For display resolutions of 1024x768 and 8bpp, a 1Mx16 DRAM is required.

Figure 2-3 shows a dedicated display memory mode data flow.

Figure 2-3. Dedicated Display Memory Mode Data Flow



2.1.3 Peripherals

The SA-1101 consists of the following peripherals:

- USB interface
  A single port USB host controller compliant with *USB Specification*, Revision 1.0 is provided.
- Parallel port
  An IEEE 1284 parallel port is provided. The pins for this device are multiplexed with the keypad.
- General-purpose input/output
  Fifteen lines of general-purpose digital input/put are provided. Six of these pins are multiplexed with the keypad interface.
- Keypad
  An 8- and a 16-bit port are available for use with a simple keypad. Each port can be configured as input or output to allow use as either row or column connections to a keypad matrix. These pins are multiplexed with the IEEE 1284 parallel port.
- PS/2 ports
  Two PS/2 ports are provided for use with keyboards, mice, trackpads or any other PS/2 compliant device.
- PCMCIA interface
  An interface is provided to allow the simple connection of the SA-1100 PCMCIA interface to two PCMCIA sockets.
- Pulse width modulation outputs
  Two pulse width modulation (PWM) outputs are provided. These are intended for use as digital-to-analog converters with the addition of filter components.

2.2 Clock Generation and Distribution

The SA-1101 input clock is a single 3.6864-MHz clock generated by the SA-1100 and sent from GPIO[27] to the CLK pin on the SA-1101. A phase-locked loop (PLL) in the SA-1101 generates the clocks required for its I/O functions, video output, and internal systems. The input clock is brought directly into the PLL's phase comparator. The VCO signal following the phase comparator generates a clock of 144 MHz. Several dividers following the VCO signal create lower-frequency clocks for the following on-chip subsystems:

- The USB host controller
  Requires a clock of 48 MHz (0.25% tolerance) and various submultiples of that frequency (1.5 MHz, 6 MHz, and 12 MHz).
- The video subsystem
  Uses a clock at 32 MHz for 640x480 display resolution, 48 MHz for 800x600 display resolution, and 72 MHz for 1024x768 display resolution.
- The AMBA System Bus (ASB) and the AMBA Peripheral Bus Interface (APB)
  Use a clock of 36 MHz.
- Shared Memory Controller (SMC) and Video Memory Controller (VMC)
  Uses a clock of 36 MHz.
- The DACs for LCD brightness/contrast control
  Use the 3.6864-MHz clock ungated so they can operate correctly even if all other sections of the component, including the PLL, are shut down to reduce power.

Table 2-1 lists the recommended clock frequencies for the correct operation of the SA-1101 functional blocks.

Table 2-1. Clock Frequencies

Clock	640x480 Display Resolution	800x600 Display Resolution	1024x768 Display Resolution
AMBA bus clock VCO	144 MHz	144 MHz	144 MHz
USB clock BCLK	36 MHz	36 MHz	36 MHz
Video clock UCLK[1]	48 MHZ	48 MHz	48 MHz
PS/2 reference clock VCLK	32 MHz	32 MHz	32 MHz
IEEE interface clock PCLK	8 MHz	8 MHz	8 MHz

Note: 1. The 12-MHz USB clock is provided from the 48-MHz UCLK.

These frequencies are controlled from the System Controller. (See Section 10.2.2, “Clock Divider Register (SKCDR)” on page 10-3.) The enables for each of these clocks also are located in the System Controller, (see Section 10.2.1, “Power Control Register (SKPCR)” on page 10-2) with the exception of the enables for BCLK and the PLL, which are available in the SMC (see Section 3.3.1, “Control Register (SKCR)” on page 3-8.)

2.3 Reset

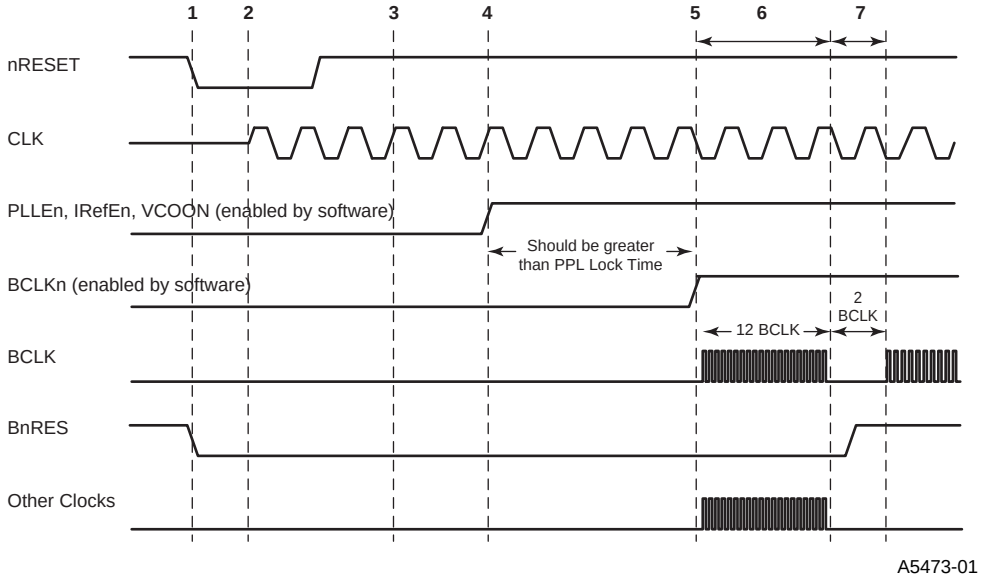
The SA-1101 is reset by asserting nRESET. When nRESET is asserted, all activity on the SA-1101 halts; when nRESET is released, the SA-1101 is in doze mode. To enable the functions of the SA-1101 and put the SA-1101 into normal mode, software must enable the PLL and BCLK. Unless indicated otherwise, all register bits are set to zero during reset.

The recommended reset sequence for the SA-1101 is as follows:

1. nRESET is asserted. This asserts BnRES (the SA-1101 system reset).
2. CLK is activated.
3. nRESET is deasserted.
4. PLLEn, IRefEn, and VCOON are asserted by software. See Section 3.3.1, “Control Register (SKCR)” on page 3-8.
5. After a period of time (greater than the LOCK time of the PLL), software asserts BCLKEn.
6. The clock controller activates BCLK for 12 cycles. This burst is routed to all clock nets in the SA-1101.
7. The Power Control Register must be set by software to allow clocking of the required parts of the SA-1101.
8. The USB must be configured before it can be used. See Section 13.1.1, “USB Reset” on page 13-1.

Figure 2-4 illustrates a reset sequence for the SA-1101.

Figure 2-4. SA-1101 Reset Sequence



2.4 Modes of Operations

The Sa-1101 operates in three modes:

- Normal
- Doze
- Sleep

2.4.1 Normal Mode

Normal mode is full operation of the SA-1101 microprocessor. All clocks are on and all functions can be used.

2.4.2 Doze Mode

Reset puts the SA-1101 into doze mode. Doze mode is the same as normal mode except that one or more of the SA-1101 functional blocks are not clocked. Clocks to some of the SA-1101 functional blocks can be selectively switched on or off under software control from the SA-1100. This feature allows the SA-1101 to dissipate lower power when not using these functional blocks.

Clocks that can be selectively switched on or off under software control are as follows:

- ICLK
  Provides all clocking for the Interrupt Controller in its normal mode. Before switching it off, select the wake-up mode. See Section 14, “Interrupt Controller” on page 14-1.
- PCLK
  Provides all clocking for the PS/2 ports. Selectively switching it on or off allows:
  - — Power down of an unused PS/2 port.
  - — Functioning of a PS/2 when BCLK is turned off.
- PICLK
  Provides all clocking for the IEEE 1284 parallel interface. Selectively switching it on or off allows:
  - — Power down of the IEEE 1284 interface, if not being used.
  - — Functioning of the IEEE 1284 interface when BCLK is turned off.
- DCLK
  Provides all clocking for the pulse width modulators (PWM) outputs. Selectively switching it on or off allows:
  - — Power down of the PWM outputs, if not being used.
  - — Functioning of the PWM outputs when BCLK is turned off.
- VCLK
  Provides all clocking for the VGA Controller. Selectively switching it on or off allows the power down of the VGA Controller, if not being used.
- UCLK
  Provides all clocking for the USB Host Controller. Selectively switching it on or off allows the power down of the USB Host Controller, if not being used.
- BCLK
  Provides all clocking for the AMBA System Bus Interface (ASB). If BCLK is turned off , only the SA-1101 Control Register can be accessed. See Section 3.3.1, “Control Register (SKCR)” on page 3-8. Before switching it off:
  - — All video activity should be stopped and VCLK should be turned off.
  - — All USB activity should be stopped and UCLK should be turned off.
  - — The frame buffer should be put into self refresh. See Section 4.3.1, “Configuration Register (VMCCR)” on page 4-6.

  With BCLK switched off:

  —Transmit/receive buffered PS/2 data and assert an interrupt (if PCLK is on).

  —Transmit/receive buffered IEEE 1284 data and assert an interrupt (if PICLK is on).

  —Output the PWM signal (if DLK is on).

  —Run the Interrupt Controller in norma mode (if ICLK is on).

  —Receive GPIO interrupts.

  —Receive keypad interrupts.

  —Receive PCMCIA interrupts.

  —Receive USBPortResume interrupt. See Section 13.1.3, “Power Management” on page 13-2 and Section 13.1.3.3, “Port Resume Interrupt” on page 13-3.

2.4.3 Sleep Mode

Sleep mode allows a reduction in system power consumption when none of the functional blocks of the SA-1101 are required. In addition to stopping all clocking activity on the SA-1101, sleep mode also

- Places the pins of the device into a minimum power configuration.
- Sets any GPIO outputs to their sleep state register values.
- Sets the PCMCIA voltage control lines to their sleep state register values.
- Sets the PCMCIA control lines according to the value in the PCMCIA sleep state register.
- Sets the frame buffer interface according to the value in the VMC Configuration Register.
- Places the frame buffer into self refresh.
- Selects the wake-up mode of the Interrupt Controller.

In sleep mode the following interrupt sources function:

- GPIO interrupts
- Keypad interrupts
- PCMCIA interrupts
- USBPortResume interrupt

Sleep mode can be initiated by:

- Software settings to the sleep bit in the SA-1101 Control Register
- Fault inputs to VDD_FLT and BAT_FLT

A state machine controls the transition to sleep mode. The state machine is clocked directly by the input clock CLK at 3.6864 MHz. The state machine has the following transitions:

- **State 0 – Active**
  This state is entered from reset. All clocks are enabled and pins are set to their rest condition. See Table 20-1.
- **State 1 – Request**
  This state is entered from the active state, by either:
  — A register write of 1 from the SA-1100 to the sleep bit of the SA-1101 Control Register (SKCR). See Section 3.3.1, “Control Register (SKCR)” on page 3-8.
  — An active BAT_FLT or an active VDD_FLT.
  This state asserts Self Refresh Request (SRReq) to the Video Memory Controller.
- **State 2 – Shutdown_1**
  This state is entered from the request state when:
  — Self Refresh Acknowledge (SRAck) is received from the Video Memory Controller confirming that it has placed the frame buffer memory interface into self refresh mode.
  — The Test Interface Controller (TIC) becomes bus master.

Note: The TIC is the default bus master of the ASB. The TIC becomes the bus master, in normal mode, if no other bus master is requesting the bus. If BAT_FLT (or VDD_FLT) occurs during a register

write from the SA-1100, this state completes and theTIC becomes a bus master. If BAT_FLT (or VDD_FLT) occurs during a USB transfer, the transfer will probably not succeed. The USB will cease to be granted on the ASB, leaving the TIC as bus master. It will then be safe to continue with the transition to sleep mode.

Shutdown_1 causes the Power Control Register to be cleared, which disables all the clocks and sets the multiplexer to keypad mode.

- **State 3 – Shutdown_2**

  This state is unconditionally entered from Shutdown_1. Because all subsidiary clocks have ceased, this state disables BCLK and the PLL.

- **State 4 – Sleep**

  This state is unconditionally entered from Shutdown_2. The following conditions are set within the SA-1101:

  — Pads are set into their associated sleep condition.

  — GPIO ouput lines are set according to the value in the GPIO sleep state register.

  — PCMCIA voltage control lines are set according to the PCMCIA sleep state register value.

  — PCMCIA control lines are set according to the PCMCIA sleep state register value.

  — The frame buffer interface is set according to the VMC Configuration Register value.

  — The video DACs and the PLL will be turned off.

- **State 5 – Wake-up**

  This state is entered from sleep mode when nCS next goes active. The pads are released from their sleep condition and return to normal operation. If BAT_FLT or VDD_FLT are asseted , a transition is made back to sleep.

  When the state machine returns to the active state, the SA-1101 Control Register (SKCR) can be accessed allowing the software to reenable the PLL and BCLK. (See Section 3.3.1, “Control Register (SKCR)” on page 3-8.) The reconfiguration of the SA-1101 for normal operation then can be carried out as required.

2.5 Test

In test mode the PLL is put into Bypass mode, so that the input frequency to the PLL (CLK) is routed directly to PLLCLK. BCLK is set to CLK/2 by setting BCLKDiv2 in the Clock Divider Register avoiding the need for clock gating on BCLK. Clock multiplexers are provided on other clock nets. See Section 10.2.2, “Clock Divider Register (SKCDR)” on page 10-3.

Table 2-2 lists the frequencies for the test clocks.

Table 2-2. Test Clock Frequencies

Clock	Normal	Test
BCLK ICLK	VCO/4	CLK/2
VCLK	VCO/4.5, VCO/3, or VCO/2	CLK
UCLK48	VCO/3	CLK/3
UCKL12	VCO/12	UTestCLK
PCLK	VCO/18	CLK/2

Interface and Shared Memory Controller 3

The SA-1100 interface and Shared Memory Controller (SMC) is the primary interface between the StrongARM® SA-1101 (SA-1101) and the StrongARM® SA-1100 (SA-1100). It uses the SA-1100 address/data buses for transferring data between devices. The two primary modes of operation are as follows:

- For SA-1101 register reads and writes
  The SA-1100 addresses the SA-1101 like a Flash ROM.
- For transferring data directly
  The SA-1101 reads the SA-1100 main memory.
- For "snooping" SA-1100 writes to a region of main memory allocated for video frame buffer
  The SA-1101 writes the SA-1100 main memory.

3.1 Signal Description

This section lists the signal descriptions for the interfaces between the SA-1101 and the SA-1100.

3.1.1 SA-1100 Memory Interface

Table 3-1 describes the signals to the external interface to the SA-1100. An “X” under column header From or DRAM indicates the modes(s) of operation for which each signal is active.

Table 3-1. SA-1100 Memory Interface Signals

Signal	From	DRAM	Description
[11:0]A	X	X	12-bit address bus
[31:0]D	X	X	32-bit data bus
nCS	X		Chip select (dedicated to SA-1101 access), asserted low
nCS	X		Chip select (dedicated to SA-1101 access), asserted low
nCS	X		Chip select (dedicated to SA-1101 access), asserted low
nOE	X	X	Output enable, asserted low for reads
MBREQ		X	Bus master request from the SA-1101 to the SA-1100
MBGNT		X	Bus master grant from the SA-1100 to the SA-1101
[1:0]nRAS		X	Row address strobe to DRAMs (one per bank)
[3:0]nCAS		X	Column address strobe to DRAMs (per-byte assignment)

3.1.2 AMBA System Bus Interface (ASB)

AMBA System Bus (ASB) includes the capability of operating in both ASB master and ASB slave modes. For register accesses, the SMC must be bus master. For USB transfers to/from the SA-1101 DRAM, the SMC is a bus slave. Table 3-2 describes the ASB signals. The following signal-type abbreviations are used in Table 3-2: input (I), output (O), bidirectional (I/O), and tristate (tri).

Table 3-2. ASB Interface Signals

Signal	Master Mode	Slave Mode	Type	Description	Notes
BCLK	X	X	I	36-MHz ASB clock	
BnRES	X	X	I	ASB reset, asserted low	
[25:0]BA	X	X	I/O	26-bit ASB address	1
[31:0]BD	X	X	I/O	32-bit data	
BWAIT	X	X	I/O	Wait, asserted high	
BWRITE	X	X	I/O	Write high, read low	
[1:0]BSIZE	X	X	I/O	00 = Byte 01 = Halfword 10 = Word (32 bits)	2
DSELSMC	X		I	Target select, from decoder	
AREQ	X		O	Master request bus ownership	
AGNT	X		I	Bus grant (from ASB arbiter)	
[1:0]BTRAN	X	X	I/O	Transfer type	3
BLOK	X		O (tri)	Lock request – drive only	4
BERROR	X		O (tri)		4
BLAST	X		O (tri)		4

NOTES:
1. For register accesses, only 12 bits of address are used. The two ASB address LSBs [1:0] are extended with zeroes to make a word-aligned byte address.
2. All register accesses have a size of 32 bits, so BSIZE = 10 when SMC is bus master. As a slave, it responds to all legal encodings of the field.
3. Normally BTRAN is only driven by a master (received by the ASB address decoder) and not received by a slave. The SMC in slave mode monitors BTRAN to know when the end of a transfer (possibly made up of several simple transfers) has taken place.
4. BLOK, BERROR, and BLAST are not used by the SMC in either mode, but must be driven to prevent the signal from floating when SMC is bus master.

3.1.3 Update FIFO

When operating in "snoop" mode, SA-1101 captures SA-1100 writes to an area of memory designated as video frame buffer. All access information – address, data, and byte mask – are sent to the update FIFO. The update FIFO acts as temporary storage, passing frame buffer update information to the SA-1101's local video frame buffer (controlled by the VMC block). The FIFO buffers any bandwidth differential between the rate of SA-1100 DRAM writes and that of the SA-1101 to its own local frame buffer. Table 3-3 describes the signals in the update FIFO interface.

Table 3-3. Update FIFO Interface Signals

Signal	Description
FifoWr	Write strobe to FIFO
[18:0]SnAddr	19-bit address (upper bits stripped off)
[31:0]VDataS	32-bit data
[3:0]SnByteEn	4-bit byte mask

3.1.4 VGA Controller

In unified display memory video mode, there is no local frame buffer memory and the SA-1101 uses the video frame buffer area of SA-1100 memory for directly refreshing its video display. The SMC receives requests (VReq) from the CRT controller to supply data to refresh the display. It requests the DRAM bus and does READ bursts, sending the data to the video FIFO.

Table 3-4 describes the VGA controller interface signals. Note that the datapath [31:0]VDataS is shared with the datapath for snoop mode.

Table 3-4. VGA Controller Interface Signals

Signal	Description
VReq	Request for a read burst from SA-1101 DRAM
[19:2]VAddr	18-bit address, word-aligned (all transfers are 32-bit)
[31:0]VDataS	32-bit data (shared datapath with snoop mode)
[3:0]VBurstL	4-bit burst count
VStbS	Strobe accompanying data – signals valid data
VAckS	Signals last data word of the burst
VMode	1 = Dedicated display memory mode 0 = Unified display memory mode

3.1.5 Test Interface Controller (TIC)

For testing, the SMC is the main interface between the internal blocks of the SA-1101 and the external world. Three signals come in from the Test Interface Controller (TIC) module to control the SMC datapath when in test mode. Table 3-5 describes the Test Interface Controller signals.

Table 3-5. Test Interface Controller Signals

Signal	Description
TicoutLen	Latches data from the ASB into a 32-bit register (TIC latch)
Ticouten	Enables TIC latch output to SA-1101 data pins
Ticinen	Enables external data from the pins onto the ASB

3.1.6 System Interface

A pair of power-quality inputs is also part of the external pin interface, VDDFLT and BAT_FLT. These inputs indicate a problem with the system power source. In the SA-1101, they disable assertion of MBREQ, causing the component to give up its possession of the shared bus. Table 3-6 describes the system interface signals.

Table 3-6. System Interface Signals

Signal	Direction	Description
nRESET	I	System reset
BAT_FLT	I	BAT_FLT pin
VDD_FLT	I	VDD_FLT pin
PLLEn	O	PLL enable bit in SKCR (PLL bypass)
BCLKen	O	BCLK enable in SKCR
Sleep	O	Sleep mode in SKCR
IRefEn	0	Enable IREF current for PLL and DACs
VCOON	0	VCO bias
ScanTestEn	0	Enable scan testing I/O for USB block
ClockTestEn	0	Multiplex control to bring clocks to I/O pins

3.2 Functional Description

This section provides a functional overview of the interfaces between the SA-1101 and the SA-1100.

3.2.1 ASB and APB Register Access

This section of the SMC responds to SA-1101 register reads and writes from the SA-1100 and turns them into ASB transfers. In this mode of operation, the SA-1101 looks like a bank of Flash ROM to the SA-1100. When nCS is asserted, it causes the SMC to initiate an ASB cycle as bus master. nCS must have a minimum pulse width. See Section 3.4, “AC Characteristics” on page 3-10.

3.2.2 USB Data Transfer

The SMC functions as an ASB slave for transfers initiated by the USB host controller. The USB takes control of ASB, becoming bus master, and targets an address in the range recognized by the ASB decoder as directed at the SA-1101 main memory. The logic generates MBREQ to SA-1100. The ASB is WAITed while bus control is acquired; with the return of MBGNT the Shared Memory Controller (SMC) takes over the bus and begins DRAM cycles, transferring data between the ASB and the SA-1100 memory.

3.2.3 Unified Display Memory Mode

The unified display memory mode, like USB-initiated transfers between the ASB and SA-1101 memory, requests control of the SA-1101 memory bus. It only does READ cycles from the memory, transferring them in a burst to the video FIFO for display. Data does not pass over the ASB.

3.2.4 SMC Arbitration

Both the VGA controller (when in unified display memory mode) and the USB can request DRAM access. If the Shared Memory Controller is idle, whoever requests DRAM access will get it. If both request it in the same cycle, an arbiter chooses one. The arbiter has a bias that can be controlled by software to favor either the video or USB requesters in the event of simultaneous requests.

3.2.5 DRAM Address Generation

The ASB (for USB requests) and the VGA controller (for display refresh requests) send addresses of 25 bits and 18 bits width, respectively. The video address is extended at the MSB end by 5 bits using the SMC register field [9:5] to make a 23-bit address; the lower 2 bits of the ASB address are stripped off and sent to the CAS generation logic to control byte enables. The top bit of the resulting 23-bit value is then sent to the RAS generation logic to control which (1 of 2) physical banks of DRAM are addressed. The remaining 22-bit value can address DRAM banks of up to 16 Mbyte (using eight 4Mx4 or two 4Mx16 DRAMs).

It is important for the SA-1101 to map its memory in the same way that SA-1100 does, so a given data word is at the same location when addressed by either device. The SA-1100 specifies its DRAM configuration and address mapping scheme using two register bits. These identify the address width for the row address (9, 10, 11, or 12 bits). The column address width is not specified.

Similarly, the SA-1101 duplicates this scheme using two register bits. It also adds two bits specifying the column address width (8, 9, or 10 bits). Mapping logic maps internal ASB or video addresses (MemAdr[25:2]) to the correct row, column, and bank group. Row and column addresses are then multiplexed during the access cycle to the 12-bit external address bus. Row addresses are also checked to see if they cross a row boundary.

3.2.6 DRAM Control Signal Generation

When granted the SA-1100's bus, the SMC enables drivers for address and control signals onto the bus. These include A[11:0], nOE, nWE, nRAS[1:0], and nCAS[3:0]. The values of nOE and nWE are mutually exclusive; nOE is asserted LOW for a read, nWE for a write (DRAMs perform "early write" type of write cycle, with nWE asserted in advance of CAS).

One of two nRAS signals asserts low, based on the top bit of the address. nCAS assertions depend on who is requesting the transfer. If it is a video display READ, all CAS signals are asserted. If it is from the USB, all CAS signals are asserted for READs but depend on the byte address and transfer size for WRITEs. Table 3-7 shows which CAS signals are enabled as a function of those variables (other combinations are illegal).

Table 3-7. DRAM Control Signals

Bsize	Byte Address	nCAS[3:0]	Transfer Type
00	00	0001	Byte transfer, lane 0
00	01	0010	Byte transfer, lane 1
00	10	0100	Byte transfer, lane 2
00	11	1000	Byte transfer, lane 3
01	00	0011	Halfword transfer, lower
11	10	1100	Halfword transfer, upper
10	00	1111	Full word transfer

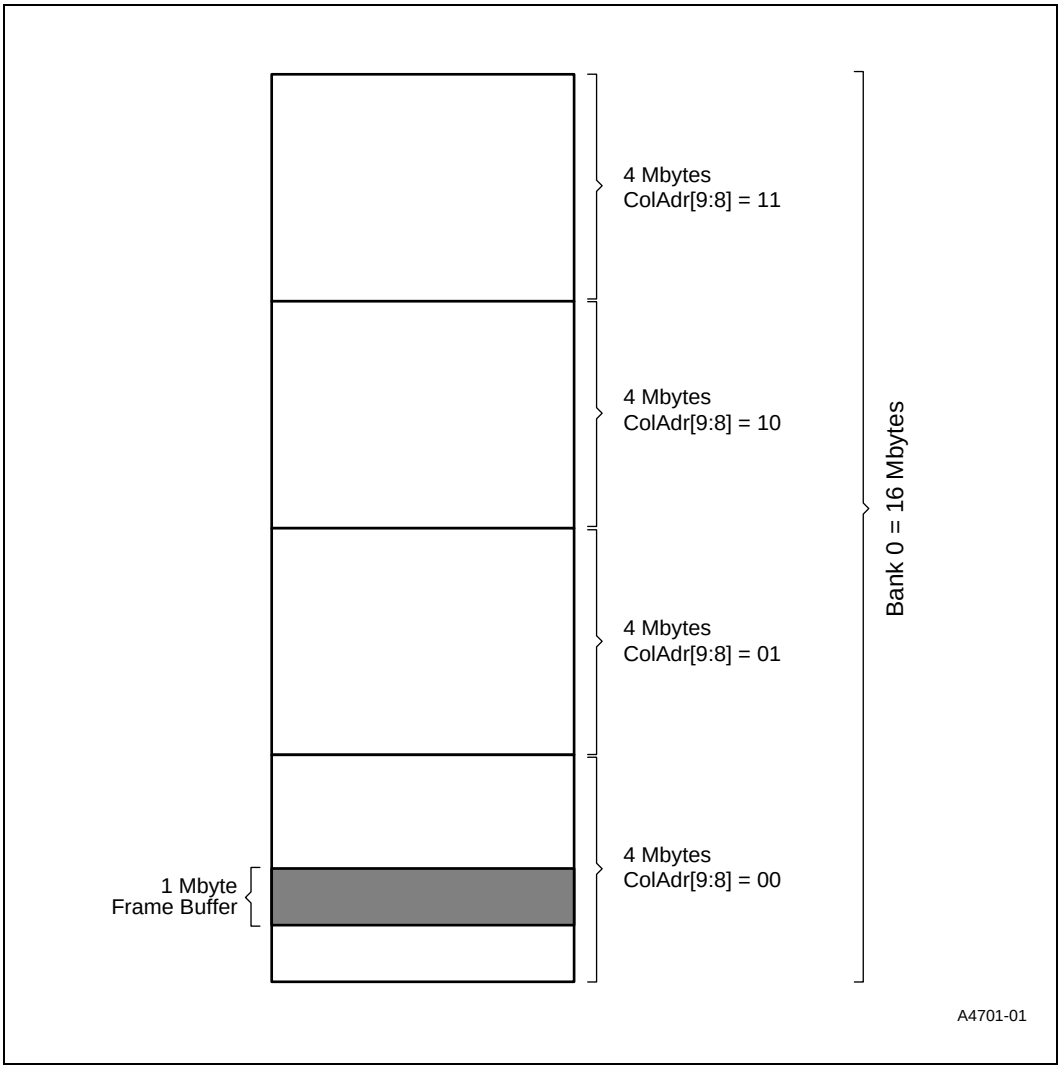
3.2.7 Dedicated Display Memory Mode – Snoop Operation

The dedicated display memory mode is a "passive" operating mode, in which WRITEs from the SA-1101 to its DRAM memory are captured if their addresses fall into a range identified as that of the video frame buffer. The data, address, and CAS signals (byte enables) are grabbed as they go by (the SA-1101 is not explicitly the target) and sent into the update FIFO. On the output of the FIFO, they are taken by the Video Memory Controller (VMC) and stored in the SA-1101's local DRAM. Data does not pass over ASB.

To make snoop-address checking faster, the logic does its in-bounds testing only on DRAM row addresses. Higher address bits are mapped to the upper three column addresses, which are only tested for a match against the three corresponding snoop register bits (26:24). For DRAMs with a 12-bit row address (DRAC = 11), the row addresses specify a 4-Mbyte space. The frame buffer can be positioned anywhere in that space (in 1-Kbyte increments) and up to 2 Mbytes in size as long as its extent does not cross the top boundary of the 4-Mbyte space.

Figure 3-1 shows an example of a 1-Mbyte frame buffer implemented with two 4Mx16 DRAMs. The particular 4-Mbyte block in which the frame buffer resides is identified by ColAdr[9:8]. The 1-Mbyte frame buffer can then be anywhere within (but cannot cross the boundary of) a 4-Mbyte block.

Figure 3-1. Snoop Addressing



The update FIFO buffers the possible mismatch between the SA-1101 writing rate and that of the local DRAM. When the FIFO nears full it can trigger an interrupt or set a register flag, polled by the SA-1101 during the drawing routine. (See the Section 3.3.3, “Snoop Register (SNPR)” on page 3-9.)

Note that “snooped” addresses are not modified as they go the Video Memory Controller. Addresses go to the dedicated display memory without change.

3.3 Programmer's Model

This section describes the registers in the interfaces between the SA-1101 and the SA-1100.

3.3.1 Control Register (SKCR)

This register provides global control over PLL operation, bus clock, sleep mode, and test mode.

Bit	Name	Function
0	PLLEn	Enables on-chip PLL. 1 = Enable 0 = Bypass
1	BCLKEn	Enables BCLK. 1 = Enable
2	Sleep	Sleep mode. 1 = Enter sleep mode
3	IRefEn	DAC Iref input enable. 1 = Enable
4	VCOON	VCO bias. 1= Enable
5	ScanTestEn	Enables scan test. 1 = Enable
6	ClockTestEn	Enables clock test. 1 = Enable

3.3.2 Shared Memory Controller Register (SMCR)

This register sets up the Shared Memory Controller for the type and size of DRAM on the SA-1100 memory bus.

Bit	Name	Function
1:0	DCAC	Number of column address bits. 00 - 8 01 - 9 10 - 10 11 - 11
3:2	DRAC	Number of row address bits. 00 - 9 01 - 10 10 - 11 11 - 12
4	—	Arbiter bias. Sets bias in favor of video or USB in case of simultaneous requests. 0 = Video 1 = USB
8:5	—	Top four bits of video memory address.

3.3.3 Snoop Register (SNPR)

This register shows information stored to define which write addresses on the SA-1100 memory bus should be snooped.

Table 3-8. Snoop Register (SNPR)

Bit	Name	Function
11:0	VFBstart	Video frame buffer starting address (in 1 Kbyte units).
22:12	VFBsize	Video frame buffer size (in 1 Kbyte units).
23	—	"Whole bank" bit. Forces column address comparators to ignore ColAdr bits 8, 9, and 10 (all values will match).
26:24	—	Column select. Match with bits 10:8 of column address.
28:27	—	Bank select. Select bank to match. 00 - Reserved 01 - Bank 0 10 - Bank 1 11 - Reserved
31	—	Enable snoop operation. 1 = Enable

3.3.4 Memory Map

Table 3-9 lists the read and write locations of the addresses in the SMC memory map.

Table 3-9. SMC Memory Map

Address	Read Location	Write Location
0x00000000	SKCR register	SKCR register
0x00140000	SMCR register	SMCR register
0x00140400	SNPR register	SNPR register

3.4 AC Characteristics

This section describes the ac characteristics of the SA-1100 interface and Shared Memory Controller.

3.4.1 Shared DRAM

Figure 3-2 shows a burst of three DRAM access.

Figure 3-2. Burst of Three DRAM Access

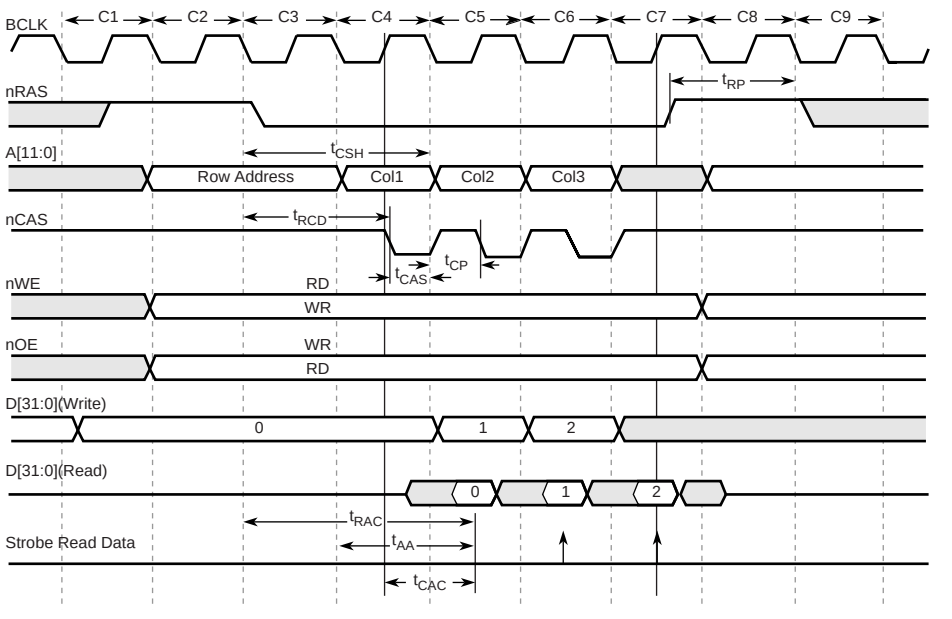


Table 3-10 lists the timing parameters for the shared DRAM. The values are based on PLL frequency of 144 MHz (BCLK = 36 MHz). DRAMs selected for use with the SA-1101 should have, for each applicable parameter, a value less than or equal to the minimum value listed.

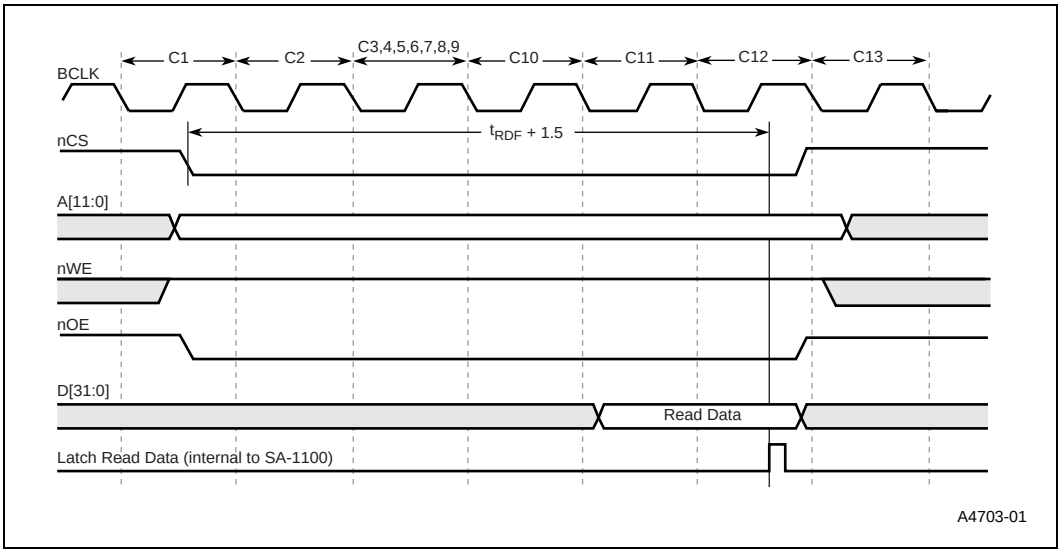
Table 3-10. Shared Memory Timing Parameters

Symbol	Parameter	Minimum	Maximum	Unit
t_{AA}	Address access time	40	44	ns
t_{CAC}	CAS access time	26	30	ns
t_{CAS}	CAS pulse width	11	16	ns
t_{CP}	CAS precharge time	11	16	ns
t_{CSH}	CAS hold after RAS	54	58	ns
t_{RAC}	RAS access time	67	72	ns
t_{RCD}	RAS-to-CAS delay	40	44	ns
t_{RP}	RAS precharge	40	44	ns

3.4.2 Register Reads

Figure 3-3 shows a register read.

Figure 3-3. Register Reads



3.4.3 Register Writes

Figure 3-4 shows a register write. Note that waveforms are not synchronous to BCLK. BCLK is shown to indicate relative duration of the signals.

Figure 3-4. Register Writes

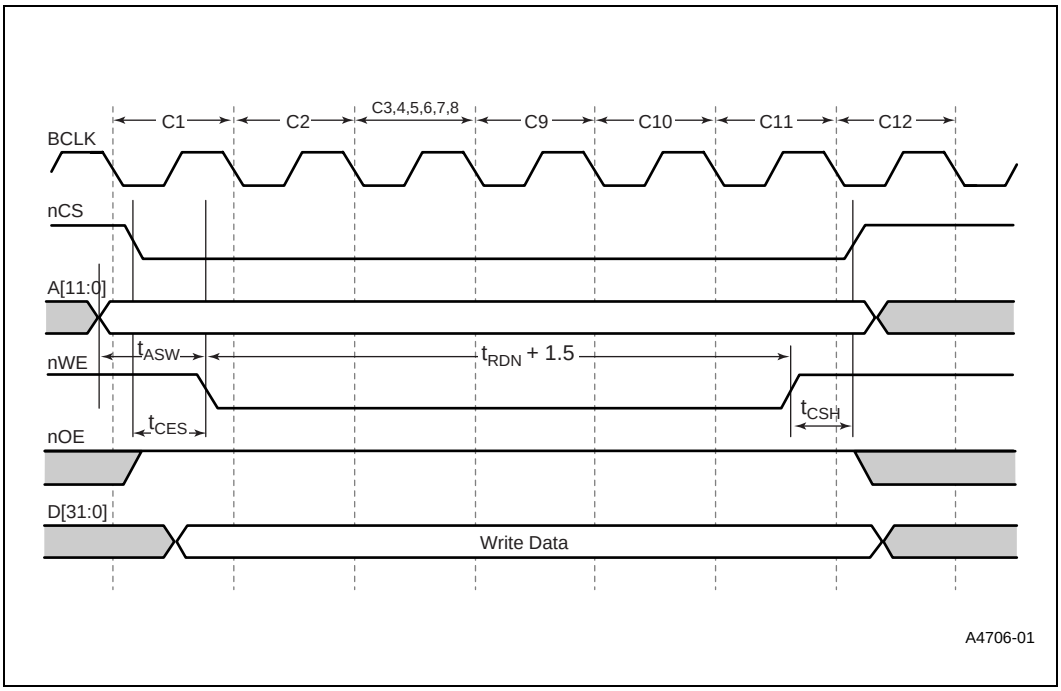


Table 3-11 lists the register read and write timing parameters.

Table 3-11. Register Read and Write Timing Parameters

Symbol	Parameter	Minimum	Maximum	Unit	Note
^{t}ASW	Address setup to nWE start	T-2	T+3	ns	1
^{t}CES	CS setup to nWE start	4T-2	4T+3	ns	1
^{t}CSH	CS hold after nWE end	2T-2	4T+3	ns	1
^{t}RDN	—	280	—		2
^{t}RDF	—	300	—		2

Note: 1. T is period in ns of the SA-1100 core (CPU) clock.
2. RDN and RDF are programmable.

Video Memory Controller 4

The Video Memory Controller (VMC) is used when the StrongARM® SA-1101 (SA-1101) operates in the dedicated display memory mode. The VMC interfaces to an external 256Kx16 or 1Mx16 DRAM, which functions as the frame buffer for a video display. Both configurations of 1Mx16 are supported: 12 row address X 8 column address, and 10 row address X 10 column address. When using the latter part, the upper address pins will still have address information on them, but they are ignored by the DRAMs.

In normal use, the VMC performs write cycles and read cycles.

- Write cycles are generated when it receives "snoop" address/data information from the Shared Memory Controller (SMC), coming through the update FIFO. This keeps the local DRAM frame buffer current with the latest data written into the StrongARM® SA-1100 (SA-1100) frame buffer in main memory.
- Read cycles are generated to refresh the video display screen. Read requests come from the VGA controller. When the video FIFO nears empty, it requests a burst of read cycles to refill it.

The VMC's interface to DRAM is 16 bits wide. Internally the SA-1101's standard data unit is 32 bits wide, so all VMC accesses use an even number of CAS cycles in a burst (2, 4, 6, ...) to read and write data, 32 bits at a time.

4.1 Signal Description

This section lists the signal descriptions for the interfaces to the Video Memory Controller.

4.1.1 External Interface to DRAM

Table 4-1 lists the signals of the external interface to DRAM.

Table 4-1. External Interface to DRAM Signals

Name	Direction	Description
[15:0]FBD	I/O	16-bit bidirectional data.
[11:0]FBA	O	12-bit address (multiplexed; can address up to 1Mx16 DRAM).
nFB_WE	O	Write enable. Asserted low.
nFB_UCAS	O	Upper CAS (column address strobe); also functions as byte enable for data bits [15:8].
nFB_LCAS	O	Lower CAS (column address strobe); also functions as byte enable for data bits [7:0].
nFB_RAS	O	Row address strobe.

Note: DRAMs include an additional input that is not driven by the SA-1101, OE (output enable, asserted LOW). In a system using the SA-1101, this DRAM pin can be tied low on the module. The DRAM will only enable its output when the appropriate sequence of RAS and CAS signals has been applied; when the VMC does a write to DRAM, it is of the "early write" type that ensures the DRAM keeps its outputs disabled, avoiding any possibility of bus contention.

4.1.1.1 AMBA Peripheral Bus Interface (APB)

The AMBA Peripheral Bus (APB) is used for register reading and writing, and for reading DRAM data back to the system in test mode. Table 4-2 lists the signals in the APB bus interface.

Table 4-2. APB Bus Interface Signals

Name	Direction	Description
BCLK	I	36-MHz ASB clock.
BnRES	I	ASB reset. Asserted low.
PSELvmc	I	SEL for APB.
[8:2]PA	I	7-bit address from APB.
[31:0]PD	I/O	32-bit bidirectional APB dat.
PSTB	I	APB strobe.
PWRITE	I	APB write.

4.1.1.2 Update FIFO

The VMC reads 55-bit data words (made up of data, address, and byte-enable fields) from the FIFO. The read is signalled by assertion of RdFifo. At the end of the read, the FIFO controller increments its read address and presents the next data word to the VMC. Table 4-3 lists the signals in the update FIFO interface.

Table 4-3. Update FIFO Interface Signals

Name	Direction	Description
[31:0]FifoDat	I	32-bit data word.
[18:0]FifoAdr	I	19-bit address.
[3:0]FifoBytEn	I	4-bit byte mask.
DataRdy	I	Signals valid data available from FIFO.
FifoGTn	I	FIFO fullness level above programmable level "n".
FifoGT8	I	FIFO fullness more than 7.
FifoEmpty	I	Update FIFO is empty.
RdFifo	O	Increment the FIFO RAM address (read port).

4.1.1.3 VGA Controller

The VGA controller requests display refresh reads from the VMC. The request includes the starting address and the number of words to be read. Since the VMC only has a 16-bit interface, there are two read cycles for each word sent back to the VGA controller. Returned data is accompanied by a strobe to indicate its validity. With the last data word, VAckD is also asserted to indicate the end of the transfer. Table 4-4 lists the signals in the VGA controller interface.

Table 4-4. VGA Controller Interface Signals

Name	Direction	Description
VReq	I	Request from video controller.
[3:0]VBurstL	I	Burst length. 0 = Send 1 word F = Send 16 words
[19:2]VAddr	I	18-bit starting address, word-aligned.
VAckD	O	Last word of transfer.
VStbD	O	Valid data on this cycle.
[31:0]VDataD	O	32 bit video data.
VMode	I	Video mode (shared or dedicated display memory); disable VMC if in unified display memory mode. 0 = Unified display memory mode 1 = Dedicated display memory mode

4.2 Functional Description

This section provides a functional overview of the interfaces to the Video Memory Controller.

4.2.1 Data Transfer Arbitration

At the heart of the VMC is a state machine that performs three types of cycles: read, write, and refresh. The read and write use identical states and only differ in the assertion of nFB_WE (write enable) to the DRAMs for writes.

The four events that can start up the state machine and initiate a data transfer are as follows:

- VReq
  Read request from the video controller.
- FifoGT8 (or FifoGTn)
  Write request from the FIFO.
- Test read
  Read request from the SA-1100, via the ASB and APB.
- RefReq
  Memory refresh request from refresh logic.

The first three of these events are handled by a simpler arbiter. Table 4-5 shows the priorities used by the arbiter.

The FIFO requests a burst write by asserting FifoGT8, which is the lowest priority request, indicating that it has eight or more words stored (this takes advantage of the bandwidth benefits of burst operation by waiting until it has eight words in it). Once underway, the burst continues until the FIFO is empty or a display refresh request (which is a higher priority) comes from the VGA controller.

Display refresh has higher priority than taking update data from FIFO in most circumstances. However, the SA-1100 may write extended bursts (snooped by the SA-1101) that exceed the VMC bandwidth available between screen refreshes. The FIFO therefore fills deeper and at some point exceeds depth "n", a programmable threshold in the update FIFO controller. The FIFO asserts FifoGTn to VMC. This has higher priority than display update requests; any display update that is underway is allowed to complete, but further requests from the video controller are ignored as long as the "n" threshold is exceeded. The VMC bursts for 32 words or until the FIFO is empty, whichever comes first.

Table 4-5. Request Arbiter Priorities

Priority	Request Type
Highest	Test access request
	Update FIFO fullness greater than n
	CRT controller display refresh request
Lowest	Update FIFO fullness greater than eight

Note: Test access is the highest priority. Test access only should be attempted when other sources of requests (except refresh) are inactive. Refresh cycles occasionally can corrupt returned data from test reads.

4.2.2 FIFO Emptying

To take advantage of DRAM burst bandwidth, the VMC does not begin transferring update (snoop) data from the update FIFO to DRAM until there are at least eight words in the FIFO (signal FifoGT8 asserted). This creates the possible situation where the FIFO was emptied by previous writes, and then the SA-1100 only writes a few pixels. To ensure that this data is transferred to the dedicated frame buffer, a timer is provided which resets to zero on any DRAM write.

When the FIFO is non-empty (but level remains less than eight), it increments the counter on every clock. When the counter value reaches a programmable value, a "timeout" asserts and kicks off a DRAM write by the VMC state machine, the same as if FifoGT8 had asserted.

4.2.3 Memory Refresh

Responding to memory refresh requests is built into the state machine, bypassing the arbiter priority logic. If a refresh request is asserted, the state machine will do the refresh whenever it returns to the idle state. Therefore refresh cannot interrupt a burst already underway, but once any burst completes the refresh will take place. Postponing a refresh has no effect on when the next request is asserted; requests are triggered by the refresh counter which has no dependencies on VMC activity.

The interval between memory refresh requests is programmable to accommodate a variety of DRAM types, including "L" versions which have extra-long refresh intervals to reduce power. Refresh is "distributed"; it refreshes one DRAM row per request, rather than doing a burst of refreshes per request.

4.2.4 Burst Transfers

Reads or writes requested by the video controller or update FIFO are normally burst transfers. The state machine will loop on a CAS state to read or write 16 bits every cycle. For writes, four bits from the update FIFO indicate which bytes are to be written. Logic in the VMC distributes the four-byte mask to the upper and lower CAS signals, over two cycles, to enable arbitrary byte masking of a 32-bit word.

If the transfer crosses a DRAM row boundary, the VMC halts the transfer, precharges the DRAM, and reasserts RAS and the new row access to resume the transfer.

4.2.5 Sleep

In sleep mode or doze mode the frame buffer contents are preserved by placing it into self refresh mode. When the SA-1101 is going into sleep, the VMC is issued with a request to place the frame buffer memory into self refresh. Once sleep state has been reached, the RAS and CAS (nFB_LCAS, nFB_RAS, nFB_UCAS) signals are held active (low) for the duration of sleep. See Section 2.4.3, “Sleep Mode” on page 2-9.

4.2.6 Reset

The SA-1101 powers-up/resets with its refresh disabled. Most DRAM require an initial pause of 200 microseconds after power-up followed by eight refresh cycles before proper device operation is achieved. The SA-1100 turns on refresh after an appropriate period of waiting; by waiting an additional period after enabling refresh and before attempting any accesses to the SA-1101 DRAM, the SA-1100 ensures that at least eight refresh cycles have taken place.

4.3 Programmer's Model

This section describes the registers in the Video Memory Controller.

4.3.1 Configuration Register (VMCCR)

This register configures the Video Memory Controller for the type of DRAM installed for dedicated frame buffer. It also controls some sleep state and test functions.

Bit	Name	Reset	Function
0	RefreshEn	0x0	Enable memory refresh. 1 = Enable
1	Config	0x0	DRAM size. 0 = 256K x 16 1 = 1M x 16
2	—	0x0	Reserved.
4:3	RefPeriod	0x0	Refresh period. 00 = 14 microseconds (approximately) 01 = 28 microseconds 10 = 112 microseconds 11 = 225 microseconds
8:5	StaleDataWait	0x8	Stale FIFO data timeout counter. 0000 = 0 clk cycles 0001 = 8 clk cycles 0010 = 16 clk cycles up to 1111 = 120 clk cycles
9	SleepState	0x0	State of frame buffer interface pins in sleep mode. See Table 16 for more information.
10	RefTest	0x0	1 = Test refresh by splitting refresh counter into two counters: the refresh low counter and the refresh high counter 0 = Refresh counter normal operation
16:11	RefLow	0x0	Refresh low counter.
23:17	RefHigh	0x0	Refresh high counter.
30:24	SDTCTest	0x0	Stale data timeout counter.
31	ForceSelfRefresh	0x0	1 = Frame buffer to self refresh; force self refresh. 0 = Frame buffer to normal operation; normal DRAM operation.

Table 4-6 describes the SleepState bit encoding.

Table 4-6. SleepState Bit Encoding

Signal	SleepState = 0	SleepState = 1
FB_A[11:0]	Tristate	High
FB_D[15:0]	Tristate	High
nFB_WE	Tristate	High

4.3.2 Memory Map

Table 4-7 lists the read and write locations in the VMC memory map.

Table 4-7. VMC Memory Map

Address	Read Location	Write Location
0x00100000	VMCCR register	VMCCR register

4.4 AC Characteristics

This section describes the ac characteristics of the Video Memory Controller.

4.4.1 Video DRAM Read and Write

Figure 4-1 shows a burst of four DRAM access.

Figure 4-1. Burst of Four DRAM Access

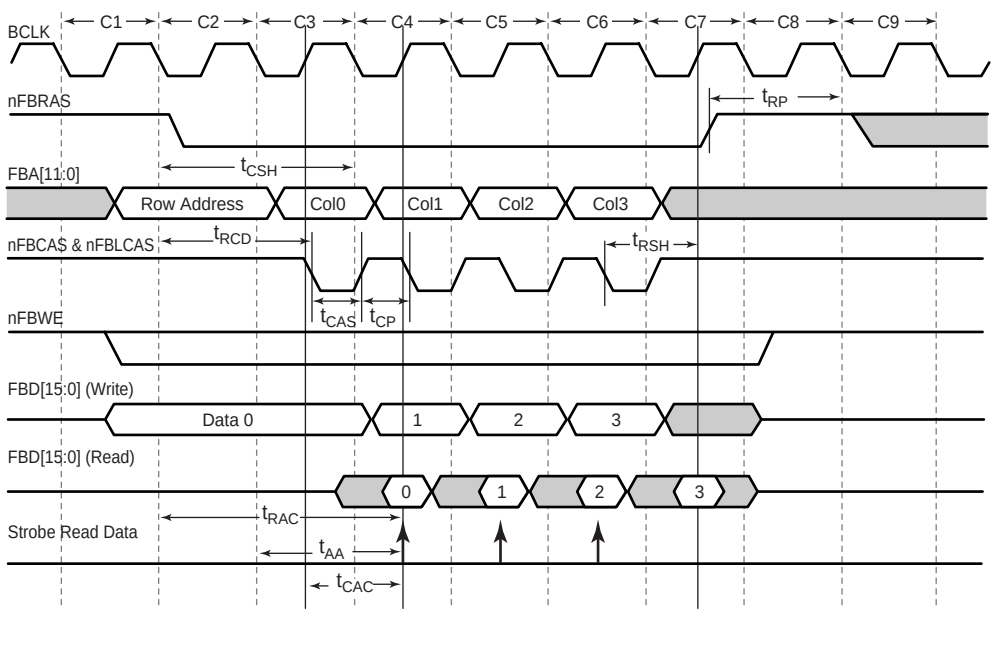


Table 4-8 lists the timing parameters for the video DRAM read and write.

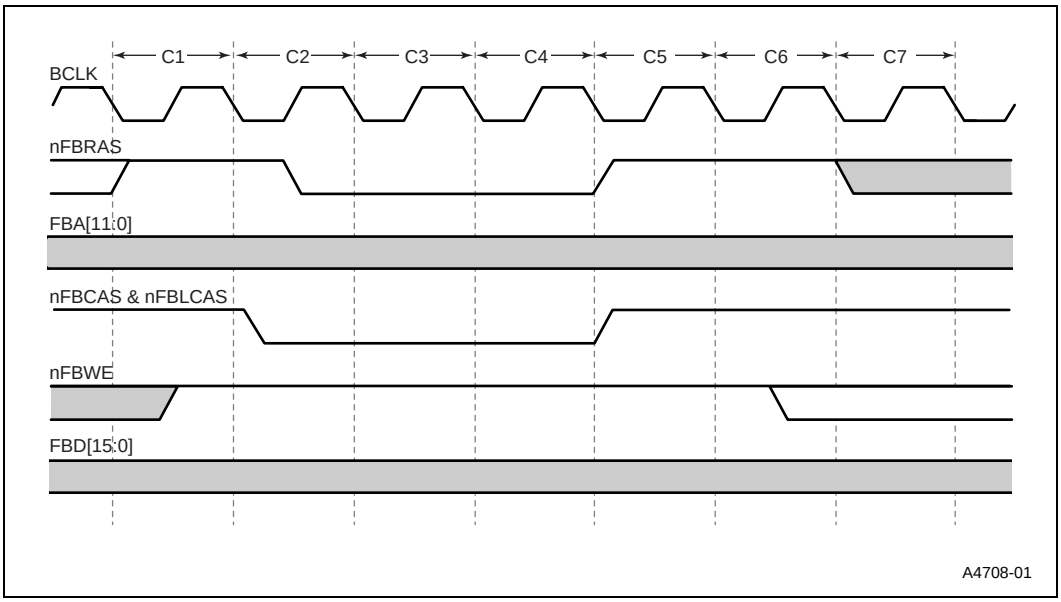
Table 4-8. Video Memory Timing Parameters

Symbol	Parameter	Minimum	Maximum	Unit
t_{AA}	Address access time	41	41	ns
t_{CAC}	CAS access time	27	27	ns
t_{CAS}	CAS pulse length	13	13	ns
t_{CP}	CAS precharge	13	13	ns
t_{CSH}	CAS hold after RAS start	55	55	ns
t_{RAC}	RAS access time	69	69	ns
t_{RCD}	RAS-to-CAS delay	42	42	ns
t_{RP}	RAS precharge	42	42	ns
t_{RSH}	RAS hold after CAS end	26	30	ns

4.4.2 Memory Refresh

Figure 4-2 shows a CAS before RAS refresh.

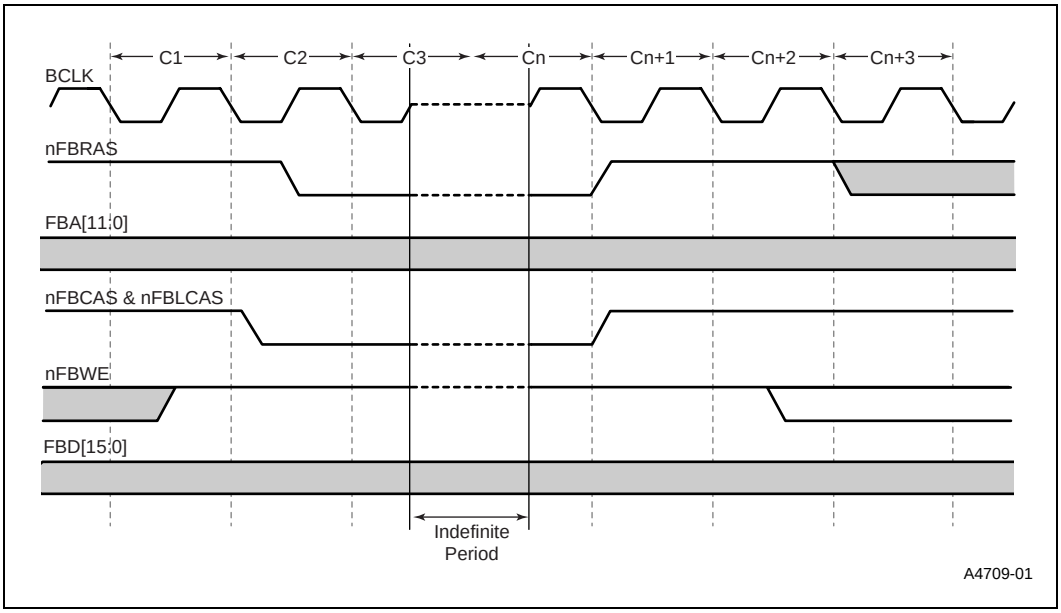
Figure 4-2. CAS Before RAS Refresh



4.4.3 Self-Refresh

Figure 4-31 shows a self-refresh.

Figure 4-3. Self-Refresh



4.5 Test

There is a test mode of operation for testing the VMC and its attached DRAM. The SA-1101's DRAM cannot be accessed by SA-1100 like its own DRAM because the access delays would be too long on read cycles. An alternate mode of access is provided, using the Flash ROM cycle types SA-1100 uses to access the SA-1101 registers. Each DRAM access is in two steps:

1. Load VMC address register (VMCAR) with the desired address to access. The VMC kicks off a read cycle and stores the returned data.
2. Read back the data from the VMC data register (VMCDR).

Testing is normally conducted using the "snoop" mode of operation to write the DRAM, and then reading it back via the two-step register read back technique. Table 4-9 lists the read and write locations in the test memory map for the VMC.

"Snoop" capture and display refresh reads should be disabled when doing test reads. Test read data occasionally can be corrupted by refresh cycles. For reliable data, reads can be repeated if a refresh cycle is enabled, or a refresh cycle momentarily can be disabled before a test read.

Table 4-9. Test Memory Map for the VMC

Address	Read Location	Write Location
0x00101000	VMCAR register	VMCAR register
0x00101400	VMCDR register	Reserved

Update FIFO 5

The function of the update FIFO is to buffer snooped write data captured at the StrongARM® SA-1100 (SA-1100) DRAM interface before passing it on to the Video Memory Controller (VMC). The FIFO contains 64 words to accommodate very fast bursts that may temporarily exceed the rate at which the VMC can write into its own local DRAM. See Section 4.2, “Functional Description” on page 4-3 for details of how the update FIFO is used by the VMC.

5.1 Signal Description

This section lists the signal descriptions for the interfaces to the update FIFO.

5.1.1 AMBA Peripheral Bus

The AMBA Peripheral Bus (APB) is used for register reading and writing, and for reading FIFO data back to the system in test mode. To keep layout area lower, only eight bits of the APB datapath connect to FIFO logic. Table 5-1 lists the signals of the APB interface.

Table 5-1. APB Interface Signals

Name	Type	Description
BCLK	I	36-MHz ASB clock
BnRES	I	ASB reset, asserted low
PSELvmc	I	SEL for APB
[3:2]PA	I	2-bit address from APB
[7:0]PD	I/O	8-bit bidirectional APB data
PSTB	I	APB strobe
PWRITE	I	APB write

5.1.2 Video Memory Controller Interface

Table 5-2 lists the signals in the Video Memory Controller Interface.

Table 5-2. VMC Interface Signals

Name	Type	Description
[31:0]FifoDat	O	32-bit data word
[18:0]FifoAdr	O	19-bit address
[3:0]FifoBytEn	O	4-bit byte mask
DataRdy	O	Signals valid data available from FIFO
FifoGTn	O	FIFO fullness level above programmable level "n"
FifoGT8	O	FIFO fullness more than 7
FifoEmpty	O	Update FIFO is empty.
RdFifo	I	Increment the FIFO RAM address (read port)

Note: Output **FifoGTn** also goes to the interrupt controller.

5.1.3 Shared Memory Controller Interface

Table 5-3 lists the signals in the Shared Memory Controller Interface.

Table 5-3. SMC Interface Signals

Name	Type	Description
FifoWr	I	Positive CAS pulse for writing snoop info into FIFO
[31:0]VDataS	I	32-bit data, passed through to FIFO RAM
[18:0]SnAddr	I	19-bit address, passed through to FIFO RAM
[3:0]SnByteEn	I	4-bit byte mask, passed through to FIFO RAM

5.1.4 Update FIFO Interface

Table 5-4 lists the signals in the Update FIFO Interface.

Table 5-4. Update FIFO Interface Signals

Name	Type	Description
[55:0]UFifoDin	O	Combined address, data, and byte mask to FIFO RAM
[55:0]UFifoDout	I	Output data from FIFO RAM's read port
[5:0]UFifoWrA	O	6-bit FIFO RAM write address
[5:0]UFifoRdA	O	6-bit FIFO RAM read address
UFifoWr	O	Write pulse to FIFO RAM

5.2 Programmer's Model

This section describes the registers in the update FIFO.

5.2.1 Update FIFO Control Register (UFCR)

This register sets the data level at which the FifoGTn flag (status bit) is set. An interrupt is generated if the interrupt is enabled.

Bit	Name	Function
6:0	FifoThreshhold	Level for FifoGTn flag
7	—	Reserved

Note: UFCR reset state is 0x20.

5.2.2 Update FIFO Status Register (UFSR)

This register shows the status of the Update FIFO.

Bit	Name	Function
0	FifoGTnFlag	FifoGTn flag 1 = Set
6:1	—	Reserved
7	FifoEmpty	FIFO is empty 1 = Empty

5.2.3 Memory Map

Table 5-5 lists the read and write locations in the memory map of the two registers in the update FIFO.

Table 5-5. UFCR and UFSR Memory Map

Address	Read Location	Write Location
0x00120000	UFCR register	UFCR register
0x00120400	UFSR register	Reserved

5.3 Test

There is a test mode of operation for testing the update FIFO. Testing is conducted using the two registers in the update FIFO.

5.3.1 FIFO Level Register

This register shows how many data words are stored in the Update FIFO.

Bit	Name	Function
6:0	FifoFullness	FIFO level.

5.3.2 FIFO Data Register

This register contains the lowest eight bits of the bottom (nearest the output port) word in the Update FIFO. It can be used to verify that the datapath from the SA-1100 to the FIFO is working and that the snoop function is programmed correctly. Note that each register-type READ from the FIFO “retires” that word and moves the READ pointer to the next word, so it can not be subsequently written to the frame buffer.

Bit	Name	Function
7:0	FifoData	FIFO data. Lower byte of data word in FIFO read port.

The read address increments each time a test read is made from the FIFO data register. Test reads cannot be mixed with normal operation.

Each test read will decrement the FIFO level by one. It is recommended to not decrement the level below zero, as the value will wrap to 63 and initiate burst writes of invalid data to the frame buffer DRAM.

5.4 Memory Map

Table 5-6 list the read and write locations in the test memory map for the update FIFO.

Table 5-6. Test Memory Map for the Update FIFO

Address	Read Location	Write Location
0x00120800	UFLVLR register	Reserved
0x00120C00	UFDR register	Reserved

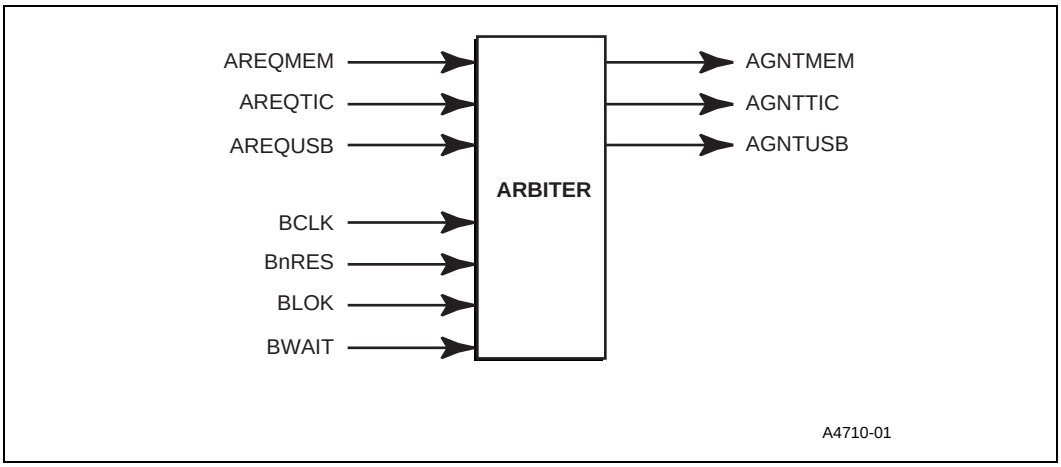
Arbiter 6

The Advanced Microcontroller Bus Architecture (AMBA) bus specification is a multimaster bus standard. As a result, a bus arbiter is needed to ensure that only one bus master has access to the bus at any particular point in time. Each bus master can request the bus; the arbiter decides which has the highest priority and issues a grant signal accordingly.

6.1 Block Diagram

Figure 6-1 shows a block diagram of the arbiter.

Figure 6-1. Arbiter Block Diagram



6.2 Signal Description

The following table describes the signals for the arbiter.

Name	Type	Description
AGNTMEM	O	Grant signal to the Shared Memory Controller. When HIGH, this signal indicates that the bus master is currently the highest priority master requesting the bus. This signal changes during the LOW phase of BCLK and remains valid through the HIGH phase.
AGNTTIC	O	Grant signal to the Test Interface Controller. When HIGH, this signal indicates that the Test Interface Controller is currently the highest priority master requesting the bus. This signal changes during the LOW phase of BCLK and remains valid through the HIGH phase.
AGNTUSB	O	Grant signal to bus master USB. When HIGH, this signal indicates that this bus master is currently the highest priority master requesting the bus. This signal changes during the LOW phase of BCLK and remains valid through the HIGH phase.
AREQMEM	I	Request from the Shared Memory Controller indicating that it requires the bus. This signal must be set up to the falling edge of BCLK.
AREQTIC	I	Request from the Test Interface Controller indicating that it requires the bus. This signal must be set up to the falling edge of BCLK.
AREQUSB	I	Request from a bus master USB indicating that the master requires the bus. This signal must be set up to the falling edge of BCLK.
BCLK	I	System (bus) clock. This clock times all bus transfers. The clock has two distinct phases: phase 1 in which BCLK is LOW, and phase 2 in which BCLK is HIGH.
BnRES	I	This signal is active LOW and indicates the reset status of the bus. It is driven by the reset controller.
BLOK	I	A shared bus lock signal driven by the currently granted bus master is used to indicate that the current transfer is indivisible from the following transfer, and that no other master should be granted the bus.
BWAIT	I	This signal is driven by the selected bus slave to indicate whether the current transfer may complete. If BWAIT is HIGH, a further bus cycle is required. If BWAIT is LOW, the transfer may complete in the current bus cycle. When no bus transfer is taking place, this signal is driven by the bus decoder. This signal is driven in the LOW phase of BCLK and is valid before the rising edge of BCLK. BWAIT is used by the arbiter block to determine when a turnaround cycle is happening on the bus.

6.3 Functional Description

This section provides a functional overview of the arbiter.

6.3.1 Request/Grant

To gain access to the AMBA System Bus (ASB), a bus master must assert its request line (AREQMEM, AREQTIC, AREQUSB). If the request is of a higher priority than any other asserted bus request lines, then the bus request line can be granted. See Table 6-1 for bus master priorities. Before AGNT is asserted, the BLOK signal must be clear indicating that no indivisible transfers are taking place.

The arbiter produces the grant signal on a BCLK cycle-by-cycle basis. However, the bus mastership does not change every cycle, and a new bus master only finally gains the bus when:

- AGNT is HIGH, indicating that the bus master is currently the highest priority.
- BWAIT is LOW, indicating that the current transfer has completed.

Each bus master (Test Interface Controller (TIC), Shared Memory Controller (SMC), and USB interface) monitors these signals to determine when it is granted use of the bus in response to its requests. Table 6-1 lists the priorities for the bus master.

Table 6-1. Bus Master Priorities

Priority	Bus Master
Highest	Test interface controller
	Shared Memory Controller
Lowest	USB interface

6.3.2 Reset

During reset, when BnRES is LOW, the arbiter grants use of the bus to the Test Interface Controller and holds all other grant signals inactive.

6.4 Programmer's Model

There are no software configurable items in the arbiter.

6.5 Test

There are no test registers in the arbiter.

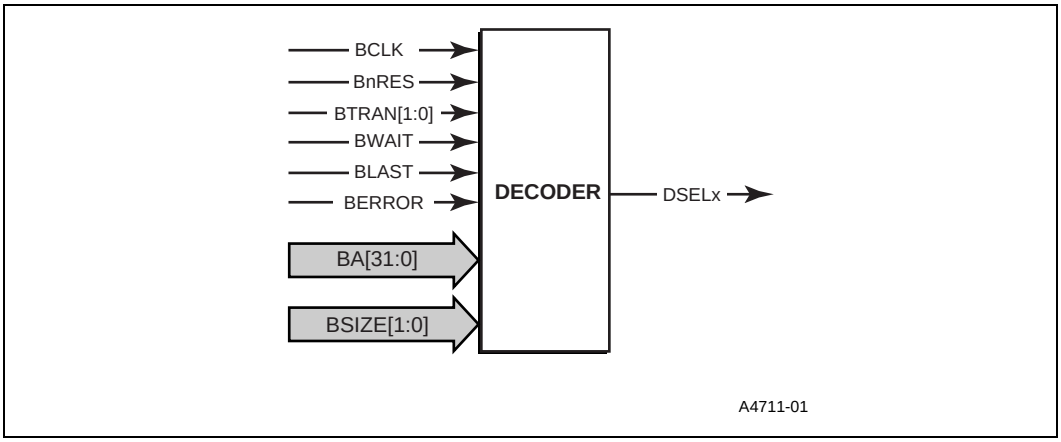
Decoder 7

The decoder performs the following three functions:

- Generates the slave select signals (DSELx) for each of the bus slaves, indicating that a transfer to that slave is required.
- Generates the slave response signals (BWAIT, BLAST and BERROR) during address-only transfers, when no slave is selected.
- Acts as a simple protection unit, which prevents attempts to access an illegal or protected area of the memory map.

Figure 7-1 shows a block diagram of the decoder.

Figure 7-1. Decoder Block Diagram



7.1 Signal Description

Table 7-1 describes the signals that interface to the decoder.

Name	Type	Description
BA[31:0] Address Bus	I	System address bus that is driven by the bus master. The addresses change during the HIGH phase before the transfer to which they refer, and remain valid until the last HIGH phase of the transfer.
BCLK Bus Clock	I	System (bus) clock. This clock times all bus transfers. The clock has two distinct phases: phase 1 in which BCLK is LOW, and phase 2 in which BCLK is HIGH.
BERROR Error Response	I/O	A transfer error is indicated by the selected bus slave or decoder using the BERROR signal. When BERROR is HIGH, a transfer error has occurred. When BERROR is LOW, the transfer is successful. This signal is also used in combination with the BLAST signal to indicate a bus retract operation. When no bus transfer is taking place, this signal is driven by the decoder. This signal is driven in the LOW phase of BCLK and is valid before the rising edge of BCLK.
BLAST Last Response	I/O	Signal is driven by the selected bus slave or decoder to indicate whether the current transfer should be the last of a burst sequence. When BLAST is HIGH, the next bus transfer must allow sufficient time for address decoding. When BLAST is LOW, the next transfer may continue a burst sequence. This signal is also used in combination with the BERROR signal to indicate a bus retract operation. When no bus transfer is taking place, this signal is driven by the decoder. This signal is driven in the LOW phase of BCLK and is valid before the rising edge of BCLK.
BnRES Reset Status	I	Active LOW signal indicates the reset status of the bus and is driven by the reset controller.
BTRAN[1:0] Transfer Type	I	Signals indicate the type of the next transaction, which may be address-only, nonsequential, or sequential. These signals are valid during the phase HIGH before the transfer to which they refer.
BWAIT Wait Response	I/O	Signal is driven by the selected bus slave or decoder to indicate whether the current transfer may complete. If BWAIT is HIGH, a further bus cycle is required. If BWAIT is LOW, the transfer may complete in the current bus cycle. When no bus transfer is taking place, this signal is driven by the bus decoder (this block). This signal is driven in the LOW phase of BCLK and is valid before the rising edge of BCLK.
DSELx Slave Select	O	Signal indicates that a slave device is selected and a data transfer is required. There is a DSELx signal for each bus slave. This signal becomes valid during the phase HIGH before the data transfer is required, and remains active until the last phase HIGH of the transfer.
BSIZE[1:0] Transfer Size	I	Signals indicate the size of the transfer which may be byte, halfword, or word. They are only needed by the decoder with decode cycles implementation. They have the same timing as the system address bus.

7.2 Functional Description

At the start of every transfer on the bus, the decoder can perform a number of actions. The decoder is able to determine when a transfer is about to start by examining the BWAIT signal, which will be LOW when the previous transfer is completing.

The actions the decoder takes depend on the type of transfer:

- Nonsequential transfer

  The decoder inserts a single decode cycle to allow the address bus to stabilize and the new address to be decoded. For the first cycle, the decoder drives BWAIT HIGH and negates all DSELx signals. In the second cycle, the decoder asserts the appropriate DSELx, and the selected slave becomes responsible for driving the slave transfer response.

- Sequential transfer with BLAST LOW

  The decoder drives the appropriate DSELx signal, and the selected slave is responsible for driving the slave transfer response.

- Sequential transfer with BLAST HIGH

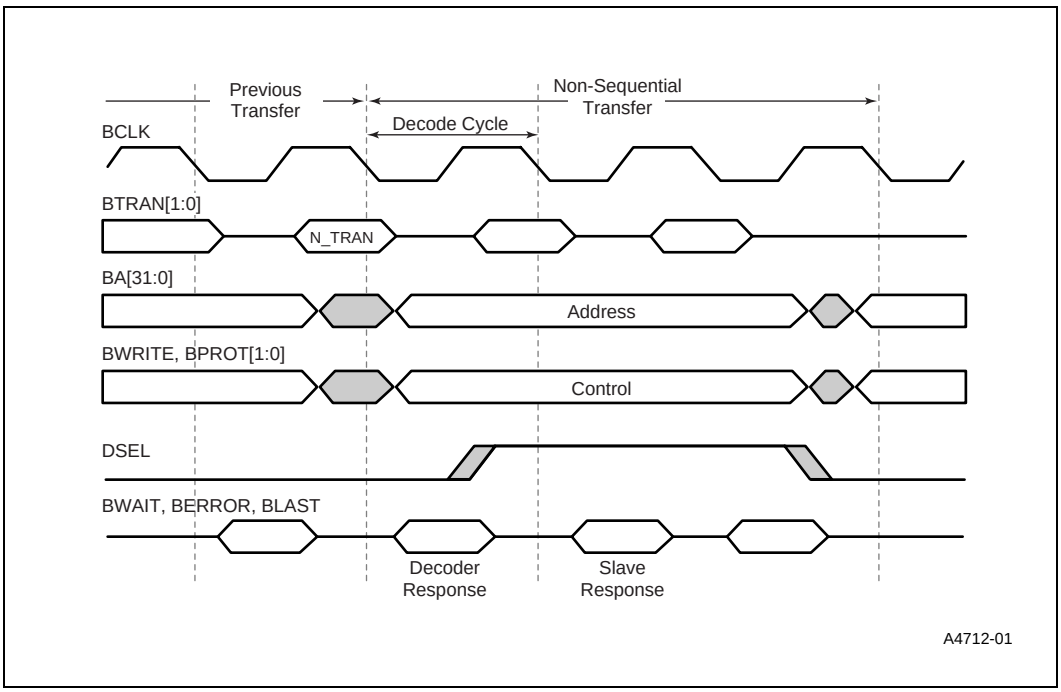
  The decoder treats this case in the same way as a nonsequential transfer.

- Address-only transfer

  The decoder does not generate any DSELx signals and drives a slave transfer response of BWAIT LOW.

Figure 7-2 illustrates a decode cyle.

Figure 7-2. Decoder with Decode Cycles



7.2.1 Slave Response

The decoder control block monitors bus activity and determines when a transfer is required to a slave device. When a transfer is required, a signal is asserted to the address decoder block, which in turn generates a slave select signal depending on the address on the bus.

When no slave is selected, the decoder must provide the slave response signals (BWAIT, BLAST and BERROR) in order for the bus to remain synchronized. The decoder will drive these signals during the address-only cycle (as indicated by the BTRAN[1:0] signals, which are driven by the bus master), and the decode cycle, which occurs at the start of every nonsequential transfer and may also be requested by a slave device using the BLAST signal.

Table 7-1 shows the response combinations of the decoder with decode cycles.

Table 7-1. Decoder with Decode Cycles Response Combinations

Condition	BWAIT	BLAST	BERROR
Address-only cycle	0	0	0
Error	0	0	1
Decode cycle	1	0	0

When a slave is selected it must provide a response on the **BWAIT**, **BERROR** and **BLAST** signals.

7.2.2 Decode Control State Machine

The decoder state machine is clocked off the falling edge of the bus clock, BCLK. Therefore it is necessary for the decoder to use latched versions of the BWAIT and BLAST signals. WAIT + RETNEXT is equivalent to LBWAIT.

The decoder will drive an address-only cycle response when in the ADDRONLY state and a decode cycle response when in the DECODE state. The decoder will not drive the slave response signals in the SLAVESEL state; this is the responsibility of the selected slave.

The decode control state machine consists of the following states:

- ADDRONLY

  No transfer. This state is entered when no data transfer is occurring on the bus. This state is also entered during a reset condition.

  The ADDRONLY state is only exited when a transfer is about to occur, as indicated by the BTRAN signal. If the transfer is nonsequential, the next state will be DECODE, but if the transfer is sequential, the next state is SLAVESEL.

- DECODE

  Decode cycle. The DECODE state is only entered for a single cycle. Assuming that there is no reset condition, the next state is SLAVESEL.

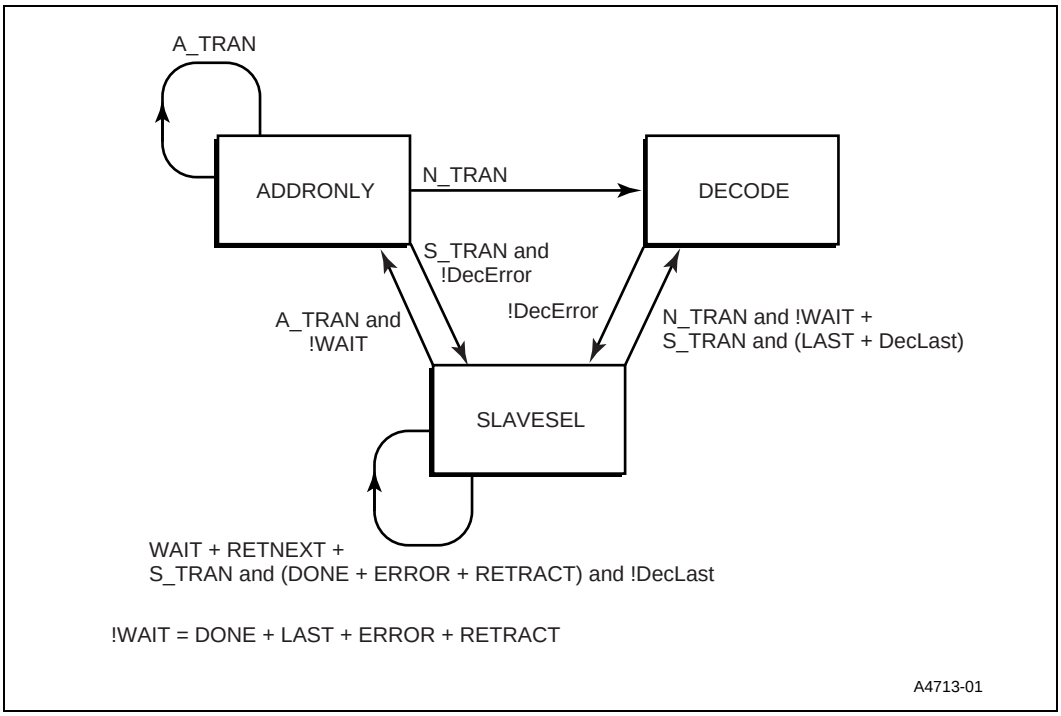
- SLAVESEL

  Slave transfer. SLAVESEL state may be entered either from DECODE or the ADDRONLY state.

  Unless there is a reset condition, the state machine will remain in the SLAVESEL state when BWAIT is HIGH. When BWAIT is LOW, the next state is ADDRONLY if the transfer type is address-only; DECODE if the transfer type is nonsequential, or if BLAST or DecLast have been asserted. However, if the transfer type is sequential and BLAST or DecLast have not been asserted, the next state remains as SLAVESEL.

Figure 7-3 shows the state machine of the decoder control block.

Figure 7-3. Decoder State Machine with Decode Cycles



7.2.3 Address Decoder

When the decoder control section indicates that a slave transfer is required, the address decoder generates a slave select signal, depending on the address on the bus.

7.2.4 Reset

During a reset condition, when BnRES is LOW, the decoder control block asynchronously removes all slave select signals.

7.2.5 Programmer's Model

There are no software configurable items in the decoder.

7.3 Test

There are no test registers in the decoder.

Test Interface Controller 8

The Test Interface Controller (TIC) is an AMBA System Bus (ASB) bus master. The TIC accepts test vectors from the external test bus (the 32-bit external data bus) and initiates bus transfers. The TIC latches address vectors from the test bus and drives the ASB address bus.

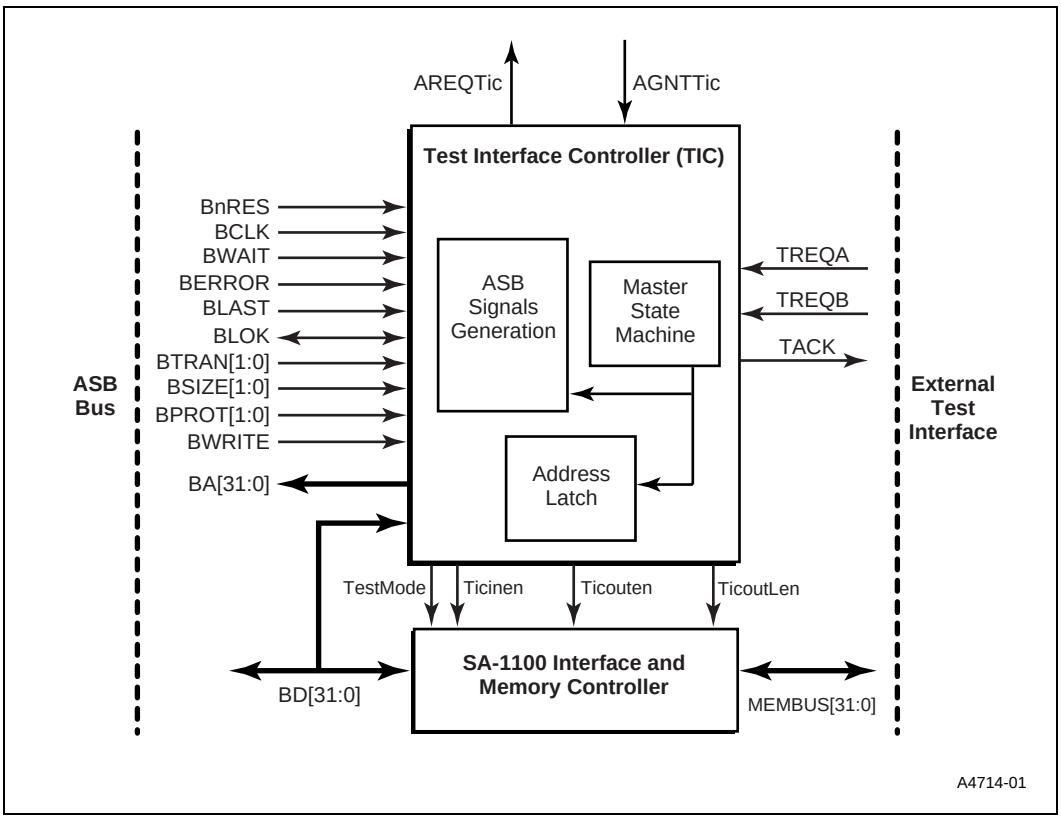
The TIC is the highest priority AMBA bus master on the StrongARM® SA-1101(SA-1101) to ensure test access under all conditions.

This TIC model does not support control vector/address incrementing (to change the address bus value, a specific address vector needs to be issued).

8.1 Block Diagram

Figure 8-1 shows the block diagram of the Test Interface Controller.

Figure 8-1. Test Interface Controller Block Diagram



8.2 Signal Description

Table 8-1 describes the signals.

Table 8-1. Signal Descriptions

Name	Type	Description
AREQTic	O	Request from the TIC, indicating that this master requires the bus. This signal must be set up to the falling edge of BCLK. The arbiter should treat this signal as the highest priority request line (over and above any complex arbitration scheme it might support).
AGNTTic	I	Grant signal that grants the bus to the test controller.
BnRES	I	Active LOW signal that indicates reset status of bus and is driven by reset controller.
BCLK	I	System (bus) clock. This clock times all bus transfers. The clock has two distinct phases — phase 1 in which BCLK is LOW and phase 2 in which BCLK is HIGH. In the example system, this clock also operates in test mode as TCLK.
BWAIT	I	Signal indicates when current transfer will complete. This signal is valid on the rising edge of BCLK. It must be used in concert with the other slave response lines BERROR and BLAST.
BERROR	I	Signal used with BWAIT and BLAST to form retract term in bus master state machine.
BLAST	I	Signal is used with BERROR and BWAIT to form the retract term.
BLOK	I	Bus lock signal. The TIC does not support locked transfers. Since the TIC should be the highest priority master, its transfers should never be interrupted. Driven LOW when the TIC is granted.
BTRAN[1:0]	O	Signals indicate the type of the next transaction, which in this master may be address-only or sequential. They are valid during the HIGH phase before the transfer to which they refer.
BSIZE[1:0]	O	Signals indicate the size of the transfer, which for this master is always word (32 bits). These signals have the same timing as the address bus.
BPROT[1:0]	O	These signals deal with address location access protection control. When generated by the TIC, these signals indicate supervisor mode transfers. They have the same timing as the address bus.
BA[31:0]	O	System address bus. The addresses become valid during the HIGH phase before the transfer to which they refer and remain valid until the last HIGH phase of the transfer.
BD[31:0]	I	In test mode, the data bus is used to load values into the TIC's address latch during "address vectors". The TIC then drives the system address bus BA with this value during subsequent single/burst read-write vectors.
BWRITE	O	When HIGH, this signal indicates a write transfer and when LOW, a read. This signal has the same timing as the address bus.
TREQA	I	Test request A. This signal is used, in combination with TREQB, to control access to the system bus from the test bus.
TREQB	I	Test request B. This signal is used, in combination with TREQA, to control access to the system bus from the test bus.
TACK	O	Test Acknowledge. This signal is used to indicate that the test interface has been granted access to the system bus. It is also used to indicate transfer delays (that is, transfers with wait cycles).
Ticinen	O	This active LOW signal indicates that the EBI should drive TBUS onto DB.
Ticouten	O	This active LOW signal indicates that the EBI should drive its latched version of BD onto the external TBUS.
TicoutLen	O	When low the BD latch in the EBI should be transparent.
TestMode	O	Indicates that the test controller has taken control of the bus.

8.3 Functional Description

The Test Interface Controller has two fundamental modes of operation:

- Default bus master
- System test

The default bus master is selected during reset or when no other masters are requesting the bus. When granted as the default master, the TIC performs no test functions but keeps the AMBA bus in a state such that:

- No data transfers occurs
  BTRAN = A-TRAN
- The bus is available to be granted to another master when requested.
  BLOK = LOW

For system test, an external tester must assert the external pins TREQA and TREQB. Table 8-2 describes the inputs and outputs to the external pins. The following abbreviations are used in Table 8-2: 0 = low, 1 = high, x = don't care.

Table 8-2. Basic TIC Operation

Inputs		Outputs	
TREQA	**TREQB**	**TACK**	**TIC Mode**
0	0	0	Acting as default bus master
1	0	0	Entering test mode
1	1	1	Test mode entered, ADDRESS vector
X	X	X	During test
0	0	1	Leaving test mode

Note: During test mode, at least one of the TREQA and TREQB inputs must be HIGH. If both inputs are LOW, this indicates end of test.

When entering test mode, the TIC requests the bus. Once it is granted as the current bus master, test mode is entered and the TACK signal is pulled HIGH.

Clocking Issues
When not in test mode, the SA-1101 is clocked by BCLK. TREQA is synchronized in the TIC by the rising edge of BCLK.

When test mode is entered, the SA-1101 will be clocked by TCLK, generated from an external source. Then TREQA and TREQB signals should be set up to the rising edge of TCLK.

The switch over from an internal BCLK source to an external TCLK source is not handled by the TIC. If an external BCLK source is used, then BCLK and TCLK can be identical.

8.3.1 Test Mode

In test mode the signal TestMode is driven HIGH. This forces BCLK to be driven from an external source (TCLK) and the Shared Memory Controller to provide a 32-bit bidirectional channel through which values can be read and written to BD. This 32-bit channel is referred to as TBUS.

The TREQA and TREQB signals should both be high on entering test mode. They are used to control the TIC, which allows values to be read or written to any address location inside the microcontroller; it can not perform read or writes to external memory as the external bus is occupied by the test bus (TBUS). This is done by applying TIC vectors, of which there are three basic types: READ, WRITE, and ADDRESS. Before a READ or WRITE vector can access a location, the appropriate address must have been loaded. Thus at the beginning of testing the first vector should be an ADDRESS type.

Table 8-3 describes the TIC vectors.

Table 8-3. TIC Vectors

TREQA	TREQB	Vector
1	1	ADDRESS. Must be first and last vector in a test sequence. Also used as a turn around cycle after a write vector.
1	0	WRITE. Must be followed by a turn around cycle.
0	1	READ.
0	0	End of test. Must be proceeded by an ADDRESS vector.

8.3.2 Vectors

The Test Interface Controllor consists of the following vectors:

- ADDRESS

  Only one ADDRESS vector is required for an address transfer. The TBUS must be driven with the address value required during the ADDRESS vector transfer. This value is latched from BD by the TIC, driven on to BA by the end of the ADDRESS vector transfer and is held for subsequent READ or WRITE vectors.

  During an ADDRESS vector, the TIC drives BTRAN as A-TRAN; no locations are accessed in the microcontroller.

- WRITE

  Only one WRITE vector is required for each write transfer. As with the ADDRESS vector, the value to be written should be driven on TBUS during the transfer for the WRITE vector. The TIC drives BTRAN as S-TRAN and asserts BWRITE, causing a write transfer to the address location setup by the last ADDRESS vector.

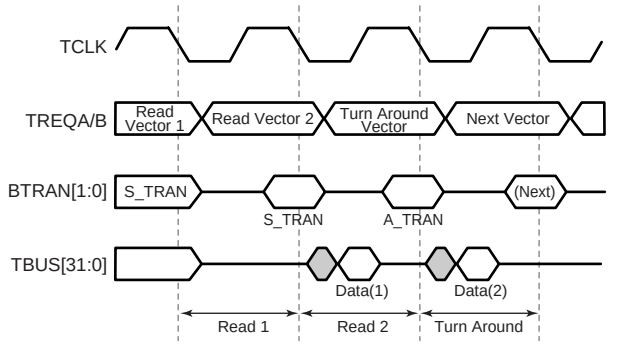
- READ

  Unlike ADDRESS or WRITE vectors, the READ vector TBUS activity does not take place during its corresponding AMBA transfer. During the read transfer period the TBUS should be undriven. The value read by the READ vector is driven out on TBUS in the cycle following the transfer. Here two reads are done (from the same address location). Since TBUS is driven in the cycle following the read 2 transfer, the READ vector 2 can not be followed by a WRITE

or an ADDRESS vector (this would require TBUS to be driven by the tester in the cycle following the read 2 transfer). Thus a turn around vector is needed. Turn around is indicated by TREQA=1, TREQB=1, which is identical to an ADDRESS vector. However, no address change occurs since TBUS is not driven and BD is not latched onto BA. The turn around vector may be followed by ADDRESS vectors (or any other type of vector), in which case, the address will change in subsequent cycles.

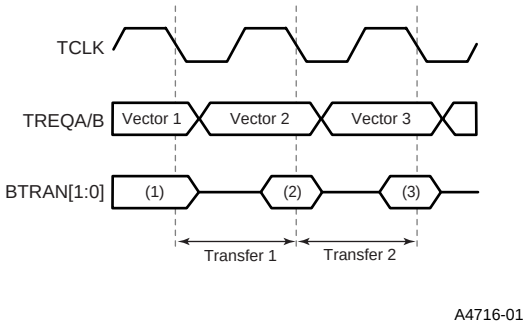
Figure 8-2 shows read vectors and turn around.

Figure 8-2. Read Vectors and Turn Around



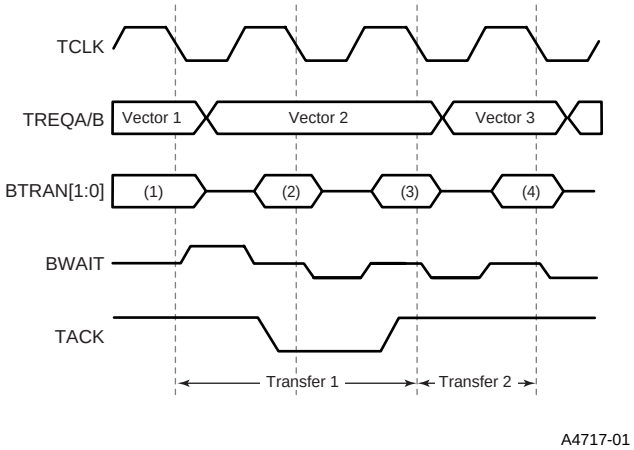
Vectors are applied in a pipelined fashion. In Figure 8-3, vector 2 is applied while the AMBA transfer triggered by vector 1 is occurring. Then in the following cycle, the transfer for vector 2 occurs. Figure 8-3 shows a simplified case.

Figure 8-3. TIC Vectors and AMBA Transfer



When the vectors are applied, the TACK signal should be monitored. This will normally be HIGH. However, if the transfer initiated by the previous vector is not complete, then TACK will go LOW off the falling edge of TCLK. When this occurs, the next vector should be held until TACK goes high. This is shown in Figure 8-4.

Figure 8-4. Vectors and Waited Transfers



Vector 1 starts a transfer with one wait cycle; vector 2 is held while the transfer completes. Although all vector types are applied as above, the timing characteristics for TBUS and the number of each vector type applied is vector dependent.

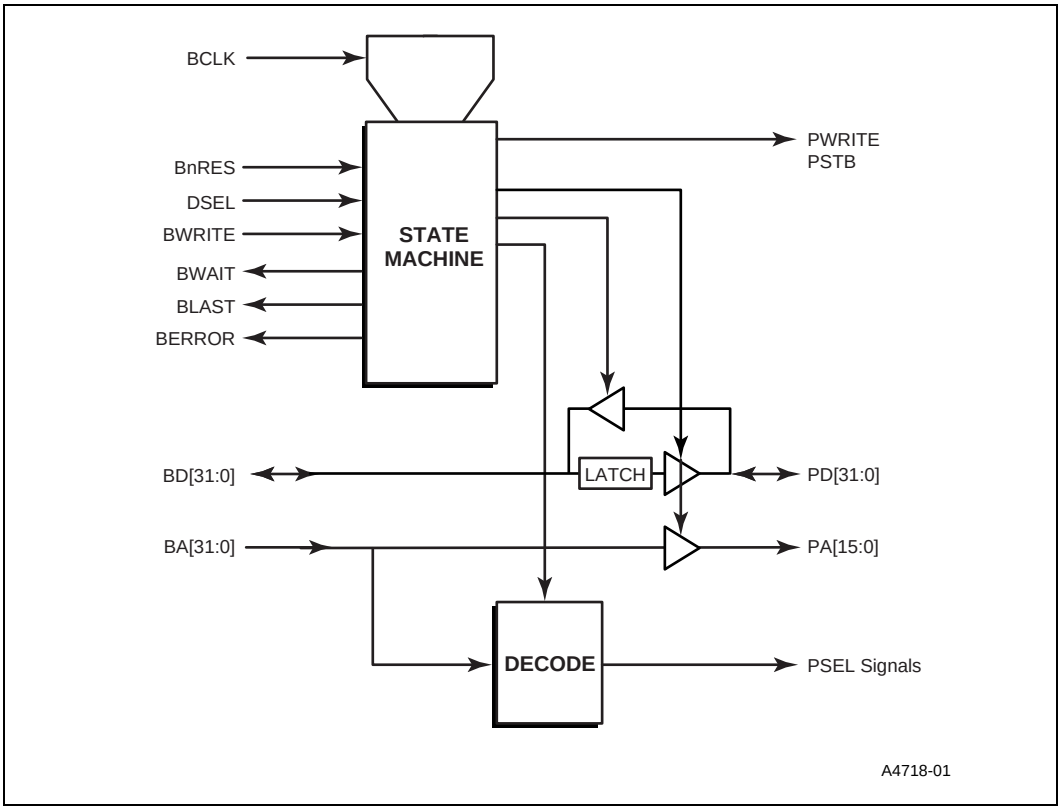
ASB-APB Bridge 9

This section describes the module that provides an interface between the AMBA System Bus (ASB) and the AMBA Peripheral Bus (APB).

9.1 Block Diagram

Figure 9-1 shows a block diagram of the ASB-APB bridge.

Figure 9-1. Block Diagram of the ASB-APB Bridge



The implementation of this block contains:

- A state machine that is independent of the device memory map
- Combinatorial address decoding logic to produce the peripheral PSEL signals

9.2 Signal Description

This section lists the signal descriptions used in the bridge module. Table 9-1 describes the signals for the ASB.

Table 9-1. Signal Descriptions

Name	Type	Source/ Destination	Description
BCLK	I	Section 3.3.1, “Control Register (SKCR)” on page 3-8	System (bus) clock. This clock times all bus transfers. The clock has two distinct phases — phase 1 in which BCLK is LOW and phase 2 in which BCLK is HIGH.
BD[31:0]	I/O	Bus master	The bidirectional system data bus. The data bus is driven by the current bus master during write transfers, and by this block during read transfers from the peripheral bus.
BnRES	I	Reset controller	An active LOW signal that indicates the reset status of the bus and is driven by the reset controller.
BWAIT	O	System decoder and current bus master	A signal that is driven by the selected bus slave to indicate if the current transfer may complete. If BWAIT is HIGH, a further bus cycle is required. If BWAIT is LOW, the transfer may complete in the current bus cycle. When no bus transfer is taking place, this signal is driven by the system decoder. When selected, the peripheral bus controller drives it in the LOW phase of BCLK. It is valid set up to the rising edge of BCLK.
BLAST	O	System decoder and current bus master	A signal that is driven by the selected bus slave to indicate if the current transfer should be the last of a burst sequence. It is always driven low. When no bus transfer is taking place, this signal is driven by the bus decoder. When selected, the peripheral bus controller drives it in the LOW phase of BCLK. It is valid set up to the rising edge of BCLK.
DSEL	I	From bus decoder	A signal that indicates that the peripheral bus controller has been selected. It becomes valid during the BCLK HIGH phase before the data transfer and remains active until the last BCLK HIGH phase of the transfer.
BERROR	O	System decoder and current bus master	A transfer error is indicated by the selected bus slave using the BERROR signal. When BERROR is HIGH, a transfer error has occurred. When BERROR is LOW, the transfer is successful. When no bus transfer is taking place, it is driven by the system decoder. When selected, the peripheral bus controller drives this signal in the LOW phase of BCLK. It is valid set up to the rising edge of BCLK.
BWRITE	I	Current bus master	This signal indicates a write cycle when HIGH and a read cycle when LOW. It has the same timing as the address bus and is driven by the bus master.

Table 9-2 describes the signals for the APB.

Table 9-2. APB Signals

Name	Type	Source/ Destination	Description
PA	O	APB peripherals	This is the peripheral address bus, which is used by individual peripherals for decoding register accesses to that peripheral. The addresses become valid before PSTB goes HIGH and remain valid after PSTB goes LOW.
PD	I/O	APB peripherals, B_D bus	This is the bidirectional peripheral data bus. The data bus is driven by this block during write cycles (when PWRITE is HIGH) and by the selected peripheral bus slave during read cycles (when PWRITE is LOW).
PSTB	O	APB peripherals	This strobe signal is used to time all accesses on the peripheral bus. The falling edge of PSTB is coincident with the falling edge of BCLK.
PWRITE	O	APB peripherals	This signal indicates a write to a peripheral when HIGH and a read from a peripheral when LOW. It has the same timing as the peripheral address bus. It becomes valid before PSTB goes HIGH and remains valid after PSTB goes LOW.

Table 9-3 describes the signals for the address decoding block.

Table 9-3. Address Decoding Block Signals

Name	Type	Source/ Destination	Description
BA	I	Current bus master	This is the system address bus, which is driven by the current bus master. The addresses change during the BCLK HIGH phase before the transfer to which they refer and remain valid until the last BCLK HIGH phase of the transfer.
PSELx	O	APB peripherals	There is one of these signals for each APB peripheral present in the system. The signal indicates that the slave device is selected and a data transfer is required. This signal has the same timing as the peripheral address bus. It becomes valid before PSTB goes HIGH and remains valid after PSTB goes LOW.

Figure 9-2 illustrates an APB write cycle.

Figure 9-2. APB Write Cycle

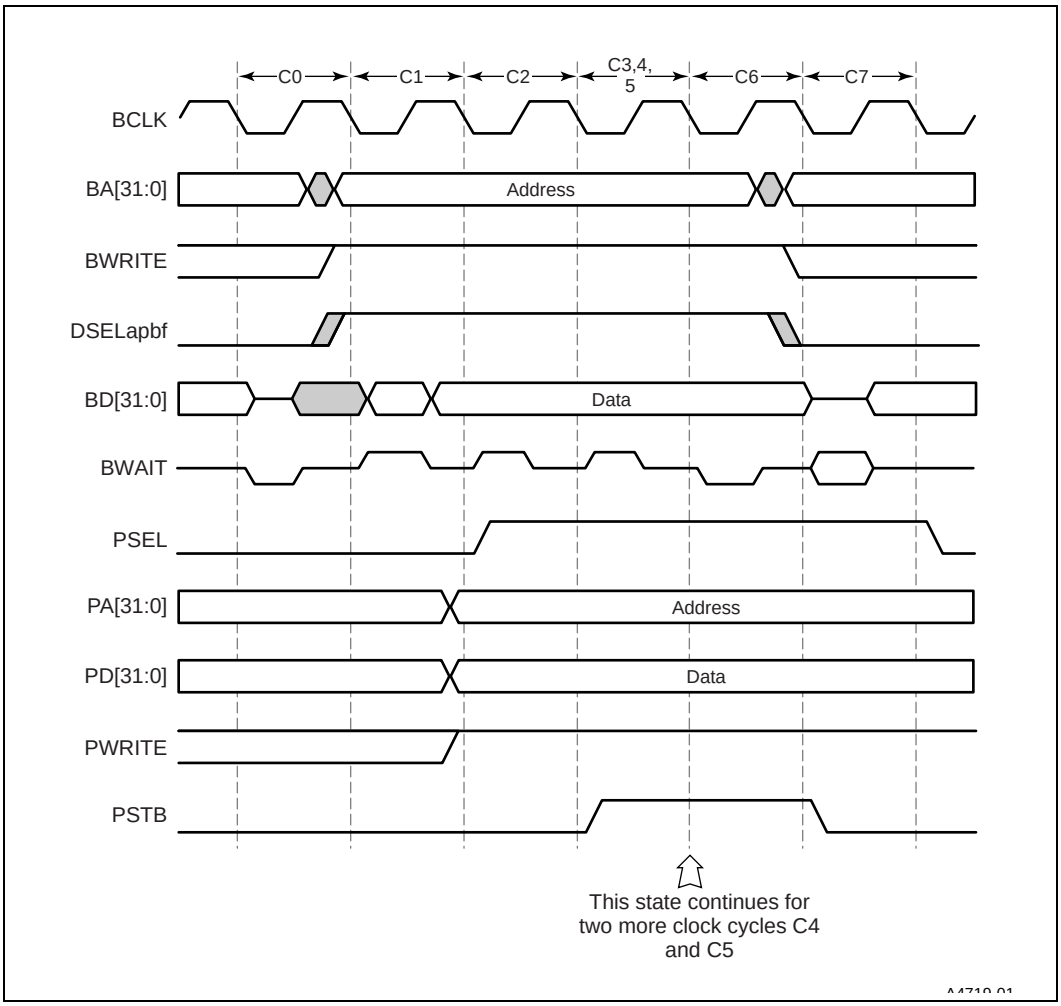
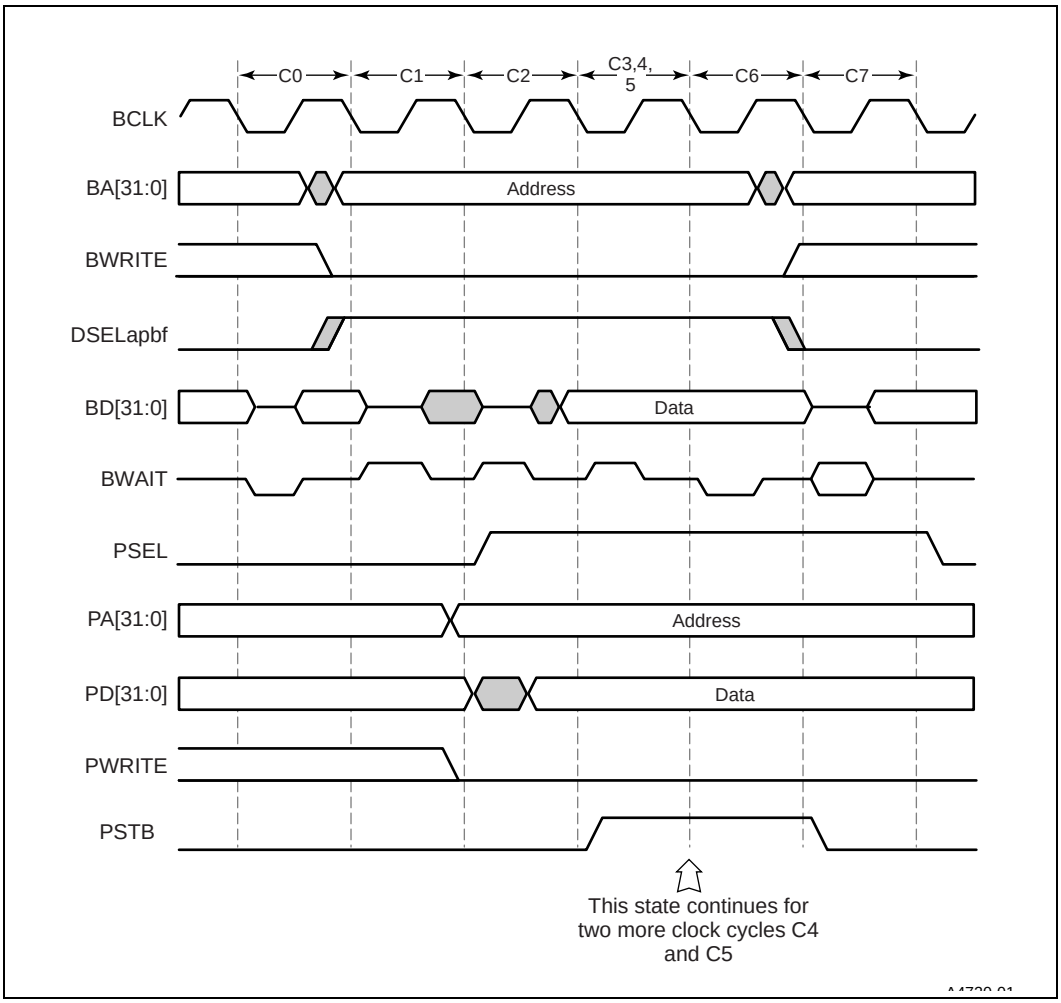


Figure 9-3 illustrates an APB read cycle.

Figure 9-3. APB Read Cycle



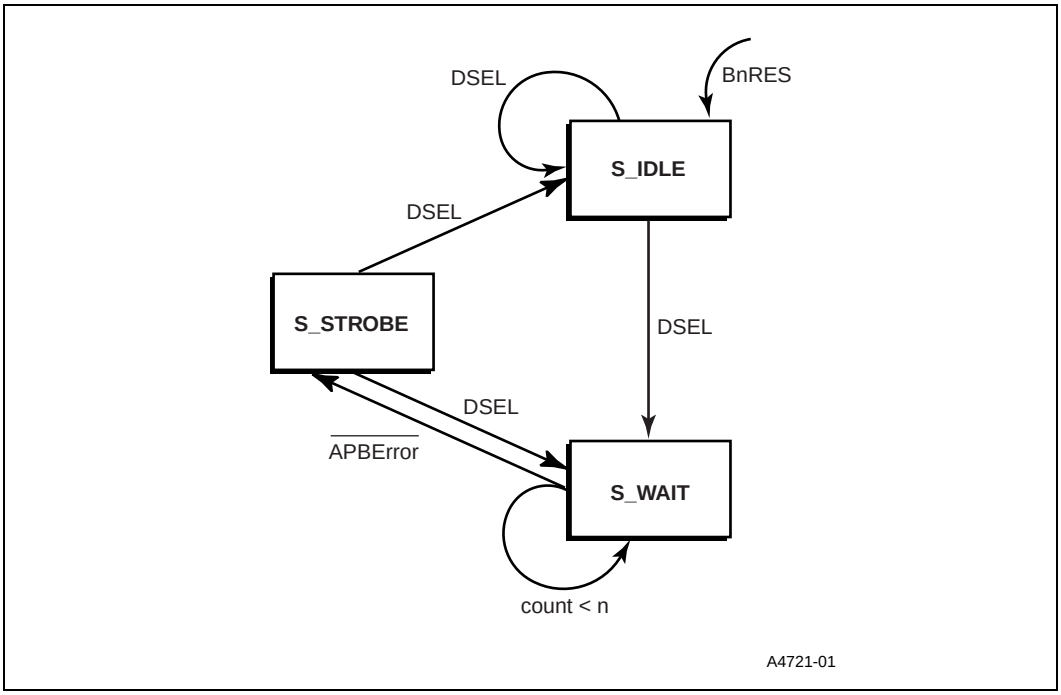
9.3 Functional Description

The APB bridge responds to transaction requests from the currently enabled bus master. The ASB transactions are converted into APB transactions. The state machine controls the depipelining of the ASB transaction, and controls the latches and drivers for the PA, PD and BD buses, also producing the PSTB signal.

Figure 9-4 shows the state machine of the APB controller. This example design uses the DSEL signal from the decoder to select the peripheral bus controller as an ASB slave. See the Decoder chapter for more information.

The individual PSELxx signals for each of the APB peripherals are decoded from BA, using the state of the state machine to enable their output.

Figure 9-4. State Machine for APB Controller



System Controller 10

The System Controller provides the clocking, power/reset functions, and mode control of the StrongARM® SA-1101 (SA-1101).

10.1 Functional Description

The System Controller, in conjuction with the Control Register, sets up the functioniality for the SA-1101. In addition, the System controller provides pulse width modulators (PWM) and video clock jitter compensation.

10.1.1 PWM Outputs

Two 8-bit resolution PWM outputs are provided. The PWM counter is clocked at CLK frequency, which gives a carrier frequency of 14.4 KHz at the recommended frequency of 3.6864 MHz for CLK. A value of 0 written to the data register gives no output and a value of 255 gives a pulse 255/256 of the carrier period.

10.1.2 Video Clock Jitter Compensation

The lowest frequency of the video clock (32 MHz), which will be used for a 640 x 480 resolution output, is achieved by dividing the PLL frequency by 4.5. This means that successive rising edges of VCLK (the video clock) will be derived from different edges of the PLL output. Because this will not have a guaranteed 50% mark/space ratio, the video clock will have a significant jitter component at 16 MHz.

A differential delay network is implemented so that 0, 1, 2, or 3 clock buffers may be inserted into the path from either branch of the generator. This will allow the jitter to be tuned out of the system.

10.2 Programmer's Model

This section describes the registers in the System Controller.

10.2.1 Power Control Register (SKPCR)

This register controls the clock enables of the SA-1101 and the setting of the multiplexer on the keypad and IEEE 1284 pins.

Bit	Name	Function
0	UCLKEn	USB 1 = Enable
1	PCLKEn	PS/2 1 = Enable
2	ICLKEn	Interrupt controller 1 = Enable
3	VCLKEn	Video controller 1 = Enable
4	PICLKEn	IEEE 1284 1 = Enable
5	DCLKEn	DACs 1 = Enable
6	nKPADEn	Multiplexer 0 = Keypad 1 = IEEE and GPIO

10.2.2 Clock Divider Register (SKCDR)

This register controls the setting of the SA-1101's PLL.

Bit	Name	Reset	Function
6:0	PLLMul	0100111	PLL Multiplier. Set to 0x27 for VCO frequency of 144 MHz.
8:7	VCLKDiv	00	Video controller clock divider. 00 – Divide VCO by 4.5 01 – Divide VCO by 3 10 – Divide VCO by 2 11 – Reserved
9	BCLKDiv	0	BCLK divider. 0 = Divide VCO by 4 1 = Divide VCO by 2 (test mode only)
10	UTESTCLKEn	0	This bit may only be set in test mode and routes the USB clock from the S1_BVD2_NSPKR pin.
16:11	DivRValue	000000	Input reference clock divider for the PLL. 00000 – Divide by 0 All other values – Reserved.
21:17	DivNValue	00000	Output clock divider for the PLL. 00000 – Divide by 0 All other values – Reserved
24:22	PLLRSH	000	PLL bandwidth control. 000 – 26.56 KHz 001 – 35.41 KHz 011 – 53.20 KHz 111 – 79.50 KHz
25	ChargePump	0	Charge pump control (PFD_CP13). Always set to 0.
26	ClkTestMode	0	Clock output test mode. 0 – Normal 1 – Test
27	ClkTestEn	0	Test clock generator. 0 – Normal 1 – Test clock enabled
30:28	ClkJitterCntl	011	Video clock jitter compensation. 000 – 3 delays in positive edge 001 – 2 delays in positive edge 010 – 1 delay in positive edge 011 – Minimum delay positive and negative 100 – 1 delay in negative delay 101 – 2 delays in negative edge 11X – 3 delays in negative edge ***Note:*** ClkJitterCntl only affects display in 640x480 resolution.
31	—	0	Reserved.

10.2.3 DAC Data Registers (DACDR1 and DACDR2)

These registers control the slow speed PWM DAC outputs.

Bit	Name	Function
7:0	DACCount	Count value. 0 – Minimum pulse width 255 – Maximum Pulse width
31:8	—	Reserved.

See Chapter 3, “Interface and Shared Memory Controller” for more information.

10.2.4 Memory Map

Table 10-1 lists the read and write locations in the memory map of the System Controller.

Table 10-1. System Controller - Read/Write Locations

Address	Read Location	Write Location
0x00000400	SKPCR register	SKPCR register
0x00040000	SKCDR register	SKCDR register
0x00060000	DACDR1 register	DACDR1 register
0x00060400	DACDR2 register	DACDR2 register

IEEE 1284 Parallel Port Interface 11

The IEEE 1284 Interface implements the IEEE specification providing a signalling method for bidirectional parallel communications with printers and other peripheral devices.

The IEEE 1284 specification defines five modes of operation:

- Compatibility Mode
- Nibble Mode
- Byte Mode
- Extended Capabilities Port (ECP) Mode
- Enhanced Parallel Port (EPP) Mode

The five modes are supported from a host viewpoint in some form, with registers defined for the control and status monitoring of the port in each mode. Interrupt request outputs have been provided that are asserted during handshaking and data transfers that occur during communication with the peripheral device.

Both forward (host-to-peripheral) and reverse (peripheral-to-host) data flows are supported through single register or FIFO access from the AMBA bus.

Note: This document assumes that the reader is familiar with the following specification:

IEEE 1284 Std-1284-1994 – *IEEE Standard Signalling Method for a Bidirectional Parallel Peripheral Interface for Personal Computers*

11.1 Signal Descriptions

This section lists the signal descriptions for the IEEE 1284 port interface. The signals descriptions for the multipurpose pins of the IEEE port interface have been separated into five tables — a table for each of the five modes of operation.

11.1.1 IEEE Port I/O: Compatibility Mode

Table 11-1 lists the IEEE port I/O signals in Compatibility Mode.

Table 11-1. IEEE Port I/O Signals in Compatibility Mode

Name	Type	Description
Data[8:1]	I/O	Forward transfer byte wide databus.
Busy	I	When HIGH, indicates that the peripheral device is not ready to receive data.
Select	I	When HIGH, signifies that the peripheral is online.
nAck	I	Pulsed LOW by the peripheral to acknowledge transfer of a data byte from the host.
nFault	I	Set LOW by the peripheral to indicate that an error has occurred.
PError	I	When HIGH, signifies error has occurred in paper path. Peripherals shall set nFault LOW whenever they set PError HIGH.
nInit	O	Pulsed LOW to reset the interface and force a return to Compatibility Mode idle phase.
nStrobe	O	When LOW, data is transferred into the input latch of the peripheral. Data is valid while nStrobe is LOW.
nSelectIn	O	When LOW, this peripheral is selected.
nAutoFd	O	Interpretation varies between peripherals. When LOW, puts some printers into auto-line feed mode.
Buffen	O	Bidirectional data pin and external buffer directional control signal.

11.1.2 IEEE Port I/O: Nibble Mode

Table 11-2 lists the IEEE port I/O signals in Nibble Mode.

Table 11-2. IEEE Port I/O Signals in Nibble Mode

Name	Type	Description
Data[8:1]	I/O	Not used.
Busy	I	Reverse data transfer phase: Data bit 3, then 7, then forward channel busy status.
Select	I	Reverse data transfer phase: Data bit 1, then 5.
nAck	I	Reverse data transfer phase: Used to qualify data being sent to the host. Reverse idle phase: Set LOW then HIGH by the peripheral to cause an interrupt indicating to the host that data is available.
nFault	I	Reverse data transfer phase: Set LOW to indicate that the peripheral has the data ready to send to the host. Then used to send data bit 0, then 4. Reverse idle phase: Used to indicate that data is available.
PError	I	Reverse data transfer phase: Data bit 2, then 6. Reverse idle phase: Set HIGH until the host requests a transfer, then follows nFault.
nInit	O	Set HIGH by the host.
nStrobe	O	Set HIGH during transfers to avoid latching data into the peripheral.
nSelectIn	O	Set HIGH by the host.
nAutoFd	O	Reverse data transfer phase: Set LOW to indicate that the host can receive peripheral-to-host data, then set HIGH to acknowledge receipt of that nibble. Following a reverse channel transfer, the interface changes to idle phase when nAutoFd is set LOW and the peripheral has no data available. Reverse idle phase: Set HIGH in response to nAck LOW pulse to reenter reverse data transfer phase. If set HIGH with nSelectIn set LOW, the IEEE 1284 idle phase is aborted, and the interface returns to Compatibility Mode.
Buffen	O	Bidirectional data pin and external buffer directional control signal.

11.1.3 IEEE Port I/O: Byte Mode

Table 11-3 lists the IEEE port I/O signals in Byte Mode.

Table 11-3. IEEE Port I/O Signals in Byte Mode

Name	Type	Description
Data[8:1]	I/O	Reverse channel byte wide data.
Busy	I	Forward channel busy status.
Select	I	Driven to indicate the response to a requested extensibility byte sent by the host.
nAck	I	Reverse data transfer phase: Used to qualify data being sent to the host. Reverse idle phase: Set LOW then HIGH by the peripheral to cause an interrupt indicating to the host that data is available.
nFault	I	Set LOW to indicate that the peripheral has data to send to the host.
PError	I	Reverse data transfer phase: Follows nFault. Reverse idle phase: Set HIGH until the host requests a data transfer, then follows nFault.
nInit	O	Set HIGH by the host.
nStrobe	O	Will be pulsed LOW during Byte Mode transfers. The peripheral shall ensure that nStrobe does not cause data to be latched when this happens.
nSelectIn	O	Set HIGH by the host.
nAutoFd	O	Reverse data transfer phase: Set LOW to indicate that the host can receive a peripheral-to-host byte, then set HIGH to acknowledge receipt of that byte. Following a reverse channel transfer, the interface transitions to idle phase when nAutoFd is set LOW and the peripheral has no data available. Reverse idle phase: Set HIGH in response to nAck LOW pulse to reenter reverse data transfer phase. If set HIGH with nSelectIn set LOW, the IEEE 1284 idle phase is aborted and the interface returns to Compatibility Mode.
Buffen	O	Bidirectional data pin and external buffer directional control signal.

11.1.4 IEEE Port I/O: Extended Capabilities Port (ECP) Mode

Table 11-4 lists the IEEE port I/O signals in Extended Capabilities Port (ECP) Mode.

Table 11-4. IEEE Port I/O Signals in Extended Capabilities Port (ECP) Mode

Name	Type	Description
Data[8:1]	I/O	Host-to-peripheral or peripheral-to-host address or data.
Busy	I	The peripheral uses this signal for flow control in the forward direction. nAck also provides a ninth data bit used to determine whether command or data information is present on the data signals in the reverse direction.
Select	I	Driven to indicate the response to a requested extensibility byte sent by the host.
nAck	I	Used in a closed-loop handshake with nAutoFd to transfer data from the peripheral to the host.
nFault	I	During ECP mode, the peripheral may drive this pin LOW to request communication with the host. The request is merely a hint; the host has ultimate control over the transfer direction. This signal provides a mechanism for peer-to-peer communication. Typically, it is used to generate an interrupt to the host. This signal is valid in the forward and reverse directions.
PError	I	The peripheral drives this signal LOW to acknowledge nInit. The host relies upon PError to determine when it is permitted to drive the data signals.
nInit	O	This signal is driven LOW to place the channel in reverse direction. While in the ECP Mode, the peripheral is only allowed to drive the bidirectional data signals when nInit is LOW and nSelectIn is HIGH.
nStrobe	O	Used in a closed-loop handshake with Busy to transfer data or address information from the host to the peripheral.
nSelectIn	O	Driven HIGH by the host while in ECP mode. Set LOW by the host to terminate ECP mode and return the Interface to Compatibility Mode.
nAutoFd	O	The host drives this signal for flow control in the reverse direction. It is used in the interlocked handshake with nAck. nAutoFd also provides a ninth bit that is used to determine whether command or data information is present on the data signals in the forward direction.
Buffen	O	Bidirectional data pin and external buffer directional control signal.

11.1.5 IEEE Port I/O: Enhanced Paralled Port (EPP) Mode

Table 11-5 lists the IEEE port I/O signals in Enhanced Parallel Port (EPP) Mode.

Table 11-5. IEEE Port I/O Signals in Enhanced Parallel Port (EPP) Mode

Name	Type	Description
Data[8:1]	I/O	Host-to-peripheral or peripheral-to-host address or data.
Busy	I	This signal should be driven inactive as a positive acknowledgement from the peripheral that transfer of data or address is completed.The signal is active LOW. It should be driven active as an indication that the device is ready for the next address or data transfer.
Select	I	This signal is manufacturer specific.
nAck	I	Used by the peripheral to interrupt the host. This signal is active HIGH and positive edge triggered.
nFault	I	This signal is manufacturer specific.
PError	I	This signal is manufacturer specific.
nInit	O	When driven LOW, this signal initiates a termination cycle that results in the interface returning to Compatibility Mode.
nStrobe	O	Set LOW to denote an address or data write operation to the peripheral. Set HIGH to denote an address or data read operation from the peripheral.
nSelectIn	O	This signal is used to denote an address cycle. It is active LOW.
nAutoFd	O	This signal is used to denote a data cycle. It is active LOW.
Buffen	O	Bidirectional data pin and external buffer directional control signal.

11.1.6 Interrupt I/O Signals

Table 11-6 lists the signal descriptions for the interrupt signals.

Table 11-6. Interrupt Signals

Name	Type	Description
IntReqTrc	O	Active HIGH signal asserted on a successful forward/reverse data or address transfer.
IntReqTim	O	Active HIGH signal asserted when the peripheral fails to provide a response to a host-driven handshake.
IntReqRav	O	Active HIGH signal asserted as a result of a peripheral input control signal changing state.
IntReqInt	O	Active HIGH signal set when the peripheral asserts the ‘Interrupt’ input in EPP mode.
IntReqEmp	O	Active HIGH signal asserted if the forward interface FIFO is emptied in the course of a data block transfer.
IntReqDat	O	Active HIGH signal asserted whenever the forward or reverse data FIFOs require service.

11.1.7 Miscellaneous Signals

Table 11-7 lists the signal descriptions for miscellaneous signals.

Table 11-7. Miscellaneous Signals

Name	Type	Description
RefClk	I	RefClk is divided down internal to the block by a programmable divider to produce an internal 8-MHz clock. RefClk must have a frequency higher than 8 MHz.

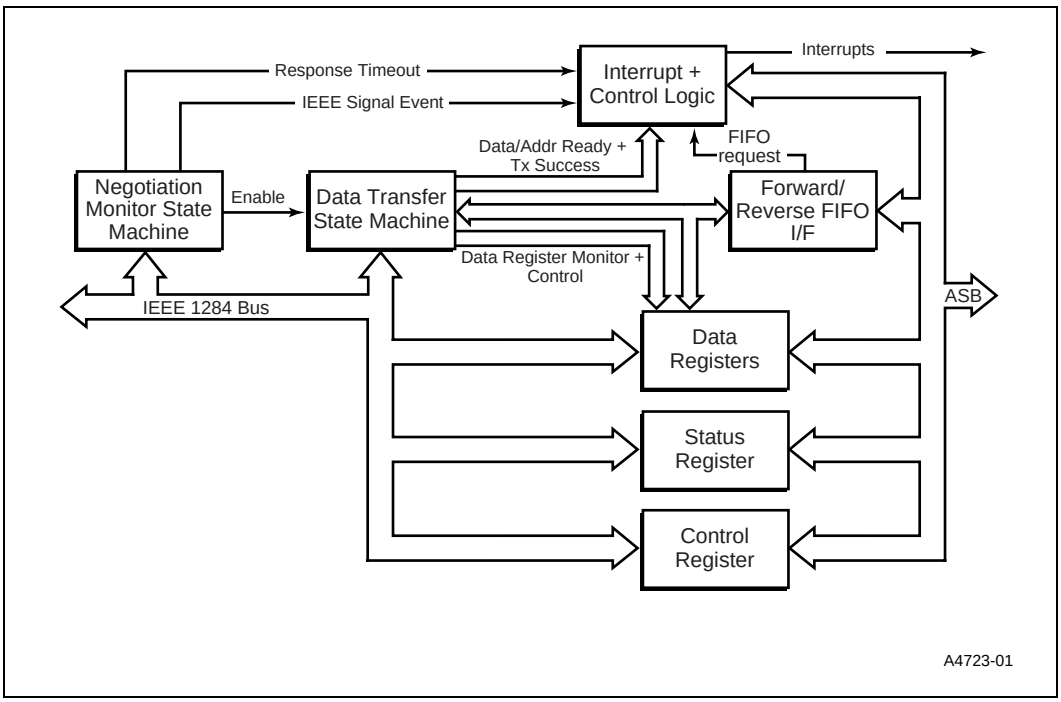
11.2 Functional Description

This section provides a functional overview of the interfaces to the IEEE 1284 port interface.

11.2.1 Block Diagram

Figure 11-1 shows a block diagram of the IEEE 1284 port interface.

Figure 11-1. IEEE 1284 Port Interface Block Diagram



11.2.2 Functionality

The Negotiation Monitor State Machine monitors IEEE bus signal events, either generated by the CPU via the AMBA bus or from the peripheral. A response (or lack of a response) from the peripheral to a CPU initiated handshake results in an interrupt request being generated.

When the host-peripheral handshake process enters the data transfer phase of one of the five IEEE modes of operation, a control register bit is set to enable the data transfer state machine. The data transfer state machine controls and monitors data flow between the IEEE data bus and the AMBA bus. Interrupt requests are generated when data is received from the peripheral or on completion of a successful forward data transfer.

Inherent within the design is a multipurpose timer. It provides a safety mechanism to timeout transactions that do not get responses from the IEEE 1284 interface, and also enables the state machines to ensure that data setup and hold times are met with respect to the IEEE 1284 strobe signals.

The methods of host-peripheral data transfer used in each of the five IEEE modes are different in each case.

11.2.3 Communication Sequence

Hosts reinitialize the interface at any time by asserting nInit LOW in conjunction with nSelectIn LOW. Resetting the interface also resets the peripheral.

The interface is always initialized to the compatibility mode — a conventional, unidirectional host-to-peripheral interface. From the compatibility mode, the host:

- Transmits data to the peripheral using the compatibility mode
- Directs the interface to a mode capable of transmitting data from the peripheral to the host

Hosts and peripherals indicate their readiness to communicate by asserting their host and peripheral logic high signals respectively.

An IEEE 1284-compliant host negotiates with the peripheral to determine whether or not the peripheral is IEEE 1284-compliant and if so, the host requests a communication mode for the peripheral; otherwise the host removes its stimulus.

The compliant IEEE 1284 device acknowledges the communication request based on its capabilities and executes or rejects the request as appropriate. At the discretion of the host, the interface returns to compatibility mode at any time.

11.2.3.1 Negotiation Sequence

If the request fails, the host returns the link to the compatibility mode and renegotiates for another transfer mode.

Table 11-8 lists the possible outcomes of the negotiation sequence.

Table 11-8. Outcomes of the Negotiation Sequence

Mode Requested	Peripheral Response	Action
Nibble or byte	The peripheral responds correctly and the peripheral-to-host data is available	The host can: • Proceed with a reverse channel data transfer using the mode requested by the host • Remain in the host busy, data available phase • Terminate and return to compatibility mode
	The peripheral responds correctly but no peripheral-to-host data is available	The host can: • Set nAutoFd LOW, which puts the interface into idle phase • Terminate and return to compatibility mode
ECP	The peripheral responds correctly.	Following a setup phase, the link enters the ECP mode forward idle phase and the: • Host can request to send data to the peripheral • Peripheral can request to send data to the host • Host can terminate ECP mode and return to compatibility mode
EPP	The peripheral responds correctly.	The link enters the EPP mode idle phase and the host can: • Write data to the peripheral • Read data from the peripheral • Write an address value to the peripheral • Read an address value from the periopheral • Terminate EPP mode and return to compatibility mode

11.2.3.2 Termination

Table 11-9 describes the types of termination.

Table 11-9. Types of Termination

Mode Requested	Host Response	Action
Nibble, byte, or ECP	Sets nSelectIn LOW and nAutoFd HIGH, if not already set.	Initiates one of two types of termination: • A handshake , which allows the peripheral to tell the host when it has returned to compatibility mode • An immediate abort, with no guarantee. If the peripheral was in a valid state, which is any state where a reverse data transfer is not in progress, the peripheral performs the handshake. Otherwise the peripheral aborts immediately and the current byte in transit is lost. However, the byte remains in the peripheral output buffer and is transmitted the next time the host performs a reverse channel transfer.
EPP	Asserts **nInit**.	The peripheral resets to its initial compatibility mode.

11.2.4 Compatibility Mode

Compatibility mode provides a byte-wide forward channel with data, clock and status lines to the peripheral device. This mode is defined by the IEEE 1284 specification to be the default for the IEEE port, and all other modes (nibble, byte, ECP, and EPP) revert to compatibility mode after host termination of the other modes.

This mode is supported through the data transfer state machine built into the IEEE 1284 Interface with a minimal amount of CPU intervention required. The state machine controls all outputs, and responds to inputs from the peripheral device, asserting the Interrupt request line when successful data transfer has been completed.

Data access can be programmed to be through single register access or FIFO interface from the AMBA bus.

Four modes are defined as capable of providing peripheral-to-host data transfers:

- Nibble mode
- Byte mode
- ECP mode
- EPP mode

11.2.5 Nibble Mode

Nibble mode provides a reverse data channel under control of the host where data bytes are transmitted as two sequential 4-bit nibbles across unused control lines. The mode is entered through a negotiation sequence with the peripheral, and a further handshaking sequence is required before data is transferred.

All negotiation and handshaking sequences are under control of the CPU through the use of a writeable control register. Peripheral responses are monitored and an interrupt request signal asserted when control lines from the peripheral change state. A readable status register is provided so that the CPU can determine the source of the Interrupt and monitor the control lines from the peripheral.

A control register bit is used to indicate that the data transfer phase has been entered and is used to activate a state machine to affect the data transfer. When the data byte from the peripheral has been read, an interrupt request is generated and the process returns to the handshake phase.

Data access can be programmed to be through single register access or FIFO interface from the AMBA bus.

11.2.6 Byte Mode

Byte mode is identical to nibble mode except that reverse data is transferred byte-wide across the bidirectional data lines provided by the IEEE port.

11.2.7 ECP Mode

ECP mode provides a byte-wide bidirectional channel controlled by an interlocked handshake between host and peripheral. Data transfers in this mode can be optionally Run Length Encoded (RLE), with a separate control line between host and peripheral indicating whether a data byte is a command word or data word.

The mode is entered through a negotiation sequence with the peripheral which is under control of the CPU through the use of a writeable control register. Peripheral responses are monitored and an interrupt request signal asserted when control lines from the peripheral change state. A readable status register is provided so that the CPU can determine the source of the interrupt and monitor the control lines from the peripheral.

A control register bit is used to indicate that the data transfer phase has been entered and is used to activate a state machine to affect data transfers. Another control bit is used to indicate the direction of data flow. Data access can be programmed to be through single register or FIFO interface from the AMBA bus.

ECP Mode has a ninth bit which may be used to tag the accompanying byte as being data or a command. For example, if Run Length Encoding (RLE) is used, it may be used to tag the byte as being a channel address, or as a *run length* of consecutive data bytes of the same value.

11.2.8 EPP Mode

EPP mode provides a byte-wide bidirectional channel controlled by the host through the use of separate time-multiplexed address and data cycles over the eight data lines of the IEEE 1284 interface.

The mode is entered through a negotiation sequence with the peripheral which is under control of the CPU through the use of a writeable control register. Peripheral responses are monitored and an Interrupt Request signal asserted when control lines from the peripheral change state. A readable status register is provided so that the CPU can determine the source of the Interrupt and monitor the control lines from the peripheral.

A control register bit is used to indicate that the EPP Idle phase has been entered and is used to activate state machines to perform one of the four following IEEE 1284 transfers:

- Data write
- Data read
- Address write
- Address read

Address transfers can be controlled by reading/writing to an address location.

Address read and write operations performed on the ASB registers cause an interrupt request to be generated if successful. This method of data transfer has been adopted due to the relatively slow peripheral response time (10µs) allowed by the IEEE specification, and saves tying up the AMBA bus for long periods of time.

11.2.9 FIFO Transfers

Data access through FIFO is supported in all the data transfer modes supported by the IEEE 1284 Interface. The FIFO interface block provides FIFO access in both forward and reverse directions. The forward direction uses a 16-word FIFO. Data throughput in the forward direction is enabled by the CPU setting a bit of a programmable register resulting in the generation of Interrupt Requests whenever the fill of the FIFO is less than 9 words or 16 words (programmable).

The reverse direction also uses a 16-word FIFO. Reverse data from the peripheral is written to the FIFO and an interrupt request is made when an 8- or 1-word fill threshold (programmable) is reached. At the end of a data block transfer, the IEEE 1284 interface is placed into Compatibility Mode which causes the last data word of the IEEE transfer to be written to the FIFO.

11.3 Programmer's Model

This section describes the registers in the IEEE 1284 parallel port interface and lists their offset addresses and widths.

Note: The reserved bits of registers should be written to as 0 and read as "Don't Care" values. The physical implementation may not be 32-bits wide.

Table 11-10 gives a summary of the registers in the IEEE 1284 parallel port interface.

Table 11-10. IEEE 1284 Parallel Port Interface Register Summary

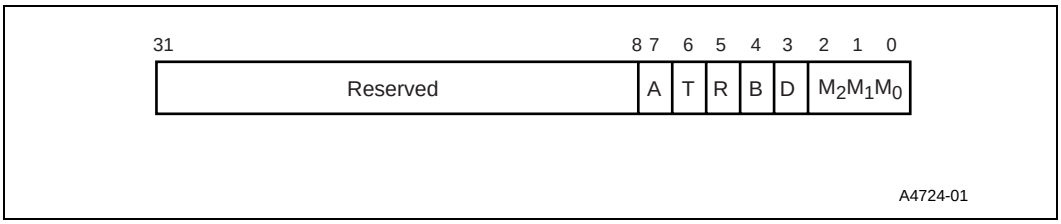
Address Offset	Type	Width	Name	Description
0x0000	R/W	8	Config	IEEE mode selection and programmable attributes
0x0400	R/W	4	Control	Controls the states of IEEE port control outputs
0x4000 – 0x7C00	R/W	32	Data	Forward (host-to-peripheral) transfer data register; 16 locations allocated for burst mode support
0x0800	R/W	8	Addr	Forward (host-to-peripheral) transfer address register
0x0C00	R	9	Status	Port input/output signal status register
0x1000	R/W*	9	IntStatus	Port interrupts status register
0x1400	R/W	2	FifoLevels	Programs the Rx and Tx FIFO interrupt generation levels
0x1800	R/W	22	InitTime	Forward (host-to-peripheral) timeout counter initial value
0x1C00	R	22	TimerStatus	Forward (host-to-peripheral) timeout counter current value
0x2000	W	0	FifoReset	A write to this location will reset the forward transfer FIFO
0x3C00	R/W	4	ReloadValue	Counter reload value for programmable RefClk divider

Note: A write of a logic 1 to any bit position clears the interrupt. A write of logic 0 leaves it unchanged.

11.3.1 IEEE Config Register

This register is an 8-bit register containing programmable bits relating to the configuration of the IEEE 1284 Interface. Figure 11-2 shows the bit coding for the IEEE config register.

Figure 11-2. IEEE Config Register Bit Coding



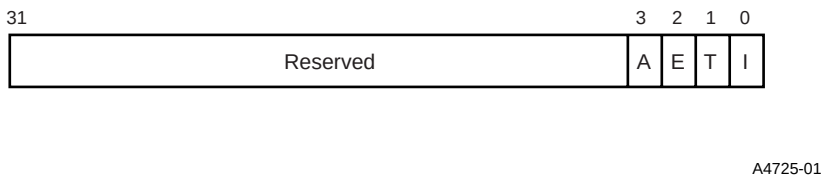
Bit	Name	Function
2:0	M[2:0]	Mode select 1XX – Compatibility Mode 000 – Nibble Mode 001 – Byte Mode 010 – ECP Mode 011 – EPP Mode
3	D	FIFO access enable 0 – Register access 1 – FIFO access
4	B	9-bit word enable 0 – Disable 9 bit mode 1 – Enable 9 bit mode
5	T	Data transfer enable 0 – Transfer disabled 1 – Transfer enabled
6	A	Data transfer direction 0 – Peripheral-to-host (reverse) 1 – Host-to-peripheral
7	E	Timer enable 0 – Timer disabled 1 – Timer enabled
R	—	Reserved

11.3.2 Control Register

This is a 4-bit register containing programmable bits that directly control the states of IEEE port control outputs. It should be noted that in some modes, control of some IEEE port outputs is handed over to state machines built in to the IEEE Interface (for example, the nStrobe output in Compatibility Mode). When this occurs, the IEEE 1284 port output does not necessarily mirror the state defined by this register. The status of IEEE 1284 port outputs can be read from the Status register in these cases.

Figure 11-3 shows the bit coding for the control register.

Figure 11-3. Control Register Bit Coding

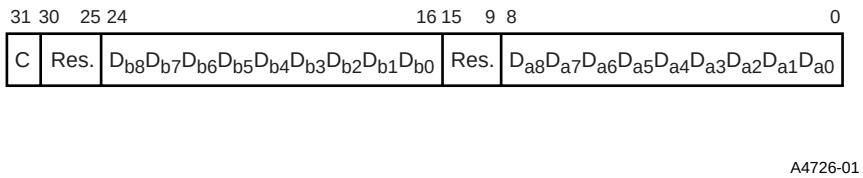


Bit	Name	Function
3	A	IEEE 1284 nAutoFd output
2	E	IEEE 1284 nSelectin output
1	T	IEEE 1284 nStrobe output
0	I	IEEE 1284 Port nInit output

11.3.3 Data Register

This register is a 32-bit register containing data relating to forward FIFO transfers used for AMBA access. Figure 11-4 shows the bit coding for the data register.

Figure 11-4. Data Register Bit Coding

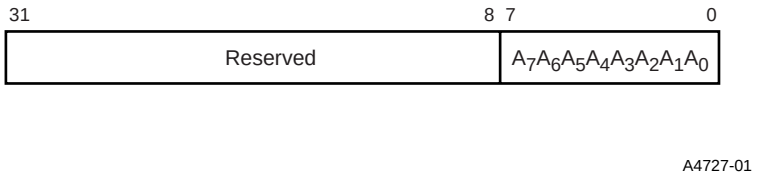


Bit	Name	Function
31	C	Byte count 0 – only byte 1 valid 1 – bytes 1 and 2 valid
30:25	—	Reserved
24:16	Db[8:0]	Data byte 2
8:0	Da[8:0]	Data byte 1

11.3.4 Addr Register

This is an 8-bit register containing forward transfer address to the peripheral device used in EPP mode. Figure 11-5 shows the bit coding for the addr register.

Figure 11-5. Addr Register Bit Coding

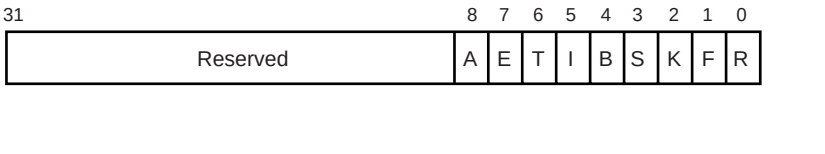


Bit	Name	Function
7:0	A[7:0]	EPP mode forward address transfer byte

11.3.5 Status Register

This is a 9-bit register containing the status of all inputs and outputs of the IEEE 1284 port at the time of reading the register. Figure 11-6 shows the bit coding for the status register.

Figure 11-6. Status Register Bit Coding

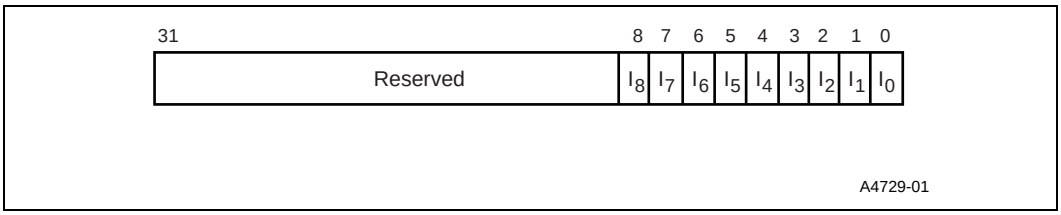


Bit	Name	Function
8	A	nAutoFd port output status
7	E	nSelectIn port output status
6	T	nStrobe port output status
5	I	nInit port output status
4	B	Busy port input status
3	S	Select port input status
2	K	nAck port input status
1	F	nFault port input status
0	R	PError port input status

11.3.6 IntStatus Register

This is a 9-bit register containing the status of all inputs and outputs of the IEEE 1284 port at the time of reading the register. Figure 11-7 shows the bit coding for the IntStatus register.

Figure 11-7. IntStatus Register Bit Coding



Bit	Name	Function
8	IntReqDat	Active HIGH signal, asserted whenever the forward or reverse FIFOs require servicing
7	IntReqEmp	Active HIGH signal, asserted if the forward interface FIFO is emptied in the course of a data block transfer
6	IntReqInt	Active HIGH signal, set when the peripheral asserts the interrupt signal in EPP mode
5	IntReqRav	Active HIGH signal, asserted as a result of a peripheral input control signal changing state
4	IntReqTim	Active HIGH signal, asserted when the peripheral fails to provide a response to a host-driven handshake
3	RevAddrComp	Active HIGH signal, asserted on completion of a successful reverse address transfer
2	RevDataComp	Active HIGH signal, asserted on completion of a successful reverse data transfer
1	FwdAddrComp	Active HIGH signal, asserted on completion of a successful forward address transfer
0	FwdDataComp	Active HIGH signal, asserted on completion of a successful forward data transfer

Note: A write of logic 1 to any bit position clears the respective interrupt. A write of logic 0 leaves it unchanged.

11.3.7 FifoLevels Register

This is a 2-bit register used to program the Rx and Tx FIFO interrupt generation levels.

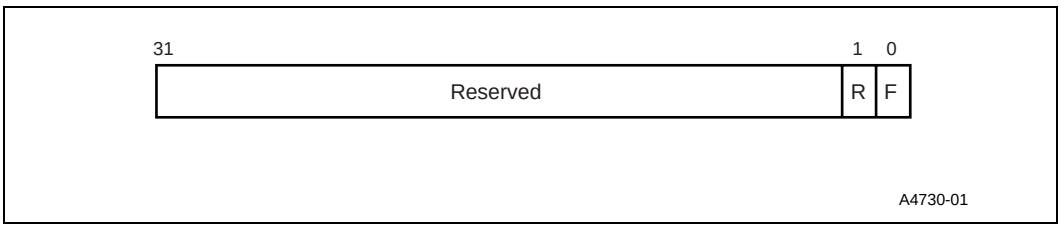
In the reverse direction, when the “R” bit is set to “1”, the interrupt signal, IntReqDat, goes HIGH whenever the Rx FIFO contains at least one full location. When the “R” bit is set to “0”, IntReqDat goes HIGH whenever the Rx FIFO contains eight or more full locations.

In the forward direction, when the “F” bit is set to “1”, the interrupt signal, IntReqDat, goes HIGH whenever the Tx FIFO has nine or more empty locations. When the “F” bit is set to “0”, IntReqDat goes HIGH whenever the Tx FIFO has 15 or more empty locations.

Note: The signal IntReqEmp goes HIGH when the Tx FIFO has 16 empty locations.

Figure 11-8 shows the bit coding for the FifoLevels register.

Figure 11-8. FifoLevels Register Bit Coding

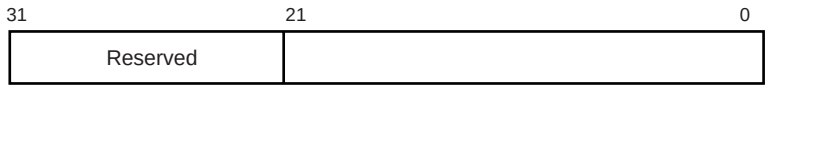


Bit	Name	Function
1	RevFifoLevel	Reverse FIFO level 0 = 8 words 1 = 1 words
0	FwdFifoLevel	Forward FIFO level 0 =16 words 1 = 9 word

11.3.8 InitTime Register

This register is a 22-bit register and defines the initial timer value following an interrupt. Figure 11-9 shows the bit coding for the InitTime register.

Figure 11-9. InitTime Register Bit Coding

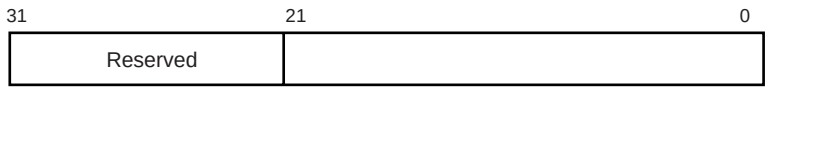


Bit	Name	Function
21:0	TimValInit	Initial timer value following an interrupt

11.3.9 TimerStatus Register

This is a 22-bit register and is the current timer value. Figure 11-10 shows the bit coding for the TimerStatus register.

Figure 11-10. TimerStatus Register BIt Coding



Bit	Name	Function
21:0	TimValStat	Current timer value

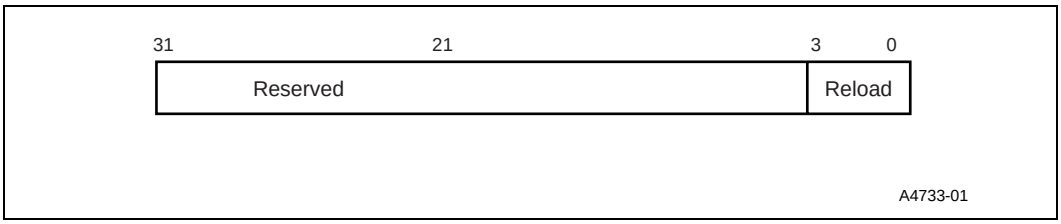
11.3.10 FIFO Reset Register

A write of any value to this address location will reset the Transmit FIFO.

11.3.11 ReloadValue Register

This is a 4-bit register containing the counter reload value for the programmable RefClk divider. Figure 11-11 shows the bit coding of the ReloadValue register.

Figure 11-11. ReloadValue Register Bit Coding



Bit	Name	Function
3:0	Reload	Reload value for programmable **RefClk** divider

11.4 Test Registers

Additional registers are provided in the IEEE 1284 parallel port interface to provide efficient test coverage. Note that these registers are for production test purposes only and should not be accessed during normal operation. Table 11-11 is a summary of the test registers.

Table 11-11. Test Registers Summary

Address Offset	Type	Width	Name	Description
0x2400	R/W	4	TestControl	Enables efficient testing of the timer counter and test clock programmability.
0x2800	R/W	8	TestDataIn	Register containing a value that can be applied to the internal data bus in test mode.
0x2C00	R/W	1	TestDataInEn	When HIGH, the value of TestDataIn register value is applied to the internal databus in test mode.
0x3000	R/W	5	TestCtrlIn	Register containing signal values that can be used to drive the control signals in test mode.
0x3400	R/W	1	TestCtrlInEn	When HIGH, the value of the TestCtrlIn register is applied to the internal control signals in test mode.
0x3800	R	8	TestDataStat	Current IEEE I/O data bus value. Note that this is not a register; it is the value on the peripheral pins for forward transfers and the data register during reverse transfers.

11.4.1 TestControl Register

The TestControl register is a 4-bit register used to control the test features of the peripheral. The large 22-bit timer counter can be partitioned into nibbles and, along with the flexibility of several clocking sources, enables efficient testing of the peripheral.

Bit	Name	Function
3	RegClk	Registered clock bit value. This bit is written with LOW/HIGH values to generate a test mode clock.
2:1	Clock Select	00 – External reference clock. 01 – Bus clock. 10 – Strobe clock (generated by read of test control register). 11 – Registered clock bit.
0	TimerTestModeEnable	When HIGH, the 22-bit timer counter is partitioned into a 2-4-4-4-4-4 nibble configuration, allowing the counter to be efficiently tested.

11.4.2 TestDataIn Register

The TestDataIn register is an 8-bit register that can be programmed with values to be applied to the internal data bus in test mode. The value of the internal data bus can also be read through this register.

Bit	Name	Function
7:0	TestDataIn	In test mode, byte-wide data can be written to and read from the data bus through this register.

11.4.3 TestDataInEn Register

The TestDataInEn register is a 1-bit register used to control whether the internal data bus is driven from the external data bus or by the value contained within the TestDataIn register.

Bit	Name	Function
0	TestDataInEn	Test data in enable 0 – External data drives the internal data bus 1 – TestDataIn register value drives the internal data bus

11.4.4 TestCtrlIn Register

The TestCtrlIn register is a 5-bit register whose value can be used to drive the major internal control signals of the peripheral in test mode. The status of the control signals can also be read through this register.

Bit	Name	Function
4	PError	Mode dependent. See the Functional Description section.
3	nFault	Mode dependent. See the Functional Description section.
2	nAck	Mode dependent. See the Functional Description section.
1	PSel	Mode dependent. See the Functional Description section.
0	Busy	Mode dependent. See the Functional Description section.

11.4.5 TestCtrlInEn Register

The TestCtrlInEn register is a 1-bit register used to control whether the major internal control signals are driven from their source or by the value contained within the TestCtrlIn register.

Bits	Name	Function
0	TestCtrlInEn	Control value in enable 0 – Control bus driven by internal control signals 1 – Control bus driven by TestCtrlIn register value

11.4.6 TestDataStat Location

This is an 8-bit read-only location that can be used to provide the current IEEE I/O data bus value of the peripheral pins. It is the value of the peripheral pins for forward transfers, and the data register value during reverse transfers.

VGA Controller 12

The VGA Controller provides the video output capabilities of the StrongARM® SA-1101 (SA-1101). The VGA Controller provides output for a monitor to complement the following LCD display capabilities of the StrongARM® SA-1100 (SA-1100):

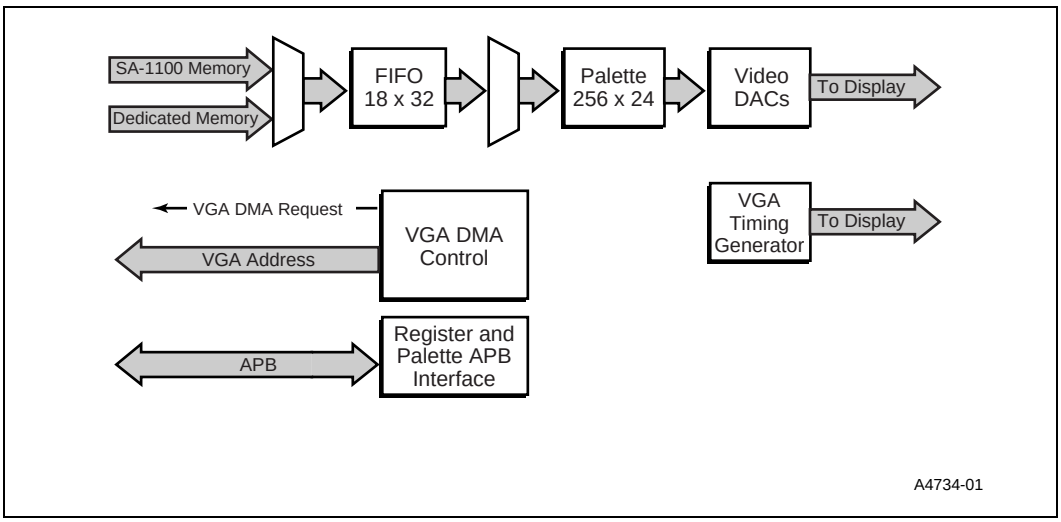
- Resolution programmable up to 1024 x 768 resolution (640x480 maximum for unified display mode)
- Programmable border color
- Programmable VGA timing
- On-chip video DACs for direct drive of a monitor
- Optional dedicated frame buffer (dedicated mode) or shared with SA-1100 frame buffer (unified mode)
- 8 bit per pixel
- 3 x 256 entry 8-bit palette RAM
- Little-endian operation

Note: The VGA Controller does not support interlaced displays. It supports non-interlaced monitor output only. There is no hardware cursor support. The VGA Controller originally was developed for WinCE* and WinCE* does not currently use a hardware cursor.

12.1 Video System Block Diagram

Figure 12-1 shows a block diagram of the video system.

Figure 12-1. Video System Block Diagram



12.2 Functional Description

This section provides a functional overview of the interfaces to the VGA Controller.

12.2.1 VGA Data Path

VGA data is received from either the SA-1100's video memory in unified display memory mode or the SA-1101's video memory in dedicated display memory mode. This data is written into an 18 deep by 32-bit wide asynchronous FIFO. The FIFO data is then extracted into a holding latch, and multiplexed down to a pixel at a time. This logical pixel data is then passed to the palette RAM. The palette RAM is composed of three 256 x 8-bit RAMs. There is a RAM array for each of the R, G and B color components of a pixel. The physical pixel data from the palette RAM is then passed to the three video DACs.

12.2.2 VGA Control Logic

The control logic for the VGA block consists of 3 main blocks.

- A video timing generator block

  Divides down from the video clock and produces the timing control signals (VSync and HSync – whether it is video data or border color to be displayed) for the monitor and video DACs.

- A DMA bus control block

  Generates the DMA address for either the shared or dedicated memory controller and generates the DMA request signal. It also controls receiving the DMA data, and writing it into the VGA FIFO.

- A register slave interface

  Resides on the AMBA Peripheral Bus (APB) and allows registers to be written from the processor. The registers and palette are APB bus slaves. The VGA control registers may not be changed while the video is running unless otherwise indicated in this specification.

The VGA palette may be written to at any time, but may only be read when active video is not being displayed.

12.3 Programmer's Model

This section describes the registers in the VGA Controller.

12.3.1 Video Control Register (VideoControl)

The video control register contains the following control bits for the VGA Controller:

- VGA Controller Enable (VGAEN)

  Stops and starts the video, with immediate effect. Note that most VGA control registers should not be altered unless this bit is set to zero.

- RED, BLUE SWAP (BGR)

  Setting this bit to 1 swaps the red and blue components of the pixels after they have been looked up in the palette. It does not affect the border color.

- Vertical Compare Interrupt Position (VCompVal)

  Allows a choice of four possible locations for the vertical compare interrupt. When the CRT scan reaches this location, an interrupt may be generated as controlled by the interrupt masks. This register may be written to while the video is running.

- VGA DMA Request Point (VGAReq) and Burst Length (VBurstL)

  Controls the DMA access to video data in the shared and dedicated DRAMs. For unified display memory mode, when there are only 2 + VGAReq words left in the FIFO, the video performs a DMA request for (VBurstL + 1) words. This implies that VBurstL must be programmed at most to be 18 - (2 + VGAReq) -1 = 15 - VGAReq. A value less than the maximum may be useful for better overall system performance. The value chosen for VGAReq depends on system performance and video resolution. The ideal values can be determined by using the VFUF interrupt. A safe value for VGAReq is 6, which works in all modes. VGAReq has a maximum value of 0xA; any value greater than this has the same effect as 0xA.

  For dedicated display memory mode, the DMA request mechanism automatically uses the optimum bandwidth of the dedicated DRAM. For this mode programmed at VGAReq to 0xA, VBurstL is not used.

- Video DMA Mode (VMode)

  Selects unified display memory mode or dedicated display memory mode.

- Palette Read Control (PalRead)

  When set to 1 the CPU may read the palette, but active video cannot be displayed. When set to 0, the CPU cannot read the palette but active video will display correctly. If it is necessary to read the palette while the video is running, this bit may be set during blanking and then back to 0 before active video starts.

The following table describes the video control register:

Bit	Name	Function
0	VgaEn	VGA Controller enable. 0 – VGA Controller disabled. 1 – VGA Controller enabled. This bit may be changed while the video is running.
1	BGR	0 – RGB normal video output. 1 – BGR red and blue swapped.
3.2	VCompVal	Generate interrupt at: 00 – Start of VSync. 01 – Start of BACK PORCH. 10 – Start of ACTIVE VIDEO. 11– Start of FRONT PORCH. These bits may be changed while the video is running.
7..4	VgaReq	FIFO emptiness level for DMA request. (0x0 to 0xA)
11..8	VBurstL	Video DMA burst length. Maximum = 15-VgaReq.
12	VMode	DMA mode. 0 = Unified display memory mode. 1 = Dedicated display memory mode.
13	PalRead	0 = Allow video to read palette. 1 = Allow CPU to read palette.
31..14	—	Reserved.

12.3.2 VGATiming 0 Register (VgaTiming0)

This register contains the following four bit-fields that are used to control horizontal VGA timing.

- Pixels-per-line (PPL)

  The pixels-per-line (PPL) bit-field is used to specify the number of pixels in each line or row on the screen. PPL is a 6-bit value that represents between 16–1024 pixels-per-line. Program the value required divided by 16, minus one.

- Horizontal Sync Pulse Width (HSW)

  The 8-bit horizontal sync pulse width (HSW) field is used to specify the pulse width of the horizontal synchronization pulse. Program the value required minus one.

- Horizontal Front Porch (HFP)

  The 8-bit horizontal front porch (HFP) field is used to specify the number of pixel clock periods between the last active video pixel and the start of the next horizontal synchronization pulse. Program to the exact value required.

- Horizontal Back Porch (HBP)

  The 8-bit horizontal back porch (HBP) field is used to specify the number of pixel clock periods between the end of the synchronization pulse and the first pixel of active video. Program the value required minus two.

Bit	Name	Description
1..0	—	Reserved.
7..2	PPL	Pixels-per-line. Number of pixels per line, divided by 16, minus one.
15..8	HSW	Horizontal sync pulse width. Number of clock periods to assert the horizontal sync pulse minus one.
23..16	HFP	Horizontal front porch. Number of clock periods of blank after the last pixel of active video, before the sync pulse occurs. Exact value.
31..24	HBP	Horizontal back porch. Number of clock periods of blank after the horizontal sync pulse, before the first set of pixels is output to the display minus two.

12.3.3 VGATiming 1 Register (VgaTiming1)

This register contains the following four bit-fields that are used to control VGA vertical timing parameters.

- Lines Per Screen (LPS)

  The lines per screen (LPS) bit-field is used to specify the number of lines. LPS is a 10-bit value that represents between 1–1024 lines per screen. The register is programmed with the number of lines per screen minus one.

- Vertical Sync Pulse Width (VSW)

  The 6-bit vertical sync pulse width (VSW) field is used to specify the pulse width of the vertical synchronization pulse as a multiple of the total horizontal period. The register is programmed with the number of lines of VSync minus one.

- Vertical Front Porch (VFP)

  The 8-bit vertical front porch (VFP) field is used to specify the number of lines to wait after the last line of active video and before the start of the next vertical synchronization pulse. Program the value required minus one

- Vertical Back Porch (VBP)

  The 8-bit vertical back porch (VBP) field is used to specify the number of lines to wait after the vertical synchronization pulse and before the start of the first line of active video (not border). Program the value required minus one.

Bit	Name	Description
9..0	LPS	Lines per screen. Number of lines per screen. Program to number of lines required minus one.
15..10	VSW	Vertical sync pulse width. Number of lines of vertical sync, minus one.
23..16	VFP	Vertical front porch. Number of inactive lines at the end of frame, before VSync period. Program to number of lines minus one.
31..24	VBP	Vertical back porch. Number of inactive lines at the start of a frame, after VSync period. Program to number of lines minus one.

12.3.4 VGA Timing 2 Register (VgaTiming2)

This register controls various functions associated with the timing and control of the VGA Controller. The VGA timing 2 register contains the following control bits:

- Invert Vsync (IVS)

  The Invert VSync (IVS) bit is used to invert the polarity of the VSync signal. When IVS=0, VSync is active LOW. When IVS=1, VSync is active HIGH.

- Invert Hsync (IHS)

  The Invert HSync (IHS) bit is used to invert the polarity of the HSync signal. When IVS=0, HSync is active LOW. When IVS=1, HSync is active HIGH.

- Composite VSync (CVS)

  When this bit is set, the VSync signal outputs a composite sync comprised of HSync XOR VSync. If the IVS bit is set, it will invert this to produce the XNOR of the syncs.

- Composite HSync (CHS)

  When this bit is set, the HSync pin outputs the logical OR of VSync and HSync. If IHS is set it will output the NOR of the syncs.

Bit	Name	Description
0	IVS	Invert VSync 0 – VSync is a negative sync. 1 – VSync is a positive sync.
1	IHS	Invert Hsync 0 – HSync is a negative sync. 1 – HSync is a positive sync.
2	CVS	Composite VSync Output XNOR of HSync and VSync on VSync pin (XOR if IVS is set).
3	CHS	Composite HSync Output AND of HSync and VSync on HSync pin (NAND if IHS is set).
31..4	—	Reserved

12.3.5 VGA Timing 3 Register (VgaTiming3)

This register is used to program the timing of the VGA border display. It consists of the following 4 8-bit fields, which specify the number of pixel clocks or line periods after the start of the porch periods that the border starts or stops:

- VGA Horizontal Border Start Register (HBS)

  The HBS contains the number of pixel clocks after the start of the horizontal back porch period, that the border display starts, minus two.

- VGA Horizontal Border End Register (HBE)

  The HBE contains the number of pixel clocks after the start of the horizontal front porch that the border color will be displayed for, minus one. The special case of zero means no border.

- Vertical Border Start Register (VBS)

  The VBS contains the number of lines after the vertical back porch begins that the border display starts, minus one.

- Vertical Border End Register (VBE)

  The VBE contains the number of lines after the vertical front porch begins that the border will be displayed for minus one. The special case of zero means no border.

Bit	Name	Description
7..0	HBS	Horizontal border start
15..8	HBE	Horizontal border end
23..16	VBS	Vertical border start
31..24	VBE	Vertical border end

12.3.6 VGA Border Color Register (VgaBorder)

This register contains the 24-bit RGB color that is shown in the border region of the display.

Bit	Name	Description
23..0	BCOL	RGB 8:8:8 border color
31..24	—	Reserved

12.3.7 VGADMA Base Address Register (VgaDBAR)

This register is a read/write register used to specify the base address of the off-chip frame buffer for the VGA. The address is a byte address, which must always be aligned to a 32-bit word and is subject to an offset in the Video Memory Controller.

In dedicated display memory mode, this is the base address in the dedicated frame buffer memory and will normally be set to all zeros. In unified display memory mode, this value together with the value in the shared memory control register defines the base address of the frame buffer in SA-1100 memory. See Chapter 3, “Interface and Shared Memory Controller”.

Bit	Name	Description
20..0	VgaDBAR	VGA DMA channel base address pointer. Byte address bottom 2 bits always zero.
31..19	—	Reserved.

12.3.8 VGADMA Channel Current Address Register (VgaDCAR)

This read-only register allows the processor to read the current value of the VGADMA channel current address register. Its value cannot be expected to be exact; it could change at any moment. However, its contents can be read to determine the approximate line that the VGA [1]ontroller is currently displaying.

Bit	Name	Description
20..0	VgaDCAR	VGADMA channel current address pointer. Byte address bottom 2 bits always zero.
31..21	—	Undefined.

1.

12.3.9 VGA Status Register (VgaStatus)

Each bit of this register is a status bit that can generate an interrupt. The VGA Status Register (VGAStatus) contains bits that signal an under-run error for the FIFO, the DMA next base update ready status, and the CRT scan reading a programmed point on the screen. Each of these hardware-detected events can generate an interrupt request to the interrupt controller.

This register contains the following control bits:

- FIFO Underflow Status (VFUF)

  The VGA FIFO underflow status (VFUF) status bit is set when the VGA FIFO under-runs. The status bit is “sticky” — it remains set after the FIFO is no longer under-running. The status bit is cleared by writing a ‘1’ to this bit of the status register.

- VGA Next Frame (VNext)

  The VGA Next Frame (VNext) is a read-only status bit that is set after the contents of the VGADMA base address register is transferred to the VGADMA current address register; it is cleared when the VGADMA base address register is written.

- VComp Interrupt (VComp)

  This bit is set when the VGA timing generator reaches the vertical region programmed in the video control register. This bit is “sticky” — it remains set until it is cleared by writing a “1” to this bit of the status register.

Bit	Name	Description
0	VFUF	FIFO underflow status bit.
1	VNext	VGA next base address update status bit. This status bit is set when the base address is transferred to the current address register at the start of frame.
2	VComp	Vertical compare status bit.

12.3.10 VGA Interrupt Mask Register (VgaInterruptMask)

Each bit of this register masks the corresponding bit in the VGA Status Register.

Bit	Name	Description
0	VFUFMask	Masks off the FIFO underflow status bit.
1	VNextMask	Masks off the VGA next base address update status bit when the base address is transferred to the current address register at the start of frame.
2	VCompMask	Masks off the vertical compare status bit.

12.3.11 Vga Interrupt Register (VgaInterrupt)

This register is the logical AND of the VGA Status Register and the VGA Interrupt Mask Register. The interrupt output from the VGA Controller is the logical OR of the bits within the VGA Interrupt Register.

Bit	Name	Description
0	VFUFInterrupt	Indicates that a VFUF interrupt has occurred.
1	VNextInterrupt	Indicates an interrupt has occurred when the base address is transferred to the current address register at the start of frame.
2	VCompInterrupt	Indicates a vertical compare interrupt has occurred.

12.3.12 VGA Palette Registers (VgaPalette)

These register entries have 24 valid bits because there are 8 bits per color gun. The registers may be written to at any time, but the reading is controlled as specified in the video control register.

Bit	Name	Description
7..0	R	Red palette data
15..8	G	Green palette data
23..16	B	Blue palette data
31..24	—	Reserved

12.3.13 DAC Control Register (DacControl)

This register controls the functions of the DAC.

Bit	Name	Description
0	DACON	1 – DACs on 0 – DACs power down
1	COMPON	1 – Test mode 0 – Normal mode
2	PEDON	Pedestal current Set to zero
3	—	Reserved
8..4	RTrim	Red current trim Set to zero
13..9	GTrim	Green current trim Set to zero
18..14	BTrim	Blue current trim Set to zero
31..19	—	Reserved

12.3.14 Register Map Locations

Table 12-1 shows the registers associated with the VGA Controller and the physical addresses used to access them.

Table 12-1. VGA Controller Register Map Locations

Address	Name	Description
VGABase + 0x0000	VideoControl	Video control register
VGABase + 0x0400	VgaTiming0	VGA timing 0 register
VGABase + 0x0800	VgaTiming1	VGA timing 1 register
VGABase + 0x0C00	VgaTiming2	VGA timing 2 register
VGABase + 0x1000	VgaTiming3	VGA timing 3 register (border timing)
VGABase + 0x1400	VgaBorder	VGA border color register
VGABase + 0x1800	VgaDBAR	VGA DMA channel base address register
VGABase + 0x1C00	VgaDCAR	VGA DMA channel current address register
VGABase + 0x2000	VgaStatus	VGA status register
VGABase + 0x2400	VgaInterruptMask	VGA interrupt mask register
VGABase + 0x2800	VgaInterrupt	VGA interrupt register
VGABase + 0x3000	DacControl	DAC control register
VGABase + 0x40000 – VGABase + 0x7FC00	VGAPalette	VGA palette programming registers

12.4 Test

There is a test mode of operation for testing the VGA Controller.

12.4.1 VGA Controller Test Register

The VGA Controller test register contains bits that allow certain VGA signals to be output for test purposes. This register should not normally be used. The register is reset to all zero, and this will result in normal operation.

Bit	Name	Description
0	TDAC	1 – DAC inputs connected to external pins
3..1	Datatest	Output video data in test mode on every 2nd clock 000 – Non test mode 001 – Red output 010 – Green output 011 – Blue output 1XX – DAC TESTOUT output
4	DAC TESTDAC	DAC's TESTDAC signal
7..5	DAC TESTOUT	DAC's TESTOUT signals (read-only)
31..8	—	Reserved

12.4.2 Memory Map

Table 12-2 shows the registers associated with the VGA Controller and the physical addresses used to access them.

Table 12-2. VGA Controller Test Registers

Address	Name	Description
VGABase+0x2C00	VgaTest	VGA test register

12.5 Characteristics

Table 12-3 lists the preliminary dc characteristics for the DACs.

Table 12-3. DAC DC Characteristics

Parameter	Minimum	Typical	Maximum	Unit
DAC resolution	—	8	—	bits
Power Supply				
Input reference current	—	77.9	—	µA
Accuracy				
DAC to DAC gain mismatch	—	—	4	%
Integral non-linearity	—	—	± 12	LSB
Differential non-linearity	—	—	± 0.5	LSB
DAC Output				
Full scale current	13	18.6	32	mA
Output disabled current	—	0	—	mA
Output compliance	0	—	1.3	V
Output capacitance	—	—	12	pF
Load resistance	—	37.5	—	Ohm

Table 12-4 lists the preliminary ac characteristics for the DACs.

Table 12-4. DAC AC Characteristics

Parameter	Minimum	Maximum	Unit	Note
DAC Switching				
Output delay	1.9	3.3	ns	
Output rise time	0.7	1.1	ns	1
Output fall time	1.0	1.4	ns	1
Pixel clock	—	80	MHz	

Note: 1. 10% to 90% time

12.6 Application

The following table lists timings for three examples of display modes. The display modes correspond to the VESA display modes of 640x480 @ 72 Hz (with a 32-MHz video clock), 800x600 @ 72 Hz (with a 48-MHz video clock), and 1024x768 @ 70 Hz (with a 72-MHz video clock). The timings are adjusted to compensate for the difference between the video clock rate and the VESA-specified clock rate for the resolution.

Register	640x480 Display Mode	800x600 Display Mode	1024x768 Display Mode
VgaTiming0	0x7f17279C	0x242D72C4	0x641382FC
VgaTiming1	0x000809df	0x16001257	0x1C0216FF
VgaTiming2	0x00000000	0x00000003	0x00000000
VgaTiming3	0x07130777	0x00000000	0x00000000
VgaControl	0x00001B41	0x00001B41	0x000018A1

12.6.1 IREF Resistor

The recommended connection of the IREF pin is with a 24-K resistor to VDD.

USB Host Interface Controller 13

The Universal Serial Bus (USB) is a cable bus that support data exchange between a host computer and a range of simultaneously accessible peripherals. The attached peripherals share USB bandwidth through a host scheduled token-based protocol. The bus allows peripherals to be attached, configured, used, and detached while the host and peripherals are in operation.

The USB Host Controller built into the StrongARM® SA-1101 (SA-1101) is Open Host Controller Interface (OHCI)-compatible, Windows95* USBD-compatible, and USB Revision 1.0-compatible. It supports both low-speed (1.5 Mbps) and high-speed (12 Mbps) USB devices. The SA-1101 drivers have not been characterized at this time.

See the *Univeral Serial Bus Specification Revision 1.0* and the *Open HCI – Open Host Controller Specification for USB* for details of the interface operation.

13.1 Functional Description

This section provides a functional overview of the USB Host Controller.

13.1.1 USB Reset

The USB Host Controller is not fully reset following an SA-1101 reset. This reset leaves the USB ForceHcReset and the USB ForceIfReset resets active. See Section 13.2.1, “Status Register” on page 13-4. To initialize the USB Host Controller, follow this sequence:

1. Start the UCLK12.
2. Wait at least 10 µs.
3. Stop the UCLK12.
4. Set the ForceIfReset bit to 0.

The USB Host Controller is initialized. Before using it to transmit or receive data:

1. Start the UCLK12.
2. Write 0x00000800 to the Test Command Setup Register (USTCSR).
3. Set the ForceHcReset to 0.

The USB Host Controller is operational.

13.1.2 USB Suspend

The USB specification defines a power conversation mechanism called suspend. Devices and hubs can be placed into the suspend state; if required, the whole system can be suspended (global suspend).

A device enters the suspend state if it does not receive an SOF (keep awake) packet greater than 3 ms. Software needs to specifically invoke the suspend state, otherwise devices will receive 1-ms keep awake SOF packets.

If there is no USB activity, the host can invoke the global suspend state, which causes all connected devices and hubs to go into their suspend states within 5 ms. The host can reenable the bus by invoking the resume state. If a device requires attention, it can cause a resume by sending the host a remote resume event (idle → K-state bus transition).

The USB specification does not define specific requirements for the suspend state, other than a device should not consume more than 500 µA.

Note: Terminate resistors draw most of this current when in the suspend state.

13.1.3 Power Management

The Open Host Controller Interface Specification for the USB defines a port power switching mechanism. A brief summary of the salient points follows:

- All ports can be continuously powered or the power can be switched.
- Power switching can be global (all ports) or individually switched.
- Power switching does not depend on the state of the port (connected, speed, enabled, and so forth).

Power-enable features only can be operated while the USB clock is running. The USB clock must be running to disable or enable the power-enable features.

The SA-1101 USB has the following features that support the power-conservation features of the Open Host Controller Interface:

- USB clock stopping
- Port power enable
- Port resume interrupt

13.1.3.1 USB Clock Stopping

The USB clock can be stopped at any time. It is recommended to stop the clock only when the USB is in the global suspend state. The global suspend state is reached when there has been no activity on the USB for greater than 5 ms and the host controller is in the suspend state. In the suspend state, the port (PAD) also will be in its low-power mode. See Section 10.2.1, “Power Control Register (SKPCR)” on page 10-2. If a device is not connected to the USB port, the host controller cannot be suspended and the port (PAD) will not be in its low-power mode.

13.1.3.2 Port Power Enable

To save power when a device is not conntected to the USB port (PAD), the port power enable can be disabled before the USB clock is stopped.

13.1.3.3 Port Resume Interrupt

The port resume interrupt does not depend on the USB clock running. It indicates that an event has occurred on the USB bus that requires the USB clock to be restarted. If the USB clock is running, the host controller interrupts handle the necessary interrupt conditions.

Three events can cause a UsbPortResume interrupt:

- A device is connected.
  Indicated by one of the USB port signals being pulled high.
- A port is disconnected.
  Indicated by both the USB port signals being pulled low.
- A remote resume signal from a currently connected dive.
  Indicated by an idle → K-state transition on the USB bus.

The port resume interrupt will go active if any one of these events occurs. Software determines when this interrupt is valid. See Section 14.2.1.1, “Source Bit Positions” on page 14-5.

13.1.4 Power-Management Routines

This section lists suggested power-management routines.

13.1.4.1 Initial Port Power-Down Sequence

To initialize a port power-down sequence, follow this sequence:

1. Wait for the device hardware reset sequence to complete. See Section 13.1.1, “USB Reset” on page 13-1.

   The USB clock is enabled.
2. Send a software reset to the host controller and wait 10 µS.
3. Clear the NoPowerSwitching bit.
4. Clear the global power enable bit.
5. Stop the USB clock.

Note: Although the UsbPortResume interrupt will be generated when a device is connected, it is necessary to set the global power enable bit before the USB port can be used.

13.1.4.2 Low-Power Mode in Suspend State Sequence

To activate the low-power mode, follow this sequence:

1. Wait for the USB to be in the global suspend state.
2. Set the port suspend bit. This may be automatic after 5 ms inactivity on the USB bus.
3. Wait for the port suspend bit to bo active.
4. Stop the USB clock.

Note: Software waits for the UsbPortResume interrupt to go active. Start the USB clock and the host controller restarts.

13.1.4.3 Low-Power Mode after Device Disconnect Sequence

To activate the low-power mode, follow this sequence:

1. Wait for port connection status to go inactive or wait for the UsbPort Resume interrupt if the USB clock is stopped. Check connection status.
2. Clear the global power enable bit.
3. Stop the USB clock.

Note: The port is powered-down. See Section 13.1.4.1, “Initial Port Power-Down Sequence” on page 13-3. Software waits for the UsbPort Resume interrupt before restarting the USB clock. If the USB clock is stopped in any other mode than described above, the host controller and any connected device may need reinitialization.

13.2 Programmer’s Model

The USB incorporates the operational registers of the Open Host Controller Interface Specification for the USB. In addition, the SA-1101 has status and reset registers that are described in this section.

13.2.1 Status Register

This register enables the user to monitor the active state of each of the interrupt sources. These bits are not latched, so care must be taken when interpreting their state.

Bit	Name	Reset	Function
6:0	—	0	See the Section 13.4.6, “Status Register (Test) (USSR)” on page 13-8.
7	IrqHciRmtWkp	0	1 = HCI remote wake-up event. This bit pulses active with a RemoteWakeUp event on one of the downstream ports of the root hub. It pulses when the host controller moves from the suspend to resume state. This signal only goes active if the RWE bit in the HcControl register is set. A latched version of this bit is available in the interrupt controller.
8	IrqHciBuffAcc	0	1 = HCI buffer active. This bit is active while the host controller is accessing the data buffer for the current TD or when an ED is being accessed (the status signal to let the application know if the host controller is reading or writing the shared memory).
9	nIrqHciM	0	0 = Normal HC interrupt active. See the Open Host Controller Interface specification.
10	nHciMFCIr	0	0 = HCI interface clear signals active. This bit pulses active when the host controller can no longer start or continue the current host controller interface (HCI) transfer (for example, during a reset while an HCI transfer is in progress). It is active for a minimum of 10 µs until the host controller reenters.
31:11	—	—	Reserved.

13.2.2 Reset Register

This register provides the user with a mechanism to individually reset the uhost core (HcReset) and the ASB master interface (IfReset). An external reset sets the ForceIfReset and the ForceHcReset bits. The ClkGenReset bit, in general, only is needed for simulation and test.

Bit	Name	Reset	Location
0	ForceIfReset	1	When set, forces a reset to the ASB master interface. Must be cleared for the interface to function. The IfReset is driven also by the system reset nSyncedBRes. This bit is reset to the active state.
1	ForceHcReset	1	When set, forces a reset to the USB Host Controller (SAND block). Must be cleared for the host controller to function. The HCReset is driven also by the system reset nSyncedBRes. This bit is reset to the active state.
2	ClkGenReset	0	When set, this bit forces the host controller clock generation block into reset. It can be used by the test interface to guarantee a deterministic behavior for the host controller block. The ClkGenReset is not driven by the system reset nSyncedBRes. This bit is reset to the inactive state.
31:3	—	—	Reserved.

13.3 Register Memory Map

Table 13-1 lists the addresses of the control bits in the USB host interface controller. See the Memory Map chapter for the base address of the USB host interface controller.

Table 13-1. USB Host Interface Register Memory Map

Address	Name
USB Base	Revision
Base + 0x0400	Control
Base + 0x0800	CommandStatus
Base + 0x0C00	InterruptStatus
Base + 0x1000	InterruptEnable
Base + 0x1400	InterruptDisable
Base + 0x1800	HCCA
Base + 0x1C00	PeriodCurrentED
Base + 0x2000	ControlHeadED
Base + 0x2400	ControlCurrentED
Base + 0x2800	BulkHeadED
Base + 0x2C00	BulkCurrentED
Base + 0x3000	DoneHead
Base + 0x3400	FmInterval
Base + 0x3800	FmRemaining
Base + 0x3C00	FmNumber
Base + 0x4000	PeriodicStart
Base + 0x4400	LSThreshold
Base + 0x4800	RhDescriptorA
Base + 0x4C00	RhDescriptorB
Base + 0x5000	RhStatus
Base + 0x5400	RhPortStatus[1]
Base + 0x11800	Status register
Base + 0x11C00	Reset register

13.4 Test Interface

The test interface allows the USB HCI-Master and ASB-Master interfaces to be tested without having to set up the USB Host Controller. The address locations are used by the tester to generate pseudo-HCI transfers over the ASB bus and out through the memory interface. The test interface only is active when both the external test pin and the TestXferSel register bit are active.

13.4.1 Transfer Address Register (USTAR)

A 26-bit value written to this location specifies the burst start address for both a read and a write transfer. This write location also strobes the HciMAdrFln signal.

13.4.2 Write Data FIFO Register (USWER)

A 32-bit value written into the Write Data FIFO causes the HciMDataFln signal to be strobed.

13.4.3 Read Data FIFO Regsiter (USRFR)

This is a 32-bit read value that access the current read data location.

13.4.4 Next Data FIFO Register (USNFR)

Reading this location increments the Read Data FIFO to the next location. The read data returned should be ignored; it could either be the current or the next FIFO location data.

13.4.5 Test Command Setup Register (USTCSR)

These bits enable the tester to set up the pseudo-HCI transfer. The TestXferSel bit must be active to enable the test interface.

Bit	Name	Function
2:0	RdBstCntr[2:0]	HCI read burst size.
6:3	ByteEnable[3:0]	HCI write byte channel active enable.
7	WriteEn	HCI write transfer qualifier. Set to active for a write cycle.
8	FifoCir	HCI force clear signal. Set to zero to simulate HCI and forcing an interface clear.
9	TestXferSel	When set to one, enables the HCI interface test operation.
10	FifoCirAtEnd	When set to one, causes the HCI force clear signal to be negated while the nTestAddrStrb is active. This enables the test interface to fully exercise the HCI-Master interface FifoCirFSM.
11	nSimScaleDownClk	When set to one, scales down the 1ms-interval clock in the host controller. Used for test and simulation.
31:12	—	Reserved.

13.4.6 Status Register (Test) (USSR)

This register provides the test with information about the status of the ASB-HCI Master interface. These status bits also are available to the user when the interface is not in test mode, in which case their meaning can be interpreted only when the interface is inactive.

Bit	Name	Function
0	nAppMDEmpty	0 = HCI-Master I/F read data FIFO empty.
1	nAppMDFirst	0 = HCI-Master I/F read data FIFO one from empty.
2	nAppMDLast	0= HCI-Master I/F write data FIFO last location available.
3	nAppMDFull	0 = HCI-Master I/F write data FIFO full.
4	nAppMAFull	0 = HCI-Master I/F transfer in progress.
5	UsbXferReq	HCI-Master I/F request handshake signal.
6	UsbXferEnd	HCI-Master I/F transfer end handshake signal.
10:7	—	See the Status Register section.
31:11	—	Reserved.

13.4.7 Test Memory Map

Table 13-2 lists the read and write locations in the test memory map for the USB.

Table 13-2. Test Memory Map for the USB

Address	Read Location	Write Location
Base + 0x10400	—	USTAR
Base + 0x10800	—	USWFR
Base + 0x10C00	USRFR	USRFR
Base + 0x11000	USNFR	—
Base + 0x11400	USTCSR	USTCSR
Base + 0x11800	USSR	—

Interrupt Controller 14

The interrupt controller generates the interrupt signals from the StrongARM® SA-1101(SA-1101) in a highly flexible manner. All interrupt sources from the system are processed with full control over the polarity of each bit. The overall interrupt structure is flat for all 64 hardware interrupt sources. The interrupt controller has two modes of operation:

- **Normal**

  The creation of interrupt edges on the SA-1101's INT pin suitable for connection to the StrongARM® SA-1100(SA_1100) interrupt controller. Full facilities are available for processing the raw interrupt sources including polarity, enabling, setting and clearing.

- **Wake-Up**

  The creation of a wake-up signal on the SA-1101's INT pin. This signal is simply the logical OR of all enabled interrupt sources. The wake-up mode enables an interrupt signal to be generated without any clocking of the interrupt controller (ICLK). It is the only mode available when the SA-1101 is in normal mode.

14.1 Functional Description

This section provides a functional overview of the interrupt controller.

14.1.1 Logic Diagrams

Figure 14-1 shows how the **INT** output is processed from the interrupt source.

Figure 14-1. Interrupt and Wake-Up Generation

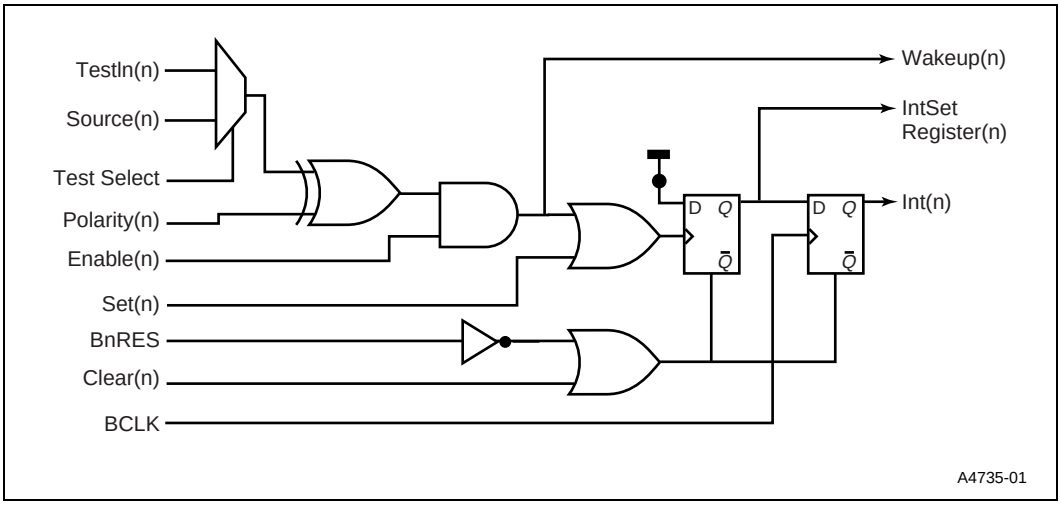
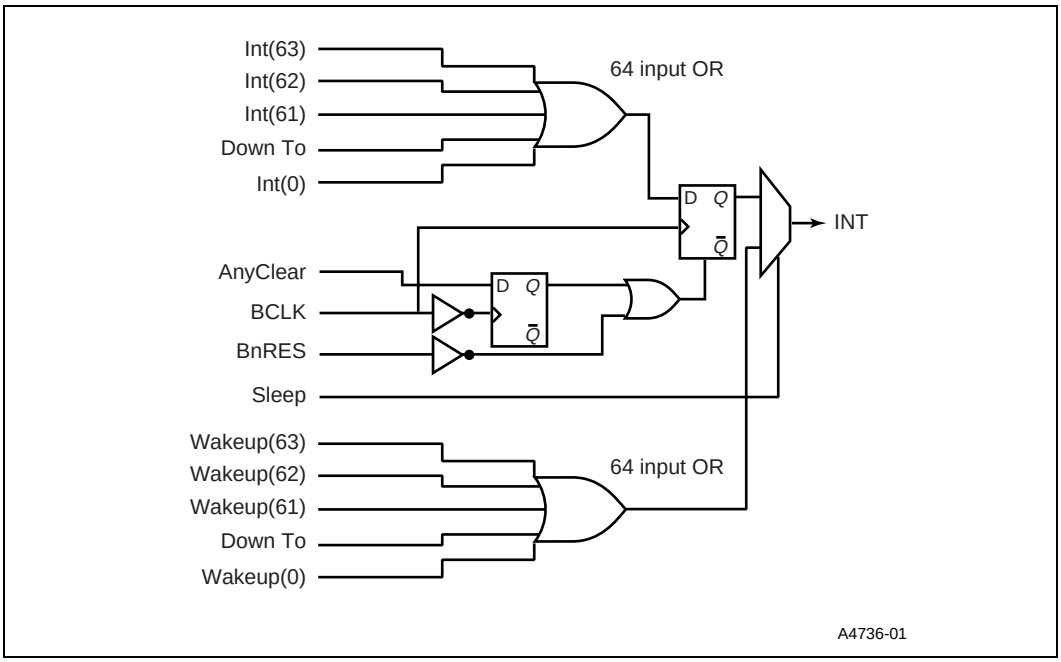


Figure 14-2 shows how the output from each bit combination circuit is then combined.

Figure 14-2. Interrupt and Wake-Up Combination



14.1.2 Interrupt Sources

The interrupt controller handles all interrupt sources throughout the SA-1101. Table 14-1 lists the interrupt sources handled by the interrupt controller.

Table 14-1. Interrupt Sources

Interrupt Number	Name	Source
7:0	GpAIn[7:0]	GPIO interface
14:8	GpBIn[6:0]	GPIO interface
15	—	Reserved
23:16	KPXIn[7:0]	Keypad interface
39:24	KPYIn[15:0]	KeyPad interface
40	MsTxInt	PS/2 mouse interface
41	MsRxInt	PS/2 mouse interface
42	TpTxInt	PS/2 trackpad interface
43	TpRxInt	PS/2 trackpad interface
44	IntReqTrc	IEEE 1284 interface
45	IntReqTim	IEEE 1284 interface
46	IntReqRav	IEEE 1284 interface
47	IntReqInt	IEEE 1284 interface
48	IntReqEmp	IEEE 1284 interface
49	IntReqDat	IEEE 1284 interface
50	videoint	VGA controller
51	fifoint	Update FIFO
52	nIrqHciM	USB controller
53	IrqHciBuffAcc	USB controller
54	IrqHciRmtWkp	USB controller
55	nHciMFClr	USB controller
56	USBError	USB controller
57	S0_READY_nIREQ	PCMCIA interface
58	S1_READY_nIREQ	PCMCIA interface
59	S0_CDVALID	PCMCIA interface
60	S1_CDVALID	PCMCIA interface
61	S0_BVD1_STSCHG	PCMCIA interface
62	S1_BVD1_STSCHG	PCMCIA interface
63	USB port resume	USB controller

14.1.3 Interrupts

When a valid interrupt occurs (correct polarity on an enabled source) a rising edge will be clocked out on the INT pin. It is recommended that this is connected to GPIO 0 or 1 on the SA-1100. If this source is enabled in SA-1100, then an IRQ or FIQ interrupt occurs depending on the value programmed into the interrupt level register on the SA-1100. The called interrupt routine can access the SA-1101's interrupt controller to find the source of the interrupt. See Section 14.2.6, "INTSTATCLR0 – INTSTATCLR1" on page 14-6.

When the interrupt has been serviced, the appropriate bit should be cleared in the SA-1100's GPIO edge detect status register. The SA-1101's INT pin then can be cleared by clearing the appropriate interrupt bit. See Section 14.2.6, "INTSTATCLR0 – INTSTATCLR1" on page 14-6.

14.1.4 Wake-Up Interrupts

In wake-up mode the logical OR of the interrupt sources is output on the INT pin.The interrupt source may be transient and when it is removed the INT pin also returns to zero. However, the source of the interrupt will have been latched within the controller. To read the source of the interrupt, the ICLK must be restarted and then the latched source will be clocked through. See Section 14.2.6, "INTSTATCLR0 – INTSTATCLR1" on page 14-6. Once the source is clocked through, which take six ICLKs, the interrupt in normal mode is asserted.

Note: If the switching back to normal mode occurs before the interrupt has propagated through the normal interrupt controller, a negative going pulse may be seen on the INT pin, which would signal another interrupt. The interrupt handling software should be designed to accommodate this condition.

14.1.5 Multiple Interrupts

The SA-1101 has a large number of possible interrupt sources and it is expected that in normal operation multiple interrupt requests will occur. The SA-1101's interrupt controller is designed to work in tandem with the SA-1100 interrupt controller to allow the user many options in dealing with this situation. When the INT line is set by one interrupt source, subsequent interrupts will not affect this line. However, once an interrupt has been serviced and the INT line has been cleared, another rising edge will be immediately clocked out. This will continue until no valid interrupt sources are asserted.

Once an SA-1101 interrupt has been requested, software can either service all pending interrupts and then clear the INT line or service the pending interrupts individually using the reassertion of the INT line to trigger another service routine.

14.2 Programmer's Model

This section describes the registers in the interrupt controller.

14.2.1 Registers

The following registers are provided. There are two copies of each 32-bit register to support the 64 interrupt sources.The register name is suffixed with either "0" or "1" to indicate the set being accessed:

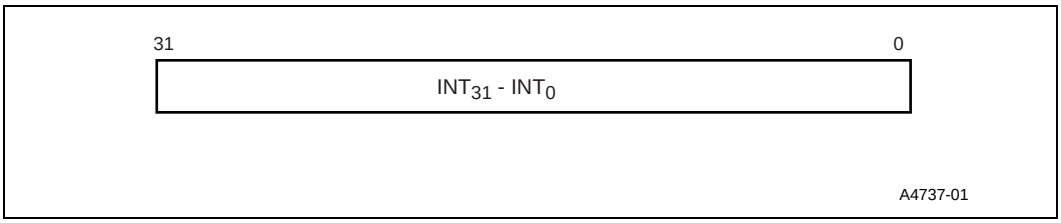
- INTTEST - Test register
- INTENABLE - Interrupt enable (mask)
- INTPOL - Polarity selection
- INTSTATCLR - Read status or write to clear
- INTSET - Read source or write to set

In addition, a single test mode select register, INTTSTSEL, is provided for test mode selection.

14.2.1.1 Source Bit Positions

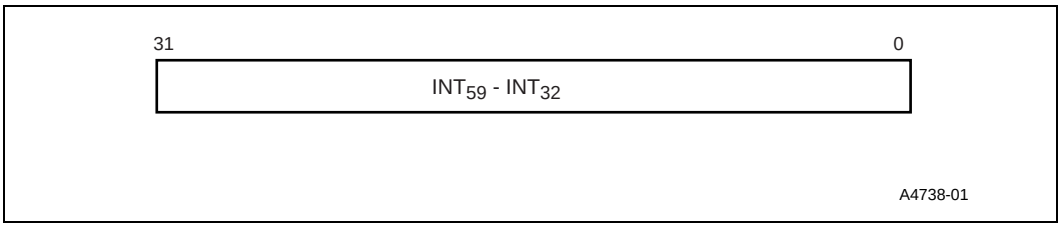
Figure 14-3 shows the bit positions for the interrupt sources in set 0 of the registers and Figure 14-4 shows the interrupt sources in set 1.

Figure 14-3. Interrupt Source Positions in Register Set 0



INT_{31}-INT_0 Hardware interrupt sources 31 to 0

Figure 14-4. Interrupt Source Positions in Register Set 1



INT_{63}-INT_{32} Hardware interrupt sources 63 to 32

14.2.2 INTTEST0 – INTTEST1

These registers should only be used for production test.

Writing to these registers controls internally generated values for each of the hard interrupt sources before the programmable polarity logic. On read, only bit 0 is significant indicating the logical OR of all the interrupt bits. Interrupts bits correspond to register bits as shown in Figure 14-3 and Figure 14-4.

14.2.3 INTENABLE0 – INTENABLE1

These registers control the masking of each status bit. Writing a zero disables that status bit from generating an interrupt and writing a one enables its effect. Reading returns the current enable register value. It is reset to zero (disabled). Interrupts bits correspond to register bits as shown in Figure 14-3 and Figure 14-4.

14.2.4 INTPOL0 – INTPOL1

These registers control the polarity of the source interrupt. Writing a zero causes the interrupt to be set on a rising edge. Writing a one causes the interrupt to be set on a falling edge. Reading the register returns the current polarity value. Interrupts bits correspond to register bits as shown in Figure 14-3 and Figure 14-4.

14.2.5 INTTSTSEL

Bit 0 of this register allows the selection of either the interrupt source or the interrupt test register (Inttest) as the raw input. A one sets the source as the test register.

Bit 1 of this register allows the selection of either normal or wake-up mode for the interrupt controller. A one sets the mode as wake-up and a zero sets the mode as normal. Interrupts bits correspond to register bits as shown in Figure 14-3 and Figure 14-4.

14.2.6 INTSTATCLR0 – INTSTATCLR1

These registers allow the status of the interrupts to be read and the interrupts to be cleared. Writing a one will clear the interrupt, writing a zero will do nothing. On read, a one indicates the interrupt is set and a zero indicates the interrupt is clear. Interrupts bits correspond to register bits as shown in Figure 14-3 and Figure 14-4.

14.2.7 INTSET0 – INTSET1

These registers allow the interrupt sources to be set. Writing a one will set the interrupt value, writing a zero will do nothing. When read, this register returns the value of the interrupt before it has been synchronized and is useful for test purposes only. Interrupts bits correspond to register bits as shown in Figure 14-3 and Figure 14-4.

14.2.8 Memory Map

Table 14-2 provides a summary of the registers.

Table 14-2. Interrupt Registers

Address Offset	Name	Reset	Description
0x1000	INTTTEST0	Yes	Write 1 = Set test interrupt source high
0x1400	INTTTEST1	Yes	Write 1 = Set test interrupt source high
0x2000	INTENABLE0	Yes	Write 0 = Disable
0x2400	INTENABLE1	Yes	Write 0 = Disable
0x3000	INTPOL0	Yes	Write 0 = Active high source
0x3400	INTPOL1	Yes	Write 0 = Active high source
0x5000	INTTSTSEL	N/A	Mode selection
0x6000	INTSTATCLR0	N/A	Read 1 = interrupt set Write 1 = Clear interrupt
0x6400	INTSTATCLR1	N/A	Read 1 = interrupt set Write 1 = Clear interrupt
0x7000	INTSET0	N/A	Write 1 = Set interrupt
0x7400	INTSET1	N/A	Write 1 = Set interrupt

14.3 Characteristics

Table 14-3 lists the characteristics for the interrupt controller.

Table 14-3. Interrupt Characteristics

Symbol	Parameter	Minimum	Maximum	Unit
t_{IL}	Interrupt latency[1]	167	194	ns
t_{IPW}	Minimum pulse width[2]	41	TBD	ns

NOTES:
1. Time from assertion of interrupt controller to assertion of **INT** pin.
2. This condition occurs when an interrupt is cleared and a further interrupt is already pending.

14.4 Application

The interrupt clock must be enabled to operate the interrupt controller in normal mode. See Section 10.2.1, “Power Control Register (SKPCR)” on page 10-2.

PS/2 Trackpad and Mouse Interfaces 15

The trackpad and mouse interfaces are identical, differing only in the names of the external pins. The interfaces are designed to communicate with a standard PS/2 trackpad, keyboard, or mouse, via a two-pin serial link. The trackpad interface uses the pins TPDATA and TPCLK, and the mouse interface uses the pins MSDATA and MSCLK, all of which are open drain.

This chapter will refer to the keyboard pins and register/signal names; the mouse and trackpad interfaces are identical.

15.1 Registers

This section describes the registers in the trackpad and mouse interfaces.

15.1.1 Control Register (KBDCR)

This register is a control that provides direct access to the CLK and DATA outputs and an enable bit to enable the interface. A write access to this location updates the control signals. A read returns the values written.

Bit	Name	Function
3	ENA	Keyboard enable bit. 0 = Disabled. 1 = Enabled.
2	N/A	Reserved. Reads as zero.
1	FKD	Force KBDATA pad low. 1 = Force the pad low regardless of the state of the keyboard FSM.
0	FKC	1 = Force KBCLK pad low regardless of the state of the keyboard FSM.

15.1.2 Status Register (KBDSTAT)

This register is a read-only register that provides status information such as busy state of the FSM, parity of last received data, and so forth.

Bit	Name	Function
7	TXE	Tx. Register empty. 0 = Not ready. 1 = Enabled, ready to transmit.
6	TXB	Tx. Busy. 1 = Currently sending data.
5	RXF	Rx. Full. 1 = RX register full, ready to be read.
4	RXB	Rx. Busy. 1 = Currently receiving data.
3	ENA	Enable bit. 1 = Function enabled.
2	RXP	Parity bit indication for last received data byte (odd parity).
1	KBD	Value on KBDATA pin (after synchronizing).
0	KBC	Value on KBCLK pin (after synchronizing and sampling by Div8 clock).

15.1.3 Transmit/Receive Data Register (KBDDATA)

This register is used both to write bytes to be transmitted across the serial link, and to read bytes received.

Bit	Name	Function
7:0	KBDAT	Tx/Rx data.

15.1.4 Clock Division Register (KBDCLKDIV)

This register is used to define the divide ratio required to generate an 8-MHz keyboard clock (KBCLK) from the RefClk input clock.

Bit	Name	Function
3:0	DivVal	Divide value. The RefClk division ratio is given by the following equation: Divide ratio = DivVal + 1

15.1.5 Clock Precount Register (KBDPRECNT)[8]

This register is used to define the maximum value of the precounter within the keyboard interface. The precounter is used to generate a 32-µs period clock from the 8-MHz keyboard clock.

The KBDPRECNT register is intended to be used in conjunction with the KBDCLKDIV register to ensure the required output frequencies are generated.

Bit	Name	Function
7:0	PreCntMax	Maximum precount value. The division ratio is given by the following equation: Divide ratio = (PreCntMax+ 1) * 2.

15.1.6 Register Memory Map

Table 15-1 lists the read and write locations in the memory map for the trackpad and mouse interfaces. See Chapter 19, “Memory Map,” for the base addresses of the trackpad and mouse interfaces.

Table 15-1. Trackpad/Mouse Register Memory Map

Address	Read Location	Write Location
Base	KBDCR register	KBDCR register
Base + 0x0400	KBDSTAT register	KBDSTAT register
Base + 0x0800	KBDDATA register	KBDDATA register
Base + 0x0C00	KBDCLKDIV register	KBDCLKDIV register
Base + 0x1000	KBDPRECNT register	KBDPRECNT register

15.2 Functional Description

The interfaces generate two interrupts each: one interrupt to indicate that the transmit buffer is empty and thus that another byte can be transmitted (KbdTxInt), and one interrupt to indicate that a byte has been received by the interface (KbdRxIn).

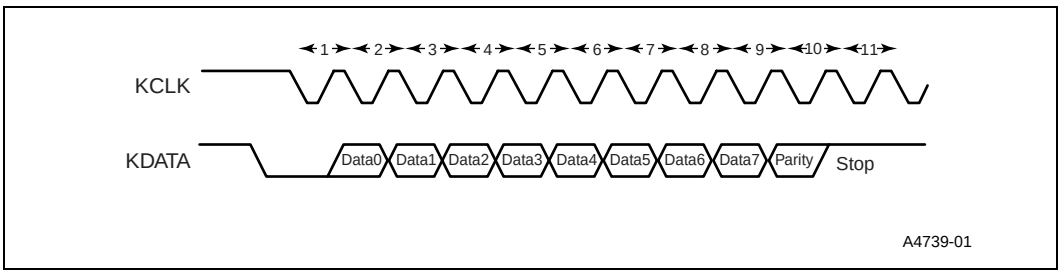
The keyboard interface is held in reset until the enable bit in the control register is set. The interface can be controlled on the basis of the interrupts generated, or by polling the status flags in the control register. The Tx interrupt is generated when the transmit buffer has been emptied and the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

Internally to the block, the KbdClk clock is divided by eight to generate a nominal 1-MHz clock frequency. The KBCLK signal is latched by the 1-MHz clock before it is used internally. This helps removed problems caused by external noise and slow moving edges.

The KBCLK signal is always driven by the keyboard, unless the StrongARM® SA-1100 (SA-1100) wishes to prevent the peripheral from transmitting (because it is about to transmit some data itself). When data is received from the peripheral, the KBDATA line is pulled low as a start bit. Each data bit is set up to the falling edge of the clock. Eight data bits are transmitted from the keyboard/mouse, followed by a parity bit (odd parity) and a HIGH stop bit.

Figure 15-1 shows the protocol of this transfer. Note that receive refers to the controller end, that is, the external peripheral is transmitting.

Figure 15-1. Keyboard/Mouse Controller Receive Protocol



When the controller would like to transmit a byte to the peripheral, the KBCLK line is pulled LOW, then allowed to float and the KBDATA line is pulled LOW, as a request to send. The keyboard then drives the clock, causing the controller to put eight bits of serial data out onto the KBDATA line. A parity bit is driven out, followed by a stop bit, and the stop bit may be acknowledged by the peripheral. (The controller does not check on the acknowledge.) Figure 15-2 shows the timing requirements of the interface.

Figure 15-2. Timing and Controller Request to Send Protocol

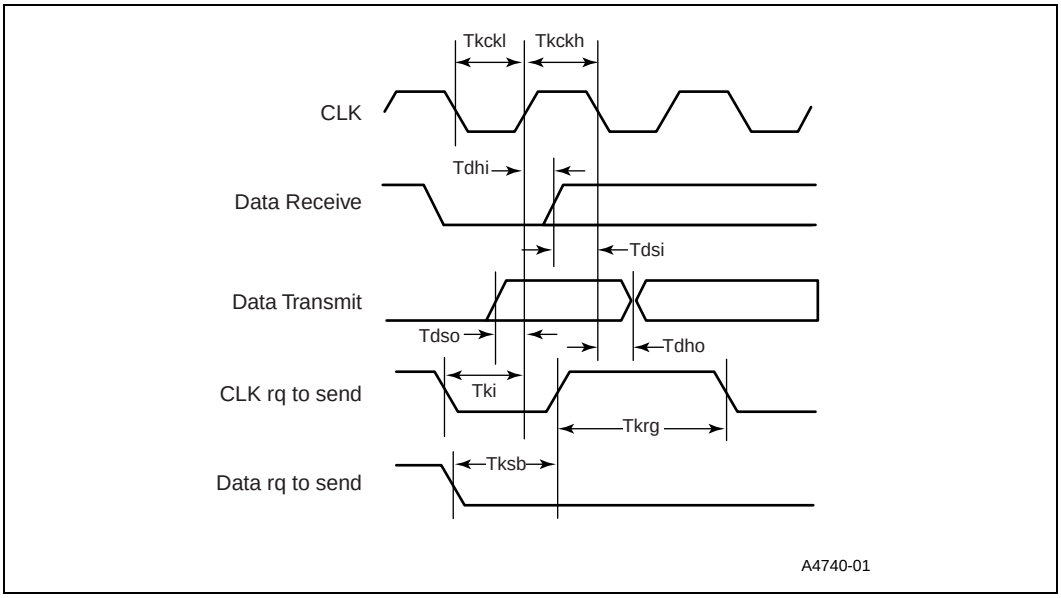


Table 15-2 shows the nominal timings for the interface signals.

Table 15-2. Keyboard/Mouse Interface Timings

Symbol	Parameters	Minimum	Maximum	Notes
Tkclk	Keyboard clock period	1 µs	100 µs	
Tkckl	Keyboard clock low time	0.5 µs	50 µs	
Tkckh	Keyboard clock high time	0.5 µs	50 µs	
Tdsi	Setup on DATA to CLK falling for receive	1 µs	Tkckh – 1 µs	
Tdhi	Hold on DATA from CLK rising for receive	1 µs	Tkckh – 1 µs	
Tdso	Setup on DATA to CLK rising for transmit	Tkckl – 1µs	Tkckl	
Tdho	Hold on DATA from CLK falling for transmit	0 ns	1 µs	
Tki	Time for which CLK is held low to request a send	63.5 µs	64.5 µs	
Tkrg	CLK low from controller to CLK low from peripheral for request to send	1 µs	—	
Tksb	CLK low to DATA low hold time for request to send	1 µs	—	1

NOTE:
1. The DATA will proceed the CLK in this implementation, so the value for Tksb shown in Figure 15-2 is negative (that is, safe).

15.3 Test Interface

A number of test registers are provided for use in production testing of the module. These registers should not be used in normal operation.

The test registers provided are as follows:

- KBDTEST1 (MSETEST1)
  Test register 1
  Control clock and counters (Write only)
- KBDTEST2 (MSETEST2)
  Test register 2
  Control test inputs (Write only)
- KBDTEST3 (MSETEST3)
  Test register 3
  View internals and outputs (Read only)
- KBDTEST4 (MSETEST4)
  Test register 4
  View internal state (Read only)

15.3.1 KBDTEST1 (MSETEST1)

[7] (+&2000)

This write-only register provides control over the clock source to the block for test purposes. In read mode, the current settings of the register are returned.

Bit	Name	Function
7	CD	Clear the divide by 8 counter. Writing a one to this location generates a pulse to reset the divide by 8 counter.
6	RC1	Registered Div8 clock (1-MHz) bit. Used in test mode as a source for the 1-MHz synchronizing clock signal by successive writes to the location.
5	MC	Manual clock. This bit can be used as a source of the 8-MHz clock signal by successive writes to the location.
4..3	C(1:0)	Clock select lines. X0 = Normal operation. The input 8-MHz clock is selected and the Div8 (1-MHz) clock is generated from the 8-MHz clock. 01 = PSEL ANDed with PSTB is driven as the internal 8-MHz clock. When this clock source is selected, every access to the block generates a positive clock pulse internally. The Div8 clock is generated from this internal 8-MHz clock in this mode, that is, the pulsed clock. 11 = The MC and RC1 bits are used as the clock sources for the 8-MHz and 1-MHz clocks respectively. To generate an internal clock transition, program these bits with the last written values inverted. For example, a write 0 followed by a write 1 creates a LOW to HIGH transition and vice versa.
2	T2	When set to 1, selects the µs16 signal for test by multimplexing to KBDCR(RXP). The µs16 signal is the output of the timer prescaler and its low time must be checked (by polling KBDCR) to be 16 µs +/- 7 µs. This allows the prescaler part of the timer to be tested in 512 vectors.
1	T1	When set to 1, allows the timeout section of the timer (that is, the last eight bits) to be clocked at eight times the normal prescaler output rate. This allows the timeout section to be tested with less vectors.
0	T0	This bit allows the middle section of the timer (for timing 64 µs) to be clocked by the 8-MHz input directly, rather than from the prescaler output. This allows the middle section to be tested in 16 vectors. It also allows the keyboard function to be run at 128 times its normal rate, so that one data transfer will take about 500 vectors. Note that without these test bits, it would take 512,000 vectors to test the timeout period.

15.3.2 KBDTEST2 (MSETEST2)

[4] (+&2400)

The second write-only test register provides control of the clock and other signals.

Bit	Name	Function
3	TICBnRES	When HIGH, this bit will cause all of the peripherals' normal mode logic/registers to be reset (EXCLUDING the test registers).
2	RKC	Registered keyboard clock bit. Used to drive the keyboard clock path internally.
1	RKD	Registered keyboard data bit. Used in test mode to drive the keyboard data path internally.
0	SEL	Multiplex select line for use in test mode. When set to 1, selects the RKC and RKD bits for use internally. Otherwise the normal block inputs are used.

15.3.3 KBDTEST3 (MSETEST3)

[8] (+&2800)

The third test register is read-only and provides observability of the pad control signals and other internal signals.

Bit	Name	Function
7	ms_16	Reflects the value from the end of timeout chain.
6	us_64	Reflects the value from the middle of timeout chain.
5	us_16	Reflects the value from the first stage of timeout chain.
4	DIV8	Reflects the value at the output of the divide by 8 counter for the Div8 clock.
3	DIn	Reflects the value of the internal KbDataIn signal.
2	CIn	Reflects the value of the internal KbClkIn signal before sampling by the Div8 (1-MHz) clock.
1	KD	Reflects the value of the nKbDataEn output signal.
0	KC	Reflects the value of the nKbClkEn output signal.

15.3.4 KBDTEST4 (MSETEST4)

[8] (+&2C00)

The fourth test register is read-only and provides observability of the internal state machine and the bit counter decode signals.

Bit	Name	Function
7	BC12	Bit counter = 12. Input to state machine.
6	BC11	Bit counter = 11. Input to state machine
5	TRES	Reset timer. State bit 5.
4	CLKOE	Clock output enable. State bit 4.
3	CRES	Reset bit counter. State bit 3.
2	RXB	Receive busy. State bit 2.
1	TXB	Transmit busy. State bit 1.
0	SRX	Set receive interrupt. State bit 0.

15.4 Application

The PS/2 clock must be enabled before the PS/2 blocks become operational. See Section 10.2.1, “Power Control Register (SKPCR)” on page 10-2.

General-Purpose I/O Interface 16

The General-Purpose I/O (GPIO) interface is an AMBA Peripheral Bus (APB) peripheral that provides 15 bits of programmable input/output divided into two ports: port A (8 bits) and port B (7 bits). Each pin is configurable as either input or output. At system reset both ports default to input.

Each line can be configured as an interrupt or wake-up source. See Section 14.1.2, “Interrupt Sources” on page 14-3.

Each port has a data register and a data direction register. The data direction register defines whether each individual pin is an input or an output. The data register is used to read the value of the GPIO pins, both input and output, as well as set the values of pins that are configured as outputs.

16.1 Functional Description

The GPIO ports provide general-purpose digital I/O capabilities for the StrongARM® SA-1101 (SA-1101). The direction of each line can be individually selected using the data direction register. The value output by each line is dependant on the SA-1101 mode. In normal mode, the values in the data write register are output on lines whose direction is set to output.

16.1.1 Sleep

In sleep mode, the values in the sleep state register are output on lines whose direction is set to output. These values remain after leaving sleep state until a new value is written to the data write register

16.2 Programmer’s Model

This section describes the registers in the general-purpose I/O interface.

16.2.1 Port A Data Write Register (PADWR)

Values written to this 8-bit register will be output on port A pins if the corresponding data direction bits are set LOW (port output). All bits are cleared by a system reset.

Bit	Function
7:0	Data to output on GPA[7:0]
31:8	Unused

16.2.2 Port B Data Write Register (PBDWR)

Values written to this 7-bit register will be output on port B pins if the corresponding data direction bits are set LOW (port output). All bits are cleared by a system reset.

Bit	Function
6:0	Data to output GPB[6:0]
31:7	Unused

16.2.3 Port A Data Read Register (PADRR)

Values read from this register reflect the external state of port A, not necessarily the value written to it.

Bit	Function
7:0	Data read from GPA[7:0]
31:8	Unused

16.2.4 Port B Data Read Register (PBDRR)

Values read from this register reflect the external state of port B, not necessarily the value written to it.

Bit	Function
6:0	Data read from GPB[6:0]
31:7	Unused

16.2.5 Port A Data Direction Register (PADDR)

Bits set in this 8-bit read/write register will select the corresponding pin in port A to become an input; clearing a bit sets the pin to output. All bits are set by a system reset so that port A is input by default.

Bit	Function
7:0	Direction of GPA[7:0} 1 = output 0 = output
31:8	Unused

16.2.6 Port B Data Direction Register (PBDDR)

Bits set in this 7-bit read/write register will select the corresponding pin in port B to become an input; clearing a bit sets the pin to output. All bits are set by a system reset so that port B is input by default.

Bit	Function
6:0	Direction of GPB[6:0} 1 = output 0 = output
31:7	Unused

16.2.7 Port A Sleep State Register (PASSR)

Values in this 8-bit read/write register will set the value of the port during sleep mode.

Bit	Function
7:0	Data to output on GPA[7:0] when in sleep mode
31:8	Unused

16.2.8 Port B Sleep State Register (PBSSR)

Values in this 7-bit read/write register will set the value of the port during sleep mode.

Bit	Function
6:0	Data to output on GPB[6:0] when in sleep mode
31:7	Unused

16.2.9 Memory Map

Table 16-1 lists the read and write locations in the GPIO memory map. See the Memory Map chapter for the base address of the GPIO.

Table 16-1. GPIO Register Memory Map

Address	Read Location	Write Location
PIO Base	PADRR register	PADWR register
PIO Base + 0x0400	PBDRR register	PBDWR register
PIO Base + 0x0800	PADDR register	PADDR register
PIO Base + 0x0C00	PBDDR register	PBDDR register
PIO Base + 0x1000	PASSR register	PASSR register
PIO Base + 0x1400	PBSSR register	PBSSR register

16.3 Test

There are no test registers in the GPIO interface.

16.4 Application

SKPCR register must be set to 1 to allow access to all of the GPIO functionality. See Section 10.2.1, “Power Control Register (SKPCR)” on page 10-2

Keypad Interface 17

The keypad interface consists of two bidirectional ports, X and Y of 8 and 16 bits respectively. Each of the pins on these ports has an open drain output and can be driven low, or left undriven by using the data write registers. The input or status of each pin can be read at any time using the data read registers.

Each line of this interface can be used as an interrupt source. See the Section 14.1.2, “Interrupt Sources” on page 14-3.

17.1 Programmer’s Model

This section describes the registers in the keypad interface.

17.1.1 Port X Data Write Register (PXDWR)

Values written to this 8-bit register are output on port X pins. To set any line to input mode, write a 1 to its bit in this register. All bits are set by a system reset so that port X is input by default.

Bit	Function
7:0	Data to output on KPX[7:0]
31:8	Unused

17.1.2 Port X Data Read Register (PXDRR)

Values read from this register reflect the external state of port X.

Bit	Function
7:0	Data to output on KPX[7:0]
31:8	Unused

17.1.3 Port Y Data Write Register (PYDWR)

Values written to this 16-bit register are output on port Y pins. To set any line to input mode, write a 1 to its bit in this register. All bits are set by a system reset so that port Y is input by default.

Bit	Function
15:0	Data to output KPY[15:0]
31:16	Unused

17.1.4 Port Y Data Read Register (PYDRR)

Values read from this register reflect the external state of port Y.

Bit	Function
15:0	Data read from KPY[15:0]
31:16	Unused

17.1.5 Register Memory Map

Table 17-1 lists the read and write locations in the keypad register memory map. See Chapter 19, “Memory Map” for the keypad interface base address.

Table 17-1. Keypad Register Memory Map

Address	Read Location	Write Location
Keypad Base	PXDRR register	PXDWR register
Keypad Base + 0x0400	PYDRR register	PYDWR register

17.2 Test

There are no test registers in the keypad interface.

17.2.1 Application

Bit 6 of the SKPCR register must be set to 0 to allow access to keypad functionality. See Section 10.2.1, “Power Control Register (SKPCR)” on page 10-2.

The interface can be used in two different modes:

- Polling

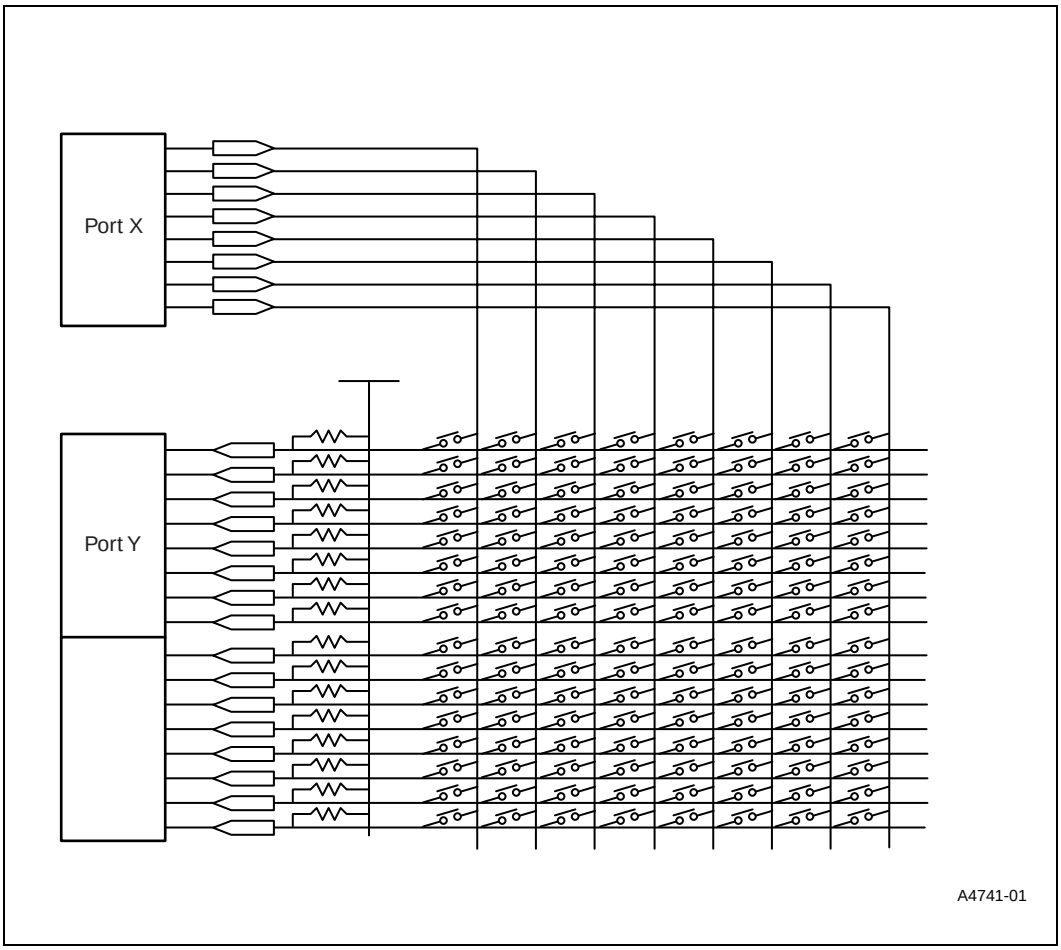
  A polling routine continuously monitors the keypad. This consists of tristating the column (X) outputs and then successively setting each bit to 0. For each successive setting, the row (Y) is read and any key press will register as a zero. Software will then have to debounce the key presses and handle multiple key presses.

- Initial Interrupt

  The column (X) outputs are all set to low. When a key is pressed, an interrupt will be asserted. The StrongARM® SA-1100 (SA-1100) can interrogate the StrongARM® SA-1101 (SA-1101) interrupt controller to find out which row (Y) was pressed. A scanning routine can be used, as for polling, to see which keys are pressed and to go through a debounce routine.

Figure 17-1 shows an example circuit.

Figure 17-1. Keypad Example



PCMCIA Interface 18

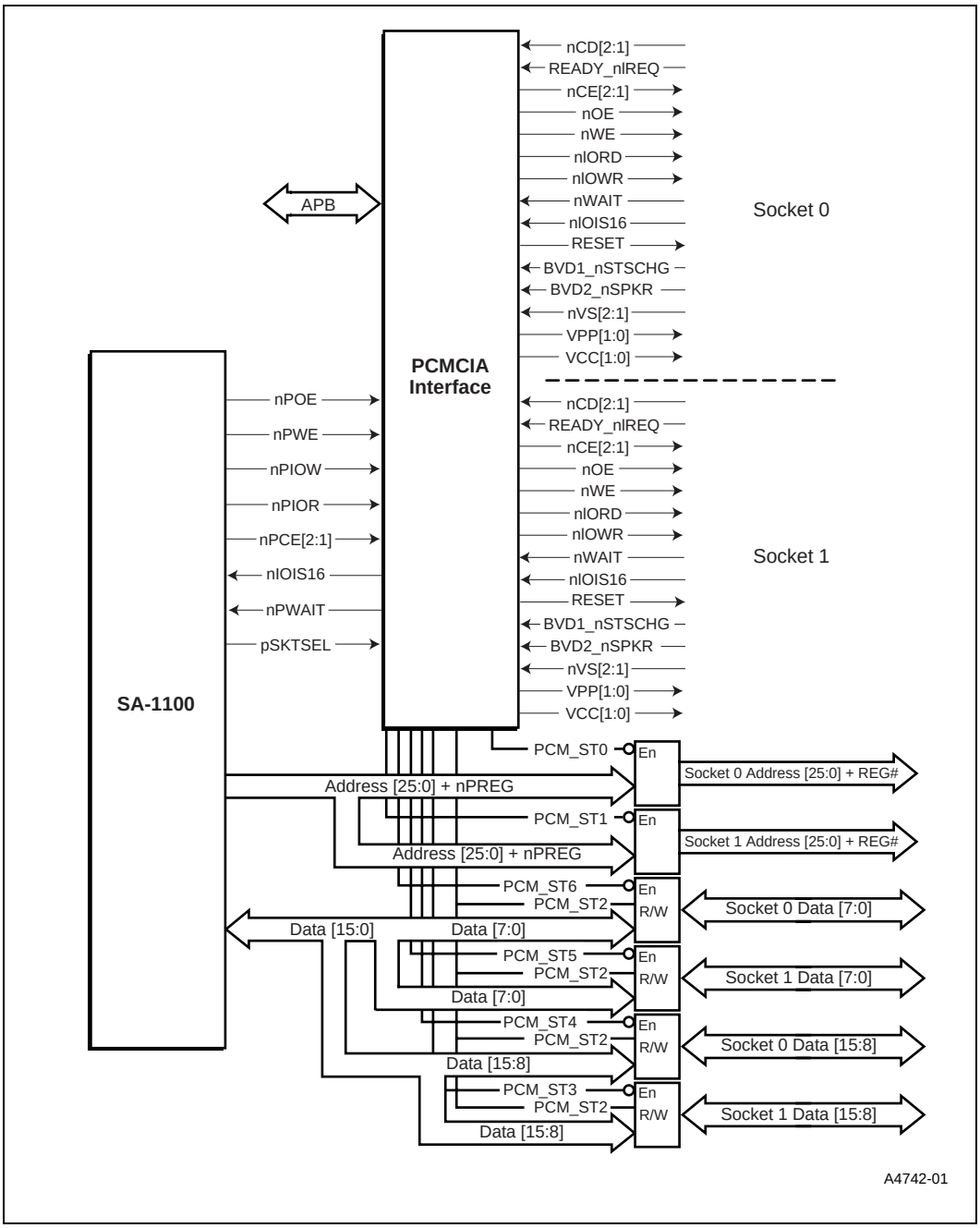
The PCMCIA interface block interfaces to two PCMCIA cards. It provides glue logic between the StrongARM® SA-1100 (SA-1100) PCMCIA interface and actual card sockets. The PCMCIA interface is provided to allow the building of two card sockets with the SA-1100 requiring only the addition of external address and data buffers.

The interface also provides facilities for controlling the state of the external pins when the StrongARM® SA-1101 (SA-1101) is in sleep mode.

18.1 PCMCIA Interface Block Diagram

Figure 18-1 shows the block diagram for the PCMCIA interface.

Figure 18-1. PCMCIA Interface Block Diagram



18.2 Functional Description

This section provides a functional overview of the PCMCIA interface.

18.2.1 Buffer Control

Seven lines are provided by the PCMCIA interface to control external data and address buffers for both sockets 0 and 1. Table 18-1 describes the buffer control signals.

Table 18-1. Buffer Control Signals

Name	Function
PCM_ST0	Socket 0 address enable
PCM_ST1	Socket 1 address enable
PCM_ST2	Data direction 0 = Read from card 1 = Write to card
PCM_ST3	Socket 1 data[15:8] enable
PCM_ST4	Socket 0 data[15:8] enable
PCM_ST5	Socket 1 data[7:0] enable
PCM_ST6	Socket 0 data[7:0] enable

The buffer control signals operate as shown in Figure 18-2 except when the SA-1101

- Enters sleep mode.
  These lines are tristated.
- Sets its float bits in the Control Register (PCCR).
  PCM_ST0, PCM_ST1, PCM_ST3, PCM_ST4, PCM_ST5, and PCM_ST6 are high.
  PCM_ST2 is not affected by the float bits.

18.2.2 Voltage Control

Four digital outputs are provided for each socket to allow the programming of an external voltage source. This allows the interface to accommodate PCMCIA cards of different operating voltages. See Table 18.4.2 “Control Register (PCCR)” on page 18-7.

18.2.3 Reset Signals

A single digital output is provided for each socket to allow the resetting of the card. Both of the signals are active high and are asserted when either the SA-1101 is in reset or the relevant bit in the Control Register is set. See Table 18.4.2 “Control Register (PCCR)” on page 18-7.

When the SA-1101 is in sleep mode the reset signals are either tristated or low, dependent upon the relevant bit in the sleep state register. See Table 18.4.3 “Sleep State Register (PCSSR)” on page 18-8.

18.2.4 Control Signals

Six control lines are provided for each socket: nOE, nWE, nIOW, nIOR, nCE1, and nCE2. The control lines operate as shown in Figure 18-2 except when the SA-1101

- Enters sleep mode.
  These lines can be tristated or driven high. See Section 18.4.3, “Sleep State Register (PCSSR)” on page 18-8
- Sets its float bits in the Control Register (PCCR).
  These control lines will float in normal and sleep mode.

18.2.5 Other Signals

The nIOIS16 and the nPWAIT signals are unaffected by the register configuration. See Figure 18-2.

18.2.6 Interrupts

Table 18-2 describes the interrupt sources provided by the PCMCIA interface.

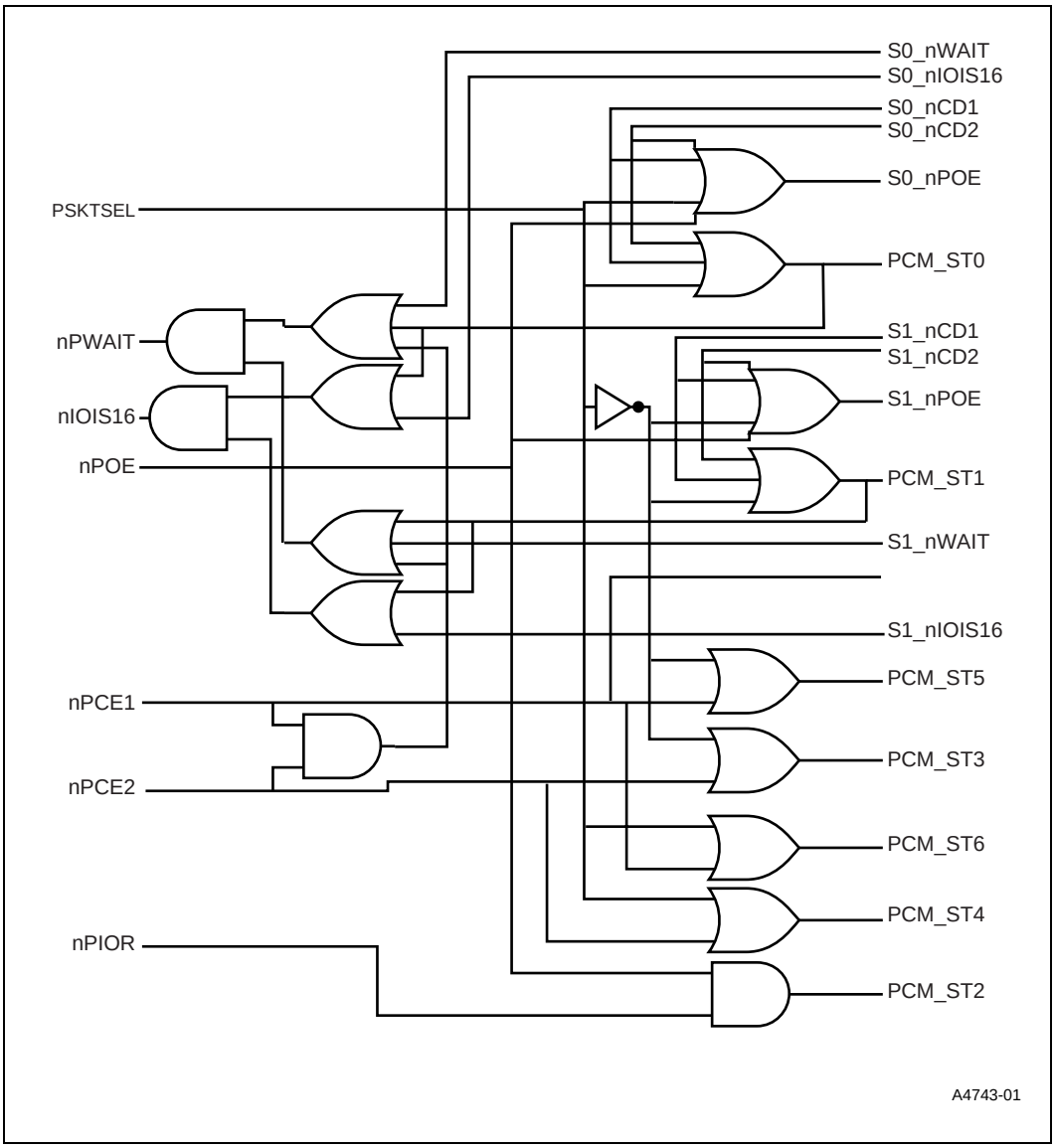
Table 18-2. Interrupt Signals

Name	Function
S0_READY_nIREQ	Socket 0 READY_nIREQ pin
S1_READY_nIREQ	Socket 1READY_nIREQ pin
S0_CDVALID	Socket 0 card detect 0 = Card fully inserted 1 = Card not fully inserted
S1_CDVALID	Socket 1card detect 0 = Card fully inserted 1 = Card not fully inserted
S0_BVD1_STSCHG	Socket 0 BVD1_STSCHG pin
S1_BVD1_STSCHG	Socket 1 BVD1_STSCHG pin

18.3 Logic Diagram

Figure 18-2 shows a diagram of the PCMCIA interface logic.

Figure 18-2. PCMCIA Interface Logic Diagram



18.4 Programmer's Model

This section describes the three registers in the PCMCIA interface.

18.4.1 Status Register (PCSR)

This register allows the reading of various signals within the PCMCIA interface.

Bit	Function
0	Socket 0 ready Status of S0_READY_nIREQ
1	Socket 1 ready Status of S1_READY_nIREQ
2	0 = Socket 0 card detect 1 and 2 valid
3	0 = Socket 1 card detect 1 and 2 valid
4	Socket 0 VS1
5	Socket 0 VS2
6	Socket 1nVS1
7	Socket 1nVS2
8	Socket 0 WP (S0_nIOIS16)
9	Socket 1 WP (S1_nIOIS16)
10	Socket 0 BVD1_nSTSCHG
11	Socket 0 BVD2_nSPKR
12	Socket 1 BVD1_nSTSCHG
13	Socket 1 BVD2_nSPKR

18.4.2 Control Register (PCCR)

This register allows the setting of various output pins in the PCMCIA interface and also allows the tristating of the control lines.

Bit	Function
0	Socket 0 VPP0
1	Socket 0 VPP1
2	Socket 0 VCC0
3	Socket 0 VCC1
4	Socket 1 VPP0
5	Socket 1 VPP1
6	Socket 1 VCC0
7	Socket 1 VCC1
8	Socket 0 reset
9	Socket 1 reset
10	S0 float 0 = Float all S0 control lines and set PCM_ST[0,4,6] high
11	S1 float 0 = Float all S1 control lines and set PCM_ST[1,3,5] high

18.4.3 Sleep State Register (PCSSR)

This register allows the setting of output pins of the PCMCIA interface during sleep mode.

Bit	Function
0	Socket 0 VCC0 0 = Low 1 = High
1	Socket 0 VCC1 0 = Low 1 = High
2	Socket 0 VPP0 0 = Low 1 = High
3	Socket 0 VPP1 0 = Low 1 = High
4	Socket 0 control lines See Table 18-3 “S0 Control Line Sleep State” on page 18-8
5	Socket 1 VCC0 0 = Low 1 = High
6	Socket 1 VCC1 0 = Low 1 = High
7	Socket 1 VPP0 0 = Low 1 = High
8	Socket 1 VPP1 0 = Low 1 = High
9	Socket 1 control lines See Table 18-4 “S1 Control Line Sleep State” on page 18-9

In sleep state, the PCMCIA control pins are put into a state dependent on the value in the S0CONT or S1CONT bits as shown in Table 18-3 and Table 18-4.

Table 18-3. S0 Control Line Sleep State

Control Line	PCSSR[4] = 0	PCSSR[4] = 1
S0_NOE	Tristate	1
S0_nIORD	Tristate	1
S0_nIOWR	Tristate	1
S0_nCE1	Tristate	1
S0_nCE2	Tristate	1
S0_nWE	Tristate	1
S0_Reset	Tristate	0

Table 18-4. S1 Control Line Sleep State

Control Line	PCSSR[9] = 0	PCSSR[9] = 1
S1_nOE	Tristate	1
S1_nIORD	Tristate	1
S1_nIOWR	Tristate	1
S1_nCE1	Tristate	1
S1_nCE2	Tristate	1
S1_nWE	Tristate	1
S1_Reset	Tristate	0

18.4.4 Memory Map

Table 18-5 shows the read and write locations in the PCMCIA interface register memory map. See Chapter 19, “Memory Map” for the PCMCIA interface base address.

Table 18-5. PCMCIA Interface Register Memory Map

Address	Read Location	Write Location
Base	PCCR register	PCCR register
Base + 0x0400	PCSSR register	PCSSR register
Base + 0x0800	PCSR register	Reserved

Memory Map 19

This chapter lists the base address for the StrongARM® SA-1101 (SA-1101) interfaces. Table 19-1 lists the memory map for the SA-1101. The base offset is relative to the base address of the chip select signal used to select the SA-1101.

Table 19-1. SA-1101 Memory Map

Base Offset	Description
00000000	Shared Memory Controller
00000400 – 0001FC00	System Controller
00020000 – 0003FC00	Arbiter
00040000 – 0005FC00	System Controller
00060000 – 0007FC00	System Controller
00080000 – 000FFC00	IEEE1284 Parallel Port
00100000 – 0011FC00	Video Memory Controller
00120000 – 0013FC00	Update FIFO
00140000 – 0015FC00	Shared Memory Controller
00160000 – 0017FC00	Interrupt Controller
00180000 – 0019FC00	USB Host Interface Controller
001A0000 – 001AFC00	PS/2 Trackpad Interface
001B0000 – 001BFC00	PS/2 Mouse Interface
001C0000 – 001DFC00	Keypad Interface
001E0000 – 001FFC00	PCMCIA Interface
00200000 – 002FFC00	VGA Control Register
00300000 – 003FFC00	General-Purpose I/O Interface
C0000000 – FFFFFFFF	Shared Memory Controller Datapath

The base offset shown in Table 19-1 is relative to the base address of the chip select signal used to select the SA-1101.

Pin and Padout 20

This chapter describes the StrongARM® SA-1101 (SA-1101) pins.

Table 20-1. Pin Descriptions (Sheet 1 of 7)

Ball	Signal	Finger	I/O Cell	Reset	Sleep	Open Drain
M1	A10	50	PB8_SLP	2	1	
M2	A11	51	PB8_SLP	2	1	
M3	A12	53	PB8_SLP	2	1	
M4	A13	49	PB8_SLP	2	1	
M5	A14	52	PB8_SLP	2	1	
N1	A15	54	PB8_SLP	2	1	
N2	A16	55	PB8_SLP	2	1	
N3	A17	58	PB8_SLP	2	1	
N4	A18	62	PB8_SLP	2	1	
N5	A19	77	PB8_SLP	2	1	
P1	A20	59	PB8_SLP	2	1	
P2	A21	60	PB8_SLP	2	1	
D16	BAT_FLT	183	PI	2	2	
B4	BLUE	250	PAS	A	A	
L3	CLK	47	PI	2	2	
K2	D0	40	PB8_SLP	2	1	
K3	D1	42	PB8_SLP	2	1	
H3	D10	27	PB8_SLP	2	1	
H4	D11	26	PB8_SLP	2	1	
H5	D12	29	PB8_SLP	2	1	
G5	D13	23	PB8_SLP	2	1	
G4	D14	22	PB8_SLP	2	1	
G3	D15	21	PB8_SLP	2	1	
G2	D16	24	PB8_SLP	2	1	
G1	D17	25	PB8_SLP	2	1	
F1	D18	17	PB8_SLP	2	1	
F2	D19	16	PB8_SLP	2	1	
K4	D2	41	PB8_SLP	2	1	
F3	D20	15	PB8_SLP	2	1	
F4	D21	14	PB8_SLP	2	1	
F5	D22	20	PB8_SLP	2	1	
E4	D23	12	PB8_SLP	2	1	
E3	D24	11	PB8_SLP	2	1	

Table 20-1. Pin Descriptions (Sheet 2 of 7)

Ball	Signal	Finger	I/O Cell	Reset	Sleep	Open Drain
E1	D25	13	PB8_SLP	2	1	
D1	D26	8	PB8_SLP	2	1	
D2	D27	7	PB8_SLP	2	1	
D3	D28	6	PB8_SLP	2	1	
C2	D29	4	PB8_SLP	2	1	
K5	D3	38	PB8_SLP	2	1	
C1	D30	5	PB8_SLP	2	1	
B1	D31	3	PB8_SLP	2	1	
J5	D4	35	PB8_SLP	2	1	
J4	D5	36	PB8_SLP	2	1	
J3	D6	37	PB8_SLP	2	1	
J2	D7	33	PB8_SLP	2	1	
J1	D8	32	PB8_SLP	2	1	
H1	D9	28	PB8_SLP	2	1	
D5	DAC_VDD	247	PPDDX	—	—	
B3	DAC_VSS	253	PPASX	—	—	
D12	FB_A0	209	PB4_SLP	7	4	
D11	FB_A1	214	PB4_SLP	7	4	
D9	FB_A10	222	PB4_SLP	7	4	
E9	FB_A11	223	PB4_SLP	7	4	
E11	FB_A2	218	PB4_SLP	7	4	
E10	FB_A3	220	PB4_SLP	7	4	
D10	FB_A4	216	PB4_SLP	7	4	
C11	FB_A5	210	PB4_SLP	7	4	
B11	FB_A6	211	PB4_SLP	7	4	
A10	FB_A7	219	PB4_SLP	7	4	
B10	FB_A8	217	PB4_SLP	7	4	
C10	FB_A9	215	PB4_SLP	7	4	
E8	FB_D0	229	PB4_SLP	7	4	
D8	FB_D1	231	PB4_SLP	7	4	
D6	FB_D10	243	PB4_SLP	7	4	
C7	FB_D11	236	PB4_SLP	7	4	
B7	FB_D12	233	PB4_SLP	7	4	
A7	FB_D13	232	PB4_SLP	7	4	
A6	FB_D14	237	PB4_SLP	7	4	
A5	FB_D15	245	PB4_SLP	7	4	
C9	FB_D2	221	PB4_SLP	7	4	
B9	FB_D3	224	PB4_SLP	7	4	
A9	FB_D4	225	PB4_SLP	7	4	

Table 20-1. Pin Descriptions (Sheet 3 of 7)

Ball	Signal	Finger	I/O Cell	Reset	Sleep	Open Drain
A8	FB_D5	228	PB4_SLP	7	4	
C8	FB_D6	230	PB4_SLP	7	4	
D7	FB_D7	235	PB4_SLP	7	4	
E7	FB_D8	234	PB4_SLP	7	4	
E6	FB_D9	241	PB4_SLP	7	4	
A13	GPA0	202	PB4S	2	5	
A14	GPA1	200	PB4S	2	5	
A15	GPA2	198	PB4S	2	5	
A16	GPA3	195	PB4S	2	5	
B13	GPA4	201	PB4S	2	5	
B14	GPA5	197	PB4S	2	5	
B15	GPA6	196	PB4S	2	5	
B16	GPA7	190	PB4S	2	5	
C13	GPB0_KPY10	203	PB4S	2	5	Open Drain for KP
C15	GPB1_KPY11	189	PB4S	2	5	Open Drain for KP
C16	GPB2_KPY12	188	PB4S	2	5	Open Drain for KP
D13	GPB3_KPY13	199	PB4S	2	5	Open Drain for KP
D14	GPB4_KPY14	187	PB4S	2	5	Open Drain for KP
D15	GPB5_KPY15	186	PB4S	2	5	Open Drain for KP
P13	GPB6	123	PB4S	2	5	
C4	GREEN	251	PAS	A	A	
A2	HSYNC	254	PB8	7	7	
L2	INT	46	PB4	8	8	
C5	IREF	248	PAS	—	—	
L5	MBGNT	45	PI	2	2	
L4	MBREQ	48	PB4	6	6	
J15	MSCLK	159	PB4S_SLP	2	1	Open Drain
J14	MSDATA	156	PB4S_SLP	2	1	Open Drain
P5	nCAS0	73	PB8_SLP	2	1	
R1	nCAS1	61	PB8_SLP	2	1	
R2	nCAS2	68	PB8_SLP	2	1	
R3	nCAS3	69	PB8_SLP	2	1	
P4	nCS	74	PI	2	2	
B12	nFB_LCAS	205	PB4_SLP	6	7	
A12	nFB_RAS	208	PB4_SLP	6	7	
E12	nFB_UCAS	181	PB4_SLP	6	7	
C12	nFB_WE	204	PB4_SLP	6	3	
R5	nIOIS16	78	PB4_SLP	8	1	
T3	nOE	72	PB8	2	2	

Table 20-1. Pin Descriptions (Sheet 4 of 7)

Ball	Signal	Finger	I/O Cell	Reset	Sleep	Open Drain
P7	nPCE1	86	PI	2	2	
N7	nPCE2	87	PI	2	2	
P6	nPIOR	82	PI	2	2	
M7	nPIOW	90	PI	2	2	
T5	nPOE	79	PI	2	2	
R7	nPWAIT	88	PB4_SLP	8	1	
N6	nPWE	85	PI	2	2	
T1	nRAS0	67	PB8_SLP	2	1	
T2	nRAS1	70	PB8_SLP	2	1	
K1	nRESET	39	PI	2	2	
K12	nTEST	155	PI	2	2	
R4	nWE	75	PB8	2	2	
T16	PCM_ST0	131	PB4_SLP	8	1	
P16	PCM_ST1	135	PB4_SLP	8	1	
N16	PCM_ST2	137	PB4_SLP	8	1	
M16	PCM_ST3	142	PB4_SLP	8	1	
L16	PCM_ST4	150	PB4_SLP	8	1	
M6	PCM_ST5	84	PB4_SLP	8	1	
T6	PCM_ST6	83	PB4_SLP	8	1	
B6	PLL_VDD	238	PPLDX	—	—	
C6	PLL_VSS	242	PPASX	—	—	
E14	PPBUFEN_KPY8	182	PB4S_SLP	2	1	Open Drain for KP
E13	PPBUSY_KPY9	178	PB4S_SLP	2	1	Open Drain for KP
G15	PPDATA0_KPX0	168	PB4S_SLP	2	1	Open Drain for KP
F16	PPDATA1_KPX1	172	PB4S_SLP	2	1	Open Drain for KP
G16	PPDATA2_KPX2	167	PB4S_SLP	2	1	Open Drain for KP
H16	PPDATA3_KPX3	162	PB4S_SLP	2	1	Open Drain for KP
H15	PPDATA4_KPX4	163	PB4S_SLP	2	1	Open Drain for KP
H14	PPDATA5_KPX5	165	PB4S_SLP	2	1	Open Drain for KP
H13	PPDATA6_KPX6	166	PB4S_SLP	2	1	Open Drain for KP
H12	PPDATA7_KPX7	164	PB4S_SLP	2	1	Open Drain for KP
F13	PPnACK_KPY4	177	PB4S_SLP	2	1	Open Drain for KP
F14	PPnAUTOFD_KPY3	176	PB4S_SLP	2	1	Open Drain for KP
E15	PPnFAULT_KPY7	180	PB4S_SLP	2	1	Open Drain for KP
G12	PPnINIT_KPY2	169	PB4S_SLP	2	1	Open Drain for KP
G13	PPnSELECTIN_KPY1	170	PB4S_SLP	2	1	Open Drain for KP
G14	PPnSTROBE_KPY0	171	PB4S_SLP	2	1	Open Drain for KP
E16	PPPERROR_KPY6	179	PB4S_SLP	2	1	Open Drain for KP
F12	PPSELECT_KPY5	175	PB4S_SLP	2	1	Open Drain for KP

Table 20-1. Pin Descriptions (Sheet 5 of 7)

Ball	Signal	Finger	I/O Cell	Reset	Sleep	Open Drain
T4	PSKTSEL	76	PI	2	2	
K16	PWM1	154	PB4_SLP	7	1	
K15	PWM2	153	PB4_SLP	7	1	
A4	RED	249	PAS	A	A	
M11	S0_BVD1_nSTSCHG	109	PI	2	2	
M10	S0_BVD2_nSPKR	102	PI	2	2	
P8	S0_nCD1	92	PI	2	2	
N8	S0_nCD2	91	PI	2	2	
M9	S0_nCE1	99	PB8_SLP	8	3	
M8	S0_nCE2	93	PB8_SLP	8	3	
R9	S0_nIOIS16	98	PI	2	2	
N10	S0_nIORD	105	PB8_SLP	8	3	
P10	S0_nIOWR	106	PB8_SLP	8	3	
N9	S0_nOE	101	PB8_SLP	8	3	
T10	S0_nVS1	103	PI	2	2	
R10	S0_nVS2	104	PI	2	2	
P9	S0_nWAIT	100	PI	2	2	
R8	S0_nWE	94	PB8_SLP	8	3	
T7	S0_READY_nIREQ	89	PI	2	2	
T8	S0_RESET	95	PB8_SLP	8	4	
P11	S0_VCC0	112	PB4_SLP	8	5	
N11	S0_VCC1	113	PB4_SLP	8	5	
T11	S0_VPP0	110	PB4_SLP	8	5	
R11	S0_VPP1	111	PB4_SLP	8	5	
T12	S1_BVD1_nSTSCHG	115	PI	2	2	
T15	S1_BVD2_nSPKR	126	PI	2	2	
N12	S1_nCD1	114	PI	2	2	
P12	S1_nCD2	118	PI	2	2	
R13	S1_nCE1	121	PB8_SLP	8	3	
T13	S1_nCE2	120	PB8_SLP	8	3	
P15	S1_nIOIS16	132	PI	2	2	
T14	S1_nIORD	124	PB8_SLP	8	3	
M13	S1_nIOWR	143	PB8_SLP	8	3	
M12	S1_nOE	119	PB8_SLP	8	3	
L12	S1_nVS1	145	PI	2	2	
L13	S1_nVS2	146	PI	2	2	
N15	S1_nWAIT	136	PI	2	2	
N14	S1_nWE	138	PB8_SLP	8	3	
N13	S1_READY_nIREQ	122	PI	2	2	

Table 20-1. Pin Descriptions (Sheet 6 of 7)

Ball	Signal	Finger	I/O Cell	Reset	Sleep	Open Drain
R14	S1_RESET	125	PB8_SLP	8	4	
P14	S1_VCC0	134	PB4_SLP	8	5	
M14	S1_VCC1	139	PB4_SLP	8	5	
L14	S1_VPP0	144	PB4_SLP	8	5	
L15	S1_VPP1	149	PB4_SLP	8	5	
J13	TPCLK	157	PB4S_SLP	2	1	Open Drain
J12	TPDATA	158	PB4S_SLP	2	1	Open Drain
K13	USB_MINUS	147	PUSBM	2	2	
K14	USB_PLUS	148	PUSBP	2	2	
A11	VDD	1	PPDI	—	—	
B5	VDD	9	PPDX	—	—	
E2	VDD	19	PPDX	—	—	
F7	VDD	31	PPDXE	—	—	
F8	VDD	43	PPDX	—	—	
F9	VDD	57	PPDX	—	—	
F10	VDD	64	PPDXE	—	—	
F11	VDD	65	PPDI	—	—	
F15	VDD	81	PPDX	—	—	
G6	VDD	96	PPDXE	—	—	
H6	VDD	107	PPDX	—	—	
H11	VDD	117	PPDX	—	—	
J11	VDD	128	PPDXE	—	—	
K6	VDD	129	PPDI	—	—	
K11	VDD	140	PPDX	—	—	
L6	VDD	152	PPDX	—	—	
L7	VDD	161	PPDXE	—	—	
L8	VDD	174	PPDX	—	—	
L9	VDD	185	PPDX	—	—	
L10	VDD	192	PPDXE	—	—	
L11	VDD	193	PPDI	—	—	
M15	VDD	207	PPDX	—	—	
R6	VDD	212	PPDX	—	—	
R12	VDD	226	PPDXE	—	—	
G11	VDD	239	PPDX	—	—	
J6	VDD	246	PPDX	—	—	
L1	VDD	256	PPDXE	—	—	
R16	VDD_FLT	133	PI	2	2	
A1	VSS	2	PPSI	—	—	
B2	VSS	10	PPSX	—	—	

Table 20-1. Pin Descriptions (Sheet 7 of 7)

Ball	Signal	Finger	I/O Cell	Reset	Sleep	Open Drain
B8	VSS	18	PPSX	—	—	
C3	VSS	30	PPSXE	—	—	
C14	VSS	34	PPSX	—	—	
D4	VSS	44	PPSX	—	—	
E5	VSS	56	PPSX	—	—	
F6	VSS	63	PPSXE	—	—	
G7	VSS	66	PPSI	—	—	
G8	VSS	71	PPSX	—	—	
G9	VSS	80	PPSX	—	—	
G10	VSS	97	PPSXE	—	—	
H2	VSS	108	PPSX	—	—	
H7	VSS	116	PPSX	—	—	
H8	VSS	127	PPSXE	—	—	
H9	VSS	130	PPSI	—	—	
H10	VSS	141	PPSX	—	—	
J7	VSS	151	PPSX	—	—	
J8	VSS	160	PPSXE	—	—	
J9	VSS	173	PPSX	—	—	
J10	VSS	184	PPSX	—	—	
K7	VSS	191	PPSXE	—	—	
K8	VSS	194	PPSI	—	—	
K10	VSS	206	PPSX	—	—	
P3	VSS	213	PPSX	—	—	
R15	VSS	227	PPSXE	—	—	
T9	VSS	240	PPSX	—	—	
J16	VSS	244	PPSX	—	—	
K9	VSS	255	PPSXE	—	—	
A3	VSYNC	252	PB4	7	7	

NOTES:
1. Input with receivers disabled. Able to float pins without resulting in large leakage. (Output tristated if applicable).
2. Normal input mode. (Output tristated if applicable.)
3. Output pins, selectable as either driven high or output tristated by registers.
4. Output pins, selectable as either driven low or output tristated by registers.
5. Input interrupt enable or output high or low selectable by registers.
6. Output active mode (in negated state).
7. Output low.
8. Output active mode (high or low depending on signals from CPU/sockets).

A. Analog pad in zero current state.

20.1 Test Pin Multiplexing

Table 20-2 lists the external pins of the SA-1101 and a description of their test function.

Table 20-2. External Pins

External Pin	Test Function	I/O	Enabled By	Description
S0_BVD2_NSPKR	ScanInA	I	nTEST=0 and ScanTest=1	USB scan in A
S0_nIORD	ScanOutA	O	nTEST=0 and ScanTest=1	USB scan out A
S1_nCD2	ScanInB	I	nTEST=0 and ScanTest=1	USB scan in B
S0_VCC0	ScanOutB	O	nTEST=0 and ScanTest=1	USB scan out B
S1_nCD1	ScanInC	I	nTEST=0 and ScanTest=1	USB scan in C
S1_RESET	ScanOutC	O	nTEST=0 and ScanTest=1	USB scan out C
S1_BVD1_NSTSCHG	ClkObserve	I	nTEST=0 and ScanTest=1	USB scan observe clock
S1_nIOIS16	ScanSel	I	nTEST=0 and ScanTest=1	USB scan select
S1_BVD2_NSPKR	UTestCLK	I	UTESTCLKEn=1	USB scan clock
PWM2	VCLK	O	nTEST=0 and ClockTest=1	VCLK test output
PWM1	BCLK	O	nTEST=0 and ClockTest=1	BCLK test output
FB_A[7:0]	DacDataIn[7:0]	I	DacTestMode=1	DAC test data in
FB_A[8]	DacClkIn	I	DacTestMode=1	DAC test clock
FB_A[11:9]	DacTESTOUT[2:0]	O	DacTestMode=1	DAC comparator outputs
GPA{7:0}	TestDataOut[15:8]	O	nTestDataEn=0	Video data out (pre-DAC)
PPDATA7_KPX7	TestDataOut[7]	O	nTestDataEn=0	Video data out (pre-DAC)
PPDATA6_KPX6	TestDataOut[6]	O	nTestDataEn=0	Video data out (pre-DAC)
PPDATA5_KPX5	TestDataOut[5]	O	nTestDataEn=0	Video data out (pre-DAC)
PPDATA4_KPX4	TestDataOut[4]	O	nTestDataEn=0	Video data out (pre-DAC)
PPDATA3_KPX4	TestDataOut[3]	O	nTestDataEn=0	Video data out (pre-DAC)
PPDATA2_KPX4	TestDataOut[2]	O	nTestDataEn=0	Video data out (pre-DAC)
PPDATA1_KPX4	TestDataOut[1]	O	nTestDataEn=0	Video data out (pre-DAC)
PPDATA0_KPX4	TestDataOut[0]	O	nTestDataEn=0	Video data out (pre-DAC)
S0_nWAIT	TREQA	I	nTEST=0	TIC test control signal
S1_nWAIT	TREQB	I	nTEST=0	TIC test control signal
nPWAIT	TACK	O	nTEST=0	TIC test acknowledge output

Package and Power 21

This chapter defines the package and power dissipation for the StrongARM® SA-1101 (SA-1101).

21.1 Package

The SA-1101 is contained in a 256 mini-BGA. Figure 21-1 shows the package details. Table 21-1 lists the dimensions in millimeters. Table 21-2 lists the StrongARM® SA-1100 (SA-1100) pins in numeric order, showing the signal type for each pin.

Figure 21-1. SA-1101 256 mBGA Mechanical Drawing

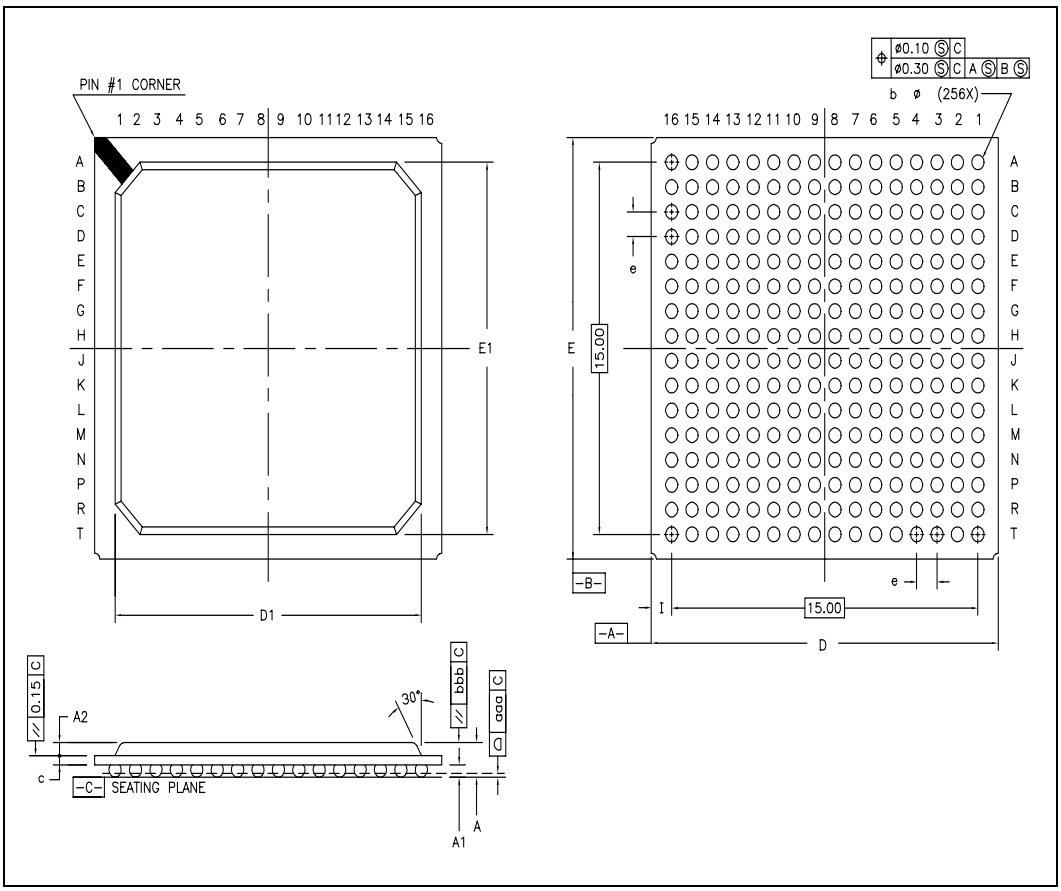


Table 21-1. 256 Mini-BGA Dimensional Attributes

Symbol	Dimension	Minimum	Nominal	Maximum
A	Package overall height	1.2	—	1.75
A1	Package standoff height	0.3	—	0.6
A2	Package thickness	0.5	—	0.85
D	Package overall width	—	17	—
D1	Package width	—	15	—
e		—	1	—
E	Package overall length	—	17	—
E1	Package length	—	15	—
l		—	1 [1]	—
b		0.4	—	0.7
c		—	.36	—
aaa		—	.15	—
bbb		—	.35	—

NOTE:

1. The value for this measurement is for reference only.

Table 21-2. SA-1101 Pinout — 256 Mini-Ball Grid Array

Pin	Signal	BGA Pad	Pin	Signal	BGA Pad	Pin	Signal	BGA Pad	Pin	Signal	BGA Pad
1	**VDD**	A11	65	**VDD**	F11	129	**VDD**	K6	193	**VDD**	L11
2	**VSS**	A1	66	**VSS**	G7	130	**VSS**	H9	194	**VSS**	K8
3	D31	B1	67	nRAS0	T1	131	PCM_ST0	T16	195	GPA3	A16
4	D29	C2	68	nCAS2	R2	132	S1_nIOIS16	P15	196	GPA6	B15
5	D30	C1	69	nCAS3	R3	133	VDD_FLT	R16	197	GPA5	B14
6	D28	D3	70	nRAS1	T2	134	S1_VCC0	P14	198	GPA2	A15
7	D27	D2	71	**VSS**	G8	135	PCM_ST1	P16	199	GPB3_KPY13	D13
8	D26	D1	72	nOE	T3	136	S1_nWAIT	N15	200	GPA1	A14
9	**VDD**	B5	73	nCAS0	P5	137	PCM_ST2	N16	201	GPA4	B13
10	**VSS**	B2	74	nCS	P4	138	S1_nWE	N14	202	GPA0	A13
11	D24	E3	75	nWE	R4	139	S1_VCC1	M14	203	GPB0_KPY10	C13
12	D23	E4	76	PSKTSEL	T4	140	**VDD**	K11	204	nFB_WE	C12
13	D25	E1	77	A19	N5	141	**VSS**	H10	205	nFB_LCAS	B12
14	D21	F4	78	nIOIS16	R5	142	PCM_ST3	M16	206	**VSS**	K10
15	D20	F3	79	nPOE	T5	143	S1_nIOWR	M13	207	**VDD**	M15
16	D19	F2	80	**VSS**	G9	144	S1_VPP0	L14	208	nFB_RAS	A12
17	D18	F1	81	**VDD**	F15	145	S1_nVS1	L12	209	FB_A0	D12
18	**VSS**	B8	82	nPIOR	P6	146	S1_nVS2	L13	210	FB_A5	C11
19	**VDD**	E2	83	PCM_ST6	T6	147	USB_MINUS	K13	211	FB_A6	B11
20	F5	D22	84	PCM_ST5	M6	148	USB_PLUS	K14	212	**VDD**	R6
21	D15	G3	85	nPWE	N6	149	S1_VPP1	L15	213	**VSS**	P3
22	D14	G4	86	nPCE1	P7	150	PCM_ST4	L16	214	FB_A1	D11
23	D13	G5	87	nPCE2	N7	151	**VSS**	J7	215	FB_A9	C10
24	D16	G2	88	nPWAIT	R7	152	**VDD**	L6	216	FB_A4	D10
25	D17	G1	89	S0_READY_nIREQ	T7	153	PWM2	K15	217	FB_A8	B10
26	D11	H4	90	nPIOW	M7	154	PWM1	K16	218	FB_A2	E11
27	D10	H3	91	S0_nCD2	N8	155	nTEST	K12	219	FB_A7	A10
28	D9	H1	92	S0_nCD1	P8	156	MSDATA	J14	220	FB_A3	E10
29	D12	H5	93	S0_nCE2	M8	157	TPCLK	J13	221	FB_D2	C9
30	**VSS**	C3	94	S0_nWE	R8	158	TPDATA	J12	222	FB_A10	D9
31	**VDD**	F7	95	S0_RESET	T8	159	MSCLK	J15	223	FB_A11	E9
32	D8	J1	96	**VDD**	G6	160	**VSS**	J8	224	FB_D3	B9
33	D7	J2	97	**VSS**	G10	161	**VDD**	L7	225	FB_D4	A9
34	**VSS**	C14	98	S0_nIOIS16	R9	162	PPDATA3_KPX3	H16	226	**VDD**	R12
35	D4	J5	99	S0_nCE1	M9	163	PPDATA4_KPX4	H15	227	**VSS**	R15
36	D5	J4	100	S0_nWAIT	P9	164	PPDATA7_KPX7	H12	228	FB_D5	A8
37	D6	J3	101	S0_nOE	N9	165	PPDATA5_KPX5	H14	229	FB_D0	E8
38	D3	K5	102	S0_BVD2_nSPKR	M10	166	PPDATA6_KPX6	H13	230	FB_D6	C8
39	nRESET	K1	103	S0_nVS1	T10	167	PPDATA2_KPX2	G16	231	FB_D1	D8
40	D0	K2	104	S0_nVS2	R10	168	PPDATA0_KPX0	G15	232	FB_D13	A7
41	D2	K4	105	S0_nIORD	N10	169	PPnINIT_KPY2	G12	233	FB_D12	B7
42	D1	K3	106	S0_nIOWR	P10	170	PPnSELECTIN_KPY1	G13	234	FB_D8	E7
43	**VDD**	F8	107	**VDD**	H6	171	PPnSTROBE_KPY0	G14	235	FB_D7	D7
44	**VSS**	D4	108	**VSS**	H2	172	PPDATA1_KPX1	F16	236	FB_D11	C7
45	MBGNT	L5	109	S0_BVD1_nSTSCHG	M11	173	**VSS**	J9	237	FB_D14	A6
46	INT	L2	110	S0_VPP0	T11	174	**VDD**	L8	238	PLL_VDD	B6
47	CLK	L3	111	S0_VPP1	R11	175	PPSELECT_KPY5	F12	239	**VDD**	G11
48	MBREQ	L4	112	S0_VCC0	P11	176	PPnAUTOFD_KPY3	F14	240	**VSS**	T9
49	A13	M4	113	S0_VCC1	N11	177	PPnACK_KPY4	F13	241	FB_D9	E6
50	A10	M1	114	S1_nCD1	N12	178	PPBUSY_KPY9	E13	242	PLL_VSS	C6
51	A11	M2	115	S1_BVD1_nSTSCHG	T12	179	PPPERROR_KPY6	E16	243	FB_D10	D6
52	A14	M5	116	**VSS**	H7	180	PPnFAULT_KPY7	E15	244	**VSS**	J16
53	A12	M3	117	**VDD**	H11	181	nFB_UCAS	E12	245	FB_D15	A5
54	A15	N1	118	S1_nCD2	P12	182	PPBUFEN_KPY8	E14	246	**VDD**	J6
55	A16	N2	119	S1_nOE	M12	183	BAT_FLT	D16	247	DAC_VDD	D5
56	**VSS**	E5	120	S1_nCE2	T13	184	**VSS**	J10	248	IREF	C5
57	**VDD**	F9	121	S1_nCE1	R13	185	**VDD**	L9	249	RED	A4
58	A17	N3	122	S1_READY_nIREQ	N13	186	GPB5_KPY15	D15	250	BLUE	B4
59	A20	P1	123	GPB6	P13	187	GPB4_KPY14	D14	251	GREEN	C4
60	A21	P2	124	S1_nIORD	T14	188	GPB2_KPY12	C16	252	VSYNC	A3
61	nCAS1	R1	125	S1_RESET	R14	189	GPB1_KPY11	C15	253	DAC_VSS	B3
62	A18	N4	126	S1_BVD2_nSPKR	T15	190	GPA7	B16	254	HSYNC	A2
63	**VSS**	F6	127	**VSS**	H8	191	**VSS**	K7	255	**VSS**	K9
64	**VDD**	F10	128	**VDD**	J11	192	**VDD**	L10	256	**VDD**	L1

Note: All **VDDX1**, **VDDX2**, and **VDDX3** pins should be connected directly to the **VDDX** power plane of the system board. **VDDP** should be connected directly to the **VDD** plane of the system board.

21.2 Power Dissipation

Table 21-3 specifies the power consumption at 3.3 V VDD for the SA-1101.

Table 21-3. SA-1101 Power Consumption at 3.3 V VDD

Parameter	Core Plus I/O IDD, Typical	DAC IDD, Typical	Total IDD, Typical	Total IDD, Maximum	Units
Run — USB off, Video off	53	—	53	60	mA
Run — USB on, Video off	65	—	65	80	mA
Run — USB on, Video 640x480	83	62	145	174	mA
Run — USB on, Video 800x600	90	62	152	182	mA
Run — USB on, Video 1024x768	110	62	172	200	mA
Doze	20	—	20	24	mA
Sleep	—	—	—	20	µA

21.2.1 Power Supply Voltages

Table 21-4 specifies the power supply voltages for the SA-1101.

Table 21-4. SA-1101 Power Supply Voltages

Parameter	Value	Units
VDD		
Minimum internal power supply voltage	3.0	V
Nominal internal power supply voltage	3.3	V
Maximum internal power supply voltage	3.6	V
VDDX		
Minimum external power supply voltage	3.0	V
Nominal external power supply voltage	3.3	V
Maximum external power supply voltage	3.6	V
PLL_VDD		
Minimum PLL power supply voltage	3.0	V
Nominal PLL power supply voltage	3.3	V
Maximum PLL power supply voltage	3.6	V

21.2.2 Absolute Maximum Ratings

Table 21-5 lists the absolute maximum ratings for the SA-1101.

Table 21-5. Absolute Maximum Ratings

Symbol	Parameter	Minimum	Maximum	Units	Note
VDD	Core supply voltage	VSS – 0.5	5.5	V	1
V_{IP}	Voltage applied to any pin	VDD – 0.5	5.5	V	1
T_S	Storage temperature	– 40	125	°C	1
T_J	Junction temperature	– 40	125	°C	1

NOTE:
1. These are stress SA-1101 ratings only. Exceeding the absolute maximum ratings may permanently damage the device. Operating the device at absolute maximum ratings for extended periods may affect device reliability.

21.2.3 DC Operating Conditions

Table 21-6 lists the functional operating dc parameters for the SA-1101.

Table 21-6. SA-1101 DC Operating Conditions

Symbol	Parameter	Minimum	Nominal	Maximum	Units	Note
V_{IHC}	Input high voltage	0.8 x VDDX	—	VDDX	V	1, 2
V_{ILC}	Input low voltage	0.0	—	0.2 x VDDX	V	1, 2
V_{OHC}	Output high voltage	0.8 x VDDX	—	VDDX	V	1, 3
V_{OLC}	Output low voltage	0.0	—	0.2 x VDDX	V	1, 3
I_{OHC}	High level output current	—	—	– 2	mA	—
T_A	Ambient operating temperature	– 0	—	70	°C	—
I_{IN}	IC input leakage current	—	10	—	µA	—
I_{OH}	Output high current (Vout = V_{DD} – 0.4 V)	—	2	—	mA	—
I_{OL}	Output low current (Vout = V_{SS} + 0.4 V)	—	2	—	mA	—
C_{IN}	Input capacitance	—	5	—	pF	—
ESD	HBM model ESD	—	1	—	KV	—

NOTES:
1. Voltages measured with respect to VSS.
2. I – CMOS-level inputs (includes I and I/O pin types).
3. O – Output, CMOS levels, tristateable.

Support, Products, and Documentation

If you need general information or support, call **1-800-628-8686** or visit Intel's website at:

http://www.intel.com

Copies of documents that have an ordering number and are referenced in this document, a product catalog, or other Intel literature may be obtained by calling **1-800-548-4725** or by visiting Intel's website for developers at:

http://developer.intel.com