Intel® StrongARM* SA-1111 Microprocessor Companion Chip

Developer's Manual

July 2000

Order No: 278242-003

Contents

Figures

Tables

Introduction 1

The Intel® StrongARM* SA-1111 Microprocessor Companion Chip (SA-1111) is a companion chip to the Intel® StrongARM* SA-1110 Microprocessor (SA-1110), providing a variety of functions suitable for use in a high-performance handheld computer system. Used with the SA-1110, this companion chip enables complete systems to be built with greatly reduced chip count, low power, and high performance.

1.1 Conventions

All fields in register descriptions that are labeled as *reserved* must be set to 0 (zero) when the data structure is created. Software should not modify or rely on these reserved field values after a data structure is created.

Certain areas of memory may be referred to as *reserved memory* in this documentation. Reserved memory- when referring to memory locations - implies that an implementation of the SA-1111 architecture may use this memory for some special purpose. For example, memory-mapped peripherals might be located in reserved memory areas on future implementations.

1.1.1 Specifying Bit and Signal Values

The terms *set* and *clear* in this documentation refer to bit values in register and data structures. If a bit is set, its value is 1; if the bit is clear, its value is 0. Likewise, setting a bit means giving it a value of 1 and clearing a bit means giving it a value of 0.

The terms *assert* and *deassert* refer to the logically active or inactive value of a signal or bit, respectively. A signal is specified as an active 0 signal when it contains a lower-case *n*. For example, the signals nRESET and S0_nWE are asserted by driving the signal to a logic 0 value.

1.1.2 Representing Numbers

All numbers in this documentation can be assumed to be base 10 unless designated otherwise. In text, binary numbers are sometimes designated with a subscript 2 (for example, 001_2). If it is obvious from the context that a number is a binary number, the "2" subscript may be omitted.

Hexadecimal numbers are designated in text with the suffix h (for example, FFFF FF5Ah) or are prefixed with 0x (for example, 0xFF).

1.1.3 Register Names

Memory-mapped registers and several of the global and local registers are referred to by their generic register names, as well as descriptive names which describe their function. Throughout this documentation, the registers' descriptive names, numbers, operands and acronyms are used interchangeably, as dictated by context.

Groups of bits and single bits in registers and control words are called either *bits*, *flags* or *fields*.

These terms have a distinct meaning in this documentation:

- **bit** —Controls a processor function; programmed by the user.
- **flag** —Indicates status. Generally set by the processor; certain flags are user programmable.
- **field** —A grouping of bits (bit field) or flags (flag field).

1.1.4 Bits in Register Descriptions

The following abbreviations are used in register descriptions to indicate the type of bit in the register.

Table 1-1. Bit Definitions

Abbreviation	Definition	Type of Bit
RW	Read and Write	Written values are maintained by the register and are readable.
WO	Write Only	The values written to this bit are not readable. A record of them must be maintained by software.
RO	Read Only	The values in this bit are readable, but cannot be modified by software.
W1C	Write 1 to Clear:	A 1 written to this bit clears its value to 0. Writing 0 to it has no effect.
WO/RO	Write Only and Read Only for two separate items in one bit field	Two separate items are accessed by writing or reading this bit field. One is read-only, the other is write-only. One example would be two entirely separate pieces of hardware accessed through the same physical address, such as the tail of an output FIFO and the front of an input FIFO.
W1S/R0	Write Strobe on Data 1 with Read Only	Writing a 1 to this field produces the described action. Writing a 0 has no effect. A value can be read from the bit field that is not necessarily the same as any value written to it.

1.2 Related Documentation

The following documents are referenced in this manual:

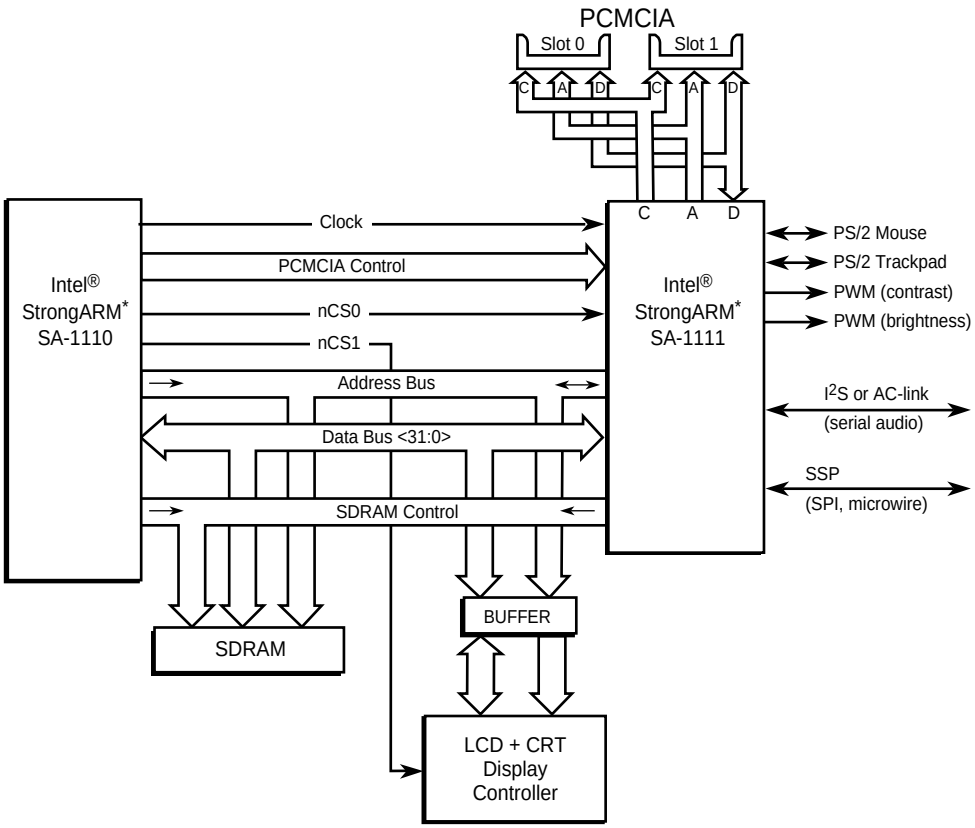
- *Universal Serial Bus Specification Revision 1.1*
- *Open HCI - Open Host Controller Specification for USB*
- *Audio Codec ’97 Component Specification*

1.3 System Block Diagram

The SA-1111 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 USB Host Controller, the SA-1111 includes extensive support for PCMCIA and Compact Flash (CF), two PS/2 ports, two industry-standard serial ports, and other I/O capabilities. It can acquire the system memory bus and do DMA transfers to system memory (SDRAM) with its high-performance memory controller.

Figure 1-1 shows how the SA-1111 can be used with the SA-1110 in a handheld computing device. The external display controller/graphics accelerator is optional, for higher performance with an LCD display, or to enable simultaneous display to a video monitor.

Figure 1-1. System Block Diagram



* Other brands and names are the property of their respective owners.

A6256-01

1.4 Signal Descriptions

Table 1-2 describes signals on SA-1111 pins.

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

Name	Type	Description
SA-1110 Processor Interface - Register Access and SDRAM Interfacing		
A<25:0>	I/O	26-bit system address bus. Bits A<24:10>, as outputs, are used for DMA access to SDRAM. Bits A<12:2> are used for processor access to SA-1111 registers and SRAM buffers. For PCMCIA access, all 26 bits are used and buffered by the SA-1111.
D<31:0>	I/O	32-bit memory data bus.
nOE	I/O	Output enable, for READ accesses.
nWE	I/O	Write enable, for WRITE accesses. In conjunction with nCAS/DQM<3:0>, specifies and controls byte writes.
nCAS/DQM<3:0>	I/O	Byte enables for SDRAM and SRAM-type WRITE transfers.
nCS	I	Chip select for enabling all SA-1111 accesses.
RDY	O	Ready signal, to system bus master - tristate or open-drain output.
MBREQ	O	Request system memory bus, to SA-1110 processor.
MBGNT	I	Memory bus grant, from SA-1110 processor.
nSDCS	O	Synchronous DRAM Chip Select, for one physical bank.
nSDRAS	O	Row Address Strobe, to SDRAM.
nSDCAS	O	Column Address Strobe, to SDRAM.
SDCLK	O	Clock to SDRAM - normally 48 MHz.
SA-1110 Interface - PCMCIA and Compact Flash Support		
nPREG	I	PCMCIA select I/O (attribute memory) space
nPOE	I	PCMCIA output enable. This PCMCIA signal is input, gated within the SA-1111, and used to perform reads from memory and attribute space.
nPWE	I	PCMCIA write enable. This signal is input, gated within the SA-1111, 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-1111, 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-1111, 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-1111, 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 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-1110.

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

Name	Type	Description
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.
PCMCIA Socket (Slot 0) Interface		
S0_nPREG	O	Socket 0 select attribute memory space.
SO_DATA<15:0>	I/O	Socket 0 data.
SO_ADDRESS<25:0>	O	Socket 0 address.
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 (2 = high byte, 1 = low byte).
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 signal/card status changed.
S0_BVD2_nSPKR	I	Socket 0 VDD voltage sense signal/audio digital speaker.
S0_nVS<2:1>	I	Socket 0 VSS voltage sense signals.
PCMCIA Power Control/GPIO		
PCMCIA_PWR/GPIO_A<3:0>	I/O	PCMCIA power control/GPIO Block A<3:0>.
Compact Flash Socket (Slot 1) Interface		
S1_nPREG	O	Socket 1 select attribute memory space.
S1_DATA<15:0>	I/O	Socket 1 data.
S1_ADDRESS<10:0>	O	Socket 1 address.
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 (2 = high byte, 1 = low byte).
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.

Table 1-2. Signal Descriptions (Sheet 3 of 4)

Name	Type	Description
S1_BVD1_nSTSCHG	I	Socket 1 VDD voltage sense signal/card status changed.
S1_BVD2_nSPKR	I	Socket 1 VDD voltage sense signal/audio digital speaker.
S1_nVS<2:1>	I	Socket 1 VSS voltage sense signals.
USB Interface		
USB_PLUS	I/O	USB transceiver plus.
USB_MINUS	I/O	USB transceiver minus.
USB_PWRCNTL/GPIO_A<4>	O	USB power control/GPIO_A bit 4
USB_PWR_SENSE	I	USB over voltage sense
PS/2 Trackpad Interface/GPIO_B Bits		
TPCLK/GPIO_B<2>	I/O - od	PS/2 track pointer clock or GPIO_B bit 2
TPDATA/GPIO_B<3>	I/O - od	PS/2 track pointer data or GPIO_B bit 3
PS/2 Mouse Interface/GPIO_B Bits		
MSCLK/GPIO_B<4>	I/O - od	PS/2 mouse clock or GPIO_B bit 4; also L3_CLK for L3 Control Bus.
MSDATA/GPIO_B<5>	I/O - od	PS/2 mouse data or GPIO_B bit 4; also L3_DATA for L3 Control Bus.
Brightness/Contrast PWM DACs/GPIO_B Bits		
PWM0/GPIO_B<0>	I/O	PWM output 0 or GPIO_B bit 0.
PWM1/GPIO_B<1>	I/O	PWM output 1or GPIO_B bit 1; also L3_MODE for L3 Control Bus.
PLL Group		
CLK	I	Master clock in, 3.6864 MHz. Connect to SA-1110 GPIO<27>.
PLL_VDD	P/G	Power for PLL.
PLL_VSS	P/G	GND for PLL.
AC-link[a] Serial Port for Audio/GPIO_C Bits		
SYS_CLK	O	System Clock for I^2S Audio Port.
BIT_CLK/GPIO_C<0>	I/O	12.288 MHz Bit Clock for Serial Audio, from external AC'97 Codec.
SYNC/GPIO_C<1>	I/O	48 KHz Sync Signal (Frame Indicator), to external AC'97 Codec.
SDATA_OUT/GPIO_C<2>	I/O	Serial Audio Data to external AC'97 Codec, or GPIO.
SDATA_IN/GPIO_C<3>	I/O	Serial Audio Data from external AC'97 Codec, or GPIO.
SSP Synchronous Serial Port/GPIO_C Bits		
SCLK/GPIO_C<4>	I/O	SSP Serial Clock to external device, or GPIO.
SFRM/GPIO_C<5>	I/O	Serial Frame signal for SSP Port, or GPIO.
TXD/GPIO_C<6>	I/O	Serial Transmit Data from SSP Port to external device, or GPIO.
RXD/GPIO_C<7>	I/O	Serial Receive Data to SSP Port from external device, or GPIO.
Miscellaneous Signals		
INT	O	Interrupt out.

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

Name	Type	Description
nRESET	I	SA-1111 hardware reset.
nTEST	I	“not Test” is a signal used for off-board production testing. It MUST be tied HIGH for normal operation. If it is driven low in a system environment, internal damage to the device may occur.
BAT_FLT	I	Battery fault or hardware-initiated Sleep.
Power and Ground		
VDD	P/G	Positive supply for the core. Pins CVDD<3:0> are allocated for this supply.
VDDX	P/G	Positive supply for the I/O pins. Pins RVDD<10:1> are allocated for VDDX.
VDD_PCMCIA	P/G	Positive supply (0, 3.3, or 5V) for PCMCIA slot. Pins PVDD<2:0> are allocated for this supply.
VDD_CF	P/G	Positive supply (0 or 3.3V) for Compact Flash (CF) slot. Pins CFVDD<2:0> are allocated for this supply.
VSS	P/G	Ground supply for the core. Pins CVSS<3:0> are allocated to this supply.
VSSX	P/G	Ground supply for all I/O pins, including PCMCIA and CF signals. Pins RVSS<17:0> are allocated to this supply.

a. Compliant with AC-Link standard 1.03.

Note: The signal types defined in Table 1-2 are as follows:

I = Input only

O = Output only

I/O - od = Input/Output - behaves like an open-drain output in PS/2 mode of operation

I/O - Input/Output

P/G= Power/Ground

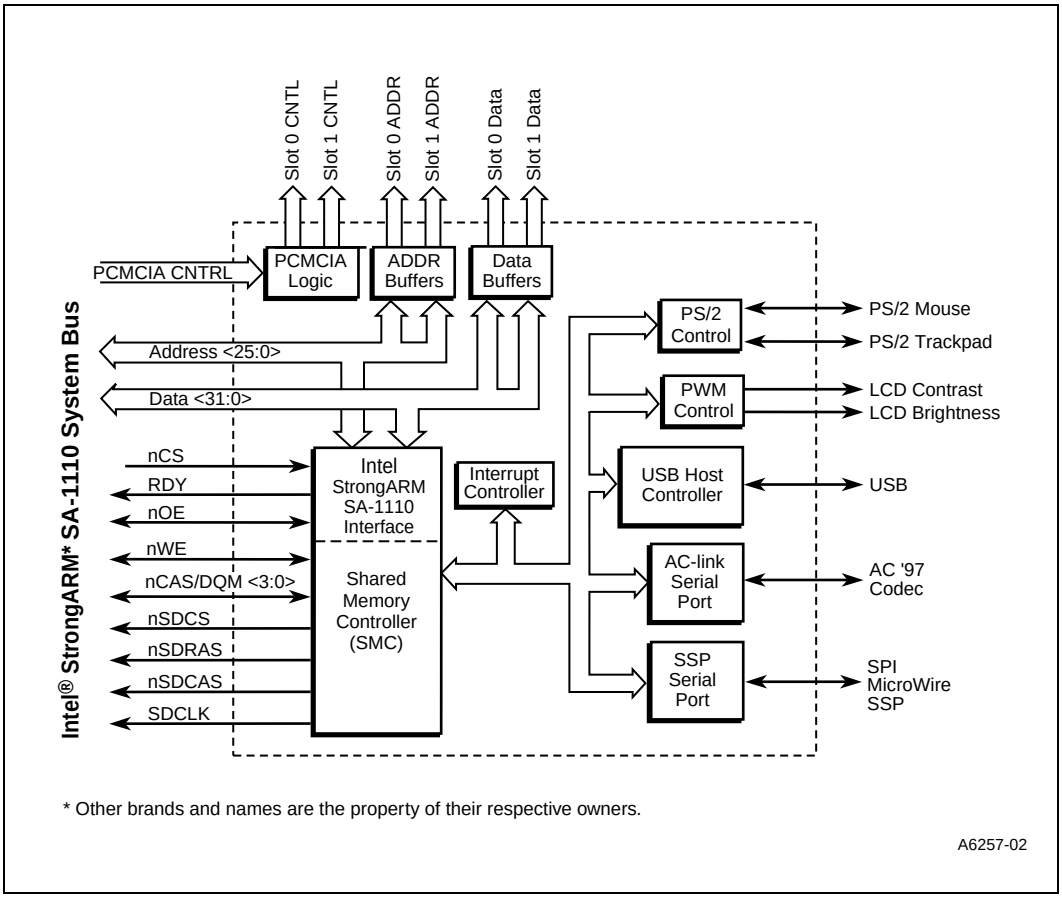
Functional Description 2

This chapter provides a functional overview of the Intel® StrongARM* SA-1111 Microprocessor Companion Chip (SA-1111), which is a companion chip to the Intel® StrongARM* SA-1110 Microprocessor (SA-1110).

2.1 Functional Blocks

Figure shows the functional blocks of the SA-1111.

Figure 2-1. SA-1111 Block Diagram



2.1.1 System Bus Interface (SBI)

The System Bus Interface (SBI) functional block connects the SA-1111 internal bus (and through that bus, all other functional blocks) to the external SA-1110 system bus. The SBI is the subsystem through which all data and control information moves between the SA-1111 and the SA-1110 processor.

The SBI is used for three distinct types of transfers:

- Register READs and WRITEs—These are SRAM-like single-word transfers, initiated by the system processor (SA-1110). Because the SBI makes use of the RDY signal, register accesses can be of varying length, depending on the latency of accessing the subsystem to which the access is directed. In particular, the USB block has long register accesses, due to synchronizing delays between timing domains, but other subsystems have shorter accesses.
- Block-data READs and WRITEs—Block-data transfers are used to fill or empty SRAM buffers serving the serial-port subsystems. Block-data transfers typically take place as SRAM-like bursts across the system bus. Bursts can be up to 8 words long. These transfers are also initiated by the system processor.
- DMA to System Memory—The SBI includes a Shared Memory Controller (SMC) sub-unit, which responds to data-transfer requests from the USBand Serial Audio Controller. When requested, it acquires control of system memory (SDRAM) and moves data between DRAMs and the functional block requesting the transfer.

2.1.2 Intel® StrongARM* SA-1111 Subsystems

The SA-1111 includes the following functional blocks:

- USB Host Controller

  A full-function USB host controller, compliant with *USB Specification, Rev. 1.1* is provided. Supported modes include both slow (1.5 Mbits/s) and fast (12 Mbits/s) speeds of operation.
- PS/2 ports

  Two PS/2 ports are provided for use with keyboards, mice, trackpads or other PS/2-compliant devices. PS/2 pins behave like open-drain I/Os when used in PS/2 mode. When not used for PS/2 attachment, the pins may be used as GPIO.
- PCMCIA and CF (Compact Flash) interface

  An interface is provided to directly drive all signals for two sockets - one for PCMCIA, one for Compact Flash - without external buffering. Full card detection and PCMCIA voltage control are provided, so both 3.3V and 5V PCMCIA cards are supported.
- Pulse width modulation outputs

  Two pulse-width-modulation (PWM) outputs are provided. These function as 8-bit-accurate D-to-A (digital-to-analog) converters, with the addition of inexpensive external filter components. They are typically used to generate brightness and contrast control voltages for LCD panels. When not used for PWM output, the pins may be used for GPIO.
- AC-link/I^2S Serial Port for Audio

  The full-duplex serial port for audio transfers serialized audio data to and from external devices using AC-Link, I^2S or “MSB-Justified” formats. These formats are used by many popular audio D-to-A converters and codecs.

  The AC-Link fully supports AC’97 codec that are compliant with AC-Link standard 1.03—some features added to later revision AC’97 codecs may not be supported. For AC’97 codecs, digitized audio can pass in both directions simultaneously using sample sizes up to

16 bits. For AC'97 codecs, the playback or audio out (SA-1111 to codec) is fixed at 48 KHz. Variable-rate sampling for playback, which is not defined in AC-Link standard 1.03, is not supported in hardware. The serial audio port supports variable sample rates for incoming audio (codec to SA-1111). Codecs compliant with AC-Link standard 1.03 or later can select the incoming audio sample rate by software control.

The audio serial port also supports I^2S or "MSB-Justified" mode requirements. Sample rates for I^2S may be 8, 11.025, 16, 22.05, 32, or 44.1 KHz. Some I^2S codecs require an additional bus for control. The L3 Bus uses three pins (PWM<1>, MSCLK, and MSDATA) to convey control information to the codec if L3 Mode is enabled. In this mode, the processor writes control bytes into the L3 register. The data is serialized and sent to the codec automatically, following L3 protocol.

The clock pin is bidirectional. The serial audio bit-rate clock may be provided by the codec (AC'97 devices), by an external oscillator, or from a programmable divider using the SA-1111 internal clock.

The Serial Audio Port can use DMA to transfer data between the external codec and system DRAM without processor intervention. If DMA is not enabled, then programmed I/O (block data transfers) can be used. Interrupts to the host processor are minimized by inclusion of dual FIFO buffers that store up to 16 samples in each direction.

If the Serial Audio function is disabled, its pins may be used for GPIO functions.

- SSP Serial Port

  Full-duplex synchronous serial interfacing is provided for attachment to modem, telecom, and other devices using serial protocols for data transfer. It supports National Microwire, TI Synchronous Serial Protocol (SSP), and Motorola Serial Peripheral Interface (SPI) serial protocols. Serial bit clock can be internally-generated up to 1.8 MHz (must be divided down from internal 3.68 MHz), or it may be supplied by an external device. Independent transmit and receive FIFOs are provided.

  The SSP Port data FIFOs may be accessed by system-initiated transfers that make use of SRAM-like block data moves, initiated by the system processor. These transfers typically require the processor to monitor FIFO levels via register polling, or using interrupts from the SSP to alert the processor to impending FIFO overflow or empty condition.

  If the SSP Serial Port function is disabled, its pins may be used for GPIO functions.

2.2 Clock Generation and Distribution

The SA-1111 input clock is a single 3.6864-MHz clock generated by the SA-1110. The clock is sent from the system processor's GPIO<27> to the CLK pin on the SA-1111. A phase-locked loop (PLL) in the SA-1111 generates the clocks required for its I/O functions and internal systems. The input clock goes to the PLL's phase comparator. The VCO 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:

- 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).
- Internal DMA bus–Operates at the frequency of the Shared Memory Controller (SMC). It uses a clock (DCLK) of 48 MHz. DMA bus support functions, like the bus arbiter, also use this clock.
- PS/2 clock–Can be programmed to function at 2, 4, or 8 MHz.
- DACs for LCD brightness/contrast control–Uses the 3.6864-MHz clock, direct and ungated, so they function even if all other sections of the device, including the PLL, are shut down to reduce power.

- SSP Serial Port–Can operate with internally-generated clock based on divided-down functionality.
- Serial Audio Port, in AC-link mode, uses 12.288 MHz supplied by the external AC'97-compatible device on its BIT_CLK input. This is divided by 256 internally and exported as the SYNC signal back to the device. In I^2S or "MSB-Justified" mode, the port supplies clocks to the external codecs (SYS_CLK and BIT_CLK are outputs), based on a programmable divider using 144 MHz as its input. The SYNC output is divided down from the BIT_CLK frequency.

These frequencies are distributed from, and controlled by, the System Controller. The enables for each of these clocks also are located in the System Controller, with the exception of enables for RCLK and the PLL, which are controlled by register bits in the System Bus Interface.

2.3 System Reset

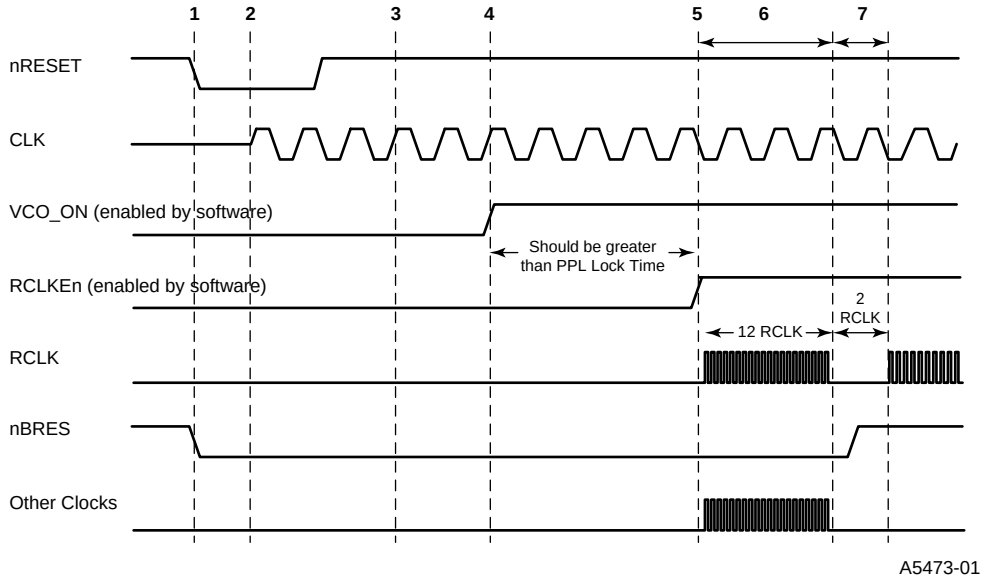
The SA-1111 is reset by asserting nRESET. When nRESET is asserted, all on-chip activity halts; when nRESET is released, the SA-1111 goes to *doze mode* (PLL operating, but clocks to some or all subsystems shut off). To enable functions of the SA-1111 and put it into *normal operational* mode, software must enable the PLL and RCLK. Unless indicated otherwise, all register bits are set to zero during reset. For information about pin states after reset, see Table 2-1.

The recommended reset sequence for the SA-1111 is as follows:

1. The nRESET is asserted. This asserts nBRES (the SA-1111 internal system reset).
2. The CLK is activated from the SA-1110. This is the clock input to the PLL's phase comparator and to the Sleep State Machine.
3. The nRESET is deasserted.
4. The VCO_ON is asserted by software, and PLL_Bypass deasserted.
5. After a period of time (greater than the LOCK time of the PLL), software asserts RCLKEn.
6. The clock controller activates RCLK for 12 cycles. This burst is routed to all clock nets in the SA-1111.
7. The Power Control Register must be set by software to allow clocking of individual functions on the SA-1111.
8. The USB must be configured before it can be used.

Figure 2-2 illustrates a reset sequence for the SA-1111.

Figure 2-2. SA-1111 Reset Sequence



2.4 Modes of Operations

The SA-1111 operates in three modes:

- Normal–The SA-1111 is fully operational.
- Doze–One or more of the system's functional blocks are not clocked resulting in the dissipation of less power.
- Sleep–All clocks are disabled resulting in a substantial reduction in power.

2.4.1 Normal Mode

Normal mode is full operation of the SA-1111 device. All clocks are on, and all functions can be used, individually enabled by clock-control bits in the System Controller.

2.4.2 Doze Mode

A software-controlled sequence of register writes can put the device into *doze mode.* This is similar to normal mode, except that one or more of the system's functional blocks are not clocked. Enabled (active) functional blocks continue to operate, while clocks to other blocks are switched off. DMA capability is disabled. This feature allows the SA-1111 to dissipate lower power when any functional blocks are not being used.

Some systems, like the PWM outputs and PS/2 ports, can operate autonomously even when the main bus clock (RCLK) is shut off. However, RCLK must be turned on for any register accesses to functional blocks beyond the SBI (System Bus Interface). SBI registers, on the other hand, are latch-based and can be written without RCLK enabled.

RCLK is an internal clock used for all register access except for the SBI registers. RCLK provides all clocking for the System Bus Interface and the internal system bus. If RCLK is turned off, only the latch-based Control Register (SKCR) in SBI block can be accessed. Before switching it off:

— All USB activity should be stopped and UCLK should be turned off.

— When the RCLK is switched off, the PS/2, SSP, USB, and audio data continues to function and can generate interrupts.

The System Controller and PWM outputs operate independently of the VCO_On signal and whether or not the PLL circuit is enabled. The System Controller can:

— Receive PCMCIA and Compact Flash interrupts.

— Receive USBPortResume interrupt.

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

- PCLK
  Provides 8 MHz clock for the PS/2 ports. Selectively switching it on or off allows:
  — Power down of an unused PS/2 port.
  — Continued functioning of a PS/2 port when RCLK is turned off.
- CLK_In
  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.
  — Continued functioning of the PWM outputs when RCLK is turned off.
- UCLK
  Provides all clocking for the USB Host Controller. Selectively switching it on or off permits powering down the USB Host Controller, if it is not being used.
- SYS_CLK and/or BIT_CLK
  An externally-supplied serial bit-rate clock at 12.288 MHz (from AC'97 Codec device or similar) to the AC-link (or I^2S) Serial Audio controller. BIT_CLK can be gated off internally to SA-1111, putting the AC-link controller into low-power (off) state. The preferred method is to shut off this subsystem and reduce power to send the appropriate power-down command, encoded in the SDATA_OUT bit stream, to the external AC'97 Codec. The Codec then proceeds through a power-down sequence and shut off BIT_CLK itself. To re-start the Codec and its BIT_CLK output, SA-1111 asserts its SYNC output for less than 1 microsecond.
  Alternatively, for I^2S (or MSB-Justified) operation, the serial audio clock is generated internally and SYS_CLK and BIT_CLK are outputs. Clock control is similar to that with AC-link, except another mechanism, which uses a separate control lines, must be used to shut down the external codec.
  The Serial Audio subsystem can continue to operate autonomously (moving data between its FIFO buffers and external codec) in Doze Mode, but the system must re-enable DMA capability if it relies on DMA to move data to and from the system DRAM.
- SCLK
  The clock for the SSP Serial Controller, which typically ranges from 8 KHz to 2 MHz. Selectively switching it on or off allows powering down of the Serial Controller.
- DCLK
  Shared Memory Controller and DMA Bus (DCLK), either 36 MHz or 48 MHz.

2.4.3 Sleep Mode

Sleep Mode allows substantial reduction in system power consumption (to several micro watts) when none of the functions of the SA-1111 are required. In addition to stopping all clock activity on the SA-1111, Sleep Mode also:

- Places the pins of the device into a minimum power configuration.
- Sets the PCMCIA voltage control lines (GPIO_A<3:0>) to their sleep state register values.
- Sets the PCMCIA address, data, and control lines according to the value in the PCMCIA sleep state register.
- Selects the wake-up mode of the Interrupt Controller.

These interrupts are ORed together and, if enabled, signal the interrupt to the system processor on the INT pin. Any interrupt may be programmed to bring the SA-1111 out of Sleep Mode, if enabled in the independent WakeUp Register. If the Sleep Mode was initiated by BAT_FLT assertion, interrupts are disabled and any existing interrupts are cleared.

Sleep Mode is initiated by:

- Software setting the sleep bit in the SA-1111 Control Register
- Power-fault signal on BAT_FLT (interrupts disabled)

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 reset condition.
- **State 1 – Request**

  This state is entered from the active state, by either:
  - Writing the value “1” to the Sleep Bit of the SA-1111 Control Register (SKCR).
  - Assertion of BAT_FLT
- **State 2 – Shutdown_1**
  - Shutdown_1 releases the system bus if MBGNT is granted and causes the Power Control Register to be cleared, which disables all clocks.
- **State 3 – Shutdown_2**

  This state is unconditionally entered from Shutdown_1. Because all subsidiary clocks have ceased, this state disables RCLK and the PLL.
- **State 4 – Sleep**

  This state is unconditionally entered from Shutdown_2. The following conditions are set within the SA-1111:
  - Pads are set into their associated sleep condition.
  - GPIO output 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 PLL is turned off.

- **State 5 – Wake-up**

  This state is entered from sleep mode anytime nCS goes active (any read or write to SA-1111). The pads, except for those in GPIO mode, are released from their sleep condition and return to normal operation. If BAT_FLT is asserted, the state machine transitions back to *Sleep Mode*.

  When the state machine returns to the active state, the SA-1111 Control Register (SKCR) can be accessed, allowing software to re-enable the PLL and RCLK. Reconfiguration of the SA-1111 for normal operation can then be carried out as required.

Note: Prior to activating the sleep bit, a zero must be written to the SKPWM register in the system controller to disable PWM outputs.

Table 2-1. Pin State After Reset and During Sleep State (Sheet 1 of 3)

Name	Type	Reset State	Sleep State
SA-1110 Processor Interface - Register Access and SDRAM Interfacing			
A<25:0>	I/O	Input	Input
D<31:0>	I/O	Input	Input
nOE	I/O	Input	Input
nWE	I/O	Input	Input
DQM<3:0>	I/O	Input	Input
nCS	I	Input	Input
RDY	O	Tri-stated	Tri-stated
MBREQ	O	Low	Low
MBGNT	I	Input	Input
nSDCS	O	Tri-stated	Tri-stated
nSDRAS	O	Tri-stated	Tri-stated
nSDCAS	O	Tri-stated	Tri-stated
SDCLK	O	Tri-stated	Tri-stated
SA-1110 Interface - PCMCIA and Compact Flash Support			
nPREG	I	Input	Input
nPOE	I	Input	Input
nPWE	I	Input	Input
nPIOW	I	Input	Input
nPIOR	I	Input	Input
nPCE<2:1>	I	Input	Input
nIOIS16	O	High	High
nPWAIT	O	High	High
PSKTSEL	I	Input	Input
PCMCIA Socket (Slot 0) Interface			
S0_nPREG	O	Tri-stated	Note 2
SO_DATA<15:0>	I/O	Input	Input

Table 2-1. Pin State After Reset and During Sleep State (Sheet 2 of 3)

Name	Type	Reset State	Sleep State
SO_ADDRESS<25:0>	O	Tri-stated	Note 3
S0_nCD<2:1>	I	Input	Input
S0_READY_nIREQ	I	Input	Input
S0_nCE<2:1>	O	Tri-stated	Note 2
S0_nOE	O	Tri-stated	Note 2
S0_nWE	O	Tri-stated	Note 2
S0_nIORD	O	Tri-stated	Note 2
S0_nIOWR	O	Tri-stated	Note 2
S0_nWAIT	I	Input	Input
S0_nIOIS16	I	Input	Input
S0_RESET	O	Tri-stated	Note 3
S0_BVD1_nSTSCHG	I	Input	Input
S0_BVD2_nSPKR	I	Input	Input
S0_nVS<2:1>	I	Input	Input
PCMCIA Power Control/GPIO			
PCMCIA_PWR/GPIO_A<3:0>	I/O	Input	Note 1
Compact Flash Socket (Slot 1) Interface			
S1_nPREG	O	Tri-stated	Note 4
S1_DATA<15:0>	I/O	Input	Input
S1_ADDRESS<10:0>	O	Tri-stated	Note 5
S1_nCD<2:1>	I	Input	Input
S1_READY_nIREQ	I	Input	Input
S1_nCE<2:1>	O	Tri-stated	Note 4
S1_nOE	O	Tri-stated	Note 4
S1_nWE	O	Tri-stated	Note 4
S1_nIORD	O	Tri-stated	Note 4
S1_nIOWR	O	Tri-stated	Note 4
S1_nWAIT	I	Input	Input
S1_nIOIS16	I	Input	Input
S1_RESET	O	Tri-stated	Note 5
S1_BVD1_nSTSCHG	I	Input	Input
S1_BVD2_nSPKR	I	Input	Input
S1_nVS<2:1>	I	Input	Input
USB Interface			
USB_PLUS	I/O	Input	Input
USB_MINUS	I/O	Input	Input
USB_PWRCNTL	O	Low	Note 6

Table 2-1. Pin State After Reset and During Sleep State (Sheet 3 of 3)

Name	Type	Reset State	Sleep State
USB_PWR_SENSE	I	Input	Input
PS/2 Trackpad Interface/GPIO_B Bits			
TPCLK/GPIO_B<2>	I/O - od	1	Note 1
TPDATA/GPIO_B<3>	I/O - od	1	Note 1
PS/2 Mouse Interface/GPIO_B Bits			
MSCLK/GPIO_B<4>/L3CLK	I/O - od	Input	Note 1
MSDATA/GPIO_B<5>/L3DATA	I/O - od	Input	Note 1
Brightness/Contrast PWM DACs/GPIO_B Bits			
PWM0/GPIO_B<0>	I/O	Input	Note 1
PWM1/GPIO_B<1>/L3MODE	I/O	Input	Note 1
PLL Group			
CLK	I	Input	Input
I2S Serial Port			
SYS_CLK	O	Low	Low
BIT_CLK/GPIO_C<0>	I/O	Input	Note 1
SYNC/GPIO_C<1>	I/O	Input	Low
SDATA_OUT/GPIO_C<2>	I/O	Input	Note 1
SDATA_IN/GPIO_C<3>	I/O	Input	Note 1
SSP Synchronous Serial Port/GPIO_C Bits			
SCLK/GPIO_C<4>	I/O	Input	Note 1
SFRM/GPIO_C<5>	I/O	Input	Note 1
TXD/GPIO_C<6>	I/O	Input	Note 1
RXD/GPIO_C<7>	I/O	Input	Note 1
Miscellaneous Signals			
INT	O	Low	Active
nRESET	I	Input	Input
nTEST	I	Input	Input
BAT_FLT	I	Input	Input

Notes:

1. These pins revert to GPIO mode of operation in Sleep State. State is determined by the corresponding GPIO Sleep State and Sleep Direction bits.
2. When the Control Register (PCCR) bit 2 is set to 0, all these pins are tri-stated. When the PCCR bit 2 is set to 1, the state of these pins depends on the state of the Sleep State register (PCSSR) bit 0:
   - — When PCSSR[0} = 0: all of these pins are tri-stated.
   - — When PCSSR[0} = 1: all of these pins are high.

3. When the Control Register (PCCR) bit 2 is set to 0, all these pins are tri-stated. When the PCCR bit 2 is set to 1, the state of these pins depends on the state of the Sleep State register (PCSSR) bit 0:
   - — When PCSSR[0} = 0: all of these pins are tri-stated.
   - — When PCSSR[0} = 1: all of these pins are low.
4. When the Control Register (PCCR) bit 3 is set to 0, all these pins are tri-stated. When the PCCR bit 2 is set to 1, the state of these pins depends on the state of the Sleep State register (PCSSR) bit 0:
   - — When PCSSR[0} = 0: all of these pins are tri-stated.
   - — When PCSSR[0} = 1: all of these pins are high.
5. When the Control Register (PCCR) bit 3 is set to 0, all these pins are tri-stated. When the PCCR bit 2 is set to 1, the state of these pins depends on the state of the Sleep State register (PCSSR) bit 0:
   - — When PCSSR[0} = 0: all of these pins are tri-stated.
   - — When PCSSR[0} = 1: all of these pins are low.
6. This depends upon the StandbyEn and PwrCtlPolLow bits in USB Reset register (seeSection 6.3.3 for more information).Table 2-2 shows the pin state while in sleep mode.

Table 2-2. USB_PwrCntl Pin State in Sleep Mode

PwrCtrPolLow	StandbyEn	Usb_PwrCntl
0	0	1
0	1	0
1	0	0
1	1	1

2.5 Test

In *test mode* the PLL is put into *Bypass* mode, so that the input frequency to the PLL (CLK) connects directly to PLLCLK (the normal PLL output to all clock dividers). Clock multiplexers are provided on RCLK and DCLK nets.

System Bus Interface (SBI) 3

The System Bus Interface is the primary interface between the Intel® StrongARM* SA-1111Microprocessor Companion Chip (SA-1111) and the Intel® StrongARM* SA-1110 Microprocessor (SA-1110). Its connection to the full SA-1110 system memory bus is used for transferring data between the host processor and SA-1111 internal functional blocks.

There are two distinct mechanisms for moving data between the external system bus and SA-1111 internal blocks:

- Register reads and writes, and block data transfers, take place using SRAM-type accesses. Register reads and writes are single-cycle events, while block data transfers are bursts of up to eight sequential data words. These transfers are initiated by the system processor.
- DMA transfers between SA-1111 and system memory may be initiated by the SA-1111 in response to a request from the internal USB Host Controller or the Serial Audio Controller. The SA-1111 acquires the system memory bus and transfers data directly to or from system DRAM.

3.1 Signal Description

This section lists the signal descriptions for the interfaces between the SA-1111 and the SA-1110.

3.1.1 Intel® StrongARM* SA-1110 System Bus Interface - Signals

Table 3-1 describes the external (pin) interface to the system memory bus.

Table 3-1. SA-1110 Memory Interface Signals - O(D) = Driven for DMA

Signal	Type	Description
A<25:0>	I and O(D)	26-bit address bus
D<31:0>	I/O	32-bit data bus
nCS	I	Chip select (dedicated to SA-1111 access), asserted low
nOE	I and O(D)	Output enable, asserted low for reads
nWE	I and O(D)	Write enable, asserted low for writes
RDY	O	Indicates SA-1111 has taken write data, or read data is available
DQM<3:0>	I and O(D)	SDRAM byte enable
MBREQ	O	Request to SA-1110 for bus ownership
MBGNT	I	Grant from SA-1110 - bus is available
nSDCS/nRAS	O(D)	SDRAM Chip Select, to any bank of SDRAM
nSDRAS	O(D)	RAS (Row Address Strobe), for SDRAM
nSDCAS	O(D)	CAS (Column Address Strobe) for SDRAM
SDCLK	O(D)	SDRAM Clock (48 MHz)

3.1.2 System Bus Interface (SBI) - Internal Buses and Signals

The SBI's internal interface to other on-chip blocks consists of two synchronous buses:

- The Register Access Bus (RAB) is used for processor-initiated transfers between the processor and on-chip registers or FIFO storage. For these transfers, the SBI is always bus master. Clock frequency for this bus is 24 MHz.

  The Register Access Bus is also the source of address and data to the Compact Flash and PCMCIA buffers.

- The DMA Bus uses PCI-like protocols to transfer data between a bus master (the USB or Serial Audio blocks) and external system DRAM. For these DMA transfers, the SBI is always a slave (target), while the initiating USB or Serial Audio block is the master. Although transfers use PCI protocols, signals like LOCK and ERROR are not supported, nor are concepts like Retry or Abort. Using the TRdy and IRdy signals enables either participant to stall the transfer.

  The bus runs at the same clock frequency as the SMC (Shared Memory Controller)—48 MHz for SDRAMs.

High-order address decoding for the RAB bus is centralized in the SBI. A central arbiter chooses the DMA Bus master.

Each bus has dual single-direction 32-bit data paths. In addition, there are two 16-bit data buses for read data from the PCMCIA and CF slots. For the RAB, incoming data for register writes and PCMCIA or CF writes transfers on RegDIn. Four byte-enable bits permit any combination of bytes to be written. Read data from registers goes out on RegDOut, from PCMCIA (Slot 0) on Slot0RdData, or from Compact Flash (Slot 1) on Slot1RdData. (Note that slot data is 16 bits wide). The SBI selects one data source and drives read data onto the system bus to the processor. When slot data is read, the upper 16 bits of the system bus are forced to zero.

For DMA transfers, the initiating master sends out read/write address and data (multiplexed like PCI) on DmaADOut. Read data from system memory transfers to the DMA master over DmaDIn.

Table 3-2 describes the internal system bus signals. The following signal-type abbreviations are used in Table 3-2: input (I) and output (O), referenced to the System Bus Interface block. Table 3-3 describes the interface to the register access bus.

Table 3-2. SBI Block - interface to DMA Bus (SBI is transfer target)

Signal	Type	Description
DCLK	I	48 MHz internal bus clock
Dma_Din<31:0>	O	32-bit read data in (to on-chip DMA master)
DmaADout<31:0>	I	32-bit address and write data out (from on-chip DMA master)
DmaBE<3:0>	I	4-bit byte enable out, to write DRAMs
Frame	I	Signals start and end of a transfer
TRdy	O	Target (SBI) ready
IRdy	I	Initiator (master) ready
DmaWr	I	DMA Command: 0 = Read, 1 = Write
AnyReq	I	OR of USB, Audio DMA requests

Table 3-3. SBI Block - interface to Register Access Bus (SBI is transfer initiator)

Signal	Type	Description	Notes
RCLK	I	RAB bus clock	
nBRES	I	Internal system bus reset, asserted low	
RegAdr<8:2>	O	7-bit register-access address (decoded locally)	1
RegDin<31:0>	O	32-bit write data, to target register or (16 bit) to slot	
RegDOut<31:0>	I	32-bit read data, from target register	
Slot0RdData<15:0>	I	16-bit read data from PCMCIA slot	
Slot1RdData<15:0>	I	16-bit read data from CF slot	
PsktSelIn	I	Select source of READ data (1=CF, 0=PCMCIA)	
nDOutEnL	I	Enable PCMCIA or CF read data out, LOW byte	
nDOutEnH	I	Enable PCMCIA or CF read data out, HIGH byte	
RegBE<3:0>	O	4-bit byte enable, to write target register or buffer	
RegWr<n>	O	Signals timing of valid write data to selected block	
RegRd<n>	O	Output enable, to read from selected block	
WAIT<n>	I	Target (slave) asserts to extend cycle	

1. All register accesses have a size of 32 bits, therefore BYT_EN = 1111 for these accesses. All reserved bits are read back as zero. USB and Audio FIFOs access 32 bits of data. Where there is only an SSP FIFO, only the lower 16 bits are valid.

3.1.3 Other Internal Interface Signals

A power-quality signal, BAT_FLT, is brought in from an external pin. This signal is used to indicate a problem with the system power source. In the SA-1111, it disables assertion of MBREQ, preventing the SBI from requesting ownership of the shared bus. It also connects to other on-chip functions like the Sleep State Machine.

Several registers bits in the SBI connect to other system blocks. Table 3-4 describes these signals.

Table 3-4. Other System Interface Signals

Signal	Direction	Description
nRESET	I	System reset; from pin
BAT_FLT	I	BAT_FLT; from pin
PLL_Bypass	O	Bypass PLL, send input CLK direct to dividers; from SKCR
RCLKen	O	RCLK enable; from SKCR
Sleep	O	Sleep mode command; from SKCR
Doze	O	Force DOZE mode
PLLON	O	Enable PLL; from SKCR
ScanTestEn	O	Enable scan testing logic; from SKCR
ClockTestEn	O	Multiplex control to bring clocks to I/O pins, for test

3.2 Functional Description

This section provides a functional overview of the external (pin) signals between the SA-1111 and the SA-1110, and the logic that controls them and transfers data to/from internal destinations.

3.2.1 Register Accesses and the Register Access Bus (RAB)

The SBI responds to register reads and writes from the SA-1110, and turns them into internal RAB transfers. In this mode of operation, the SA-1111 looks like SRAM to the processor, to which it can perform single-cycle or burst accesses (see the following figures.)

When nCS is asserted, it causes the SBI to initiate an RAB transfer as RAB bus master. The type of transfer (read or write) is signaled by the processor using nOE or nWE assertion. After nCS and nOE/nWE assertion, data is placed on the internal 32-bit RAB datapath by the SBI (for writes) or by the target (for reads), enabled by RegRd<n>. An access directed to SA-1111 is identified in the SBI first by the assertion of nCS. If it is not directed to one of the registers in the SBI itself (which does not use the RAB bus), the access becomes an RAB transfer. High-order address bits A<25:9> are centrally-decoded in the SBI, and signal Stb or RdEn is sent from the decoder to the addressed block. Within the block, local address-decode logic, using A_In<8:2>, identifies which of several registers (or FIFO buffer) may be addressed. Only address signals required for this local decoding need connect to the block.

Table 3-5 lists the memory map. The *base offset* is relative to the base address of the chip select signal (nCS) used to select the SA-1111.

The address space is subdivided evenly between the twelve functional units. Each unit is allocated a 512-byte (128 words) block of memory space, regardless of how many registers or FIFO buffers the unit actually has.

Note: Only the base offset for the System Bus Interface is fully decoded and requires bits 25:13 to be all zeros. All other register addresses have bits 25:13 aliased (bits 25:13 are **not** decoded).

Table 3-5. SA-1111 Memory Map

Base Offset	Description
00000000 – 000001FF	System Bus Interface
00000200 – 000003FF	System Controller (SK)
00000400 – 000005FF	USB Host Controller
00000600 – 000007FF	Serial Audio Controller
00000800 – 000009FF	SSP Serial Port
00000A00 - 00000BFF	PS/2 Trackpad Interface
00000C00 - 00000DFF	PS/2 Mouse Interface
00001000 – 000011FF	GPIO Blocks A, B, C
00001200 – 000013FF	Reserved
00001400 – 000015FF	Reserved
00001600 – 000017FF	Interrupt Controller
00001800 – 000019FF	PCMCIA Interface

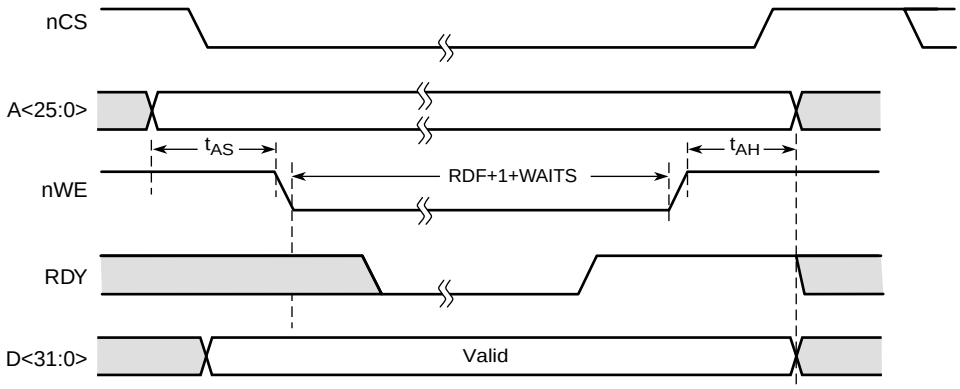
Note: A major difference between SA-1101 and SA-1111 is that the former only connected to address bits <21:10>, while the SA-1111 connects to all 26 address bits. This permits register addresses to be spaced at 4-byte intervals, rather than 1K bytes as required for the SA-1101.

Signal Stb indicates the availability of Write data on Data_In to the target block; RdEn tells the target to enable Read data onto Data_Out bus. If the target needs more time, it asserts the Wait signal. Data flows through the SBI, unclocked, between external system bus and the RAB data bus. Once data has been latched (for a write) or is ready to be sampled (read), the SBI asserts RDY to the system processor. The processor, after some delay, terminates the cycle by deasserting nOE/nWE and nCS. Register accesses always transfer single (32-bit) words. Cycles may be of variable length, depending on the timing of RDY asserted by SA-1111 back to the processor.

Note the RAB has two modes of access. For fixed-latency transfers, use of RDY should be disabled in the SA-1111 control register (SKCR). All accesses have this length.

Figure 3-1, Figure 3-2, Figure 3-3, and Figure 3-4 illustrate the variable latency mode of operation.

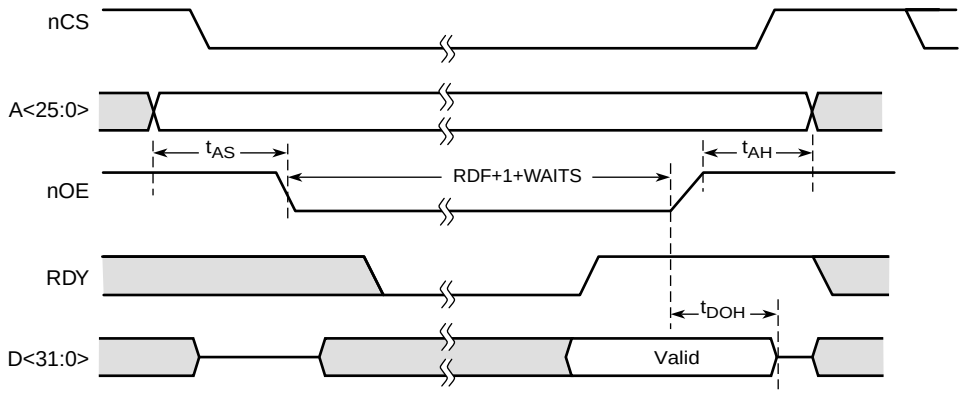
Figure 3-1. Register Write, Single Cycle with Variable Latency



NOTE: RDF measured in SA-1110 Memory Clock periods.
RDF must be set to >90 nS

A6620-01

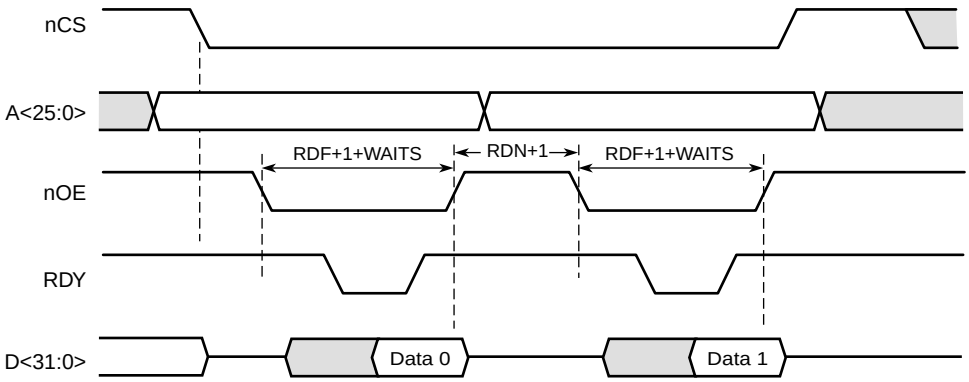
Figure 3-2. Register Read, Single Cycle with Variable Latency



NOTE: RDF measured in SA-1110 Memory Clock periods.
RDF must be set to >90 nS

A6621-01

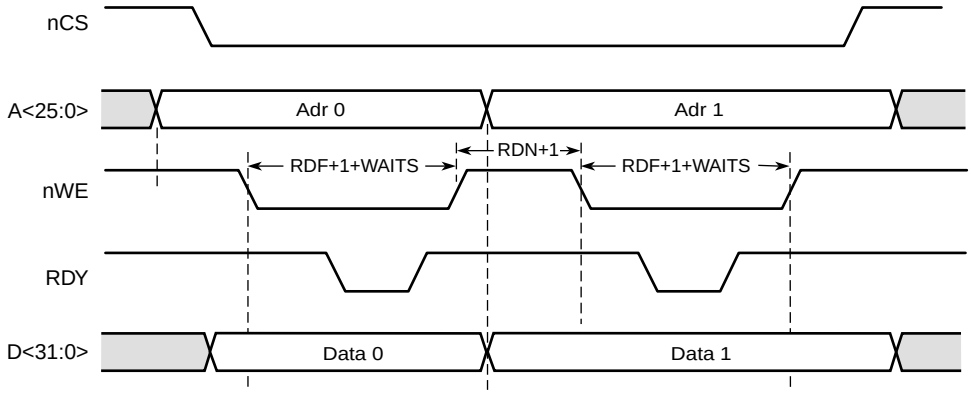
Figure 3-3. Register Read, Burst of 2 with Variable Latency



NOTE: RDN must be ≥ 2

A6622-01

Figure 3-4. Register Write, Burst of 2 with Variable Latency



NOTE: RDN must be ≥ 2

A6623-01

Table 3-6 lists the register read and write timing parameters with variable latency.

Table 3-6. Register Read and Write Timing Parameters - Variable Latency

Symbol	Parameter	Minimum	Maximum	Unit	Note
^{t}AS	Address setup to nWE or nOE start	5T-2	5T+3	ns	1
^{t}CES	nCS setup to nWE or nOE start	4T-2	4T+3	ns	1
^{t}AH	Address hold after nWE or nOE end	T-2	—	ns	—
^{t}CSH	nCS hold after nWE or nOE end	2T-2	4T+3	ns	1
^{t}nWE	Duration of nWE or nOE assertion	2T(RDF +1+Waits)		ns	2
^{t}nCS	Duration of nWE or nOE deassertion	2T(RDN+1)		ns	2

Note: 1. T is period in ns of the SA-1110 core (CPU) clock.
2. RDN and RDF are programmable in the SA-1110.

Figure 3-5 shows a register read with fixed latency mode.

Figure 3-5. Register Reads with Fixed Latency Mode of Register Access

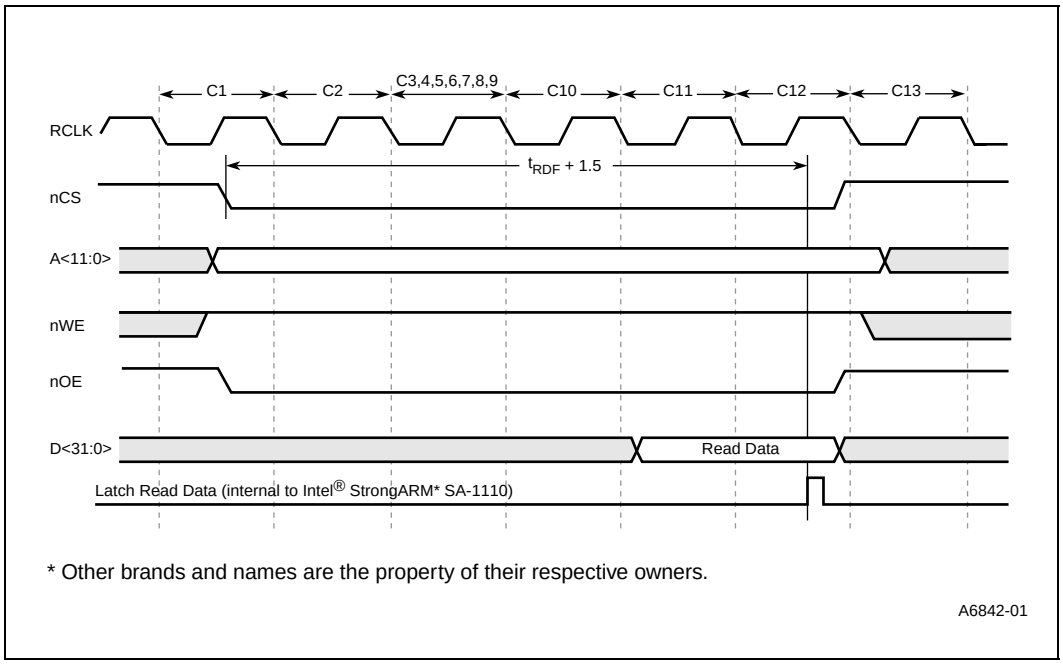


Figure 3-6 shows a register write with fixed latency mode. Note that waveforms are not synchronous to RCLK. RCLK is shown to indicate relative duration of the signals.

Figure 3-6. Register Writes with Fixed Latency Mode of Register Access

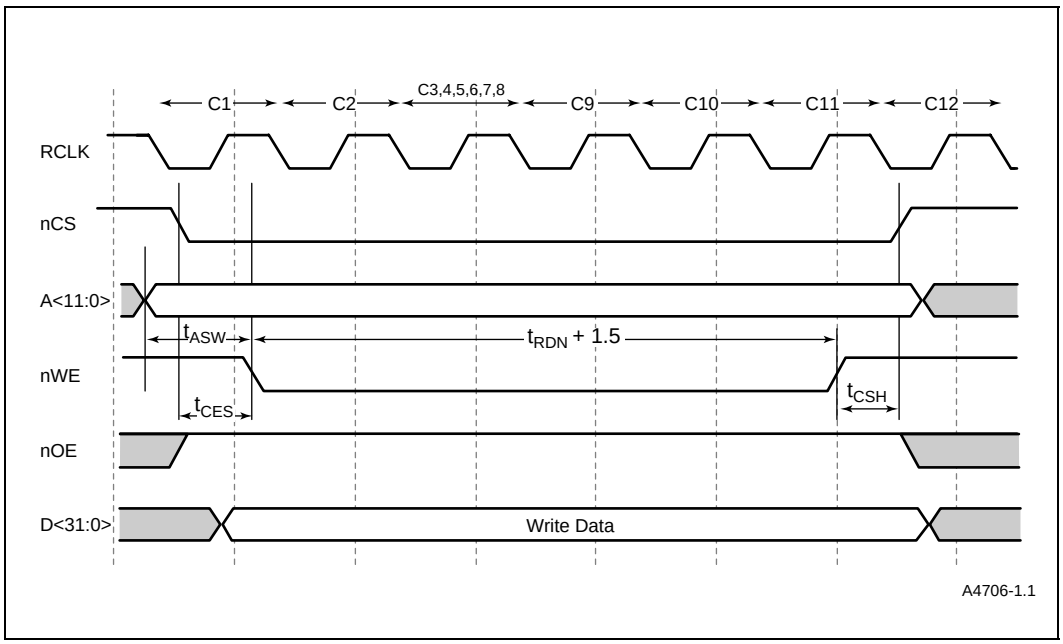


Table 3-7 lists the register read and write timing parameters with fixed latency.

Table 3-7. Register Read and Write Timing Parameters - RDY disabled

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-1110 core (CPU) clock.
2. RDN and RDF are programmable in the SA-1110.

3.2.2 PCMCIA and CF Access

The SA-1111 has internal buffers for a complete set of address, data, and control signals for two slots – one PCMCIA, and one Compact Flash (CF). Writes to either slot send data through the SBI to the data output pins using RegDIn data path. For reads, two 16-bit buses return data from the slots to the SBI, from which it is driven on the system bus pins to the SA-1110. PCMCIA reads use Slot0RdData, and CF reads use Slot1RdData bus. These are multiplexed in the SBI, using signal PsktSelIn to indicate which bus is being read. One or both bytes of the returning 16-bit word may be enabled; the disabled byte's data pins are forced to zero. Signals nDOutEnL (for bits <7:0>) and nDOutEnH (for bits <15:0>) specify which byte to enable to the SA-1110. The upper 16 bits of the 32-bit datapath are forced to zero.

3.2.3 DMA Access to System Memory

The USB block requires DMA access to system memory. The Serial Audio Port can also (programmably) operate in DMA mode, in which they may transfer data directly to and from the system DRAM. A stream of incoming data from a USB or audio peripheral can fill buffers local to the block; when enough data is stored it must be transferred to system memory. Similarly, a request for data in system memory may come from a USB peripheral, and the USB block fetches the requested data from DRAM.

3.2.3.1 Acquiring Internal and External System Buses

The USB or Serial Audio block first requests the internal bus by asserting its request (DMA_Req<n>) to the central arbiter. A version of this signal also goes to the SBI, telling the SBI to request and acquire the external bus. The SBI asserts MBREQ to the SA-1110 processor. When the system bus is idle, the processor responds with MBGNT and tri-states all required bus signals (address and DRAM control) so the SA-1111 can drive them. Meanwhile, the arbiter grants ownership of the internal DMA bus to the requester, and the new bus master asserts Frame and drives the starting address of the transfer onto DmaADOut. The address is latched in the SMC's address register/counter.

The Shared Memory Controller (SMC) begins driving external memory signals to prevent them from floating, then asserts nSDRAS with the starting Row Address.

If the transfer is a DMA write, the bus master drives the first data word onto the bus and asserts *IRDY* to tell the SMC there is valid data on the bus. The SMC clocks the data into a register and drives it onto the system data bus along with nSDCAS, write command, and Column Address. It asserts TRDY to the bus master to indicate it has taken the data. If the transfer is a DMA read, the SMC asserts nSDCAS, Column Address, and read command to SDRAM. It waits the appropriate number of cycles (two or three, depending on programmed CAS latency), latches returning READ data from SDRAM, and asserts TRDY to the bus master to indicate valid read data is available.

The USB block (or other DMA requester) is the initiator and uses IRDY to regulate transfers over the internal system bus; the SMC is the target and uses TRDY.

3.2.3.2 DMA Bus Arbitration

The DMA bus has a central arbiter. When the USB or Serial Audio controller wants to transfer data to or from system memory, it must become DMA bus master. The central arbiter receives the request as DMA_Req<n> from the requesting function block. The arbiter grants bus ownership by asserting DMA_Gnt<n> if the bus is not in use, or it waits until the current transfer is done. The default bus owner is the SAC controller; however all blocks must request the bus when they want it. If there are simultaneous requests, the arbiter grants ownership to the highest-priority requester controller.

Once granted, the winning controller drives the address onto the internal address bus to the SBI, which uses it for DRAM addressing. There is no re-mapping of physical address in the SBI; the address sent to DRAM is identical to the address placed on the DMA bus.

3.2.3.3 Shared Memory Controller (SMC)

Once it has acquired the bus, the requesting subsystem begins data read or write cycles across the DMA bus. These are translated into DRAM cycles (SDRAM) by the SMC, driving the appropriate DRAM control and address signals to the system memory DRAMs. The SMC inserts any required wait states (using TRDY) on the DMA bus for READ transfers from system memory.

The SMC opens the addressed bank and row of DRAM memory. It keeps the bank/row “open” (RAS asserted) until the completion of the transfer. The transfer takes the form of a burst of data reads or writes, in which a burst may consist of one to eight words transferred. Arbitrary byte writes are supported. At the DRAMs, byte writes are controlled by the DQM bits. Reads are always 32-bit (all bytes enabled); the requesting controller ignores unwanted bytes.

For SDRAM accesses, only a subset of the external address bus is driven with the valid memory address. This subset is A<24:10>. All other addresses are driven by low-drive I/O pads so that these signals do not float.

SDRAM setup is the responsibility of the SA-1110. After power-on, the memory controller sets up the SDRAM’s CAS latency, burst length, and burst type. SDRAMs are programmed by the SA-1110 processor to operate with “Burst Length = 1.” This does not mean they do not burst; it just means the SMC must supply a command and address every cycle in order to maintain consecutive full-speed (burst) data transfers.

If a burst crosses a row boundary, the SMC closes the row, pre charges (deasserts RAS), and restarts the transfer at the next row. This is all done transparently from the USB/Audio Controller point of view; it sees a pause during the data transfer (caused by *TRDY* deassertion) but it does not need to be aware of DRAM row boundaries.

DRAM refresh is the responsibility of the SA-1110 memory controller, not the SMC. Because SMC bus possession is short, and the SBI guarantees return of the bus to the processor after every possession, data integrity is not compromised. The SA-1110 memory controller has a continuously-running refresh timer; when it times out, it sets a flag that forces a refresh once DRAM possession is returned. While the flag is set, the timer keeps running, so the average refresh rate is unchanged regardless of SA-1111 ownership frequency or durations.

Note that CAS latency (2 or 3) must be programmed the same in the SMC as it is for the SA-1110, even though the SDRAM clock is a different frequency and the SDRAM could theoretically work with a different value. Since the SA-1110 sets up the SDRAM CAS latency (CL) value, the same value must be used by the SMC. CL specifies number of clock delays for READ data after the read command is clocked into the SDRAM.

Although the SA-1110 supports 16-bit and 32-bit bus widths to SDRAM, the SA-1111 only works with 32-bit DRAM width.

3.2.3.4 DRAM Address Generation

The SMC can control a variety of DRAM and SDRAM configurations. The SA-1110 processor and the SA-1111 must both be set up correctly (and identically), so that data goes into locations accessible to both parts, and so that the same physical address on internal buses refers to the same location in DRAM. To correctly address DRAM, the memory's number of row address bits must be loaded into an SMC register. Table 3-8 lists the DRAM address configuration and register settings.

Table 3-8. DRAM Address Configuration and Register Settings

DRAM Size	DRAM Organization	Address Row and Column Width	DRAC Setting
16 Mbit	1M x 16	10 x 10	001
		12 x 8	011
	2M x 8	11 x 10	010
		12 x 9	011
	4M x 4	11 x 11	010
		12 x 10	011
64 Mbit	4M x 16	11 x 11	010
		12 x 10	011
		13 x 9	100
		14 x 8	101
	8M x 8	12 x 11	011
		13 x 10	100
		14 x 9	101
	16M x 4	12 x 12	011
		13 x 11	100
		14 x 10	101
256 Mbit	16M x 16	15 x 9	110
	32M x 8	15 x 10	110

3.2.3.5 DRAM Control Signal Generation

On system initialization, the system must load the SMC register indicating what type of DRAM is being used. Bit 0 of the SMC Control Register (DTIM) must always be programmed with a one.

For SDRAM control, the SMC generates the following set of signals (see Figure 3-7 and Figure 3-8):

- SDCLK – 48 MHz clock
- nSDRAS – Row Address Strobe
- nSDCAS – Column Address Strobe
- nOE – Output enable
- nWE – Write enable
- nSDCS – chip select for SDRAMs
- DQM<3:0> byte enables

Signal SDCKE (Clock Enable) is always driven from the SA-1110. It is deasserted during the clock hand over period when bus ownership transfers between masters, then reasserted to enable the new owner's clock.

The SMC also drives signal nOE during DMA transfers. The behavior of nOE is programmable for use in systems with different bus architectures. The nOE behavior may be enabled or disabled. For example, a design may limit system bus loading by placing buffers and transceivers between the SA-1110 (with its system memory) and other peripherals, including the SA-1111. For DMA cycles, the SA-1111 must move data through the transceivers in either direction, depending on whether it is a read or write transfer. Signal nOE may be used to control transceiver direction if it is enabled. When enabled, it is asserted low on reads to SDRAM, and remains high for writes.

Other systems may not require nOE assertion for DMA transfers. If it is disabled, nOE is still driven by the SA-1111 but remains high (deasserted) for both reads and writes.

Figure 3-7. SDRAM Read Cycle, Burst of 2 - CAS Latency = 3

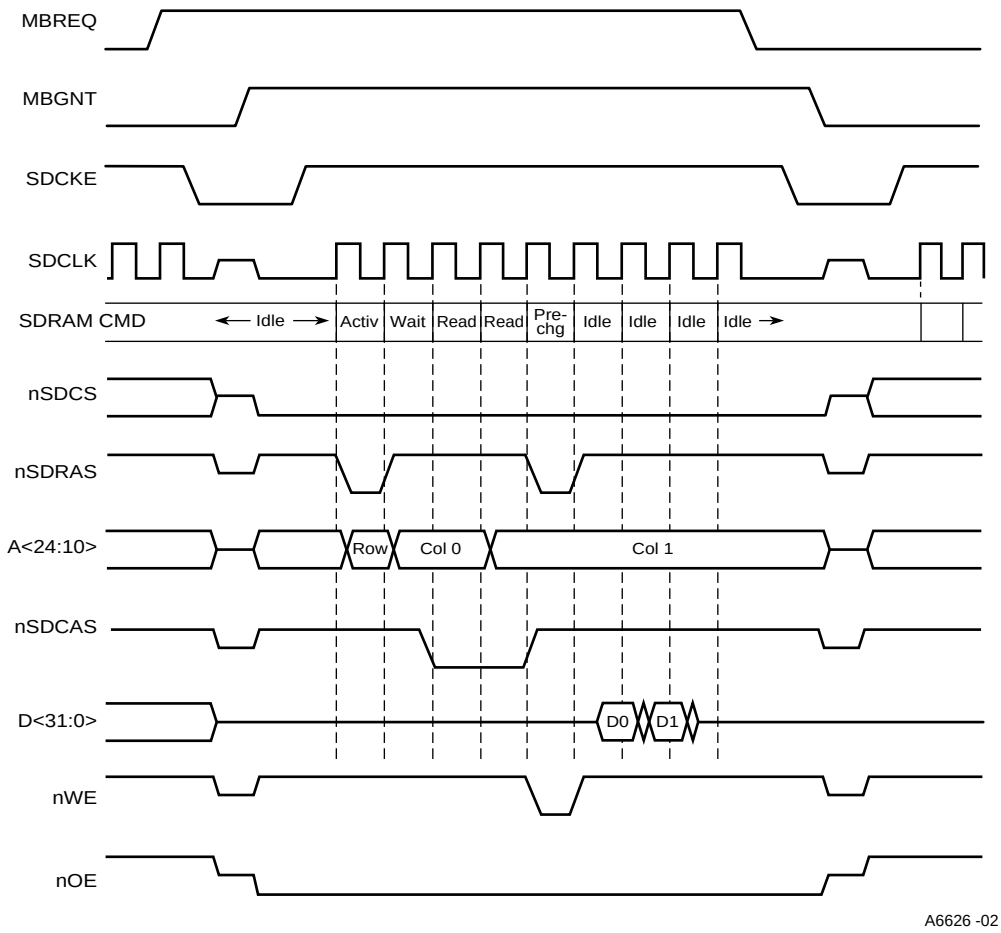
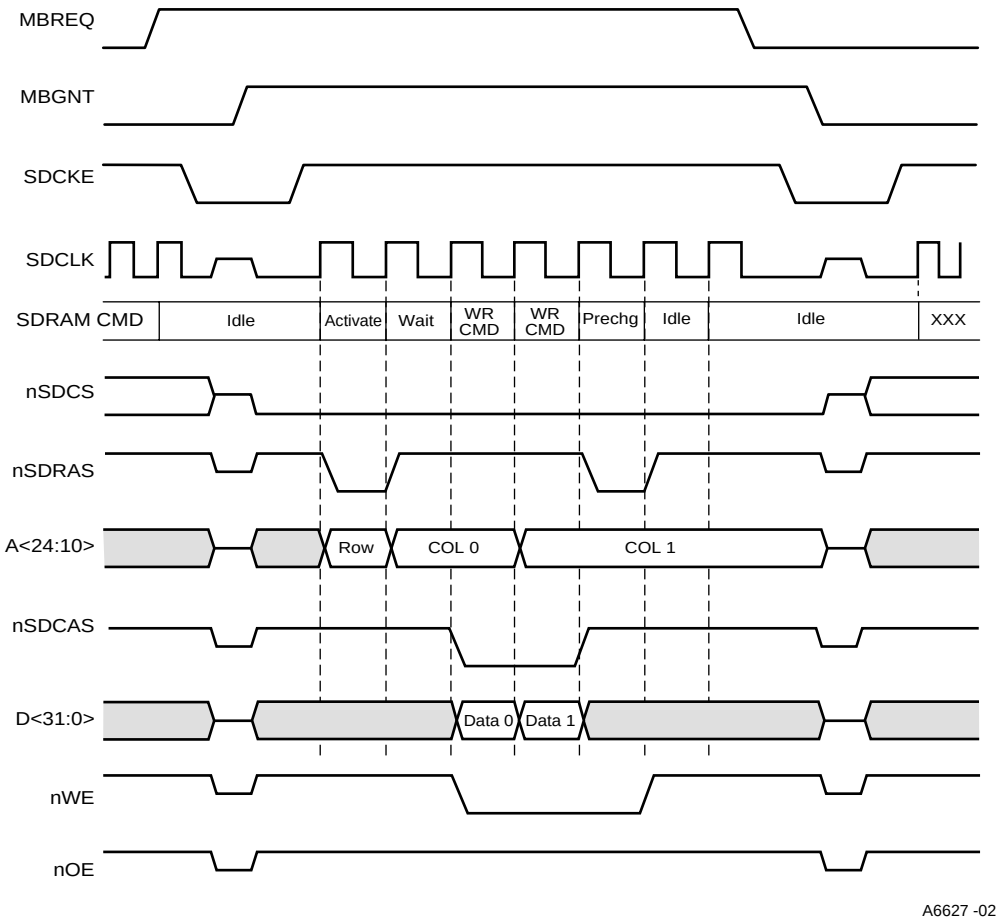


Figure 3-8. SDRAM Write Cycle, Burst of 2



3.3 SBI Registers

This section describes the registers in the SBI.

3.3.1 Control Register (SKCR)

This register provides global control over PLL operation, bus clock, sleep mode, test mode, RDY response, and the mapping of SA-1111 into high-order address ranges.

Table 3-9. SKCR Bit Descriptions (Sheet 1 of 2)

0x00000000 SKCR SA-1111 Companion Chip

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																			nOE_EN	UsbIOTestEN	PllTestEn	OPPC	SeLAC	RdyEn	ClockTestEn	ScanTestEn	VCO_OFF	Doze	Sleep	RCLKEn	PLL_Bypass
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0

Bits	Name	Type	Description
0	PLL_Bypass	RW	Specifies on-chip PLL or external source for clocks 1 = Enable 0 = Bypass
1	RCLKEn	RW	Enables Internal System Bus Clocks (RCLK and DCLK). 1 = Enable
2	Sleep	WO	Force entry into Sleep mode **1 = Enter sleep mode**[a] 0 = No effect.
3	Doze	WO	Force entry into Doze mode **1 = Enter doze mode**[a] 0 = No effect.
4	VCO_OFF	RW	VCO on/off - enables or disables system PLL for clock generation 1= Off 0 = On
5	ScanTestEn	RW	Enables scan test. Takes effect only in test mode. 1 = Enable
6	ClockTestEn	RW	Enables clock test. Takes effect only in test mode. 1 = Enable
7	RdyEn	RW	Enable RDY response (for SA-1110) vs. fixed-length register accesses 1 = Enable
8	SeLAC	RW	Audio Feature Select 1 = AC Link 0 = 1^2S
9	OPPC	RW	Out only pad control. Takes effect only in test mode. 1= Tri-states all the output pads.
10	PllTestEn	RW	Enables Pll test. Takes effect only in test mode.

Table 3-9. SKCR Bit Descriptions (Sheet 2 of 2)

0x00000000	SKCR	SA-1111 Companion Chip

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																			nOE_EN	UsbIOTestEN	PllTestEn	OPPC	SeLAC	RdyEn	ClockTestEn	ScanTestEn	VCO_OFF	Doze	Sleep	RCLKEn	PLL_Bypass
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0

Bits	Name	Type	Description
11	UsbIOTestEN	RW	Enables USB IO cell test. Takes effect only in test mode.
12	nOE_EN	RW	Enable nOE assertion on DMA read cycles from SDRAM: 0 = Enable 1 = Disable
31:13	—	—	Reserved.

a. Entering this mode has wide ranging effects. For more information about sleep mode, see Section 2.4.3

3.3.2 Shared Memory Controller Register (SMCR)

This register provides control of the SMC and DRAM configuration information.

Table 3-10. SMCR Bit Descriptions (Sheet 1 of 2)

0x00000004	SMCR	SA-1111 Companion Chip

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																										CLAT	DRAC			MBGE	DTIM
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	0	1

Bits	Name	Type	Description
0	DTIM	RW	Specifies type of DRAM: 0 = Undefined 1 = SDRAM
1	MBGE	RW	Shared memory grant enable. 0=MBGNT is gated and disabled. 1=MBGNT is enabled

Table 3-10. SMCR Bit Descriptions (Sheet 2 of 2)

0x00000004 SMCR SA-1111 Companion Chip

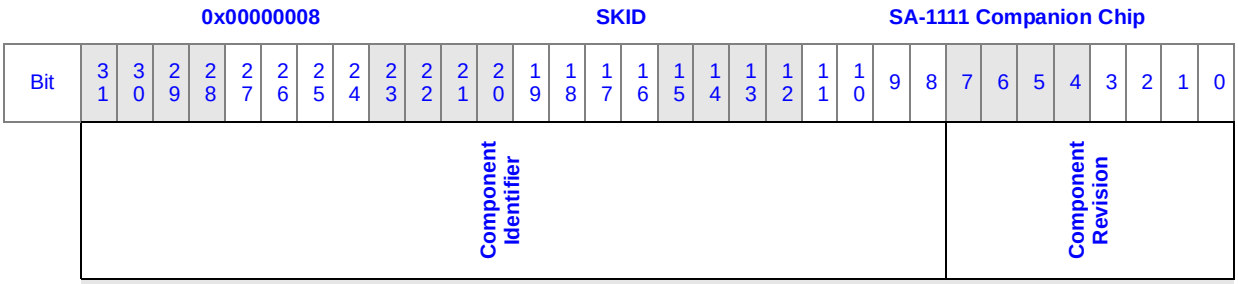
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																										CLAT	DRAC			MBGE	DTIM
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	0	1

Bits	Name	Type	Description
4:2	DRAC	RW	Number of DRAM row address bits: 000 = 9 001 = 10 010 = 11 011 = 12 100 = 13 101 = 14 110 = 15
5	CLAT	RW	For SDRAMs, specifies CAS latency for READs: 1 = CAS latency = 3 0 = CAS latency = 2
31:6	—	—	Reserved.

3.3.3 ID Register (SKID)

This is a read-only register that identifies the component as an SA-1111 companion chip and lists the revision of the component.

Table 3-11. SKID Bit Descriptions



Bits	Name	Type	Description
7:0	Component Revision	RO	Specifies the revision of the component in two hex value; bits 7:4 are the silicon revision level, bits 3:0 are the metal revision level.
31:8	Component Identifier	RO	Specifies the identity of the SA-1111 as: 0x690CC2XX (where XX represents bits 7:0)

3.3.4 Memory Map

Table 3-12 lists the read and write locations of the addresses in the SBI memory map.

Table 3-12. SBI Memory Map

Address	Read Location	Write Location
0x00000000	SKCR register	SKCR register
0x00000004	SMCR register	SMCR register
0x00000008	SKID register	(read only)

SA-1111 Internal Bus Architecture 4

Two buses carry data traffic between the System Bus Interface and the on-chip functional blocks:

- The Register Access Bus (RAB) conveys all register reads and writes, and processor-initiated transfers of serial audio or SSP data;
- The DMA Bus is a multi-master bus, used by the USB Host Controller, or the Serial Audio Controller for transferring blocks of data between system DRAM and their associated peripheral devices.

4.1 Register Access Bus

The Register Access Bus (RAB) is a synchronous single-master bus. Transfers over the RAB are initiated by the Intel® StrongARM* SA-1110 Microprocessor (SA-1110). The System Bus Interface (SBI) is always the master. Other on-chip blocks are targets.

4.1.1 RAB Central Address Decoder

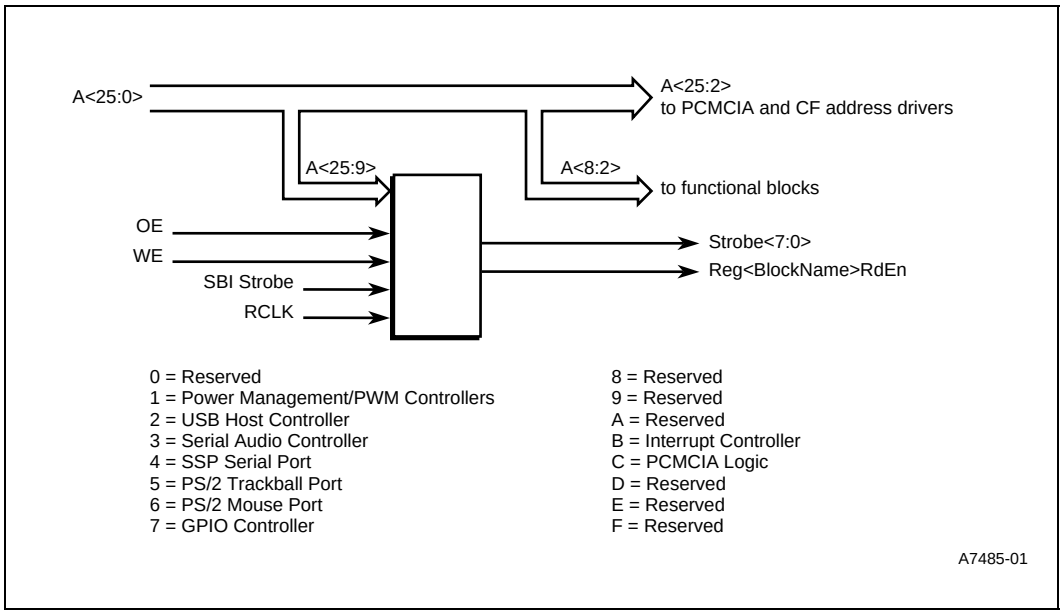
SA-1111 uses a centralized address decoder for the Register Access Bus (RAB). The decoder functionally is part of the System Bus Interface (SBI) block. The decoder makes use of high-order address bits (A<25:9>), driven by the processor to the Intel® StrongARM* SA-1111 Microprocessor Companion Chip (SA-1111) during register accesses or block-data transfers, to identify the targeted block. The decoder performs the following functions:

- Generates the target-specific write strobe (Reg<BlockName>Stb) and read-enable (Reg<BlockName>RdEn) signals to each bus target, indicating that a write or read transfer to that slave is underway.
- Acts as a simple protection unit, which prevents attempted accesses to an illegal or protected area of the on-chip memory space.

Low-order addresses (A<8:2>) are driven from the SBI to transfer targets. These are used locally within the target to decode specific register or FIFO buffer addresses. High-order bits <25:9> are sent only to the PCMCIA and CF address output pads, but do not go to any functional blocks.

Figure 4-1 shows a block diagram of the decoder.

Figure 4-1. Decoder Block Diagram



Each functional block is allocated a memory space of 128 words (512 bytes). Address bits A<9> and above select the functional block in the decoder; bits A<8:2> are passed to the functional blocks for local register decoding.

Table 4-1 lists the functional block assignments. Note that accesses to registers within the SBI itself do not result in RAB activity.

Table 4-1. RAB Target Blocks/Address Assignment

Address A<12:9>	Functional Block Assignment
0	Reserved
1	Power Management/PWM Controllers
2	USB Host Controller
3	Serial Audio Controller
4	SSP Serial Port
5	PS/2 Trackpad Port
6	PS/2 Mouse Port
7	GPIO Controller
8	Reserved
9	Reserved
A	Reserved
B	Interrupt Controller
C	PCMCIA Logic
D	Reserved
E	Reserved
F	Reserved

4.1.2 Signal Description

Table 4-2 describes RAB signals.

Table 4-2. Register Access Bus Signal Description

Name	Description
RCLK	RAB (bus) clock; normally 24 MHz
nBRES	Active LOW signal to reset all blocks in SA-1111
RegAddr<8:2>	Low-order target address bus that is driven by the SBI
RegDin<31:0>	Data bus for WRITE data to target register or buffer
RegDOut<31:0>	Data bus for READ data, from target register or buffer
RegBE<3:0>	Byte enable, to control which bytes are written at target register or buffer
Reg<BlockName>Stb	Write strobe - indicates valid WRITE data on D_IN bus
Reg<BlockName>RdEn	Read enable - indicates target should put READ data on D_OUT bus
Reg<BlockName>Wait	Wait (not READY) signal from target - to extend duration of transfer

4.1.3 RAB Cycles

Register Access Bus transfers are initiated by the system processor. The external bus signals are interpreted by the SBI and converted to RAB cycles. Register reads and writes are single-cycle transfers, while buffer accesses may take the form of a burst with up to 8 words transferred.

4.1.4 RAB Protocol

For an RAB Write, incoming data is placed on RegDln by the SBI, and Reg<BlockName>Stb asserts to the targeted destination block for one cycle. The target register or buffer clocks in the data. If it can not accept data for any reason, it asserts its Reg<BlockName>Wait signal before Reg<BlockName>Stb assertion; the SBI holds the data on the bus and delay Reg<BlockName>Stb until Wait deasserts.

For an RAB Read, low-order address bits are driven by the SBI onto RegAddr<8:2> with local address information, and Reg<BlockName>RdEn is asserted to the targeted functional block. The block selects the specific register or buffer which has been addressed, and enables it onto the 32-bit RegDOut bus. All 32 bits are driven even if the targeted register/buffer has a smaller data width. The functional block continues driving data until Reg<BlockName>RdEn is deasserted by the SBI. If the block requires more than a cycle to access and drive valid data on RegDOut, it asserts Reg<BlockName>Wait and holds it until valid data is finally on the bus. The SBI uses Wait to control the external RDY signal to the system processor.

The Address Decoder always inserts at least one “dead cycle” between assertions of Reg<BlockName>RdEn to different targets, to prevent overlap of tristate drivers on RegDOut lines.

4.2 DMA Bus

Table lists the DMA bus interface signals.

Table 4-3. DMA Bus Interface Signals

Name	Source	Description
DCLK	CLK module	32- or 48-MHz DMA bus clock.
SacReq	DMA initiator	Audio Controller Request DMA transfer, to bus arbiter
SacGnt	Arbiter	Bus Grant for Audio Controller, from arbiter
UsbReq	DMA initiator	Usb Request DMA transfer, to bus arbiter
UsbGnt	Arbiter	Bus Grant for USB, from arbiter
DmaWr	DMA initiator	DMA command: 0 = Read, 1 = Write
DmaDin<31:0>	SBI	Read data, from DRAMs to SAC
DmaADOut<31:0>	DMA initiator	DMA address or write data, from SAC to DRAMs
DmaLenBE<3:0>	DMA initiator	DMA burst length and Byte Enable, for data transfers
FRAME	DMA initiator	Frame - DMA Bus protocol signal
IRDY	DMA initiator	Initiator ready
TRDY	SBI	Target (SBI) ready

4.2.1 DMA Bus Arbiter

The DMA Bus is a synchronous multi master bus. The bus may be acquired and controlled by the USB Host Controller or the Audio Port. The functional unit controlling the bus is responsible for driving address and control signals, and units which are not masters at that time must tristate (turn off) their drivers connected to the same signals. A bus arbiter is needed to ensure that only one bus master controls the bus at a given time.

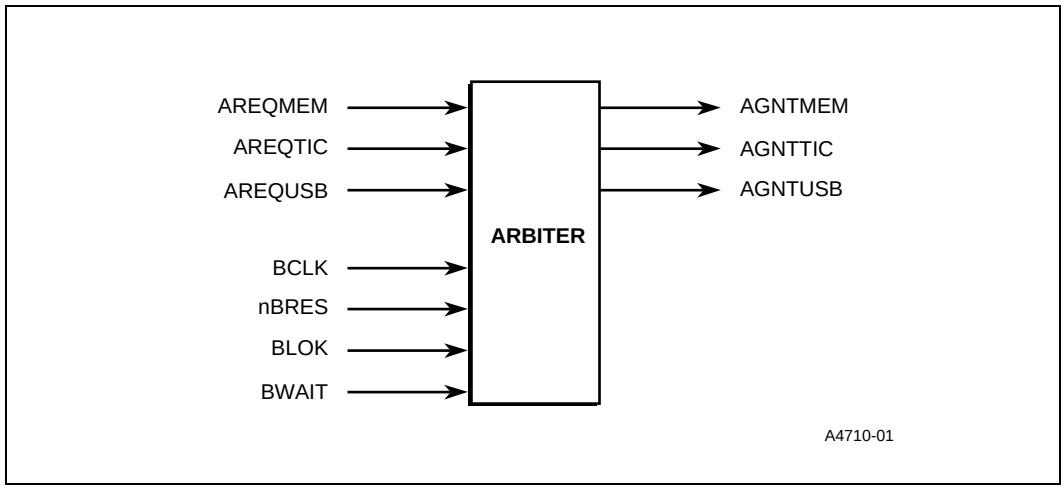
Each bus master can request the bus at any time. If a transfer is underway, the arbiter delays issuing a responding bus grant until the current transfer is complete. In the event of multiple simultaneous requests, the arbiter decides which has the highest priority and issues a grant signal accordingly.

Each potential master has its own request output, and grant input, connecting to the arbiter.

4.2.1.1 DMA Arbiter Block Diagram

Figure 4-2 shows a block diagram of the arbiter.

Figure 4-2. Arbiter Block Diagram



4.2.1.2 Arbiter Signals

Table 4-4 describes the signals for the arbiter.

Table 4-4. Arbiter Signals

Name	Type	Description
UsbReq	I	Request from the USB Host Controller indicating it requires the bus. This signal must be set up to the rising edge of DCLK.
SacReq	I	Request from the Serial Audio Controller indicating it requires the bus. This signal must be set up to the rising edge of DCLK.
UsbGnt	O	Grant signal to the USB Controller. When HIGH, this signal indicates that the USB block is currently the highest priority master requesting the bus. This signal changes during the LOW phase of DCLK and remains valid through the HIGH phase.

Table 4-4. Arbiter Signals

Name	Type	Description
SacGnt	O	Grant signal to the Serial Audio Controller. When HIGH, this signal indicates that the Serial Audio block is currently the highest priority master requesting the bus. This signal changes during the LOW phase of DCLK and remains valid through the HIGH phase.
DCLK	I	DMA Bus Clock. This clock times all bus transfers. The clock has two distinct phases: phase 0 in which DCLK is HIGH, and phase 1 in which DCLK is LOW.
nBRES	I	This signal is asserted LOW to reset the arbiter and other bus control functions.
Frame	I	Asserted by the DMA masters to indicate start and end of a transfer.
Trdy	I	Target Ready–Driven by the transfer target (System Bus Interface - SBI) to indicate readiness to send READ data, or to take WRITE data.
Irdy	I	Initiator Ready–Asserted by the DMA Master to indicate readiness to send WRITE data, or to receive READ data.
AnyReq	O	Any Request–This is ORed signal from UsbReq and SacReq.

4.2.1.3 Arbiter Functional Description

There are two potential DMA bus masters in the SA-1111, so the arbiter must decide which of the two can be the next master in the event of simultaneous requests. To gain access to the DMA Bus, a potential bus master must assert its request line ((x)Req). If the bus is not busy and the request is higher priority than any other asserted bus request, then the request is granted to that requester. If the bus is busy, the arbiter waits until the transfer underway is completed, then issues (x)Gnt to the highest-priority requester. A transfer, once underway, is never interrupted by another requester. Requesters may remove their (x)Req signal as soon as it gets the grant ((x)Gnt).

When a (x)Gnt signal is issued to a new bus master, there is always at least one "dead" cycle after deassertion of the previous (x)Gnt to prevent the possibility of tristate overlap on the bus.

A requester becomes the new bus master when its (x)Gnt is returned. It then asserts Frame and drives address, burst length, and command onto the DMA Address/Command Bus lines. The ensuing transfer then uses PCI-like protocols, with the new bus master as Initiator (controlling IRdy) and the System Bus Interface in the role of Target, controlling TRdy.

Table 4-5 lists the priorities for the arbiter in the event of simultaneous requests. The USB Host Controller has the highest priority and Serial Audio Controller is the default bus master.

Table 4-5. Bus Master Priorities

Priority	Bus Master
Highest	USB Host Controller
Lowest	Serial Audio Controller

4.2.1.4 Arbiter Reset

During reset, when nBRES is LOW, the arbiter returns bus ownership to the default owner, the serial Audio Controller. Audio bus driver logic must drive DmaADOut, DmaWr, DmaLenBE, Frame and IRdy signals at all times when its grant (SacGnt) is asserted, even when a transfer is not actively underway.

4.2.2 DMA Bus Cycles

Once a requester has been granted the bus, it drives the target DMA address onto the address/data out bus DmaAOut and asserts Frame. The Shared Memory Controller (SMC) begins a DRAM access to system DRAMs (the external DRAM bus has already been acquired by the SMC). Multiple words may be transferred in either direction, up to a burst length of 16. The bus master and SBI (which is the target) handshake each word using IRdy and TRdy signals. At the end of the transfer, the bus master deasserts Frame. When (x)Gnt is removed, it stops driving DmaADOut, DmaWr, DmaLenBE, and IRdy.

4.2.3 DMA Bus Protocols

If the transfer is a write to DRAM, the bus master puts data on DmaADOut and asserts IRdy to signal the availability of valid data. The SMC clocks the data into SDRAM and returns TRdy to indicate that it is ready for new data. Either party in the transfer - the SMC or the bus master - can delay a transfer by withholding its (x)Rdy signal.

For reads to DRAM, the bus master asserts IRdy when the transfer begins. The SMC launches a DRAM read cycle and waits the required latency time to get data back from DRAM. When valid data returns, the SMC drives it down the DmaDIn bus to the waiting bus master and asserts TRdy. The master latches the data and continues to signal IRdy if it wants more.

If the sequence of DRAM accesses crosses a row boundary, the SMC must pause the burst while it closes the current row, initiates DRAM precharge, and begins accessing the next DRAM row. It holds TRdy deasserted while this sequence is underway.

System Controller 5

The System Controller provides the clocking, power/reset functions, and mode control of the Intel® StrongARM* SA-1111 Microprocessor Companion Chip (SA-1111).

5.1 Functional Description

The System Controller, in conjunction with the Control Register, sets up the functionality for the SA-1111. In addition, the System controller provides two pulse-width modulation (PWM) outputs which, when combined with simple external components, function as 8-bit A-to-D converters generating programmable output voltages.

5.1.1 System Power Control

Power Control is enabled by clock gating to the various functional blocks. Blocks with individually controllable clocks are:

- USB Host Controller (48 MHz UCLK)
- SSP Serial Port (3.68 MHz SCLK)
- Serial Audio Controller (12.288 MHz AC-link clock, or programmable SYS_CLK and L3)
- PS/2 Ports (1-8 MHz PCLK clock)
- Shared Memory Controller and DMA Bus (DCLK), (48 MHz)
- RAB (24 MHz RCLK)
- PWM generators (3.6864 MHz Clk_In)

5.1.2 PWM Outputs

Two 8-bit resolution PWM outputs are provided. The PWM counter is clocked at CLK frequency (the input to SA-1111 PLL), which gives a carrier frequency of 14.4 KHz with the standard input 3.6864 MHz. A value of 0 written to the data register gives no output (output remains at VSS), and a value of 255 gives a pulse duty cycle of 255 clock periods high, 1 clock period low.

If either or both PWM functions are not needed, the pins may be used for GPIOs. PWM0 pin is used for GPIO_B<0>, while PWM1 pin is used for GPIO_B<1>. Refer to Chapter 10 for information on enabling and programming GPIO functions.

The clock to the PWM generators (Clk_In) may be gated off to reduce power. In the current system, the SA-1110 switches the 3.6864 MHz reference clock to 32 KHz clock while going into the sleep mode. Programming SKPWM register to zero and disables the PWM outputs prior to switching the clock. Asserting BAT_FLT also disables the PWM outputs.

5.2 Programmer's Model

This section describes the registers in the System Controller.

5.2.1 Power Control Register (SKPCR)

This register controls the clock enables for the functional blocks in the SA-1111. Note that SCLK or SYS_CLK may be brought in from an external source, but is still gated on and off by its control bit. All bits are cleared by reset, so all of the peripheral clocks are disabled at reset and are enabled by software as and when they are needed.

Table 5-1. SKPCR Bit Descriptions

0x0000200																															
SKPCR																															
SA-1111 Companion Chip																															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																							PWMCLKEn	DCLKEn	PTCLKEn	PMCLKEn	SCLKEn	L3CLKEn	I²SCLKEn	ACCLKEn	UCLKEn
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Type	Description
0	UCLKEn	RW	USB Host Controller Clock 1 = Enable
1	ACCLKEn	RW	Audio Controller AC Link Clock 1 = Enable
2	I^2SCLKEn	RW	Audio Controller I^2S Clock Enable 1 = Enable
3	L3CLKEn	RW	Audio Controller L3 Clock Enable 1 = Enable
4	SCLKEn	RW	SSP (Serial Port) Controller Clock 1 = Enable
5	PMCLKEn	RW	PS/2 Mouse Port Clock 1 = Enable
6	PTCLKEn	RW	PS/2 Track pad Clock 1 = Enable
7	DCLKEn	RW	Shared Memory Controller and DMA Bus Clock - 32 or 48 MHz 1 = Enable
8	PWMCLKEn	RW	PWM Clock enable 1 = Enable
31:9	—	—	Reserved.

5.2.2 Clock Divider Register (SKCDR)

This register controls the setting of the SA-1111's PLL.

Table 5-2. SKCDR Bit Descriptions

0x0000204																		SKCDR												SA-1111 Companion Chip											
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																	OPSel	OPDiv		IPDiv					FBDiv						
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1	0	0

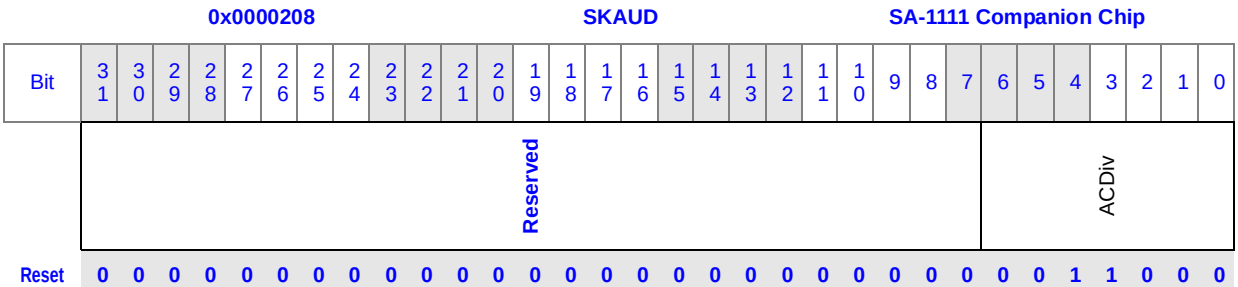
Bits	Name	Type	Description
6:0	FBDiv	RW	FeedBack Divider. Set to 0x4c for VCO frequency of 144 MHz.
11:7	IPDiv	RW	Input Divider. Set to 0 for VCO frequency of 144 MHz.
13:12	OPDiv	RW	Output divider. Set to 0 for PLL frequency of 144 MHz. 00 = PLL output is same as VCO output 01 = PLL output is VCO output divided by 4 10 = PLL output is VCO output divided by 2 11 = PLL output is VCO output divided by 8
14	OPSel	RW	Output Selector. Takes effect while in PLL bypass mode. 0 = In Bypass PLL mode, output clock is the same as the input clock 1 = In Bypass PLL mode, output clock is the inverse of the input clock
31:15	—	—	Reserved.

VCO output = VCO input * (FBDiv/IPDiv). Program FBDiv and IPDiv with (required value minus 2). For example, to program input divider with the value of 8, set IPDiv bits to 6, (8 minus 2).

5.2.3 Audio Clock Divider Register (SKAUD)

This register controls the generation of SYS_CLK and BIT_CLK for Audio module. SYS_CLK is generated by dividing PLL clock by the divider value.

Table 5-3. SKAUD Bit Descriptions



Bits	Name	Type	Description
6:0	ACDiv	RW	Audio Clock Divider. Program (required value –1). Set to 0x18 for sampling frequency of 22.05 KHz
31:7	—	—	Reserved.

Table 5-4 lists Audio Sampling Frequencies as a function of PLL Divider Value.

Table 5-4. Audio Sampling Frequencies

PLL Divider Value (decimal)	Audio Sampling Frequency
25	22.05 KHz
35	16.0 KHz
51	11.025 KHz
70	8.0 KHz

5.2.4 PS/2 Mouse Clock Divider Register (SKPMC)

This register controls the generation of PS/2 clocks. PCLK is generated by dividing PLL clock by the divider value.

Table 5-5. SKPMC Bit Descriptions

0x000020C **SKPMC** **SA-1111 Companion Chip**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																								PMCDiv							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1

Bits	Name	Type	Description
7:0	PMCDiv	RW	PS/2 Clock Divider. Program (required value – 1). Set to 0x11 for PS2 frequency of 8 MHz.
31:8	—	—	Reserved.

5.2.5 PS/2 Track Pad Clock Divider Register (SKPTC)

This register controls the generation of PS/2 clocks. PCLK is generated by dividing PLL clock by the divider value.

Table 5-6. SKPTC Bit Descriptions

0x0000210 **SKPTC** **SA-1111 Companion Chip**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																								PTCDiv							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1

Bits	Name	Type	Description
7:0	PTCDiv	RW	PS/2 Clock Divider. Program (required value – 1). Set to 0x11 for PS2 frequency of 8 MHz.
31:8	—	—	Reserved.

5.2.6 PWM0 Enable Register (SKPEN0)

This register enables the PWM0 module. When PWM0 is enabled, it takes control of the pin from GPIO module and sends PWM0 output on it.

Table 5-7. SKPEN0 Bit Descriptions

0x0000214 SKPEN0 SA-1111 Companion Chip

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																															PWMOEN
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Type	Description
0	PWMOEN	RW	PWM0 Enable 1 = Enable.
31:1	—	—	Reserved.

5.2.7 PWM0 Clock Register (SKPWM0)

This register controls the duty cycle of PWM clocks. Refer to the PWM output section for details. This register should be written with the value of zero prior to going into sleep mode.

Table 5-8. SKPWM0 Bit Descriptions

0x0000218 SKPWM0 SA-1111 Companion Chip

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																								PWMOCK							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Type	Description
7:0	PWMOCK	RW	PWM Clock Duty Cycle Control. Reset value of 0 disables the clock.
31:8	—	—	Reserved.

5.2.8 PWM1 Enable Register (SKPEN1)

This register enables the PWM1 module. When PWM1 is enabled, it takes control of the pin from GPIO module and sends PWM1 output on it.

Table 5-9. SKPEN1 Bit Descriptions

0x000021C — SKPEN1 — SA-1111 Companion Chip

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																															PWM1EN
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Type	Description
0	PWM1EN	RW	PWM1 Enable 1 = Enable.
31:1	—		Reserved.

5.2.9 PWM1 Clock Register (SKPWM1)

This register controls the duty cycle of PWM clocks. Refer to Section 5.1.2 for details. This register should be written with the value of zero prior to going into sleep mode.

Table 5-10. SKPWM1 Bit Descriptions

0x0000220 — SKPWM1 — SA-1111 Companion Chip

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserve																								PWM1CK							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Type	Description
7:0	PWM1CK	RW	PWM Clock Duty Cycle Control. Reset value of 0 disables the clock.
31:8	—	—	Reserved.

5.2.10 Memory Map

Table 5-11 lists the read and write locations in the memory map of the System Controller.

Table 5-11. System Controller - Read/Write Locations

Address	Read Location	Write Location
0x0000200	SKPCR register	SKPCR register
0x0000204	SKCDR register	SKCDR register
0x0000208	SKAUD register	SKAUD register
0x000020C	SKPMC register	SKPMC register
0x0000210	SKPTC register	SKPTC register
0x0000214	SKPEN0 register	SKPEN0 register
0x0000218	SKPWM0 register	SKPWM0 register
0x000021C	SKPEN1 register	SKPEN1 register
0x0000220	SKPWM1 register	SKPWM1 register

USB Host Interface Controller 6

The Universal Serial Bus (USB) is a bus cable that supports serial data exchange between a host computer and a variety of simultaneously-accessible peripherals. The attached peripherals share USB bandwidth through a host-scheduled token-based protocol. Peripherals may be attached, configured, used, and detached, while the host and other peripherals continue operation.

The USB Host Controller built into the Intel® StrongARM* SA-1111 Microprocessor Companion Chip (SA-1111) is Open Host Controller Interface (OHCI)-compatible, Windows95* USBD-compatible, and USB Revision 1.1-compatible. It supports both low-speed (1.5 Mbps) and high-speed (12 Mbps) USB devices.

See the *Universal Serial Bus Specification Revision 1.1* and the *Open HCI – Open Host Controller Specification for USB* for details of the interface operation.

6.1 Functional Description

The USB Host Controller follows USB protocols to set up and operate peripherals on the USB. In addition, it controls the transfer of data between USB peripherals (attached to the bus) and system memory. This section provides a functional overview of the USB Host Controller.

6.1.1 USB Reset

The USB Host Controller is not fully reset following an SA-1111 reset. This reset leaves the USB ForceHcReset and the USB ForceIfReset asserted. To initialize the USB Host Controller, follow this sequence:

1. Start the UCLK12.
2. Wait at least 10 µS.
3. Clear both the ForceIfReset and the ForceHcReset bits.

The USB Host Controller is now operational.

6.1.2 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 explicitly invoke the suspend state; otherwise devices automatically 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 re-enable 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 requiring devices to draw less than 500 µA supply current.

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

6.1.3 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 block's clock is running. The USB clock must be running to disable or enable power-enable features.

The SA-1111 USB has the following features that support power-conservation features of the Open Host Controller Interface:

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

6.1.3.1 USB Clock Stopping and Power Enable

The USB clock can be stopped at any time. However, stopping the clock is only recommended 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 more than 5 ms, and the host controller is in the *suspend* state. In the suspend state, the port (PAD) also is in low-power mode, but still is able to detect the USB port resume interrupt.

If a device is not connected to the USB port, the host controller cannot be suspended, and the port (PAD) is not in its low-power mode. To save power when a device is not connected to the USB port (PAD), the *port power enable* can be disabled before the USB clock is stopped (refer to OHCI specification).

6.1.3.2 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 interrupt handles any interrupt conditions. For the port resume interrupt to function both the USB pad's single-ended receiver and the port power supply must be powered on.

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 device.
  Indicated by an idle → K-state transition on the USB bus.

6.1.3.3 Summary of Low Power Operation

Table 6-1 lists the power status of each of the potential power drain components in the USB interface for the various modes of operation.

Table 6-1. Operational Power Status

Mode of Operation	Notes	Power Status				
		USB Core	Tx Driver	Differential Receiver	Single-Ended Receiver	Port Power Supply
Fully operational	1	Powered	Powered	Powered	Powered	Powered
Global suspend	2, 3, 6	Under user control	Low power	Low power	Powered	Powered
Port power disabled (as specified in OHCI spec).	3,4	Under user control	Low power	Low power	Low power	Low power
Sleep mode active	5,2	Under user control	Low power	Low power	Under user control	Under user control
Standby (low power)	7, 4, 2	Low power	Low power	Low power	Under user control	Under user control

Notes:

1. Fully operational is defined as: USB Host controller is in the operational state, the Sleep mode is not active, and the port power is enabled.
2. The user may stop the USB clock (see Section 5.2.1) and still be able to receive a USB port resume interrupt when a device sends a remote resume interrupt or a device is disconnected.
3. When the port power status bit in the HC core is disabled the USB port is considered to be disabled, and therefore not in use, all components are put into low-power mode. The user may stop the USB clock to further reduce power drain (see Section 5.2.1).
4. When the USB pads singled-ended receiver is in low power mode the remote resume interrupt will not function, this may also be the case if the device does not receive power from the USB port.
5. When the Sleep mode is active the single-ended pad receiver together with the port power supply power enable are controlled by the StandbyEn control bit; only if this bit is cleared will the port resume interrupt function.
6. A device must be connected before the HC can be put into the suspend state.
7. To disable the USB port the USB clock must be stopped. The single ended port receiver and USB power supply is powered on if the *Port Power Enable* was enabled before the clock was stopped.

6.1.4 Suggested Power-Management Routines

This section lists suggested power-management routines.

6.1.4.1 Initial Port Power-Down Sequence

To initiate a port power-down sequence, follow this procedure:

1. Wait for the device hardware reset sequence to complete. For more information on USB reset, see Section 6.1.2. The USB clock is enabled.
2. Send a software reset to the host controller and wait 10 µS.
3. Set the *global power enable* bit.
4. Clear the StandbyEn bit.
5. Stop the USB clock.
   The USB is now in *Standby* mode.

Note: Software waits for the UsbPortResume interrupt before restarting the USB clock.

6.1.4.2 Low-Power Mode in Suspend State Sequence

To put the USB system into low-power mode, follow this sequence:

1. Wait for the USB to go to *global suspend* state.
2. Set the *port suspend* bit. This may be automatic after 5 ms of inactivity on the USB bus.
3. Wait for the *port suspend* bit to go active.
   The USB is now in *suspend mode.*
4. Clear the StandbyEn bit.
5. Stop the USB clock.
   The USB is now in *Standby* mode.

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

6.1.4.3 Low-Power Mode after Device Disconnect Sequence

To activate low-power mode, follow this sequence:

1. Wait for *port connection status* to go inactive, or wait for the Usb Port Resume interrupt if the USB clock is stopped. Check connection status.
2. Clear the StandbyEn bit.
3. Stop the USB clock.
   The USB is now in *Standby* mode.

Note: The port is powered down. For more information on the initial port power down sequence, see Section 6.1.4.1. Software waits for the Usb Port Resume interrupt before restarting the USB clock.

If the USB clock is stopped in any mode other than that described above, the host controller and any connected device may need re-initialization.

6.1.4.4 Disable USB Port

To initiate a port disable sequence, follow this procedure:

1. Wait for the device hardware reset sequence to complete. For more information on USB Reset, see Section 6.1.1.
   The USB clock is enabled
2. Send a software reset to the host controller and wait 10 µS.
3. Clear the *global power enable* bit.
   The USB is now in *Port Power Disabled* mode.
4. Set the StandbyEn bit.
5. Stop the USB clock.
   The USB is now in *Standby* mode and the port power is disabled.

Note: The UsbPortResume interrupt is not generated when a device is connected unless the StandbyEn bit is cleared. To set the global power enable bit, it is necessary to first start the USB clock.

6.2 USB Block Interfaces

6.2.1 External Pin Interface

The USB Controller connects to four pins on the SA-1111 device. These are:

- USB_PLUS—One of two differential USB I/O signals, to USB peripherals.
- USB_MINUS—The second differential USB I/O signal, to USB peripherals.
- USB_PWRCNTL —Normally this pin is driven from the Port Power status bit in the HcRhPortStatus register, but when the sleep mode is active or the USB clock is disabled it is driven from the SleepStandbyEn bit. This signal may be used to control an external power-switching device supplying power to USB peripherals (Port Power Supply).
- USB_PWR_SENSE—This is a dedicated active high input which is normally used for sensing port over current fault conditions on the USB power supply. It can initiate an interrupt to the system processor if enabled (refer to Section 6.3.1 for more information). This signal also drives the Over Current status signal to the USB HC core (refer to the OHCI register specification).

6.2.2 Internal RAB Bus Interface

The Register Access Bus (RAB) is used for register reading and writing. Table 6-2 lists the signals in the RAB bus interface.

Note: The following registers may only be accessed as 32-bit words; byte and half-word writes are illegal and may cause erroneous behavior.

Table 6-2. RAB Interface Signals

Name	Direction	Description
RCLK	I	24-MHz RAB clock.
nBRES	I	RAB reset, asserted low.
RegUsbStb	I	Write Strobe to USB Controller
RegUsbRdEn	I	Output Enable, for Reads to USB Controller
RegAddr<8:2>	I	7-bit local address from RAB.
RegDin<31:0>	I	32-bit write data, from system bus to USB Controller
RegDout<31:0>	I	32-bit read data, from USB Controller to system bus
RegBe<3:0>	I	Byte Enable, for data writes using RAB transfers
RegUsbWait	O	Wait signal to SBI

6.2.3 Internal DMA Bus Interface

With DMA transfers enabled, the USB Controller requests the system bus when it needs to transfer data to or from system DRAM. Once it receives the grant, it drives addresses and data onto the DMA bus to the SBI. The SBI's memory controller translates those into DRAM cycles and completes the transfer between the internal DMA bus and system DRAM. Table 6-3 lists the DMA bus interface signals.

Table 6-3. DMA Bus Interface Signals

Name	Direction	Description
DClk	I	32- or 48-MHz DMA bus clock.
UsbReq	O	Request DMA transfer, to bus arbiter
UsbGnt	I	Bus Grant, from arbiter
DmaLenBe	O	Multiplexed address burst length and data write byte enable
DmaWr	O	DMA command: 0 = Read, 1 = Write
DmaDin<31:0>	I	32-bit read data, from DRAMs to USB Controller
DmaADout<31:0>	O	Multiplexed 32-bit address and write data, from USB Controller to DRAMs
Frame	O	Frame - DMA Bus protocol signal
IRdy	O	Initiator (USB Controller) ready
TRdy	I	Target (SBI) ready

6.2.4 Interrupt Generation

Six possible sources of interrupts from the USB interface generate the single UsbInt signal (which is driven to the Interrupt Controller block). These six sources are:

- HCI interrupt; as specified in the OHCI specification.

- Remote Wake-up interrupt; only operational when the USB clock is enabled.
- HCI Buffer Active interrupt; activated when the HC performs a DMA access to external memory.
- HCI transfer abort interrupt; activated when the HC aborts a DMA transfer.
- USB device over current interrupt; activated when a device sinks more current than allowed by the USB specification.
- Port Resume interrupt; See Section 6.1.3.2 for details of operation. This interrupt is only enabled when the USB clock has been stopped.

Each interrupt source has associated status and test bits, the status bit allows the user to inspect the status of the condition causing the interrupt (also allows a polling operation to be used). The test bit enables the signal to be stimulated for test.

6.3 Programmer's Model

The USB incorporates operational registers of the Open Host Controller Interface Specification for the USB. In addition, the SA-1111 has USB-related status and reset registers that are described in this section.

6.3.1 Status Register

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

Table 6-4. Status Register Bit Descriptions (Sheet 1 of 2)

0x0000518 Status SA-1111 Companion Chip

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																				UsbPwrSense	NHciMFClr	nIrqHciM	IrqHciBuffAcc	IrqHciRmtWkp	Reserved							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Type	Description
6:0	—	—	Reserved.
7	IrqHciRmtWkp	RO	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.

Table 6-4. Status Register Bit Descriptions (Sheet 2 of 2)

0x0000518 Status SA-1111 Companion Chip

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																				UsbPwrSense	NHciMFCIr	nIrqHciM	IrqHciBuffAcc	IrqHciRmtWkp	Reserved							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Type	Description
8	IrqHciBuffAcc	RO	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	RO	0 = Normal HC interrupt active. See the Open Host Controller Interface specification.
10	NHciMFCIr	RO	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.
11	UsbPwrSense	RO	1 = Port over current. This bit reflects the state of the USB Power Sense pin. This status bit is also reflected in the Over Current Condition status bit of the HCI controller when its clock is running.
31:12	—	—	Reserved.

6.3.2 Interrupt Test Register

This register is only used for test, it enables the interrupt path to be exercised independently of the interrupt source. While in test mode and with the Interrupt Test bit (UsbIntTest) is set, these bits directly drive the interrupt signals to the Interrupt controller block.

Table 6-5. Interrupt Test Register Bit Descriptions

0x0000520 Interrupt Test SA-1111 Companion Chip

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																				UsbPortResumeTest	NHciMFCirTest	NirqHciMtest	IrqHciBuffAccTest	IrqHciRmtWkpTest	Reserved							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Type	Description
6:0	—	—	Reserved.
7	IrqHciRmtWkpTest	WO	Force HCI remote wake-up interrupt 1= Force interrupt strobe. 0 = No effect.
8	IrqHciBuffAccTest	WO	Force HCI buffer active interrupt. 1= Force interrupt strobe. 0 = No effect.
9	NirqHciMtest	WO	Force normal HC interrupt. 1= Force interrupt strobe. 0 = No effect.
10	NHciMFCirTest	WO	Force HCI interface transfer abort interrupt. 1= Force interrupt strobe. 0 = No effect.
11	UsbPortResumeTest	WO	Force USB port resume interrupt. 1= Force interrupt strobe. 0 = No effect.
31:12	—	—	Reserved.

6.3.3 Reset Register

This register provides a mechanism to individually reset the uhost core (HcReset) and the DMA master interface (IfReset). An external reset sets the ForceIfReset and ForceHcReset bits. The ClkGenReset bit is only used for simulation and test.

Table 6-6. Reset Register Bit Descriptions

0x000051C — **Reset Test** — **SA-1111 Companion Chip**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																								PwrCtrlPolLow	PwrSensePolLow	SleepStandbyEn	UsbintTest	SimScaleDownClk	ClkGenReset	ForceHcReset	ForceIfReset
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

Bits	Name	Type	Description
0	ForceIfReset	RW	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	RW	When set, forces a reset to the USB Host Controller. 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	RW	When set, this bit resets the Host Controller clock generation block. It can be used by the test interface to guarantee 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.
3	SimScaleDown Clk	RW	When set to one, scales down the 1ms-interval clock in the host controller. Only used for test and simulation, must be set to zero for normal operation
4	UsbintTest	RW	When set to one and Test mode is active, this bit enables the interrupt test bits that directly force the interrupt signals the interrupt controller.
5	SleepStandby En	RW	This bits operation is only valid when the Sleep mode is active; when set to one, disables the USB single-ended receiver and the USB port power supply.
6	PwrSensePol Low	RW	This bit selects the polarity of the USB_PWR_SENSE input signal. When set active, the signal is considered active low.
7	PwrCtrlPolLow	RW	This bit selects the polarity of the PwrCtrlPolLow output signal. When set active, the signal is considered active low.
31:8	—	—	Reserved.

6.4 Register Memory Map

Table 6-7 lists the addresses of the control bits in the USB host interface controller. See Section 3.2.1 for the base address of the USB host interface controller.

Table 6-7. USB Host Interface Register Memory Map

Address	Name
0x0000400	Revision
0x0000404	Control
0x0000408	CommandStatus
0x000040C	InterruptStatus
0x0000410	InterruptEnable
0x0000414	InterruptDisable
0x0000418	HCCA
0x000041C	PeriodCurrentED
0x0000420	ControlHeadED
0x0000424	ControlCurrentED
0x0000428	BulkHeadED
0x000042C	BulkCurrentED
0x0000430	DoneHead
0x0000434	FmInterval
0x0000438	FmRemaining
0x000043C	FmNumber
0x0000440	PeriodicStart
0x0000444	LSThreshold
0x0000448	RhDescriptorA
0x000044C	RhDescriptorB
0x0000450	RhStatus
0x0000454	RhPortStatus<1>
0x0000518	Status register
0x000051C	Reset register
0x0000520	Interrupt test register
0x0000530–0x000055C	Read Data FIFO RAM test access

Serial Audio Controller 7

The Serial Audio Controller (SAC) is the functional block through which serialized digital audio passes between the system (Intel® StrongARM* SA-1110 and memory) and an external codec. Typically the codec contains one or more A-to-D converters and a similar number of D-to-A converters.

The system can record an audio source by passing incoming audio through an A-to-D converter and storing the digitized samples in system memory. Sampled audio may be compressed by the host processor before committing to memory, to reduce storage requirements.

For playback, or for the production of synthesized audio, the processor retrieves and decompresses stored audio samples (or creates audio samples algorithmically) and sends them to the Serial Audio Controller. The SAC passes them on to the external D-to-A converter via its serial audio link, where they are converted to an analog audio waveform.

The Serial Audio Controller supports several industry-standard formats for the transfer of digitized audio information between the Intel® StrongARM* SA-1111 Microprocessor Companion Chip (SA-1111) and an external codec. These formats are AC-link, I^2S, and MSB-Justified. The Serial Audio Controller can operate full-duplex (simultaneous traffic in both directions). Four pins connect the controller to external codecs: a bit-rate clock (which can use either internal or external source), a formatting or “Left/Right” control signal, and the two serial audio pins (one for each direction). An additional pin provides SYS_CLK/RESET# signal to the codec.

The Serial Audio Controller can use on-chip DMA capability to transfer audio data directly between the external codec and system DRAM, without processor intervention. Or, the processor can transfer data in either direction using bursts of SRAM-like accesses to audio data FIFOs.

For I^2S systems, additional pins are required to control the external codec. Some codecs use a control bus (L3 Bus) which requires three signals. Pins PWM<1>, MSCLK, and MSDATA are dedicated to L3 Bus function. They can be toggled under software control as GPIO outputs, or they can use on-chip logic to output L3-Bus protocols by writing bytes into the L3-bus register.

This chapter describes the operation and signal definition for the Serial Audio Controller (SAC) functional block.

Note: For information about Serial Audio Controller timing, see Section 13.3.

7.1 Signal Description

This section lists the signals which interface to the Serial Audio Controller.

7.1.1 External Interface to Serial Audio Controller

Table 7-1 lists the external signals between the SAC and an external Codec device. Alternatively, if the SAC function is not being used, the pins may function as GPIOs (refer to Chapter 10 for information on alternate mode of operation).

Table 7-1. External Interface to Codec

Name	Direction	Description
SYS_CLK	O	System Clock for I^2S systems or nBRES and RESET# (AC-link)
BIT_CLK	I/O	12.288 MHz (AC'97, AC-link) or other bit-rate clock
SYNC	O	Frame indicator (AC-link) or L/R identifier (I^2C or MSB-Justified)
SDATA_OUT	O	Serial audio data out to codec, for digital-to-analog conversion
SDATA_IN	I	Serial audio data in, from codec

SYS_CLK is an internally-supplied or externally-supplied clock for I^2S systems. Normally it is based on the PLL output, divided by an integer value between 25 and 70, to generate SYS_CLK. Table 7-6 lists the available sampling frequencies. It is then divided by 256 to generate the audio sampling frequency (typically between 8 and 22 KHz). SYS_CLK is used by AC-link for the RESET# signal.

BIT_CLK is an externally-supplied clock for AC-link. Typically (AC-link) it is supplied by the AC'97 codec, which has its own oscillator and attached 24.576 MHz crystal. The clock specifies the bit rate at which serial audio data moves between the codec and the SA-1111. An alternate mode of operation (I^2S or "MSB-Justified") may use an external clock or an internally-supplied bit clock. If the clock is generated internally, the BIT_CLK pin is an output. Table 7-6 lists the available sampling frequencies based on an internal clock source. BIT_CLK is 64 times the sampling frequency for I^2S operation or MSB-Justified operation.

SYNC signals the beginning of an audio sample frame for AC-link format. For I^2S or "MSB-Justified" formats, it indicates Left/Right sample identity.

SDATA_IN and SDATA_OUT are the serial audio data streams, one for each direction. See "Serial Audio Formats" for details on data encoding in the stream.

7.1.1.1 Optional L3 Bus Interface

For interfacing to I^2S codecs, other pins are required for transferring control information. Typically two or three GPIO signals may be used for codec control. If the codec uses L3 Bus protocol, three pins (PWM<1> = L3_MODE, MSCLK = L3_CLK, and MSDATA = L3_DATA) are dedicated to L3 Bus interfacing. The processor can load the L3 Bus register one byte at a time, and the data is transferred over the L3 bus automatically. Table 7-2 shows the assignment of L3 Bus functions to specific SA-1111 pins.

Table 7-2. L3 Bus Interface Pins

Name	Direction	Description
L3_CLK	O	Used by target codec to clock in serial address and data bits Uses MSCLK/GPIO_B<4> pin, when L3 mode enabled
L3_DATA	I/O	Bidirectional serial data path for codec control and status information Uses MSDATA/GPIO_B<5> pin, when L3 mode enabled
L3_MODE	O	Mode signal, to distinguish between address and data modes of operation Uses PWM/GPIO_B<1> pin, when L3 mode enabled

7.1.2 Internal Interfaces

7.1.2.1 Register Access Bus (RAB)

The Register Access Bus (RAB) is used for register reading and writing. Table 7-3 lists the signals in the RAB bus interface.

Table 7-3. RAB Interface Signals

Name	Direction	Description
RCLK	I	24-MHz RAB clock
nBRES	I	RAB reset, asserted low
RegAddr<8:2>	I	7-bit local address from RAB.
RegDIn<31:0>	I	32-bit write data, from system bus to SAC
RegDOut<31:0>	I	32-bit RAB read data, from SAC to system bus
RegBE<3:0>	I	Byte Enable, for data writes using RAB transfers
RegSacStb	I	Write Strobe to Serial Audio Controller
RegSacRdEn	I	Output Enable, for Reads to Serial Audio Controller

7.1.2.2 DMA Bus Interface

With DMA transfers enabled, the SAC requests the system bus when it needs to transfer data to or from system DRAM. Once it receives the grant, it drives addresses and data onto the internal bus to the SBI. The SBI's memory controller translates those into DRAM cycles and completes the transfer between the internal bus and DRAM. Table lists the DMA bus interface signals.

Table 7-4. DMA Bus Interface Signals

Name	Direction	Description
DCLK	I	32- or 48-MHz DMA bus clock
SacReq	O	Request DMA transfer, to bus arbiter
SacGnt	I	Bus Grant, from arbiter
DmaWr	O	Write command to SBI and DRAMs
DmaDIn<31:0>	I	32-bit read data, from DRAMs to SAC
DmaADout<31:0>	O	32-bit write data, from SAC to DRAMs
DmaLenBE<3:0>	O	Byte Enable, for data writes using DMA transfers
Frame	O	Frame - DMA Bus protocol signal
IRdy	O	Initiator (SAC) ready
TRdy	I	Target (SBI) ready

7.1.2.3 Other Signals

There are four input clocks and eleven interrupt output signals. Table 7-5 lists the other signals.

Table 7-5. Other Signals

Name	Direction	Description
SACMDSL	I	SAC Operation Mode Select signal from Control Register SKCR
BIT_CLK	I	Bit Rate Clock for I^2S or MSB-Justified systems
SYS_CLK	I	System Clock for I2S or MSB-Justified systems
CK3M6	I	3.6864 MHz clock for wake up sequence control.
TFS	O	Transmit FIFO Service Request
RFS	O	Receive FIFO Service Request
TUR	O	Transmit FIFO Under-run
ROR	O	Receive FIFO Overrun
CADT	O	Command Address/Data Transfer is Done
SADR	O	Status Address/Data Receive is Done
RSTO	O	Read Status Time Out
TDBDA	O	DMA Transmit Buffer Done A
TDBDB	O	DMA Transmit Buffer Done B
RDBDA	O	DMA Receive Buffer Done A
RDBDB	O	DMA Receive Buffer Done B

7.2 Serial Audio Controller Operation

Audio data is transferred between the external system bus and an external codec through the SBI, the internal bus, and a buffer in the SAC. Transfers may be initiated by the host processor (RAB Bus burst cycles) or by the SAC (DMA transfers).

7.2.1 RAB Bus Data Transfers

Outgoing data (from system to codec) is written by the host processor through the SBI to the SAC's Transmit buffer. The write takes the form of an SRAM-like burst, with one to eight words being transferred per burst. The buffer works as a FIFO, and is seen as a block of eight 32-bit locations by the processor. The SAC then takes the data from the buffer, serializes it, and sends it over the serial wire (SDATA_OUT) to the codec, where it can be converted to an analog waveform.

Incoming data from the external codec (on SDATA_IN) is converted to parallel words and stored in the Receive FIFO buffer. A programmable "fullness" threshold, when passed, triggers an interrupt to the Interrupt Controller (and thence, if enabled, to an interrupt input on the SA-1110). The interrupt service routine responds by identifying the source of the interrupt and then doing an SRAM-like burst read from the inbound FIFO buffer.

FIFO status is available to the system processor in the SAC Status Registers (SASR0 and SASR1). The amount of data in the FIFO is directly readable, as well as bits signifying FIFO full, FIFO empty, and FIFO above or below a programmable threshold. FIFO levels may be maintained either by polling routines or by interrupts triggering SAC service routine.

The host processor differentiates between the two FIFOs by whether it does a READ or a WRITE transfer. Read bursts automatically target the Receive FIFO, while write bursts take data from the Transmit FIFO. From a memory map point of view, they appear at the same block of addresses as Serial Audio Data Register (SADR).

7.2.2 DMA Audio Data Transfers

The Serial Audio Controller can transfer data directly to and from system DRAM if DMA operation is enabled. When data is to be transferred (generally in response to a Transmit or Receive FIFO condition), the SAC requests the bus using DMA_Req<1>. The SBI acquires mastership of the external DRAM bus, signals back to the arbiter, and the arbiter asserts DMA_Gnt<1> to the SAC.

The SAC drives address onto the bus, asserts DWrite if it's a write transfer to DRAM, and then drives data onto the bus, or latches incoming data from DRAM if it is a read. Data transfers can be read or write bursts up to eight words long. After a transfer burst, the SAC must give up its request so the bus can be returned to SA-1110 ownership. When the complete block is transferred, it sets the appropriate DMA_Done bit in the status register. An interrupt will be signaled to the system processor if it is enabled.

The DMA can also be set up to loop continuously on a block-data transfer from memory to the external codec by setting "Enable Interface Loop Back Function" bit. The *DMA Done* interrupt still asserts at the end of each loop, but may be disabled if desired.

DMA Address and Control registers must be set up in advance. DMA addresses, transfer size, burst size, and DMA enables are set up independently for transmit and receive data transfers. Interrupts for "DMA_Done" flags are set up in status register (SASR0 and SASR1).

The SA-1111 can address one bank of SDRAM, up to the architectural limit of 128 MBytes. 25 bits can address the resulting 32M words. Maximum transfer size is 8 Kbytes, and can be specified to byte-level resolution.

7.2.3 Serial Audio Clocks and Sampling Frequencies

For I^2S and MSB-justified formats, the Serial Audio Controller supports standard audio sample rates (typically 8.0, 11.025, 16.0, 22.05, 32.0 and 44.1 KHz) using an internally-supplied clock based on the 144-MHz PLL. Table 7-6 shows actual audio sampling (fs) and BIT_CLK frequencies for I^2S or MSB-Justified interface mode. There are small differences of frequency numbers from the standard sample rate because the PLL 144-MHz (143.7696 MHz) clock is generated using a 3.6864-MHz input clock. Alternatively, an external clock source may be used to obtain exact standard sampling frequencies.

Table 7-6 lists the PLL divider values with respect to the audio sampling frequencies. The PLL divider values are controlled by the SKAUD register, which is described in Section 5.2.3.

Table 7-6. Audio Sampling Frequency as a Function of PLL Divider Value

PLL Divider Value (decimal)	Ideal Sampling Frequency (fs)	Actual Sampling Frequency (fs)	SYS_CLK Frequency	BIT_CLK Frequency
70	8.0 KHz	8.0 KHz	2.05 MHz	513 KHz
51	11.025 KHz	11.01 KHz	2.82 MHz	705 KHz
35	16.00 KHz	16.05 KHz	4.11 MHz	1.03 MHz
25	22.05 KHz	22.46 KHz	5.75 MHz	1.44 MHz
18	32.0 KHz	31.2 KHz	8.0 MHz	2.0 MHz
12	44.1 KHz	43.2 KHz	11.06 MHz	2.76 MHz

For AC-Link (AC'97) operation, the sampling rate is programmable in the codec for incoming audio, and fixed at 48 KHz for playback.

7.3 Data Formats

7.3.1 Parallel Data Formats: Buffer and DRAM Storage

FIFO buffers are 16 words deep x 32 bits wide. This stores 32 samples per channel in each direction, good for about 1 millisecond of stereo audio at 16 KHz sample rate.

Audio data are stored with two samples (Left + Right) per 32-bit word, even if samples are smaller than 16 bits. "Left" channel data occupies bits <15:0>, while the "Right" channel data uses bits <31:16> of the 32-bit word. Within each 16-bit field, the audio sample is left-justified, with unused bits packed as zeroes on the right-hand (LSB) side.

In external DRAM, the mapping of stereo samples is the same as in the FIFO buffers ("Left" channel data occupies bits <15:0>, while the "Right" channel data uses bits <31:16> of the 32-bit word). However, single-channel audio occupies a full 32-bit word per sample, using either the upper or lower half of the word, depending on whether it's considered a "Left" or "Right" sample.

7.3.2 Serial Data Formats for Transfer to/from Codecs

Four pins are used to transfer data between the SA-1111 and external codecs or modems. Several serial data formats are supported.

7.3.2.1 AC-link Serial Data

AC-link specifies a full-duplex serial audio interface between an external codec and a digital controller (SAC). It is used to attach to one or more external AC'97-compliant codecs. The external device generates a 12.288 MHz clock and sends it as BIT_CLK to the SAC. In the SAC, the clock is divided by 256 to yield a 48 KHz audio-sample clock. This is used internally, and sent back to the codec as SYNC. This signal defines the start of each "frame". Within a frame are 13 slots, each slot (except the first) containing 20 bits. The timing of BIT_CLK and SYNC are applied by controllers at each end to sample the two serial data streams - one inbound (SDATA_IN), the other outbound (SDATA_OUT) - and capture serial audio bits going in both directions (see Figure 7-1 and Figure 7-2).

Note: Although AC-link specifies up to 20-bit audio samples, only samples of 16 bits and smaller are supported by this logic and its related FIFO buffers.

Figure 7-1. AC-Link Audio Output Frame

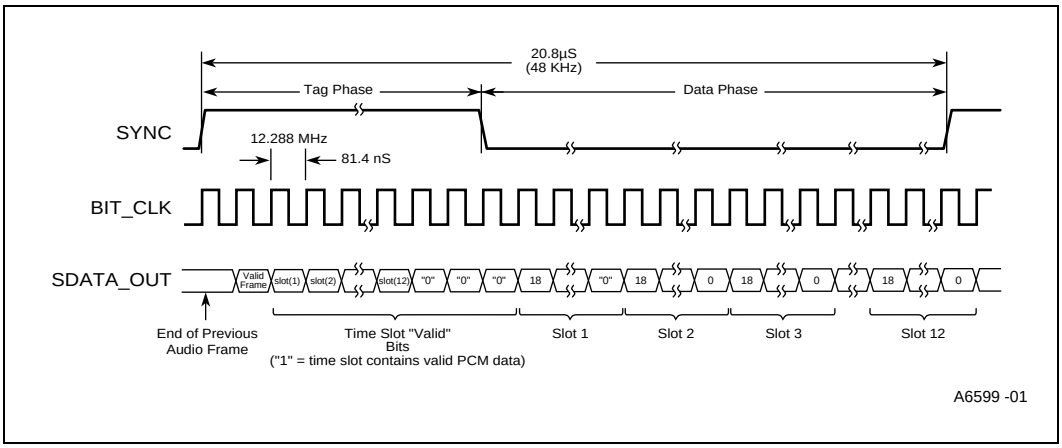
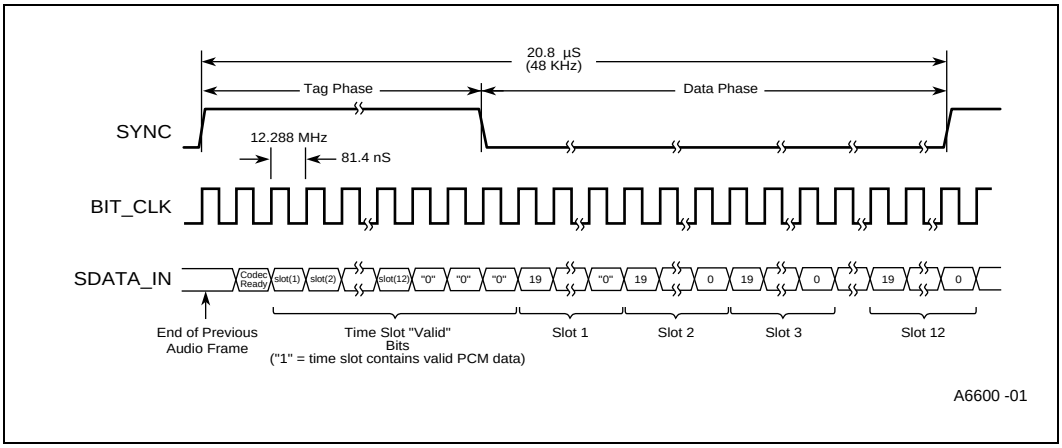


Figure 7-2. AC-Link Audio Input Frame



7.3.2.2 I²S and MSB-Justified Serial Audio Formats

I²S and MSB-Justified are similar protocols for digitized stereo audio transmitted over a serial path. They work with a variety of clock rates, which can be obtained by dividing the 144 MHz PLL clock by a programmable divider, or from an external clock source.

SYS_CLK is normally derived from the PLL clock, divided by a programmable divider (divide integers between 25 and 70 to generate a frequency between approximately 2 MHz and 5.6 MHz. Its frequency is always 256 times the audio sampling frequency.

BIT_CLK supplies the serial audio bit rate, the basis for the external codec bit-sampling logic. Its frequency is 64 times the audio sampling frequency. Divided by 64, the resulting 8 KHz to 22 KHz signal signifies timing for “Left” and “Right” serial data samples passing on the serial data paths. This Left/Right signal is sent to the codec on SYNC pin. Each phase of the Left/Right signal is

accompanied by one serial audio data sample on the data pins SDATA_IN and SDATA_OUT. Figure 7-4 provides timing diagrams that show formats for I^2S and MSB-justified modes of operations.

Figure 7-3. I^2S Data Formats (16 bits)

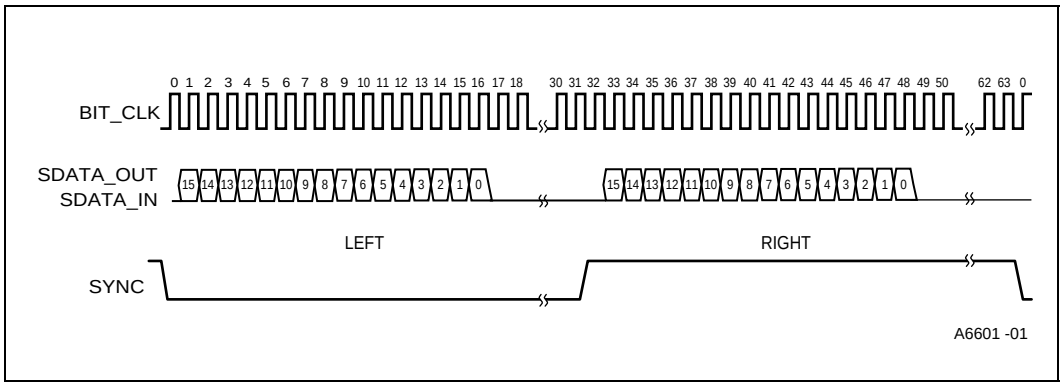
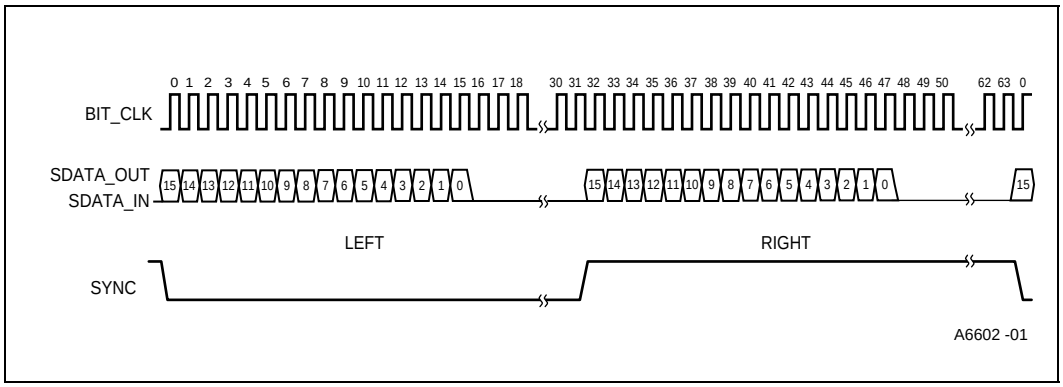


Figure 7-4. MSB-Justified Data Formats (16 bits)



7.4 Serial Audio Controller Registers

There are 24 registers in the Serial Audio Controller block: three control, one data, two status, two interrupt control (clear and test), two L3 control bus, four AC-link control, and ten DMA control registers.

- Control registers are used to program common control, alternate mode specific control, and AC-link specific control registers.
- The Data Register is mapped as one 32-bit word address, which physically points to either of two 32-bit registers. One register is for WRITES, and transfers data to the Transmit FIFO; the other is for READS, and takes data from the Receive FIFO.
- The Status Registers signal the state of the FIFO buffers and the status of the interface which is selected by the common control register.
- The Interrupt Registers are Interrupt Clear Register and Interrupt Test Register.
- The L3 Control Registers are Address Register and Data Register for L3 Control Bus Interface.

- The AC-link Control Registers are Command Address, Command Data, Status Address, and Status Data for AC-link interface.
- The DMA Control Registers are Control/Status Register, Buffer Start Address Registers, and Buffer Count Registers for each DMA Transmit and DMA Receive Transfer.

7.4.1 Serial Audio Control Registers

Three control registers are used to program Serial Audio Controller common control, alternate mode (I^2S and MSB-Justified Interface) specific control, and AC-link Interface specific control registers.

7.4.1.1 Serial Audio Common Control Register (SACR0)

This register controls common functions for Serial Audio Controller (SAC). The control register for alternate mode (I^2S and MSB-Justified Interface) specific functions and AC-link Interface specific functions are described in following 2 sections.

- ENB–This bit enables SAC function.

Note: The following bits are effective when the ENB bit is "1" (the SAC function is enabled).

- BCKD–This bit specifies input/output direction of BIT_CLK.
- RST–This bit specifies resetting the SAC Control and FIFOs when this is on.
- TFTH–The 4-bit data value specifies Transmit FIFO interrupt or DMA threshold level. This value should be programmed to the threshold value minus one.
- RFTH–The 4-bit data value specifies Receive FIFO interrupt or DMA threshold level.

Warning: Using values that are too large for TFTH and RFTH can result in an overflow condition in the FIFOs, which are 16 levels deep. Using values that are too small can result in an underflow condition in the FIFOs. The optimum value is dependent upon latency and the bandwidth, and how fast the request can be serviced.

Note: The power-up/reset default value of this register is 7700h.

Table 7-7. SACR0 Bit Descriptions

0x0000600 SACR0 SA-1111 Companion Chip																															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																RFTH				TFTH				Reserved				RST	BCKD	Reserved	ENB
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	1	1	1	0	0	0	0	0	0	0	0

Bits	Name	Type	Description
0	ENB	RW	Enable SAC function: 0 = Pins function as GPIOs (SAC function is disabled) 1 = Pins function as Serial Audio Controller (SAC function is enabled)
1	—	RW	Reserved
2	BCKD	RW	Specify BIT_CLK pin direction: 0 = Output 1 = Input
3	RST	RW	Reset the SAC Control and FIFOs except this register: 0 = Not reset 1 = Reset is Active to Other SAC Registers
7:4	—	—	Reserved
11:8	TFTH	RW	Transmit FIFO interrupt or DMA threshold; set to value 0 – 15. This value should be set to the desired threshold value minus one. Larger values provide a longer latency than smaller values. Latency values of nine or greater have their bursts separated into two bursts. Using threshold values that are too large can result in an overflow condition. Using threshold values that are too small can result in an underflow condition. The optimum value, which is system dependent, results in the FIFO being half full.
15:12	RFTH	RW	Receive FIFO interrupt or DMA threshold; set to value 0 – 15. This value should be set to the desired threshold value minus one. Larger values provide a longer latency than smaller values. Latency values of nine or greater have their bursts separated into two bursts. Using threshold values that are too large can result in an overflow condition. Using threshold values that are too small can result in an underflow condition. The optimum value, which is system dependent, results in the FIFO being half full.
31:16	—	—	Reserved

7.4.1.2 Serial Audio Alternate Mode (I^2S/MSB-Justified) Control Register (SACR1)

This register controls alternate mode (I^2S and MSB-Justified Interface) specific functions when SACMDSL bit of SKCR selects alternate mode.

- AMSL–This bit selects Operation Mode of Alternate Mode, whether I^2S or MSB-Justified Operation.
- L3EN–This bit enables L3 Control Bus function.

- L3MB–This bit indicates L3 Control Bus Data is Multiple Byte Transfer.
- DREC–This bit disables Recoding Function of I^2S or MSB-Justified Interface.
- DRPL–This bit disables Replaying Function of I^2S or MSB-Justified Interface.
- ENLBF–This bit enables L3 Control Bus and I^2S/MSB Interface Loop Back Function which is for test only.

Note: The power-up/reset default value of this register is 0000h.

Table 7-8. SACR1 Bit Descriptions

0x0000604 **SACR1** **SA-1111 Companion Chip**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																										ENLBF	DRPL	DREC	L3MB	L3EN	AMSL
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Type	Description
0	AMSL	RW	Specify Alternate Mode (I^2S or MSB-Justified) Operation: 0 = Select I^2S Operation Mode 1 = Select MSB-Justified Operation Mode
1	L3EN	RW	Enable L3 Control Bus function: 0 = Pins PWM<1>, MSCLK, MSDATA function as named, or as GPIOs 1 = Pins function as L3 bus signals
2	L3MB	RW	L3 Control Bus Data Multiple Byte Transfer: 0 = L3 Control Bus Data is Last Byte Transfer 1 = L3 Control Bus Data is Multiple Byte Transfer
3	DREC	RW	Disable Recording Function of I^2S or MSB-Justified Interface: 0 = Recording Function is Enabled 1 = Recording Function is Disabled
4	DRPL	RW	Disable Replaying Function of I^2S or MSB-Justified Interface: 0 = Replaying Function is Enabled 1 = Replaying Function is Disabled
5	ENLBF	RW	Enable L3 Control Bus and I^2S/MSB Interface Loop Back Function: 0 = L3 Control Bus and I^2S/MSB Interface Loop Back Function is Disabled 1 = L3 Control Bus and I^2S/MSB Interface Loop Back Function is Enabled
31:6	—	—	Reserved

7.4.1.3 Serial Audio AC-link Control Register (SACR2)

This register controls AC-link specific functions when SACMDSL bit of SKCR selects AC-link mode.

- TS3V–This bit indicates Transmit Slot 3 Data Out is Valid (Left Channel Data Out is Valid).
- TS4V–This bit indicates Transmit Slot 4 Data Out is Valid (Right Channel Data Out is Valid).
- WKUP–This bit initiates "Wake Up AC'97 Codec Sequence" by Activating SYNC Signal.
- DREC–This bit disables Recoding Function of AC-link Interface.
- DRPL–This bit disables Replaying Function of AC-link Interface.
- ENLBF–This bit enables AC-link Interface Loop Back Function, which is for test only.
- RESET#–This bit specifies a hardware RESET# signal to SYS_CLK pin.

Note: The power-up/reset default value of this register is 0000h.

Table 7-9. SACR2 Bit Descriptions (Sheet 1 of 2)

0x0000608 **SACR2** **SA-1111 Companion Chip**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																									RESET#	ENLBF	DRPL	DREC	WKUP	TS4V	TS3V
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Type	Description
0	TS3V	RW	Transmit Slot 3 Valid (Left Channel Data Out is Valid): 0 = Transmitting Time Slot 3 Data is not Valid 1 = Transmitting Time Slot 3 Data is Valid
1	TS4V	RW	Transmit Slot 4 Valid (Right Channel Data Out is Valid): 0 = Transmitting Time Slot 4 Data is not Valid 1 = Transmitting Time Slot 4 Data is Valid
2	WKUP	WO/RO	Wake Up AC'97 Codec: 0 = Wake Up AC'97 Codec is Done (Reset by H/W after Waking Up) 1 = Wake Up AC'97 Codec by Activating SYNC Signal
3	DREC	RW	Disable Recording Function of AC-link Interface: 0 = Recording Function is Enabled 1 = Recording Function is Disabled
4	DRPL	RW	Disable Replaying Function of AC-link Interface: 0 = Replaying Function is Enabled 1 = Replaying Function is Disabled

Table 7-9. SACR2 Bit Descriptions (Sheet 2 of 2)

0x0000608 SACR2 SA-1111 Companion Chip

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																									RESET#	ENLBF	DRPL	DREC	WKUP	TS4V	TS3V
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Type	Description
5	ENLBF	RW	Enable AC-link Interface Loop Back Function (for test only): 0 = AC-link Interface Loop Back Function is Disabled 1 = AC-link Interface Loop Back Function is Enabled
6	RESET#	RW	Specify RESET# signal to SYS_CLK output: 0 = RESET# signal is active low 1 = RESET# signal is not active
31:7	—	—	Reserved

Note: The power-up/reset default value of this register is 0000h.

7.4.2 Serial Audio Status Register (SASR0)

There are two status registers, the Serial Audio Status Register SASR0 is used for the I^2S and MSB-Justified Interface and FIFO Status (SASR0), the other is for AC-link Interface and FIFO Status (SASR1). Bits 15 to 0 are common FIFO status bits with SASR1. This status register is a read-only register.

Table 7-10. SASR0 Bit Descriptions (Sheet 1 of 2)

0x000060C SASR0 SA-1111 Companion Chip

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved														L3RD	L3WD	RFL				TFL				Reserved	ROR	TUR	RFS	TFS	BSY	RNE	TNF
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description
0	TNF	RO Transmit FIFO not full: 0 - Transmit FIFO is full 1 - Transmit FIFO is not full

Table 7-10. SASR0 Bit Descriptions (Sheet 2 of 2)

0x000060C — SASR0 — SA-1111 Companion Chip

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved														L3RD	L3WD	RFL				TFL				Reserved	ROR	TUR	RFS	TFS	BSY	RNE	TNF
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name		Description
1	RNE	RO	Receive FIFO not empty: 0 - Receive FIFO is empty 1 - Receive FIFO is not empty
2	BSY	RO	SAC Busy: 0 - SAC is idle or disabled 1 - SAC currently transmitting or receiving a frame
3	TFS	RO	Transmit FIFO Service Request: 0 - Transmit FIFO level exceeds TFL threshold, or SAC disabled 1 - Transmit FIFO level is at or below TFL threshold (Interruptible)
4	RFS	RO	Receive FIFO Service Request: 0 - Receive FIFO level below RFL threshold, or SAC disabled 1 - Receive FIFO level is at or above RFL threshold (Interruptible)
5	TUR	RO	Transmit FIFO Under-run: 0 - Transmit FIFO has not experienced an under-run 1 - Attempted data write to full Transmit FIFO (Interruptible)
6	ROR	RO	Receive FIFO Overrun: 0 - Receive FIFO has not experienced an overrun 1 - Attempted data write to full Receive FIFO (Interruptible)
7	—	—	Reserved
11:8	TFL	RO	Transmit FIFO Level: Number of entries in Transmit FIFO
15:12	RFL	RO	Receive FIFO Level: Number of entries in Receive FIFO
16[a]	L3WD	RO	L3 Control Bus Data Write Done: 0 - This status bit is reset by Clear Register (SASCR) writing 1 - L3 Control Bus Data Write Data is Done (Interruptible)
17[a]	L3RD	RO	L3 Control Bus Data Read Done: 0 - This status bit is reset by Clear Register (SASCR) writing 1 - L3 Control Bus Data Read Data is Done (Interruptible)
31:18	—	—	Reserved

a. Bits 16 and 17 are I^2S and MSB-justified interface specific status bits.

7.4.3 Serial Audio Status Registers (SASR1)

Serial Audio Status Register SASR1 is used for the AC-link Interface and FIFO Status. Bits 15 to 0 are common FIFO status bits with SASR0. This status register is a read-only register.

Bits 16 to 22 are AC-link Interface specific status bits.

Table 7-11. SASR1 Bit Descriptions (Sheet 1 of 2)

0x0000610 SASR1 SA-1111 Companion Chip

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved									RSV4	RS3V	CRDY	CLPM	RSTO	SADR	CADT	RFL				TFL				Reserved	ROR	TUR	RFS	TFS	BSY	RNE	TNF
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	
0	TNF	RO	Transmit FIFO not full: 0 - Transmit FIFO is full 1 - Transmit FIFO is not full
1	RNE	RO	Receive FIFO not empty: 0 - Receive FIFO is empty 1 - Receive FIFO is not empty
2	BSY	RO	SAC Busy: 0 - SAC is idle or disabled 1 - SAC currently transmitting or receiving a frame
3	TFS	RO	Transmit FIFO Service Request: 0 - Transmit FIFO level exceeds TFL threshold, or SAC disabled 1 - Transmit FIFO level is at or below TFL threshold (Interruptible)
4	RFS	RO	Receive FIFO Service Request: 0 - Receive FIFO level below RFL threshold, or SAC disabled 1 - Receive FIFO level is at or above RFL threshold (Interruptible)
5	TUR	RO	Transmit FIFO Under-run: 0 - Transmit FIFO has not experienced an under-run 1 - Attempted data write to full Transmit FIFO (Interruptible)
6	ROR	RO	Receive FIFO Overrun: 0 - Receive FIFO has not experienced an overrun 1 - Attempted data write to full Receive FIFO (Interruptible)
7	—	—	Reserved
11:8	TFL	RO	Transmit FIFO Level: Number of entries in Transmit FIFO
15:12	RFL	RO	Receive FIFO Level: Number of entries in Receive FIFO
16	CADT	RO	0 - This status bit is reset by Clear Register (SASCR) writing 1 - AC-link Command Address and Data are Received is done (Interruptible)
17	SADR	RO	0 - This status bit is reset by Clear Register (SASCR) writing 1 - AC-link Command Address and Data are Received is done (Interruptible)

Table 7-11. SASR1 Bit Descriptions (Sheet 2 of 2)

0x0000610 — SASR1 — SA-1111 Companion Chip

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved									RSV4	RS3V	CRDY	CLPM	RSTO	SADR	CADT	RFL				TFL				Reserved	ROR	TUR	RFS	TFS	BSY	RNE	TNF
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name		Description
18	RSTO	RO	Read Status Time Out: 0 - Read Status Time Out not occurred 1 - Read Status Time Out is detected, 4 Frames after Read Status Command (Interruptible)
19	CLPM	RO	0 - BIT_CLK from AC'97 Codec is active (after Wake Up) 1 - Command for AC'97 Codec Low Power Mode is detected
20	CRDY	RO	0 - Codec Ready Status Bit (Input Frame Slot0 bit-15) is not set 1 - AC'97 Codec is Ready for Normal Operation
21	RS3V	RO	Received Slot 3 Valid (Left Channel Data In is Valid): 0 = Receive Time Slot 3 Data is not Valid 1 = Receive Time Slot 3 Data is Valid
22	RS4V	RO	Received Slot 4 Valid (Right Channel Data In is Valid): 0 = Received Time Slot 4 Data is not Valid 1 = Received Time Slot 4 Data is Valid
31:23	—	—	Reserved

7.4.4 Serial Audio Status Clear Register (SASCR)

The Serial Audio Status Clear Register (SASCR) is in the Status Control Register. The bit positions correspond to each status source bit position in the Status register.

Table 7-12. SASCR Bit Descriptions

0x0000618															SASCR														SA-1111 Companion Chip							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved													STD	RDD	DTS	Reserved									ROR	TUR	Reserved				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Type	Description
4:0	—	—	Reserved
5	TUR	WO	Clears transmit FIFO under-run status bit, which is bit 5 of SASR0 and SASR1 1 = Clears target bit. 0 = No effect.
6	ROR	WO	Clears Receive FIFO overrun status bit, which is bit 6 of SASR0 and SASR1 1 = Clears target bit. 0 = No effect.
15:7	—	—	Reserved
16	DTS	WO	Clears L3C/AC-link Data Sent status bit, which is bit 16 of SASR0 and SASR1 1 = Clears target bit. 0 = No effect.
17	RDD	WO	Clears L3C/AC-link Data Read Done status bit, which is bit 17of SASR0 and SASR1 1 = Clears target bit. 0 = No effect.
18	STO	WO	Clears AC-link Read Status Time Out status bit, which is bit 18 of SASR1 1 = Clears target bit. 0 = No effect.
31:19	—	—	Reserved

7.4.5 L3 Control Bus Address Register (L3CAR)

There are two L3 Control Bus Registers: L3 Control Bus Address Register (L3CAR) and L3 Control Bus Data Register (L3CDR). The L3CAR provides the 8-bit address value that is transmitted on the L3 bus.

Table 7-13. L3CAR Bit Descriptions

0x000061C **L3CAR** **SA-1111 Companion Chip**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																							ADR								
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Type	Description
7:0	ADR	RW	8-bit address value to be transmitted on L3 bus. Transfer type is encoded into 2 LSBs of address.
31:8	—	—	Reserved

7.4.6 L3 Control Bus Data Register (L3CDR)

There are two L3 Control Bus Registers: L3 Control Bus Address Register (L3CAR) and L3 Control Bus Data Register (L3CDR). The L3CDR provides the 8-bit data value that is transmitted and received on the L3 bus.

Table 7-14. L3CDR Bit Descriptions

0x0000620 **L3CDR** **SA-1111 Companion Chip**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																								DAT							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Type	Description
7:0	DAT	RW	8-bit data value transmitted/received on L3 bus
31:8	—	—	Reserved

Most L3 transfers are writes to codec registers. The address register is written first, so address and command (embedded in 2 LSBs of the address byte) are sent to the target codec.

- If both address LSBs are not “01”, it is a write transfer to the codec. When the L3 Data Register is subsequently written, the register’s contents are transmitted serially to the codec.
- If both address LSBs are “01”, it is a read request from the codec. Following the address transmission, SA-1111 begins clocking L3_CLK and tristates (stops driving) L3_DATA. The codec drives the serial data line, returning serial data which is clocked into the L3 Receive Data Register. A read from L3CDR transfers the read data byte to the processor.
- After each byte is transferred, status bit L3WD or L3RD in SASR0 is set. If its corresponding interrupt is enabled, it generates an interrupt to the system processor.

7.4.7 AC-link Control Registers

There are four AC-link Control Registers: AC-link Command Address Register (ACCAR), AC-link Command Data Register (ACCDR), AC-link Status Address Register (ACSAR), and AC-link Status Data Register (ACSDR).

7.4.8 AC-link Command Address Register (ACCAR)

AC-link Command Address and Command Data Transfer is executed automatically after two data writing to ACCAR and ACCDR. When AC-link Command Address and Command Data Transfer is done, CADT bit of SASR1 status register and its interrupt request are set.

Table 7-15. ACCAR Bit Descriptions

0x0000624 — **ACCAR** — **SA-1111 Companion Chip**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved												CAR																			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Type	Description
19:0	CAR	RW	20-bit read/write register, 20-bit data for AC-link Command Data Port (Slot 1of Output Frame). **Bits 11 to 0 are reserved.**
31:20	—	—	Reserved

7.4.9 AC-link Command Data Register (ACCDR)

AC-link Command Address and Command Data Transfer are executed automatically after two data writings to ACCAR and ACCDR. When AC-link Command Address and Command Data Transfer is done, CADT bit of SASR1 status register and its interrupt request are set.

Table 7-16. ACCDR Bit Descriptions

0x0000628 ACCDR SA-1111 Companion Chip

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved												CDR																			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Type	Description
19:0	CDR	RW	20-bit read/write register, 20-bit data for AC-link Command Data Port (Slot 2 of Output Frame). **Bits 3 to 0 are reserved.**
31:20	—	—	Reserved

7.4.10 AC-link Status Address Register (ACSAR)

When AC-link Status Address and Status Data are Received, SADR bit of SASR1 status register and make interrupt request are set and Status Address and Status Data are held in ACSAR and ACSDR.

Table 7-17. ACSAR Bit Descriptions

0x000062C ACSAR SA-1111 Companion Chip

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved												SAR																Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Type	Description
3:0	—	—	Reserved
19:4	SAR	RO	16-bit read only register, 16-bit data from AC-link Status Address Port (Slot 1 of Input Frame). Echo of register index for which data is being returned.
31:20	—	—	Reserved

7.4.11 AC-link Status Address Register (ACSDR)

When AC-link Status Address and Status Data are Received, SADR bit of SASR1 status register and make interrupt request are set and Status Address and Status Data are held in ACSAR and ACSDR.

Table 7-18. ACSDR Bit Descriptions

0x0000630 ACSDR SA-1111 Companion Chip

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved												SDR																Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Type	Description
3:0	—	—	Reserved
19:4	SDR	RO	16-bit read only register, 16-bit data from AC-link Status Data Port (Slot 2 of Input Frame). The Status Data Port delivers 16-bit control register read data (bits 20 to 4 of this register).
31:8	—	—	Reserved

7.4.12 Serial Audio DMA Transmit Control/Status Register (SADTCS)

Table 7-19. SADTCS Bit Descriptions (Sheet 1 of 2)

0x0000634 SADTCS SA-1111 Companion Chip

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																								TBIU	TDSTB	TDBDB	TDSTA	TDBDA	Reserved		TDEN
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Type	Description
0	TDEN	RW	Serial Audio DMA Transmit Enable. 1= Enable. 0 = No effect.
2:1	—	—	Reserved.
3	TDBDA	W1C	Serial Audio DMA Transmit Buffer Done A (interruptible).

Table 7-19. SADTCS Bit Descriptions (Sheet 2 of 2)

0x0000634 SADTCS SA-1111 Companion Chip

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																								TBIU	TDSTB	TDBDB	TDSTA	TDBDA	Reserved		TDEN
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Type	Description
4	TDSTA	WO	Serial Audio DMA Transmit Buffer Start Transfer A; Clears TDBDA. 1= Starts transfer. 0 = No effect.
5	TDBDB	W1C	Serial Audio DMA Transmit Buffer Done B (interruptible).
6	TDSTB	WO	Serial Audio DMA Transmit Buffer Start Transfer B, Clears TDBDB. 1= Starts transfer. 0 = No effect.
7	TBIU	RO	Serial Audio DMA Transmit Buffer (A or B) In Use.
31:8	—	—	Reserved.

7.4.13 Serial Audio DMA Transmit Buffer Start Address Register A (SADTSA)

Table 7-20. SADTSA Bit Descriptions

0x0000638 SADTSA SA-1111 Companion Chip

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TDSA																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Type	Description
31:0	TDSA	RW	Serial Audio DMA Transmit Start Address A, read/write register. It contains the starting memory address for buffer area A. **The LSB 2 bits must be 00.** Data Transfer unit is four bytes and it counts four at a time.

7.4.14 Serial Audio DMA Transmit Buffer Count Register A (SADTCA)

Table 7-21. SADTCA Bit Descriptions

0x000063C																															
SADTCA																															
SA-1111 Companion Chip																															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																			TDCA												
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Type	Description
12:0	TDCA	RW	Serial Audio DMA Transmit Transfer Count A, read/write register. It contains the transfer count in bytes for memory buffer area A. **The LSB 2 bits must be 00.** Data Transfer unit is four bytes and it counts four at a time.
31:13	—	—	Reserved.

7.4.15 Serial Audio DMA Transmit Buffer Start Address Register B (SADTSB)

Table 7-22. SADTSB Bit Descriptions

0x0000640																															
SADTSB																															
SA-1111 Companion Chip																															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TDSB																														
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Type	Description
31:0	TDSB	RW	Serial Audio DMA Transmit Start Address B, read/write register. It contains the starting memory address for buffer area B. **The LSB 2 bits must be 00.** Data Transfer unit is four bytes and it counts four at a time.

7.4.16 Serial Audio DMA Transmit Buffer Count Register B (SADTCB)

Table 7-23. SADTCB Bit Descriptions

0x0000644 SADTCB SA-1111 Companion Chip

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																						TDCB									
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Type	Description
12:0	TDCB	RW	Serial Audio DMA Transmit Transfer Count B, read/write register. It contains the transfer count in bytes for memory buffer area B. **The LSB 2 bits must be 00.** Data Transfer unit is four bytes and it counts four at a time.
31:13	—	—	Reserved.

7.4.17 Serial Audio DMA Receive Control/Status Register (SADRCS)

Table 7-24. SADRCS Bit Descriptions Bit Descriptions (Sheet 1 of 2)

0x0000648 SADRCS SA-1111 Companion Chip

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																								RBIU	RDSTB	RDBDB	RDSTA	RDBDA	Reserved		RDEN
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Type	Description
0	RDEN	RW	Serial Audio DMA Receive Enable. 1= Enable. 0 = No effect.
2:1	—	—	Reserved.
3	RDBDA	W1C	Serial Audio DMA Receive Buffer Done A (interruptible).
4	RDSTA	WO	Serial Audio DMA Receive Buffer Start Transfer A; Clears RDBDA. 1= Starts transfer. 0 = No effect.
5	RDBDB	W1C	Serial Audio DMA Receive Buffer Done B (interruptible).

Table 7-24. SADRCS Bit Descriptions Bit Descriptions (Sheet 2 of 2)

0x0000648	SADRCS	SA-1111 Companion Chip

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																								RBIU	RDSTB	RDBDB	RDSTA	RDBDA	Reserved		RDEN
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Type	Description
6	RDSTB	WO	Serial Audio DMA Receive Buffer Start Transfer B, Clears RDBDB. 1= Starts transfer. 0 = No effect.
7	RBIU	RO	Serial Audio DMA Receive Buffer (A or B) In Use.
31:8	—	—	Reserved.

7.4.18 Serial Audio DMA Receive Buffer Start Address Register A (SADRSA)

Table 7-25. SADRSA Bit Descriptions

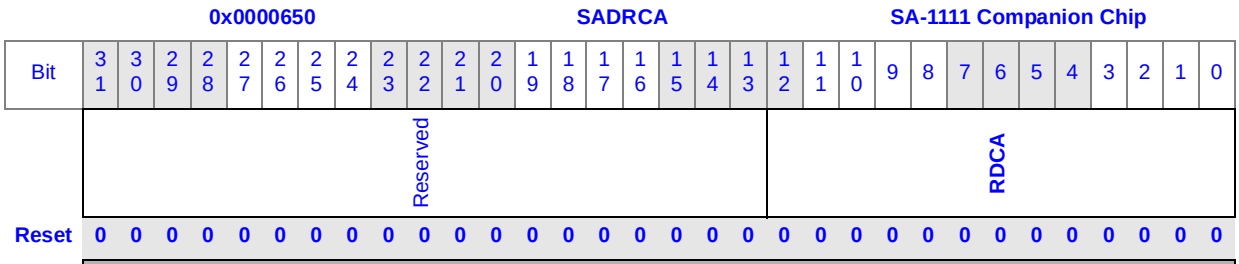
0x000064C	SADRSA	SA-1111 Companion Chip

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RDSA																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Type	Description
31:0	RDSA	RW	Serial Audio DMA Receive Start Address A, read/write register. It contains the starting memory address for buffer area A. **The LSB 2 bits must be 00.** Data Transfer unit is four bytes and it counts four at a time.

7.4.19 Serial Audio DMA Receive Buffer Count Register A (SADRCA)

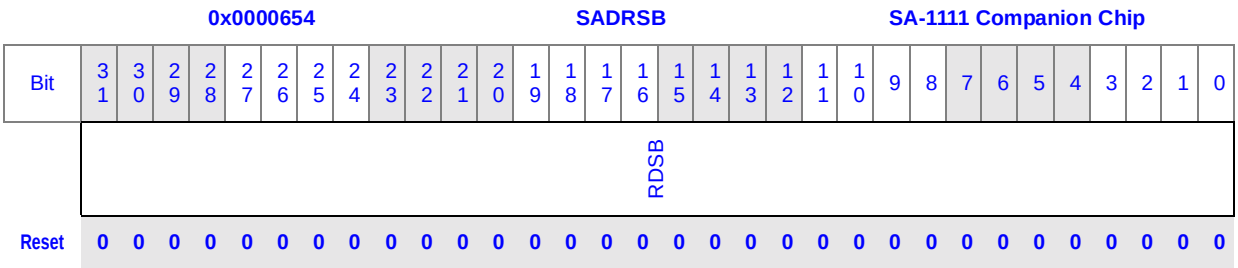
Table 7-26. SADRCA Bit Descriptions



Bits	Name	Type	Description
12:0	RDCA	RW	Serial Audio DMA Receive Transfer Count A, read/write register. It contains the transfer count in bytes for memory buffer area A. **The LSB 2 bits must be 00.** Data Transfer unit is four bytes and it counts four at a time.
31:13	—		Reserved.

7.4.20 Serial Audio DMA Receive Buffer Start Address Register B (SADRSB)

Table 7-27. SADRSB Bit Descriptions



Bits	Name	Type	Description
31:0	RDSB	RW	Serial Audio DMA Receive Start Address B, read/write register. It contains the starting memory address for buffer area B. **The LSB 2 bits must be 00.** Data Transfer unit is four bytes and it counts four at a time.

7.4.21 Serial Audio DMA Receive Buffer Count Register B (SADRCB)

Table 7-28. SADRCB Bit Descriptions

0x0000658 — SADRCB — SA-1111 Companion Chip

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																				RDCB											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Type	Description
12:0	RDCB	RW	Serial Audio DMA Receive Transfer Count B, read/write register. It contains the transfer count in bytes for memory buffer area B. **The LSB 2 bits must be 00.** Data Transfer unit is four bytes and it counts four at a time.
31:13	—	—	Reserved.

7.4.22 Serial Audio Interrupt Test Register (SAITR)

Writing "1" to the corresponding bit position to the Serial Audio Interrupt Test Register generates a interrupt strobe signal to Interrupt Controller for test purposes. All bit symbols come from the status registers: I^2S and MSB-Justified Interface and FIFO Status (SASR0), AC-link Interface and FIFO Status (SASR1), DMA Transmit Control/Status (SADTCS), and DMA Receive Control/Status (SADRCS). This is a write-only register.

Table 7-29. SAITR Bit Descriptions (Sheet 1 of 2)

0x000065C — SAITR — SA-1111 Companion Chip

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																				RDBDB	RDBDA	TDBDB	TDBDA	Reserved	RSTO	SADR	CADT	ROR	TUR	RFS	TFS
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description
0	TFS	WO Transmit FIFO Service Request 1= Force interrupt. 0 = No effect.
1	RFS	WO Receive FIFO Service Request 1= Force interrupt. 0 = No effect.
2	TUR	WO Transmit FIFO Under-run 1= Force interrupt. 0 = No effect.

Table 7-29. SAITR Bit Descriptions (Sheet 2 of 2)

0x000065C — SAITR — SA-1111 Companion Chip

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																				RDBDB	RDBDA	TDBDB	TDBDA	Reserved	RSTO	SADR	CADT	ROR	TUR	RFS	TFS
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	
3	ROR	WO	Receive FIFO Overrun 1= Force interrupt. 0 = No effect.
4	CADT	WO	Command Address/Data Transfer is Done 1= Force interrupt. 0 = No effect.
5	SADR	WO	Status Address/Data Receive is Done 1= Force interrupt. 0 = No effect.
6	RSTO	WO	Read Status Time Out 1= Force interrupt. 0 = No effect.
7	—	—	Reserved
8	TDBDA	WO	DMA Transmit Buffer Done A1= Force interrupt. 0 = No effect.
9	TDBDB	WO	DMA Transmit Buffer Done B 1= Force interrupt. 0 = No effect.
10	RDBDA	WO	DMA Receive Buffer Done A 1= Force interrupt. 0 = No effect.
11	RDBDB	WO	DMA Receive Buffer Done B 1= Force interrupt. 0 = No effect.
31:12	—	—	Reserved

7.4.23 Serial Audio Data Register (SADR)

The SAC Data Register (SADR) is a block of eight 32-bit locations that can be accessed by 32-bit data transfers. The address block represents two physical registers. One register is for WRITES, and transfers data to the Transmit FIFO; the other register is for READS, and takes data from the Receive FIFO.

Table 7-30. SADR Bit Descriptions

0x0000680 to0x00006BC																SADR												SA-1111 Companion Chip															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0										
	DTH																DTL																									
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0										

Bits	Name	Type	Description
15:0	DTL	WO/RO	Serial Audio Data Word to be read from Receive FIFO or to be written to Transmit FIFO. This is PCM 16-bit Left Channel Data.
31:16	DTH	WO/RO	Serial Audio Data Word to be read from Receive FIFO or to be written to Transmit FIFO. This is PCM 16-bit Right Channel Data.

7.4.24 Serial Audio Controller: Register Memory Map

Table 7-31 lists the serial audio controller register memory map and the physical addresses to access them.

Table 7-31. Serial Audio Controller Register Memory Map (Sheet 1 of 2)

Address	Mnemonic	Full Name
0x0000600	SACR0	Serial Audio Common Control Register
0x0000604	SACR1	Serial Audio Alternate Mode (I^2S/MSB-Justified) Control Register
0x0000608	SACR2	Serial Audio AC-link Control Register
0x000060C	SASR0	Serial Audio I^2S/MSB-Justified Interface and FIFO Status Register
0x0000610	SASR1	Serial Audio AC-link Interface and FIFO Status Register
0x0000614	—	Reserved
0x0000618	SASCR	Serial Audio Status Clear Register
0x000061C	L3CAR	L3 Control Bus Address Register
0x0000620	L3CDR	L3 Control Bus Data Register
0x0000624	ACCAR	AC-link Command Address Register
0x0000628	ACCDR	AC-link Command Data Register
0x000062C	ACSAR	AC-link Status Address Register
0x0000630	ACSDR	AC-link Status Data Register
0x0000634	SADTCS	Serial Audio DMA Transmit Control/Status Register

Table 7-31. Serial Audio Controller Register Memory Map (Sheet 2 of 2)

0x0000638	SADTSA	Serial Audio DMA Transmit Buffer Start Address Register A
0x000063C	SADTCA	Serial Audio DMA Transmit Buffer Count Register A
0x0000640	SADTSB	Serial Audio DMA Transmit Buffer Start Address Register B
0x0000644	SADTCB	Serial Audio DMA Transmit Buffer Count Register B
0x0000648	SADRCS	Serial Audio DMA Receive Control/Status Register
0x000064C	SADRSA	Serial Audio DMA Receive Buffer Start Address Register A
0x0000650	SADRCA	Serial Audio DMA Receive Buffer Count Register A
0x0000654	SADRSB	Serial Audio DMA Receive Buffer Start Address Register B
0x0000658	SADRCB	Serial Audio DMA Receive Buffer Count Register B
0x000065C	SAITR	Serial Audio Interrupt Test Register
0x0000680 to 0x00006BC	SADR	Serial Audio Data Register (a block of 32-bit 16locations)

SSP Serial Port 8

The SSP Serial Port is a full-duplex synchronous serial interface. It can connect to a variety of external analog-to-digital (A/D) converters, audio and telecom codecs, and many other devices which use serial protocols for transferring data. It supports National's Microwire*, Texas Instruments' synchronous serial protocol (SSP), and Motorola's serial peripheral interface (SPI) protocol.

The SSP operates in master mode (the attached peripheral functions as a slave), and supports serial bit rates from 7.2 KHz to 1.84 MHz. Serial data formats may range from 4 to 16 bits in length. Two on-chip RAM buffers function as independent FIFOs for data, one for each direction. The buffers are 16 entries deep x 16 bits wide.

Buffers may be burst-loaded or emptied by the system processor using SRAM-like burst transfers, from one to eight words per transfer. Each 32-bit word from the system fills one entry in FIFO using the lower half 16-bits of a 32-bit word.

8.1 Signal Description

This section lists the signals which interface to the SSP Serial Port.

8.1.1 External Interface to SSP, SPI, or Microwire Peripherals

Table 8-1 lists the external signals between the SSP and an external peripheral. Alternatively, the four pins may be configured as GPIOs if the SSP function is not used (see Chapter 10).

Table 8-1. External Interface to Codec

Name	Direction	Description
SCLK	O (SSP mode)	Serial bit-rate clock
SFRM	O (SSP mode)	Frame indicator
TXD	O (SSP mode)	Transmit Data (serial data out)
RXD	I (SSP mode)	Receive Data (serial data in)

SCLK is the bit-rate clock, driven from SSP port to the peripheral. Normally it toggles only when data is actively being transmitted and received.

SFRM is the framing signal, indicating beginning and end of a serialized data word.

TXD and RXD are the outbound (Transmit) and inbound (Receive) serialized data words, respectively. Word length is a function of the selected serial data format (DSS in Control Register 0).

If SSP operation is disabled, the four SSP pins are available for GPIO use. See Chapter 10 for details on configuring pin direction and interrupt capabilities.

8.1.2 Internal Interfaces

8.1.2.1 Register Access Bus (RAB)

The Register Access Bus (RAB) is used for register reading and writing. Table 8-2 lists the signals in the RAB bus interface.

Table 8-2. RAB Interface Signals

Name	Direction	Description
RCLK	I	24-MHz RAB clock.
nBRES	I	RAB reset, asserted low.
RegAddr<8:2>	I	7-bit local address from RAB.
RegDIn<31:0>	I	32-bit write data to SSP registers or Transmit (lower 16 bits are valid data).
RegDOut<31:0>	O	32-bit read data from SSP registers or Receive (lower 16 bits are valid data).
RegBE<3:0>	I	4-bit Byte Enable for SSP register/Transmit FIFO writes.
RegSspStb	I	Write strobe.
RegSspRdEn	I	Output Enable for register/Receive FIFO reads.

8.1.2.2 Data Block Transfers

Block-data transfers are made through on-chip buffers organized as FIFOs, using the D_IN or D_OUT buses. The mechanism for transferring block data between FIFO and the system is SRAM-like block data transfers are initiated by the Intel® StrongARM* SA-1110 Microprocessor (SA-1110). Control uses RAB bus signals as listed in Table 8-2 to read and write bursts of data.

8.2 SSP Operation

Serial data is transferred between the system and an external peripheral through FIFO buffers in the SSP Port. Transfers are initiated by the host processor, using an SRAM-like burst of one to eight words, to/from system memory. Operation is full duplex - separate buffers and serial data paths permit simultaneous transfers in both directions.

8.2.1 Processor-Initiated Data Transfer

Transmit data (from system to peripheral) is written by the host processor through the SBI to the SSP block's "Transmit" buffer. The write takes the form of an SRAM-like burst, with one to eight words being transferred per burst. The buffer works as a FIFO, and is seen as a block of sixteen 16-bit locations by the processor. The SSP Port then takes the data from the buffer, serializes it, and sends it over the serial wire (TXD) to the peripheral.

Receive data from the peripheral (on RXD) is converted to parallel words and stored in the "Receive" FIFO buffer. A programmable "fullness" threshold, when passed, triggers an interrupt to the Interrupt Controller (and thence, if enabled, to an interrupt input on the SA-1110). The interrupt

service routine responds by identifying the source of the interrupt and then doing an SRAM-like burst read from the inbound FIFO buffer. All buffer reads and writes use the internal system bus to transfer data between SBI and the buffer.

The host processor differentiates between the two FIFOs by whether it does a read or a write transfer. Read bursts automatically target the Receive FIFO, while write bursts write data to the Transmit FIFO. From a memory map point of view, they are at sixteen addresses (actually, a block of 16 contiguous addresses).

FIFO buffers are 16 words deep x 16 bits wide for read and write. This stores up to 16 samples per buffer.

8.3 Data Formats

8.3.1 Serial Data Formats for Transfer to/from Peripherals

Four pins are used to transfer data between the SA-1111 and external devices. Although there are several formats for serial data, they have the same basic structure:

- SCLK–Defines the bit rate at which serial data is driven onto, and sampled from, the bus.
- SFRM–Defines the boundaries of a basic data “unit,” comprised of multiple serial bits.
- TXD–Is the serial datapath for outbound data, from system to peripheral.
- RXD–Is the serial datapath for inbound data, from peripheral to system.

A data frame may contain from 4 to 16 bits, depending on the format selected. Serial data is transmitted MSB first. Three formats are supported: Motorola SPI, Texas Instruments synchronous serial protocol (SSP), and National Microwire.

For SPI and Microwire formats, SFRM functions as a chip select to enable the external device (target of the transfer), and is held active-low during the data transfer. For SSP format, SFRM is pulsed high for one (serial) data period at the start of each frame.

The function and use of the serial clock SCLK is different for each format:

- For Microwire, both data sources switch (change to the next bit) outgoing data on the falling edge of SCLK, and sample incoming data on the rising edge.
- For SSP, data sources switch outgoing data on the rising edge of SCLK, and sample incoming data on the falling edge
- SPI gives the user the choice of which edge of SCLK to use for switching outgoing data, and for sampling incoming data. In addition, the user can move the phase of SCLK, shifting it one-half period earlier or later.

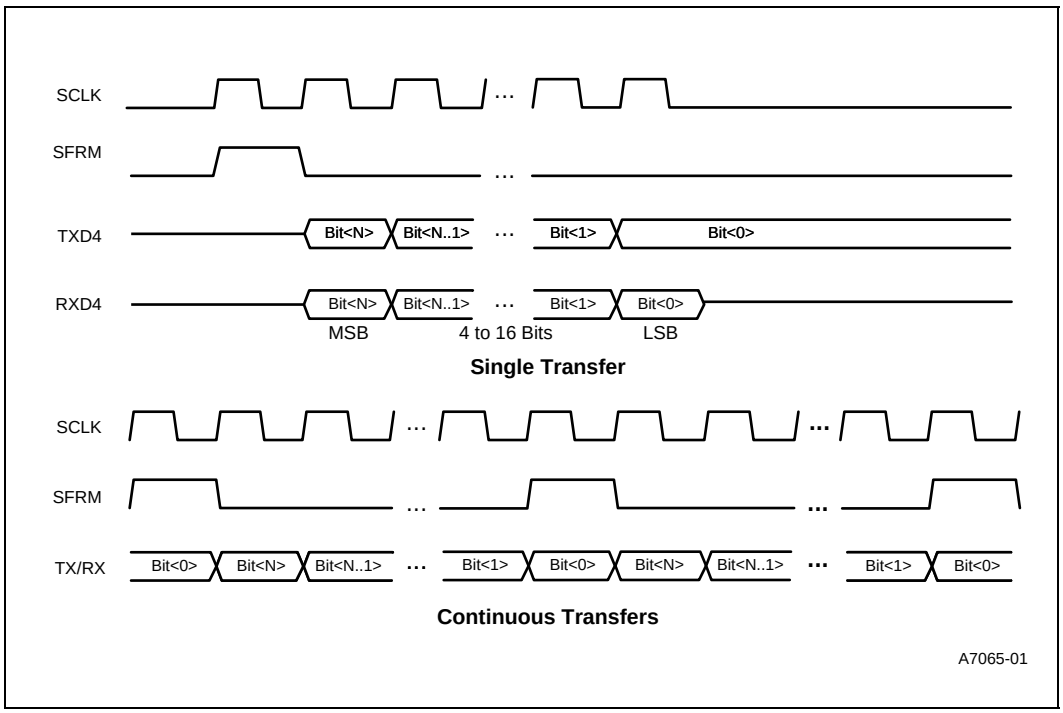
While SSP and SPI are full-duplex protocols, Microwire uses a half-duplex master-slave messaging protocol. At the start of a frame, a single-byte control message is transmitted from the controller to the peripheral; no data is sent by the peripheral. The peripheral interprets the message and responds with requested data, one clock after the last bit of the requesting message. Return data - part of the same frame - can be from 4 to 16 bits in length. Total frame length is 13 to 25 bits.

Note that the serial clock (SCLK), if driven by the SSP Port, only toggles while active data transfer is underway. At other times it is held in “inactive” state, as defined by the specified protocol under which it operates.

8.3.1.1 SSP Format - Detail

When outgoing data in the SSP controller is ready to transmit, SFRM asserts for one clock period. On the following clock, data to be transmitted is driven on TXD a bit at a time, with the most significant bit first. Similarly, the peripheral drives data on the RXD pin. Word length may be from 4 to 16 bits. All transitions take place on the rising edge of SCLK and data sampling is done on the falling edge. At the end of the transfer, TXD retains the value of the last bit sent (bit 0) through the next idle period. If the SSP Port is disabled or reset, TXD goes low.

Figure 8-1. SSP Synchronous Serial Frame Format



8.3.1.2 SPI Format - Detail

There are four possible sub-modes, depending on the edges selected for switching driven data and for sampling received data, and the selection of the phase of SFRM (see Section 8.6.2.4 and Section 8.6.2.5 for complete description of each mode).

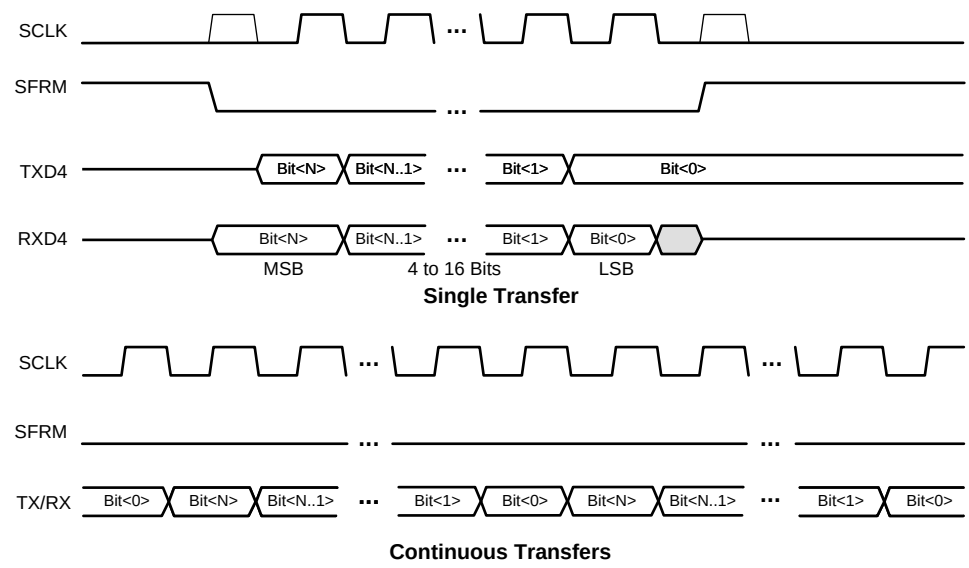
For SPI format, SCLK and TXD are low, and SFRM is high, in idle mode. When transmit (outgoing) data is ready, SFRM goes low and stays low for the remainder of the frame. The MSB of serial data is driven onto TXD a half-cycle later, and halfway into the first bit period SCLK asserts high and continues toggling for the remaining data bits. Data transitions on the falling edge of SCLK. From 4 to 16 bits may be transferred per frame.

With the assertion of SFRM, receive data is simultaneously driven from the peripheral on RXD, MSB first. Data transitions on SCLK falling edges, and is sampled by the controller on rising edges.

At the end of the frame, SFRM is deasserted high one clock period after the last bit has been latched at its destination, and the completed incoming word is shifted into the “incoming” FIFO. The peripheral can tristate RXD after sending the last (LSB) bit of the frame. TXD retains the last value transmitted when the controller goes into idle mode, unless the SSP port is disabled or reset (which forces TXD to zero).

For back-to-back transfers, start and completion are like those of a single transfer, but SFRM does not deassert between words. Both transmitter and receiver know the word length, and keep track internally of the start and end of words (frames). There are no “dead” bits; the least significant bit of one frame is followed immediately by the most significant bit of the next.

Figure 8-2. SPI Frame Format



Note:
The phase and polarity **SCLK** can be configured for four differnet modes. This example shows just one of those modes.

A7062-01

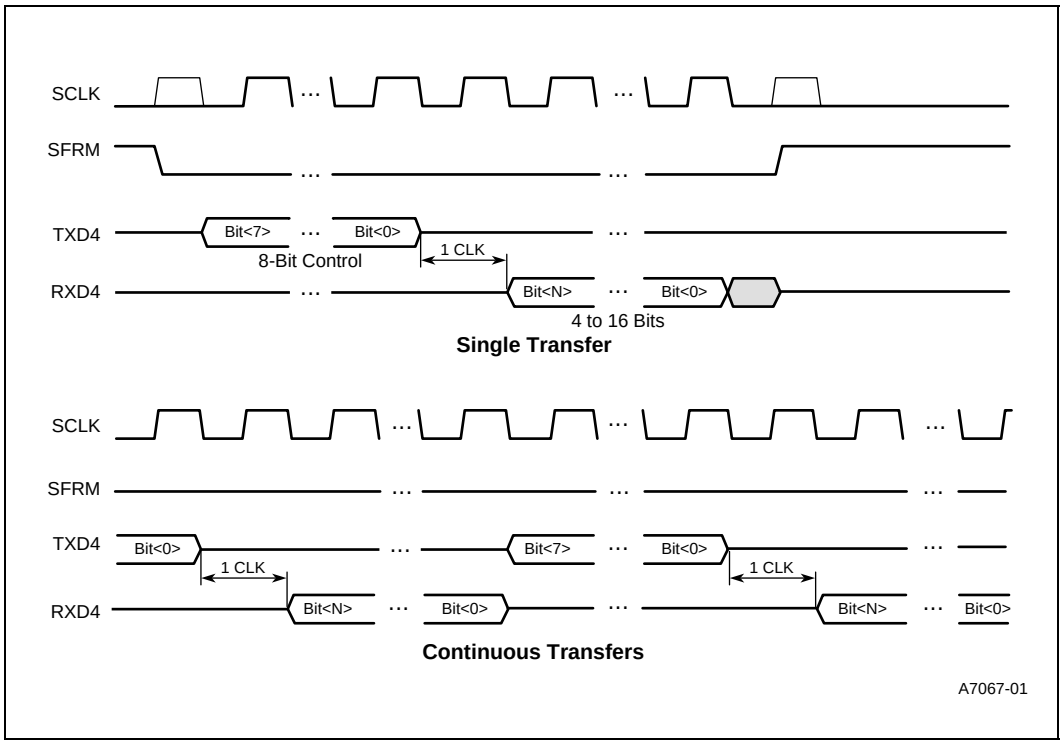
8.3.1.3 Microwire* Format - details

Format is similar to SPI, except transmission is half- instead of full-duplex, using master-slave message-passing.

Idle state has SCLK low and SFRM high. Each serial transmission begins with SFRM assertion (low), followed by an 8-bit command word sent from controller to peripheral on TXD. RXD, controlled by the peripheral, remains tristated. SCLK goes high midway into the command MSB and continues toggling at the bit rate. One bit-period after the last command bit, the peripheral returns the serial data requested, MSB first, on RXD. Data transitions on falling edge of SCLK and is sampled on the rising edge. The last falling edge of SCLK coincides with the end of the last data bit (LSB) on RXD, and it remains low after that (if it is the only, or last word, of the transfer). SFRM deasserts high one-half clock period later.

The start and end of a series of back-to-back transfers are like those of a single transfer; however SFRM remains asserted (low) throughout the transfer. The end of a data word on RXD is followed immediately by the start of the next command byte on TXD, with no “dead” time.

Figure 8-3. National Microwire* Frame Format



8.3.2 Parallel Data Formats for Buffer Storage

Data in the buffers is always stored with one sample per 16-bit word regardless of the format’s data word length. Within each 16-bit field, the stored data sample is right-justified, with the word’s LSB in bit 0, and unused bits packed as zeroes on the left-hand (MSB) side.

8.4 Buffer Operation

There are two separate and independent buffers for “incoming” (from peripheral) and “outgoing” (to peripheral) serial data. Buffers are filled or emptied by an SRAM-like burst initiated by the system processor. Bursts may be from one to eight 32-bit words in length, thus transferring up to eight data samples per burst.

Although the system bus is 32 bits wide, single samples may be transferred. Only lower half two bytes of 32 bits word have valid data and upper half two bytes are not used by the SSP Controller.

Each buffer consists of a dual-port SRAM with control circuitry to make it work as a FIFO, with independent read and write ports.

Buffer filling and emptying may be performed by the system processor in response to an interrupt from FIFO logic. Each FIFO has a programmable threshold at which an interrupt is triggered. When the threshold value is exceeded, an interrupt is generated which, if enabled, signals the host processor to empty an “inbound” FIFO or to refill an “outbound” FIFO. See Section 8.6.2 for a description of the threshold and how to set it in the SSP Control Register.

The system can also poll status bits to learn how full a FIFO is.

8.5 Baud-Rate Generation

The baud (or bit-rate clock) is generated internally by dividing the standard input clock (3.6864 MHz) with a programmable divider. This feeds a programmable divider to generate baud rates from 7.2 KHz to 1.8432 Mbits per second.

8.6 SSP Serial Port Registers

There are four registers in the SSP block: two control, one data, and one status register.

- Control registers are used to program the baud rate, data length, frame format, data transfer mechanism, and port enabling. In addition, they permit setting the FIFO “fullness” threshold that triggers an interrupt.
- The Data Register is mapped as a block of eight (32-bit) locations, which physically points to either of two 32-bit registers. One register is for writes, and transfers data to the Transmit FIFO; the other is for reads, and takes data from the Receive FIFO. A write cycle, or burst write, loads successive words into the SSP Write Register, from the lower half two bytes of a 32-bit word to the Transmit FIFO. A read cycle, or burst read, similarly takes data from the SSP Read Register, and the Receive FIFO reloads it with available data bits it has stored. As in write operations, only the lower two bytes are valid.

  Although a read or write burst increments its address, the three address LSBs are ignored by SSP control logic; all accesses to the eight-word block access the Read or Write Register. The eight-location block is aligned to an eight-word boundary. Bursts from the processor will not cross the boundary.

  The FIFOs are independent buffers to allow full duplex operation.
- The Status Register signals the state of the FIFO buffers: whether the programmable threshold has been passed (Transmit/Receive Buffer service request), and a value showing the actual “fullness” of the FIFO. Flags bits indicate when the SSP is actively transmitting data, when the Transmit Buffer is not full, and when the Receive Buffer is not empty. Error bits signal overrun errors.

8.6.1 SSP Control Register 0

The SSP control register 0 (SSPCR0) contains four different bit fields which control various functions within the SSP.

8.6.1.1 Data Size Select (DSS)

The four-bit data size select (DSS) field is used to select the size of the data transmitted and received by the SSP. Data can be 4 to 16 bits in length. When data is programmed to be less than 16 bits, received data is automatically right-justified and the upper bits in the receive FIFO are zero-filled by receive logic. Transmit data must be right-justified by the user before being placed in the transmit FIFO; however the upper unused bits are ignored by the SSP's transmit logic. Although it is possible to program data sizes of one, two, and three bits, these sizes are reserved and produce unpredictable results in the SSP.

When National Microwire frame format is selected, this bit field selects the size of the received data. Note that the size of the transmitted data is always eight bits in this mode.

8.6.1.2 Frame Format (FRF)

The two-bit frame format (FRF) field is used to select which frame format to use: Motorola SPI (FRF=00), Texas Instruments synchronous serial (FRF=01), or National Microwire (FRF=10). Note that FRF=11 is reserved and produces unpredictable results.

8.6.1.3 Synchronous Serial Port Enable (SSE)

The SSP enable (SSE) bit is used to enable and disable all SSP operation. When SSE=0 the SSP is disabled; when SSE=1 the SSP is enabled. All of its clocks are powered down to minimize power consumption when the SSP is disabled. Note that SSE is the only control bit within the SSP which is reset to a known state. It is cleared to zero to ensure the SSP is disabled following a reset.

When the SSE bit is cleared during active operation, the SSP is disabled immediately, causing the current frame which is being transmitted to be terminated. Clearing SSE also resets the SSP's FIFOs. However the SSP's control and status registers are not reset. The user must ensure these registers are properly reconfigured before re-enabling the SSP.

When the SSP is disabled, its four pins may be used as GPIOs (GPIO block C, bits 4-7). They may be configured as inputs or outputs by control registers described in Chapter 10. In *Sleep* mode, the state of the pins is controlled by the GPIO sleep register; SSP register settings have no effect.

8.6.1.4 Serial Clock Rate (SCR)

The eight-bit serial clock rate (SCR) bit-field is used to select the baud, or bit rate, of the SSP. A total of 256 different bit rates can be selected, ranging from a minimum of 7.2 Kbps to a maximum of 1.8432 Mbps. The serial clock generator uses the 3.6864 MHz clock produced by the on-chip PLL (143.7696 divided by 39) divided by a fixed value of four, and then the programmable SCR value to generate the serial clock (SCLK). The resultant clock is driven on the SCLK pin and is used by the SSP's transmit logic to drive data on the TXD pin, and to latch data on the RXD pin. Depending on the frame format selected, each transmitted bit is driven on either the rising or falling edge of SCLK, and is sampled on the opposite clock edge.

The following table shows the bit locations corresponding to the four different control bit fields within SSP control register 0. Note that the SSE bit is the only control bit which is reset to a known state, to ensure the SSP is disabled following a reset. The reset state of all other control bits is unknown and must be initialized before enabling the SSP. Also note that writes to bit 6 are ignored and reads return zero.

Table 8-3. SSP Control Register 0 Bit Descriptions

0x0000800 — SSPCR0 — SA-1111 Companion Chip

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																SCR								SSP_EN	Reserved	FRF		DSS			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Type	Description
3:0	DSS (Data Size Select)		0000 - Reserved, undefined operation 0001 - Reserved, undefined operation 0010 - Reserved, undefined operation 0011 - 4-bit data 0100 - 5-bit data 0101 - 6-bit data 0110 - 7-bit data 0111 - 8-bit data 1000 - 9-bit data 1001 - 10-bit data 1010 - 11-bit data 1011 - 12-bit data 1100 - 13-bit data 1101 - 14-bit data 1110 - 15-bit data 1111 - 16-bit data
5:4	FRF (Frame Format)		00 - Motorola SPI 01 - Ti Synchronous Serial (SSP) 10 - National Microwire 11 - Reserved, undefined operation
6	—	—	Reserved
7	SSP_EN		Synchronous Serial Port Enable: 0 - SSP operation disabled (pins may function as GPIOs) 1 - SSP operation enabled
15:8	SCR (Serial Clock Rate)		Value (0 to 255) used to generate transmission rate of SSP. Bit rate = 3.6864×10^6 / 2x(SCR+1) where SCR is decimal integer
31:16	—	—	Reserved

8.6.2 SSP Control Register 1

The SSP Control Register 1 (SSPCR1) contains six bit fields which control various SSP functions.

8.6.2.1 Receive FIFO Interrupt Mask (RIM)

The Receive FIFO Interrupt Mask (RIM) bit is used to mask or enable the Receive FIFO service request interrupt. When RIM=0, the interrupt is masked, and the state of the Receive FIFO Service Request (RFS) bit within the SSP Status Register is ignored by the interrupt controller. When

RIM=1, the interrupt is enabled, and whenever RFS is set (one) an interrupt request is made to the interrupt controller. Note that programming RIM=0 does not affect the current state of RFS or the receive FIFO logic's ability to set and clear RFS, it only blocks the generation of the interrupt request.

8.6.2.2 Transmit FIFO Interrupt Mask (TIM)

The Transmit FIFO Interrupt Mask (TIM) bit is used to mask or enable the transmit FIFO service request interrupt. When TIM=0, the interrupt is masked and the state of the Transmit FIFO Service Request (TFS) bit within the SSP Status Register is ignored by the interrupt controller. When TIM=1, the interrupt is enabled, and whenever TFS is set (one) an interrupt request is made to the interrupt controller. Note that programming TIM=0 does not affect the current state of TFS or the transmit FIFO logic's ability to set and clear TFS, it only blocks the generation of the interrupt request

8.6.2.3 Loop Back Mode (LBM)

The loop back mode (LBM) bit is used to enable and disable the ability of the SSP transmit and receive logic to communicate internally. This function is for test purposes only. When LBM=0, the SSP operates normally. The transmit and receive data paths are independent and communicate via their respective pins. When LBM=1, the output of the transmit serial shifter is directly connected to the input of the receive serial shifter internally.

8.6.2.4 Serial Clock Polarity (SPO)

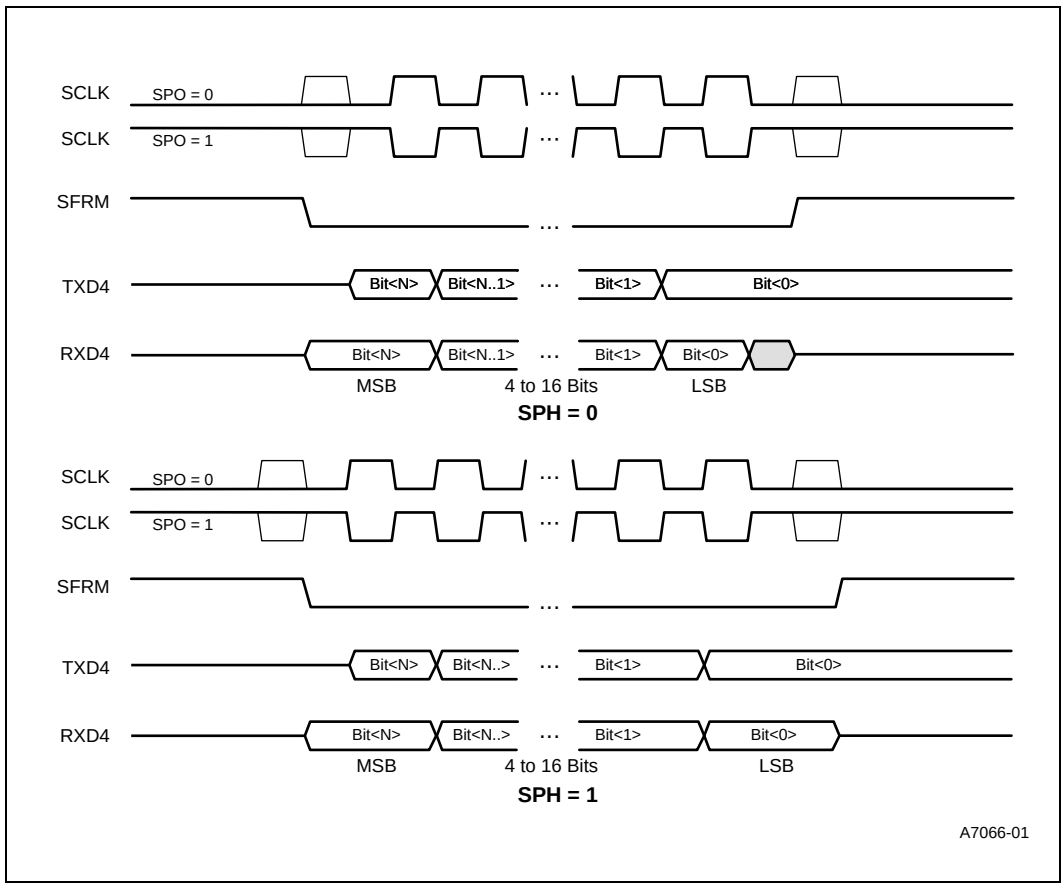
The serial clock polarity (SPO) bit selects the polarity or active/inactive state of the serial clock (SCLK) pin when Motorola SPI format is selected (FRF=00). When SPO=0, the inactive or idle state of SCLK is low. Thus when the SSP is not actively transmitting/receiving data, the SCLK pin is held low. When SPO=1, the inactive or idle state of SCLK is high. Thus when the SSP is not actively transmitting/receiving data, the SCLK pin is held high. The programming of SPO alone does not determine which SCLK edges are used to drive and latch data to or from the transmit and receive pins. The programming of SPO and the serial clock phase (SPH) bit determines this. Note that SPO is ignored in all other modes except Motorola SPI format (FRF=0).

8.6.2.5 Serial Clock Phase (SPH)

The serial clock phase (SPH) bit selects the phase relationship of the serial clock (SCLK) signal with the serial frame (SFRM) signal when Motorola SPI format is selected (FRF=00). When SPH=0, SCLK remains in its inactive state (as programmed by SPO) for one full SCLK period duration after SFRM is asserted (drive low). SCLK continues to transition during the entire frame and is driven to its inactive state one-half SCLK period duration before SFRM is negated (driven high). When SPH=1, SCLK remains in its inactive state (as programmed by SPO) for one-half SCLK period duration after SFRM is asserted (drive low). SCLK continues to transition during the entire frame and is driven to its inactive state one full SCLK period duration before SFRM is negated (driven high). Using SPH and SPO together determine when SCLK is active during the assertion of SFRM and which edge of SCLK is used to drive data to the transmit pin as well as latch data from the receive pin. When SPO and SPH are the same value (both 0 or both 1), transmit data is driven on the falling edge of SCLK and receive data is latched on the rising edge of SCLK. Alternatively, when SPO and SPH are of opposite value (one 0 and the other 1), transmit data is driven on the rising edge of SCLK and receive data is latched on the falling edge of SCLK. Note that SPH is ignored in all other modes, except Motorola SPI format (FRF=00).

Figure 8-4 shows the pin timing for all four programming combinations of SPO and SPH. Note that SPO inverts the polarity of the SCLK signal, and SPH determines the phase relationship between SCLK and SFRM, shifting the SCLK signal one-half phase relationship between SCLK and SFRM, shifting the SCLK signal one-half phase to the left or right during the assertion of SFRM.

Figure 8-4. Clock Phase Formats for SPO and SPH Programming



8.6.2.6 Transmit FIFO Interrupt Threshold (TFT)

This 4-bit value sets the level at (or below) which the FIFO controller triggers a service interrupt. The value is programmed to the threshold value minus one. The reset value is 7.

8.6.2.7 Receive FIFO Interrupt Threshold (RFT)

This 4-bit value sets the level at (or above) which the FIFO controller triggers a service interrupt. The value is programmed to the threshold value minus one. The reset value is 7.

Note: The following table shows bit locations corresponding to control bit fields in SSP Control Register 1. The reset state of all bits is undefined, and they must be initialized before enabling the SSP. Note that writes to reserved bits are ignored, and reads of these bits return zero.

Table 8-4. SSP Control Register 1 Bit Descriptions

0x0000804	SSPCR1	SA-1111 Companion Chip

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																	RFT				TFT				Reserved		SPH	SPO	LBM	Reserved	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Type	Description
1:0	—	—	Reserved
2	LBM (Loop-Back Mode)	RW	0 - Normal serial port operation enabled 1 - Output of transmit serial shifter connected to input of receive serial shifter, internally
3	SPO	RW	Serial clock polarity[a]. 0 – The inactive or idle state of SCLK is low. 1 – The inactive or idle state of SCLK is high.
4	SPH	RW	
6:5	—	—	Reserved
10:7	TFT (Transmit FIFO Threshold)	RW	Sets threshold level at which Transmit FIFO asserts interrupt –defaults to 0111. This level should be set to the threshold value minus 1.
14:11	RFT (Receive FIFO Threshold)	RW	Sets threshold level at which Receive FIFO asserts interrupt–defaults to 0111. This level should be set to the threshold value minus 1.
31:15	—	—	Reserved

a. For additional information about SPI formats, see the Motorola website.

8.6.3 SSP Data Register (SSPDR)

The SSP Data Register is a block of 16-bit locations that can be accessed by 32-bit data transfers (actually uses only the lower 16 bits). Transfers can be from one to eight words. The address block represents two physical registers: the first is temporary storage for data on its way out through the Transmit FIFO, the other is temporary storage for data coming in through the Receive FIFO.

As the register is accessed by the system, FIFO control logic transfers data automatically between register and FIFO as fast as the system moves it. Data in the FIFO shifts up or down to accommodate the new word (unless it's an attempted write to a full Transmit FIFO). Status bits are available to show the system whether either buffer is full, above or below a programmable threshold, or empty.

For outbound data transfers (write from system to SSP peripheral), the register may be loaded (written) by the system processor anytime it is empty.

When a data size of less than 16-bits is selected, the user should right-justify data written to the transmit FIFO. Transmit logic ignores any unused bits. Received data less than 16-bits is automatically right-justified in the receive buffer. When the SSP is programmed for National Microwire frame format, the default size for transmit data is eight-bits (the most significant byte is ignored), the receive data size is controlled by the programmer.

The following table shows the location of the SSP data register. Note that both FIFOs are cleared when the block is reset, or by writing a zero to SSE (SSP disabled).

Table 8-5. SSP Data Register Bit Descriptions

0x0000840 to 0x000087C SSPDWR SSPDRR SA-1111 Companion Chip

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																DATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Type	Description
15:0	DATA (Low Word)	WO/RO	Data word to be written to/read from Transmit/Receive FIFO.
31:16	—	—	Reserved.

8.6.4 SSP Status Register

The SSP status register (SSPSR) contains bits which signal overrun errors as well as the transmit and receive FIFO service requests. Each of these hardware-detected events signal an interrupt request to the interrupt controller. The status register also contains flags that indicates when the SSP is actively transmitting characters, when the transmit FIFO is not full, and when the receive FIFO is not empty (no interrupt generated).

Bits which cause an interrupt signal the interrupt request as long as the bit is set. Once the bit is cleared, the interrupt is cleared. Read/write bits are called status bits, read-only bits are called flags. Status bits are referred to as “sticky” (once set by hardware, must be cleared by software). Writing a one to a sticky status bit clears it, writing a zero has no effect. Read-only flags are set and cleared by hardware; writes have no effect. Additionally some bits which cause interrupts have corresponding mask bits in the interrupt controller registers and are indicated in the section headings that follow.

8.6.4.1 Transmit FIFO Not Full Flag (TNF) (read-only, non-interruptible)

The transmit FIFO not full flag (TNF) is a read-only bit which is set whenever the transmit FIFO contains one or more entries which do not contain valid data. TNF is cleared when the FIFO is completely full. This bit can be polled when using programmed I/O to fill the transmit FIFO over its half-way mark. This bit does not request an interrupt.

8.6.4.2 Receive FIFO Not Empty Flag (RNE) (read-only, non-interruptible)

The receive FIFO not empty flag (RNE) is a read-only bit which is set when ever the receive FIFO contains one or more entries of valid data and is cleared when it no longer contains any valid data. This bit can be polled when using programmed I/O to remove remaining bytes of data from the receive FIFO since CPU interrupt requests are only made when four or more bytes reside within the FIFO (three, two, or one bytes may remain at the end of a frame). This bit does not request an interrupt.

8.6.4.3 SSP Busy Flag (BSY) (read-only, non-interruptible)

The SSP busy (BSY) flag is a read-only bit which is set when the SSP is actively transmitting and/or receiving data, and is cleared when the SSP is idle or disabled (SSE=0). This bit does not request an interrupt.

8.6.4.4 Transmit FIFO Service Request Flag (TFS) (read-only, maskable interrupt)

The Transmit FIFO service request flag (TFS) is a read-only bit which is set when the transmit FIFO is equal or below the threshold value and requires service to prevent an underrun. TFS is set whenever the number of valid data entries in the transmit FIFO is equal or below the transmit FIFO threshold value (TFT), and cleared when there are more entries of valid data than the TFT. When the TFS bit is set, an interrupt request is made unless the transmit FIFO interrupt request mask (TIM) bit is cleared. For more information, see Section 11.

8.6.4.5 Receive FIFO Service Request Flag (RFS) (read-only, maskable interrupt)

The receive FIFO service request flag (RFS) is a read-only bit which is set when the receive FIFO is nearly filled and requires service to prevent an overrun. RFS is set whenever the number of the valid data entries in the receiver FIFO is equal or greater than the receive FIFO threshold value (RFT), and cleared when there are less entries of valid data than the threshold (RFT). When the RFS bit is set, an interrupt request is made unless the receive FIFO interrupt request mask bit is cleared.

8.6.4.6 Receiver Overrun Status (ROR)(read/write, non-maskable interrupt)

The receiver overrun status bit (ROR) is a read/write bit which is set when the receive logic attempts to place data into the receive FIFO after it has been completely filled. Each time a new piece of data is received, the set signal to the ROR bit is asserted, and the newly received data is discarded. This process is repeated for each new piece of data received until at least one empty FIFO entry exists. When the ROR bit is set, an interrupt request is made. Writing a one to this bit resets ROR status and its interrupt request.

8.6.4.7 Transmit FIFO Level

This 4-bit value shows how many valid entries are currently in the Transmit FIFO.

8.6.4.8 Receive FIFO Level

This 4-bit value shows how many valid entries are currently in the receive FIFO.

The following table shows the bit locations corresponding to the status and flag bits within the SSP status register. All bits are read-only except ROR which is read/write. Writes to TNF, RNE, BSY, TFS, and RFS have no effect. The reset state of ROR is unknown and must be initialized before enabling the SSP. Note that writes to reserved bits are ignored and reads to those bits return zeros.

Table 8-6. SSP Status Register Bit Descriptions

0x0000810 SSPSR Status SA-1111 Companion Chip

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																RFL				TFL				ROR	RFS	TFS	BSY	RNE	TNF	Reserved	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Type	Description
1:0	—	—	Reserved
2	TNF - Transmit FIFO not full	RO	0 - Transmit FIFO is full 1 - Transmit FIFO is not full
3	RNE - Receive FIFO not empty	RO	0 - Receive FIFO is empty 1 - Receive FIFO is not empty
4	BSY - SSP Busy	RO	0 - SSP is idle or disabled 1 - SSP currently transmitting or receiving a frame
5	TFS - Transmit FIFO Service Request	RO	0 - Transmit FIFO level exceeds TFL threshold, or SSP disabled 1 - Transmit FIFO level is at or below TFL threshold, request interrupt
6	RFS - Receive FIFO Service Request	RO	0 - Receive FIFO level exceeds RFL threshold, or SSP disabled 1 - Receive FIFO level is at or below RFL threshold, request interrupt
7	ROR - Receive FIFO Overrun	WO/RO	0 - Receive FIFO has not experienced an overrun 1 - Attempted data write to full Receive FIFO, request interrupt
11:8	TFL - Transmit FIFO Level	RO	Number of entries in Transmit FIFO
15:12	RFL - Receive FIFO Level	RO	Number of entries in Receive FIFO
31:16	—	—	Reserved

8.6.5 SSP Interrupt Test Register (SSPITR)

Writing a "1" to the corresponding bit position in the SSP Interrupt Test register generates an interrupt strobe signal to the Interrupt Controller for test purposes. All bit symbols come from the SSP status register. All bit locations correspond to the status and flag bits within the SSP interrupt test register.

Table 8-7. SSPITR Bit Descriptions

0x0000814 — SSPITR Status — SA-1111 Companion Chip

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																								ROR	RFS	TFS	Reserved				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Type	Description
4:0	—	—	Reserved
2	TFS	WO	Transmit FIFO service request 1 = Force Interrupt strobe. 0 = No effect.
3	RFS	WO	Receive FIFO service request 1 = Force Interrupt strobe. 0 = No effect.
4	ROR	WO	Receive FIFO overrun 1 = Force Interrupt strobe. 0 = No effect.
31:8	—	—	Reserved

8.6.6 SSP Register Address Map

The following table shows the registers associated with the SSP and the physical addresses used to access them.

Table 8-8. SSP Register Address Map

Address	Mnemonic	Full Name
0x0000800	SSPCR0	SSP Control Register 0
0x0000804	SSPCR1	SSP Control Register 1
0x0000808	—	Reserved
0x000080C	—	Reserved
0x0000810	SSPSR	SSP Status Register
0x0000814	SSPITR	SSP Interrupt Test Register
0x0000840 to 0x000087C	SSPDWR (Write) SSPDRR (Read))	SSP Data Write Register or SSP Data Read Register

PS/2 Trackpad and Mouse Interfaces 9

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, using 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 (when in PS/2 mode of operation).

All four pins also function as GPIOs, being multiplexed with the I/O of an on-chip GPIO block. This chapter limits its scope to description of PS/2 operation. The two PS/2 blocks are identical, and so are their signal pairs at the chip pins.

9.1 Registers

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

9.1.1 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.

Table 9-1. KBDCR Bit Descriptions

0x0000A00, 0x0000C00 — KBDCR — SA-1111 Companion Chip

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																												ENA	Reserved	FKD	FKC
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Type	Description
0	FKC	RW	1 = Force MSCLK and TPCLK pads low regardless of the state of the keyboard FSM.
1	FKD	RW	Force MSDATA and TPDATA pads low. 1 = Force the pads low regardless of the state of the keyboard FSM.
2	—	—	Reserved
3	ENA	RW	Keyboard enable bit. 0 = Disabled. 1 = Enabled.
31:4	—	—	Reserved

9.1.2 Status Register (KBDSTAT)

All the bits with the exception of STP in this register are read-only. Each bit provides status information such as busy state of the FSM, parity of last received data, and so forth. STP bit when set also generates an interrupt. Writing a one to this bit clears the interrupt.

Table 9-2. KBDSTAT Bit Descriptions

0x0000A04, 0x0000C04 — KBDSTAT — SA-1111 Companion Chip

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																							STP	TXE	TXB	RXF	RXB	ENA	RXP	KBD	KBC
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Type	Description
0	KBC	RO	Value on KBCLK pin (after synchronizing and sampling by Div8 clock).
1	KBD	RO	Value on KBDATA pin (after synchronizing).
2	RXP	RO	Parity bit indication for last received data byte (odd parity).
3	ENA	RO	Enable bit. 1 = Function enabled.
4	RXB	RO	Rx Busy. 1 = Currently receiving data.
5	RXF	RO	Rx Full. 1 = RX register full, ready to be read.
6	TXB	RO	Tx Busy. 1 = Currently sending data.
7	TXE	RO	Tx Register empty (interruptible). 0 = Not ready. 1 = Enabled, ready to transmit.
8	STP	WO/RO	Stop bit error (interruptible). 0 = Stop bit received correctly. 1 = Did not receive stop bit at the end of transmission.
31:9	—	—	Reserved

9.1.3 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.

Table 9-3. KBDDATA Bit Descriptions

0x0000A08, 0x0000C08 — KBDDATA — SA-1111 Companion Chip

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																								KBDAT							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Type	Description
7:0	KBDAT	WO/RO	Tx and Rx data.
31:8	—	—	Reserved

9.1.4 Clock Division Register (KBDCLKDIV)

This register is used to define the divide ratio to generate an 1 MHz sampling clock. The KBCLK signal coming from the PS/2 device is latched by the sampling clock before using internally. This helps removed problems caused by external noise and slow moving edges.

Table 9-4. KBDCLKDIV Bit Descriptions

0x0000A0C, 0x0000C0C — KBDCLKDIV — SA-1111 Companion Chip

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																														DivVal	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Type	Description
1:0	DivVal	RW	Divide value. 00 = Divided by 8. Incoming KbdClk is 8 MHz. 01 = Divided by 4. Incoming KbdClk is 4 MHz. 10 = Divided by 2. Incoming KbdClk is 2 MHz. 11 = Reserved.
31:2	—	—	Reserved

9.1.5 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.

Table 9-5. KBDPRECNT Bit Descriptions

0x0000A10, 0x0000C10 — KBDPRECNT — SA-1111 Companion Chip

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																								PreCntMax							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Type	Description
7:0	PreCntMax	RW	Maximum precount value. The division ratio is given by the following equation: Divide ratio = (PreCntMax+ 1) * 2.
31:8	—	—	Reserved

9.1.6 KBD Interrupt Test Register (KBDITR)

This is the KBD Interrupt Test register (KBDITR). Writing a "1" to the corresponding bit position in this register generates an interrupt strobe signal to the Interrupt Controller for test purposes. All bit symbols come from the KBDSTAT status register.

Table 9-6. KBDITR Bit Descriptions (Sheet 1 of 2)

0x0000A14, 0x0000C14 — KBDITR — SA-1111 Companion Chip

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																							STP	TXE	Reserved	RXF	Reserved				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Type	Description
4:0	—	—	Reserved
5	RXF	WO	Rx full. 1 = Force interrupt strobe. 0 = No effect.
6	—	—	Reserved
7	TXE	WO	Tx Register empty 1 = Force interrupt strobe. 0 = No effect.

Table 9-6. KBDITR Bit Descriptions (Sheet 2 of 2)

0x0000A14, 0x0000C14 — KBDITR — SA-1111 Companion Chip

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																							STP	TXE	Reserved	RXF	Reserved				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Type	Description
8	STP	WO	Stop bit error. 1 = Force interrupt strobe. 0 = No effect.
31:9	—	—	Reserved

9.1.7 Register Memory Map

Table 9-7 and Table 9-8 list the read and write locations in the memory map for the trackpad and mouse interfaces. See Section 3.2.1 for the base addresses of the trackpad and mouse interfaces.

Table 9-7. Trackpad Register Memory Map

Address	Read Location	Write Location
0x0000A00	KBDCR register	KBDCR register
0x0000A04	KBDSTAT register	KBDSTAT register
0x0000A08	KBDDATA register	KBDDATA register
0x0000A0C	KBDCLKDIV register	KBDCLKDIV register
0x0000A10	KBDPRECNT register	KBDPRECNT register
0x0000A14	KBDITR register	KBD Interrupt Test register

Table 9-8. Mouse Register Memory Map

Address	Read Location	Write Location
0x0000C00	KBDCR register	KBDCR register
0x0000C04	KBDSTAT register	KBDSTAT register
0x0000C08	KBDDATA register	KBDDATA register
0x0000C0C	KBDCLKDIV register	KBDCLKDIV register
0x0000C10	KBDPRECNT register	KBDPRECNT register
0x0000C14	KBDITR register	KBD Interrupt Test register

9.2 Functional Description

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

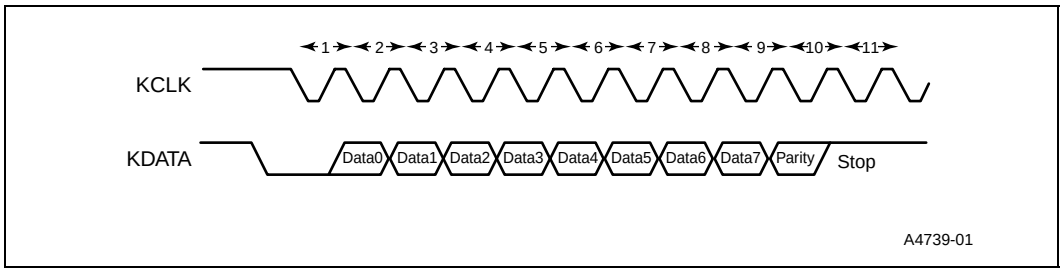
The keyboard interface is held in reset until the enable bit in the control register is set. The interface can be controlled on the basis of the interrupts generated, or by polling the status flags in the control register. The Tx interrupt is generated when the transmit buffer has been emptied and the 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 remove problems caused by external noise and slow moving edges.

The KBCLK signal is always driven by the keyboard, unless the SA-1110 wishes to prevent the peripheral from transmitting (such as when 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 9-1 shows the protocol of this transfer. Note that “receive” refers to the controller end—that is, the external peripheral is transmitting.

Figure 9-1. Keyboard/Mouse Controller Receive Protocol



When the controller wants to transmit a byte to the peripheral, the KBCLK line goes LOW, is 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 onto the KBDATA line. A parity bit is driven out, followed by a stop bit; the stop bit may be acknowledged by the peripheral. (The controller does not check on the acknowledge.) Figure 9-2 shows the timing requirements for the interface.

Figure 9-2. Timing and Controller Request to Send Protocol

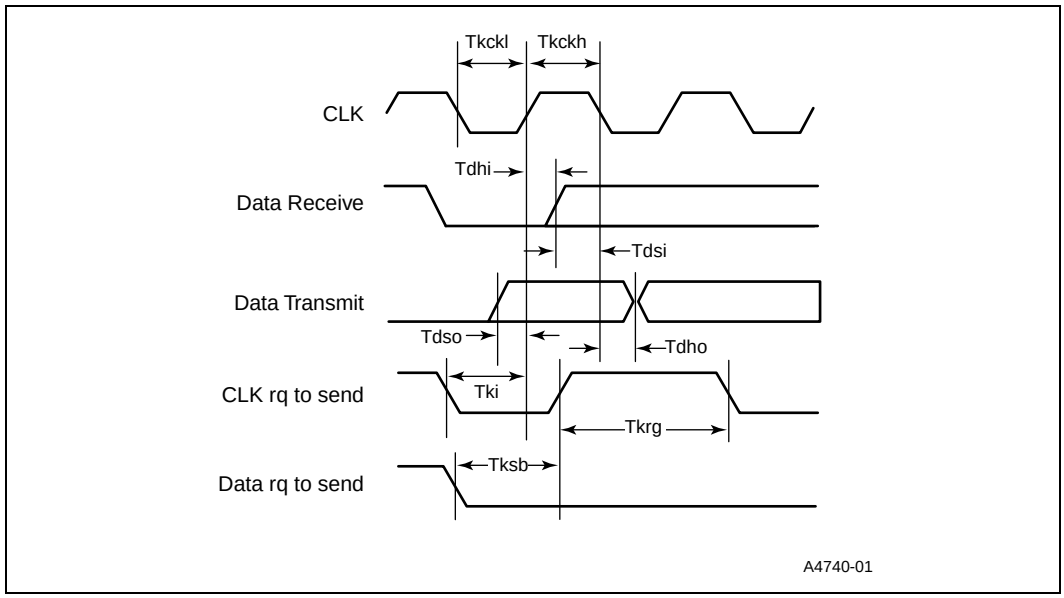


Table 9-9 shows the nominal timings for the interface signals.

Table 9-9. 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 proceeds the CLK so the value for Tksb shown in Figure 9-2 is negative (that is, safe).

General-Purpose I/O Interface 10

The General-Purpose I/O (GPIO) interface is a functional block that provides up to 19 bits of programmable input/output. The 19 bits are distributed across three GPIO units, each unit providing up to eight bits of GPIO. The blocks are designated as GPIO_A, GPIO_B, and GPIO_C. Individual pins are identified within each block - for example, GPIO_A<3> would be bit 3 of block A. Each pin is configurable as either input or output. At system reset, all pins default to *input* mode of operation.

As inputs, each pin can be configured as an interrupt source, wake-up source, or both. Interrupts are masked in hardware if the pins are configured as outputs.

Each GPIO block has a Data Value Register and a Data Direction Register. The Data Direction Register defines whether each individual pin is an input or an output. The Data Value Register is used both to read the level on the GPIO pin, either as an input or as an output, and to set the values of pins that are configured as outputs.

Most GPIO pins have alternate functions, and their use as GPIO or the alternate function is determined by the alternate function. When the “alternate” function is enabled the associated GPIO interrupts are automatically disabled.

Four GPIO_A<3:0> bits are always available as GPIO, but are normally used in “output” mode to control an external PCMCIA or USB power controller. All four PS/2 pins may be optionally used as GPIOs, and serial port I/Os - both the Audio Port and the SSP Port - are available as GPIOs if their “normal” port functions are not used. In addition, the two PWM (D-to-A) outputs are configurable as GPIOs.

10.1 Functional Description

GPIO ports provide general-purpose digital I/O capabilities for the SA-1111. If a pin’s GPIO function is enabled (alternate function is disabled), the direction of the signal (input or output) can be individually specified. If the pin is set to “output”, its value (1 or 0) may then be specified.

The state of enabled GPIO pins also depends on the SA-1111 mode. In *normal* mode, values written to the Data Value Register are asserted on pins whose direction has been set to “output”. The state of each GPIO pin can be determined by reading the Data Value Register. This is true regardless of whether the pin is configured as an input or an output.

In *sleep* mode, the direction of any GPIO may be specified independently of its direction in normal mode. Values in the Sleep Value Register are asserted on pins whose direction is set to “output”. Register values remain after leaving *sleep* state.

As inputs, any GPIO can be additionally configured to be an interrupt source, or a wakeup source, or both. These capabilities are programmed by bits set in the Interrupt Controller’s configuration registers. The interrupt event - rising or falling edge, for example - is also programmable on a per-bit basis.

10.2 GPIO Pin List and Description

Table 10-1 lists the GPIO pins.

Table 10-1. GPIO Pin List

GPIO Bit	Function
GPIO_A<3:0>	4-bit Control Interface to external power controller for PCMCIA, Compact Flash
PWM / GPIO_B<0>	Pulse-width-modulation D-to-A converter 0, or GPIO block B, bit 0
L3MODE/PWM / GPIO_B<1>	Pulse-width-modulation D-to-A converter 1, or GPIO block B, bit 1
TPCLK / GPIO_B<2>	PS/2 Trackpad clock, or GPIO block B, bit 2
TPDATA / GPIO_B<3>	PS/2 Trackpad data, or GPIO block B, bit 3
L3CLK/MSCLK / GPIO_B<4>	L3CLK, or PS/2 Mouse clock, or GPIO block B, bit 4
L3DATA/MSDATA / GPIO_B<5>	L3DATA, or PS/2 Mouse data, or GPIO block B, bit 5
BIT_CLK / GPIO_C<0>	AC-link/I^2S clock, or GPIO block C, bit 0
SYNC / GPIO_C<1>	AC-link/I^2S SYNC (L/R) signal, or GPIO block C, bit 1
SDATA_OUT/ GPIO_C<2>	AC-link/I^2S serial data out, or GPIO block C, bit 2
SDATA_IN / GPIO_C<3>	AC-link/I^2S serial data in, or GPIO block C, bit 3
SCLK / GPIO_C<4>	SSP Serial Port Clock, or GPIO block C, bit 4
SFRM / GPIO_C<5>	SSP Serial Port Frame, or GPIO block C, bit 5
TXD / GPIO_C<6>	SSP Serial Port Data Out, or GPIO block C, bit 6
RXD / GPIO_C<7>	SSP Serial Port Data In, or GPIO block C, bit 7

Note: The GPIO_B<4:5> pins are multiplexed with two other functions, the L3 (see Section 7.1.1.1) and the PS/2 mouse function. If the L3 function is enabled, the L3 function takes precedence over the GPIO_B and the PS/2 mouse functions and controls these pins. If the L3 function is not enabled, the PS/2 mouse function takes precedence over the GPIO_B function. If the L3 and the PS/2 mouse functions are not enabled, the GPIO_B controls these pins.
This same pattern applies to the GPIO_B<1> pin. If the L3 function is enabled, the L3 function takes precedence over the GPIO_B and the PWM functions and controls the pin. If the L3 function is not enabled, the PWM function takes precedence over the GPIO_B<1> pin. If the L3 and the PWM functions are not enabled, the GPIO_B controls these pins.

10.3 Programmer's Model

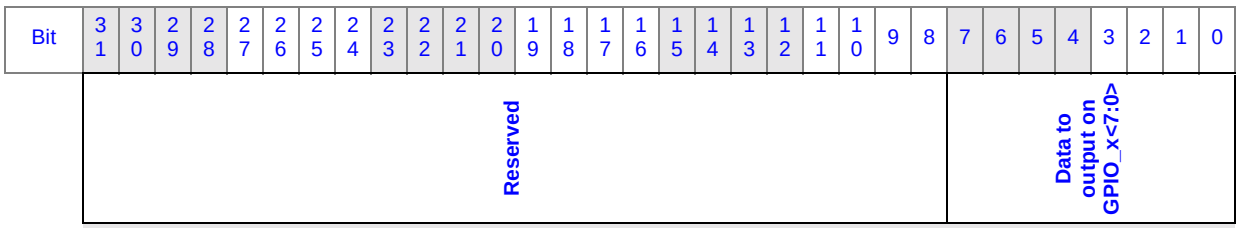
This section describes the registers in the general-purpose I/O interface. The description shows each block as an 8-bit unit, and applies to all GPIO blocks, regardless of the actual number of bits that connect to pins from the block.

10.3.1 Data Value Register (Px_DWR - Write)

Values written to this 8-bit register are output on port x pins if the corresponding data direction bits are set LOW (pin set to “output”). All bits are cleared (set to zero) by a system reset.

Table 10-2. Px_DWR Bit Descriptions

0x0001004, 0x0001014, 0x0001024 Px_DWR SA-1111 Companion Chip



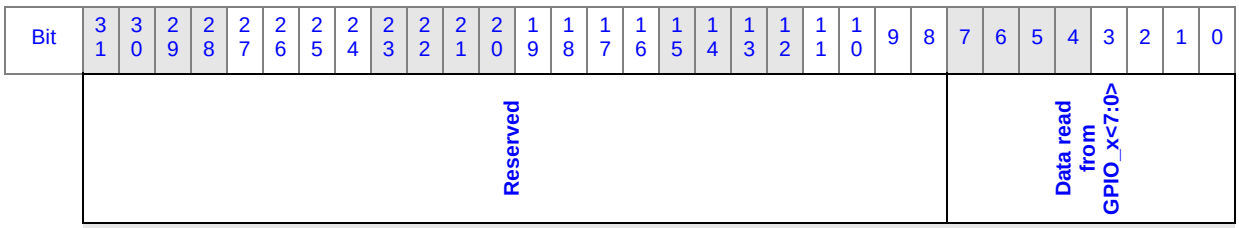
Bits	Name	Type	Description
7:0	Write Data	RW	Data to output on GPIO_x<7:0> (x = A, B, C)
31:9	—		Unused

10.3.2 Data Value Register (Px_DRR - Read)

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

Table 10-3. Px_DRR–Read Bit Descriptions

0x0001004, 0x0001014, 0x0001024 Px_DRR–Read SA-1111 Companion Chip



Bits	Name	Type	Description
7:0	Read Data	RO	Data read from GPIO_x<7:0> (x = A, B, C)
31:9	—	—	Unused

10.3.3 Data Direction Register (Px_DDR)

Bits set in this 8-bit read/write register select the corresponding pin in the GPIO block to become an input; clearing a bit makes the pin an output. All bits are *set* by system reset, so a port is input by default.

Table 10-4. Px_DDR Bit Descriptions

0x0001000, 0x0001010, 0x0001020 — Px_DDR — SA-1111 Companion Chip

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																								Direction of GPIO_x<7:0>							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Type	Description
7:0	Direction	RW	Direction of GPIO_x<7:0> (x = A, B, C): 1 = input 0 = output
31:9	—	—	Unused

10.3.4 Sleep State Register (Px_SSR)

The value in this 8-bit read/write register sets the value of the port during *sleep* mode.

Note: These bits take effect on the state of the relevant pins when the system goes into *sleep* mode regardless of whether the pins are programmed for the GPIO function or for the alternate function.

Table 10-5. Px_SSR Bit Descriptions

0x000100C, 0x000101C, 0x000102C	Px_SSR	SA-1111 Companion Chip

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																								Data to output on GPIO_x<7:0> when in sleep mode							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Type	Description
7:0	Output Data	RW	Data to output on GPIO_x<7:0> when in sleep mode
31:8	—		Unused

10.3.5 Sleep Direction Register (Px_SDR)

This register takes effect only when the system is in sleep mode. Bits set in this 8-bit read/write register select the corresponding pin in the GPIO block to become an input; clearing a bit makes the pin an output. All bits are *set* by system reset, so a port is input by default.

Table 10-6. Px_SDR Bit Descriptions

0x0001008, 0x0001018, 0x0001028	Px_SDR	SA-1111 Companion Chip

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Unused																								Direction of GPIO_x<7:0>							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Type	Description
7:0	Direction	RW	Direction of GPIO_x<7:0> (x = A, B, C): 1 = input 0 = output
31:9	—	—	Unused

10.3.6 Memory Map

Table 10-7 lists the read and write locations in the GPIO memory map. See the Section 3.2.1 for the base address of the GPIO.

Table 10-7. GPIO Register Memory Map

Address	Name	Function
0x0001000	PA_DDR	GPIO Block A Data Direction
0x0001004	PA_DRR (read); PA_DWR (write)	GPIO Block A Data Value Register
0x0001008	PA_SDR	GPIO Block A Sleep Direction
0x000100C	PA_SSR	GPIO Block A Sleep State
0x0001010	PB_DDR	GPIO Block B Data Direction
0x0001014	PB_DRR (read); PB_DWR (write)	GPIO Block B Data Value Register
0x0001018	PB_SDR	GPIO Block B Sleep Direction
0x000101C	PB_SSR	GPIO Block B Sleep State
0x0001020	PC_DDR	GPIO Block C Data Direction
0x0001024	PC_DRR (read); PC_DWR (write)	GPIO Block C Data Value Register
0x0001028	PC_SDR	GPIO Block C Sleep Direction
0x000102C	PC_SSR	GPIO Block C Sleep State

10.4 Test

There are no test registers in the GPIO interface.

Interrupt Controller 11

The interrupt controller generates interrupt signals from the Intel® StrongARM* SA-1111 Microprocessor Companion Chip (SA-1111) in a highly flexible manner. The logic provides full control over the polarity of each bit. The overall interrupt structure is flat for all hardware interrupt sources. The interrupt controller has two independent modes of operation:

- **Normal interrupt:**

  Creates an interrupt signal on the SA-1111's INT output pin, suitable for connection to the SA-1110 interrupt inputs. Full facilities are available for processing the raw interrupt sources including polarity, enabling, setting, and clearing. Some interrupts may be enabled (active) while SA-1111 is in *Sleep* mode.

- **Wake-Up:**

  Generates a wake-up signal on the SA-1111's INT output pin. In normal use the Intel® StrongARM* SA-1110 Microprocessor (SA-1110) would respond by initiating the wake-up sequence in the SA-1111. This signal is the logical OR of all enabled wake-up sources. Wake-up inputs can be enabled and polarity-specified independently of the same signal's Interrupt Enable status.

11.1 Functional Description

This section provides a functional overview of the interrupt controller.

11.1.1 Logic Diagrams

Figure11-1 shows how the INT output is generated from the interrupt source. For low power considerations, a interrupt is detected asynchronously on its rising or falling edge using a D-type flip flop and a gated clock (gRclk) is used for synchronization when reading the interrupt status. Therefore, no free running clock is needed. Based on the polarity of each interrupt, each interrupt is asynchronously latched on the rising edge or falling edge of the raw interrupt source (IntRaw<n>). The latched interrupts (IntLatched<n>) are then logical ORed together to generate the SA-1111 interrupt output (INT). When reading the interrupt status, a gated clock gRclk is provided and the latched interrupts (IntLatched<n>) are doubled synchronized to clock out the interrupt status (IntStatus<n>).

Figure 11-1. Interrupt and Wake-Up Generation

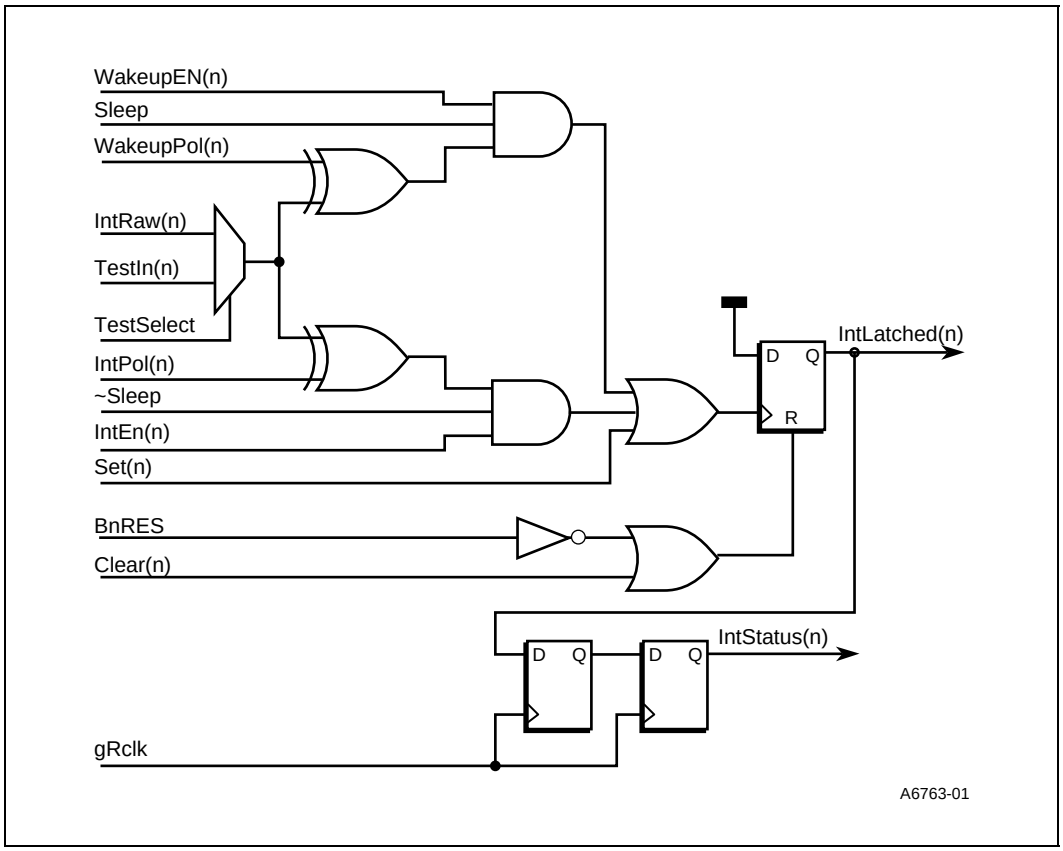
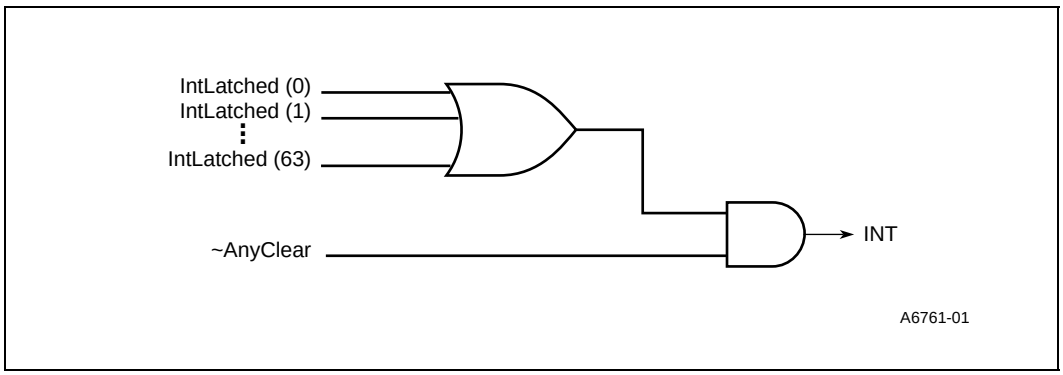


Figure11-2 shows how the output from each bit combination circuit is then combined to generate a single SA-1111 interrupt (INT). Here INT is generated asynchronously. Since the SA-1110 is rising edge sensitive to the interrupt pin INT, any interrupt status clear forces INT to be low as long as AnyClear is asserted. If there are still pending interrupts, INT goes back to high again as soon as AnyClear is deasserted. This results in producing another rising edge on INT if there are pending interrupts.

Figure 11-2. Interrupt and Wake-Up Combination



11.1.2 Application

The interrupt clock must be enabled to operate the interrupt controller in normal mode.

11.2 Interrupt Sources

The interrupt controller handles all interrupt sources throughout the SA-1111. Table 11-1 lists the interrupt sources handled by the interrupt controller.

Table 11-1. Interrupt Sources (Sheet 1 of 2)

Interrupt Number	Name	Source
3:0	GpAInt<3:0>	GPIO_A block inputs (4 total)[a]
9:4	GpBInt<5:0>	GPIO_B block inputs (6 total)[a]
17:10	GpCInt<7:0>	GPIO_C block inputs (8 total)[a]
18	MsTxInt	PS/2 mouse transmit interrupt[b]
19	MsRxInt	PS/2 mouse receive interrupt[b]
20	MsStopErrint	PS/2 mouse stop bit error interrupt[b]
21	TpTxInt	PS/2 trackpad transmit interrupt[b]
22	TpRxInt	PS/2 trackpad receive interrupt[b]
23	TpStopErrint	PS/2 trackpad stop bit error interrupt[b]
24	SspXmtint	Ssp Transmit FIFO empty/under threshold[b]
25	SspRcvint	Ssp Receive FIFO full/over threshold[b]
26	SspROR	Ssp Receive FIFO overrun[b]
27:31	Reserved	—
32	AudXmtDmaDoneA	Audio Transmit DMA Buffer A Done[b]
33	AudRcvDmaDoneA	Audio Receive DMA Buffer A Done[b]
34	AudXmtDmaDoneB	Audio Transmit DMA Buffer B Done[b]
35	AudRcvDmaDoneB	Audio Receive DMA Buffer B Done[b]
36	AudTFSR	Audio Transmit FIFO empty/under threshold[b]
37	AudRFSR	Audio Receive FIFO full/over threshold[b]
38	AudTUR	Audio Transmit FIFO under-run interrupt[b]
39	AudROR	Audio Receive FIFO overrun[b]
40	AudDTS	Audio L3/ACLink Data Sent interrupt[b]
41	AudRDD	Audio L3/ACLink Data Read Done interrupt[b]
42	AudSTO	Audio ACLink read status timeout interrupt[b]
43	USBPwr	USB Controller - power sense input[b]
44	nIrqHciM	USB Controller[b]
45	IrqHciBuffAcc	USB Controller[b]
46	IrqHciRmtWkp	USB Controller[b]
47	nHciMFCir	USB Controller[b]

Table 11-1. Interrupt Sources (Sheet 2 of 2)

Interrupt Number	Name	Source
48	USB port resume	USB Controller[b]
49	S0Readynint	PCMCIA interface[b]
50	S1Readynint	PCMCIA interface[b]
51	S0CDValid	PCMCIA interface[b]
52	S1CDValid	PCMCIA interface[b]
53	S0_Bvd1Stschg	PCMCIA interface[b]
54	S1_Bvd1Stschg	PCMCIA interface[b]
55:63	Reserved	—

a. GPIO interrupts, which are implementation dependant, can have their polarity set to zero (rising) or one (falling).
b. Intel recommends that all non-GPIO interrupts have their polarity set to their reset value of zero (rising).

11.3 Interrupts

When a valid interrupt occurs (of the correct polarity, on an enabled source) a rising-edge signal is generated on the INT pin. Connection of this signal to GPIO 0 or GPIO 1 on the SA-1110 is recommended. If this source is enabled in SA-1110, then an IRQ or FIQ interrupt occurs, depending on the value programmed into the processor's interrupt level register. The called interrupt routine can access the SA-1111's interrupt controller to find the source of the interrupt. See Section 11.5.6 for more information.

When the interrupt has been serviced, the appropriate bit should be cleared in the SA-1110's GPIO edge-detect status register. The SA-1111's INT pin then can be cleared by clearing the appropriate interrupt bit and the corresponding interrupt bit in the peripheral module. See Section 11.5.6 and corresponding peripheral chapters for more details.

11.3.1 GPIO Interrupts

GPIO interrupts are a special case. Virtually all GPIOs share their I/O pins with alternate functions (SSP Serial Port, Audio Controller, etc.). If the alternate function is disabled, the GPIO function enabled, direction set to *input*, and enabled as an interrupt source, passes interrupts on to the INT pin where they can be sensed by the system processor.

If, on the other hand, the same GPIO pin is switched to the alternate function, that pin's interrupt capability is disabled (as a GPIO). This is true regardless of its GPIO interrupt programming.

11.3.2 Wake-Up Interrupts

Most signals available as interrupts may also be used as wake-up initiators, active when the device is in *Sleep* state. They can be enabled and their activating polarity set independently of the Interrupt settings for the same signal.

When an enabled wake-up signal is detected, the logical OR of all potential sources is output on the INT pin. The source of the interrupt is latched within the controller. In response to the wake up interrupt to read the source of the interrupt, gRclk must be restarted and then the latched source is clocked through. See Section 11.5.6 for more information. Once the source is clocked through the interrupt in normal mode is asserted.

Wake-up interrupts are only active when the *Sleep* state has been initiated by software. The other *Sleep*-initiating event, assertion of BAT_FLT, disables all interrupts. On recovery from BAT_FLT induced sleep, the system typically resets the state of the device, including interrupts.

11.3.3 Multiple Interrupts

The SA-1111 has many interrupt sources; in normal operation, multiple interrupt requests occur. The SA-1111's interrupt controller is designed to work in tandem with the SA-1110 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 its level. However, once an interrupt has been serviced and the interrupt cleared internally, the INT pin deasserts in response, and then asserts again to reflect still-pending interrupts. This continues until all interrupts have been serviced.

Once an SA-1111 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.

11.4 Testability

Internally, SA-1111 interrupt controller has 2 sets of test registers. It can generate values for each interrupt sources before and after the programmable polarity logic and enable logic. In addition, each peripheral also has an interrupt test register which can generate values for each of its own interrupt sources under test mode.

11.5 Programmer's Model

This section describes the registers in the interrupt controller.

11.5.1 Registers

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

- INTTEST - Test register
- INTEN - 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.

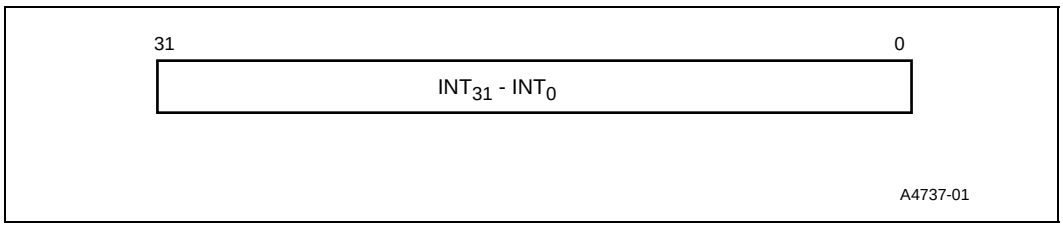
Two additional sets of registers are provided for wake-up signal enabling and polarity.

- WAKE_EN - Individually enable wake-up sources
- WAKE_POL - Specify the polarity of the wake-up source

11.5.1.1 Source Bit Positions

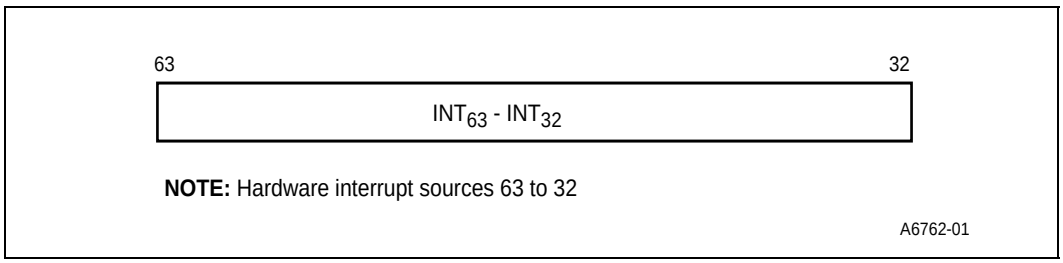
Figure 11-3 shows the bit positions for the interrupt sources in set 0 of the registers and Figure 11-4 shows the interrupt sources in set 1 (see Table 11-1 for valid interrupts).

Figure 11-3. Interrupt Source Positions in Register Set 0



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

Figure 11-4. Interrupt Source Positions in Register Set 1



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

11.5.2 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 11-3 and Figure 11-4.

11.5.3 INTEN0 – INTEN1

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 11-3 and Figure 11-4.

11.5.4 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 11-3 and Figure 11-4.

Note: Intel recommends that all non-GPIO interrupts have their polarity set to their reset value of zero (rising).

11.5.5 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 11-3 and Figure 11-4.

11.5.6 INTSTATCLR0–1

The INSTATCLR0-1 register is a block of four 32-bit locations that can be accessed with 32-bit operations. The address block represents four physical registers. Two are for reads to obtain the status of the interrupt controller, the other two are for writes to clear the status of the interrupt controller.

On read, a one indicates the interrupt is set and a zero indicates the interrupt is clear. Writing a one clears the interrupt, writing a zero does nothing. Interrupts bits correspond to register bits as shown in Figure 11-3 and Figure 11-4.

11.5.7 INTSET0 – INTSET1

These registers allow the interrupt sources to be set. Writing a one sets the interrupt value, writing a zero does 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 11-3 and Figure 11-4.

11.5.8 WAKE_EN0 - WAKE_EN1

These registers individually enable inputs to be specified as wake-up sources. Note that wake-up interrupts are only active during software-induced *Sleep* state (not BAT_FLT-induced).

11.5.9 WAKE_POL0 - WAKE_POL1

These registers enable polarity of wakeup signal to be specified for individual wake-up sources.

11.5.10 Memory Map

Table 11-2 provides a summary of the registers.

Table 11-2. Interrupt Registers

Address Offset	Name	Reset	Description
0x1600	INTTTEST0	Yes	Write 1 = Set test interrupt source high
0x1604	INTTTEST1	Yes	Write 1 = Set test interrupt source high
0x1608	INTEN0	Yes	Write 0 = Disable
0x160C	INTEN1	Yes	Write 0 = Disable
0x1610	INTPOL0	Yes	Write 0 = Active high source
0x1614	INTPOL1	Yes	Write 0 = Active high source
0x1618	INTTSTSEL	Yes	Mode selection
0x161C	INTSTATCLR0-1	NA	Read 1 = interrupt set Write 1= Clear interrupt
0x1624	INTSET0	Yes	Write 1 = Set interrupt
0x1628	INTSET1	Yes	Write 1 = Set interrupt
0x162C	WAKE_EN0	Yes	Write 1 = Enable interrupt
0x1630	WAKE_EN1	Yes	Write 1 = Enable interrupt
0x1634	WAKE_POL0	Yes	Write 1 = Set interrupt polarity = high
0x1638	WAKE_POL1	Yes	Write 1 = Set interrupt polarity = high

11.6 Characteristics

Table 11-3 lists the characteristics for the interrupt controller.

Table 11-3. Interrupt Characteristics

Symbol	Parameter	Minimum	Unit
t_{IL}	Interrupt latency[1]	Cp to Q of a D flip flop plus a few gates	ns
t_{IPW}	Minimum pulse width[2]	1 gRclk	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.

PCMCIA Interface 12

The PCMCIA interface block provides control logic and a complete set of signal buffers for one PCMCIA card and one CF (Compact Flash) card. Alternatively, with an external address buffer it can support two PCMCIA cards. The control logic and built-in buffers eliminate ten or more external “glue” and buffer/transceiver components, providing a highly-integrated and lower power solution for the creation of complete systems.

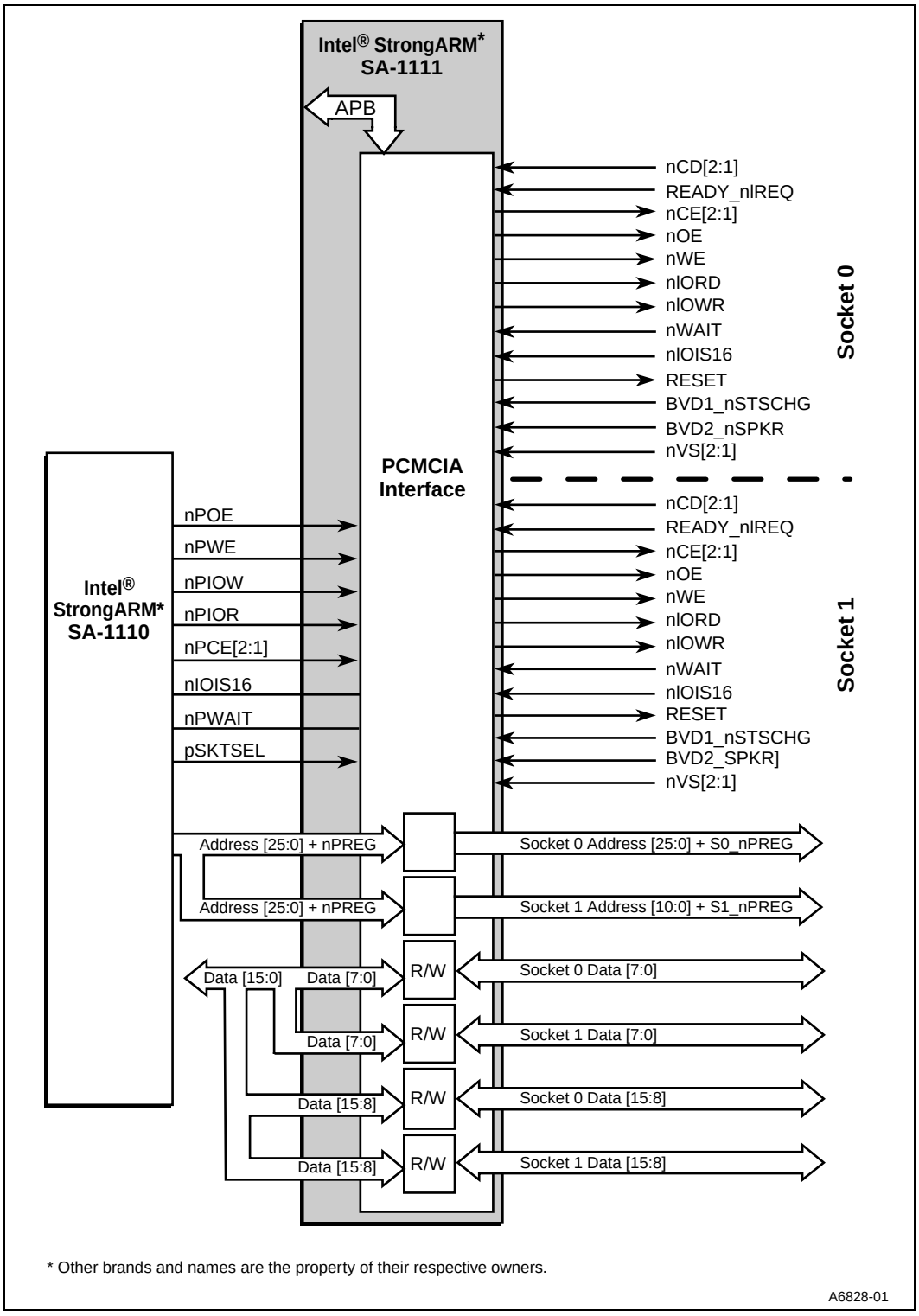
Four GPIOs are available to control an external power-selection and switching device, which supplies power to the card sockets.

The interface also provides facilities for controlling the state of the pins when the Intel® StrongARM* SA-1111 Microprocessor Companion Chip (SA-1111) is in *sleep* mode.

12.1 PCMCIA Interface Block Diagram

Figure 12-1 shows the block diagram for the PCMCIA interface.

Figure 12-1. PCMCIA Interface Block Diagram



12.2 Functional Description

12.2.1 PCMCIA/CF Address and Data Buffers

Buffers are built into the SA-1111 for all address and data signals to one PCMCIA slot and one CF slot. The PCMCIA slot supports both 3.3V and 5.0V PCMCIA cards.

When the device is in *Sleep* mode, the state of PCMCIA and CF address and data pins changes to a programmable value (Hi-Z or logic LOW).

12.2.2 Voltage Control

Four GPIO pins are provided for programming an external voltage source. This allows the interface to accommodate PCMCIA cards of different operating voltages. These pins are controlled by GPIO register writes (see Chapter 10).

12.2.3 Reset Signals

A single logic output is provided for each socket to allow resetting the PCMCIA or CF card. Both signals are active high, and are asserted when the SA-1111 is in *reset*, or the relevant bit in the Control Register is set (refer to Section 12.6.2).

When the SA-1111 is in *sleep* mode the reset signals are either tristated (off) or low, depending on the relevant bit in the sleep state register (refer to Section 12.6.3).

12.2.4 Control Signals

Seven control lines are driven to each socket: nPREG, nOE, nWE, nIOW, nIOR, nCE1, and nCE2. The control lines operate except when the SA-1111:

- Enters sleep mode.
  These lines can be tristated or driven high (refer to Section 12.6.3).
- Sets its float bits in the Control Register (PCCR).
  These control lines float in *normal* mode, when appropriately set via control register bits, and in and *sleep* mode.

Two signals are returned, through the SA-1111, from each card socket to the processor. These are:

- Sx_nIOIS16 (x = 0 or 1) is an acknowledgement from the card that the current address is a valid 16-bit address;
- Sx_nPWAIT (x = 0 or 1) is driven low by the card to extend the length of a transfer.

These pins are inputs on the card interface of the SA-1111; the signal from the selected card is routed to a pin on the SA-1111's processor interface, where it is an output, driven to the processor. Since they are inputs at the card interface, they are unaffected by *Sleep* mode or any register configuration.

12.2.5 Interrupts

Table 12-1 describes the interrupt sources provided by the PCMCIA interface.

Table 12-1. Interrupt Signals

Name	Function
S0_READY_nIREQ	Socket 0 READY_nIREQ pin
S1_READY_nIREQ	Socket 1 READY_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 (voltage detect) pin
S1_BVD1_STSCHG	Socket 1 BVD1_STSCHG (voltage detect) pin

12.3 Detailed Signal Descriptions

This section provides detailed signal descriptions for the PCMCIA interface block diagram shown in Figure 12-1.

12.3.1 Intel® StrongARM* SA-1111 Interface Signals.

Table 12-2 describes the SA-1110 interface signals.

Table 12-2. SA-1110 Interface Signals (Sheet 1 of 2)

Signal Name	Description
PKTSEL	PCMCIA socket select. This signal is an input and 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.
Address <25:0>	Address Inputs. This bus conveys the address for PCMCIA and Compact Flash slots.
Data <15:0>	Bi-directional Data Bus. This bus is driven by the PCCard/Compact Flash Block. Only when either one of nPCE<2:1> is asserted and either nPIOR or nPOE are asserted.
nPCE<2:1>	Card Enable. These signals are inputs, gated within the block, and used to select a PCMCIA card. Bit 2 enables the high byte lane and bit 1 enable the low byte lane.
nIOIS16	IO is 16 bit. This signal is an output and generated as following: IF PSKTSEL is low and Card is fully inserted in socket zero, then nPIOIS16 = S0_nIOIS16; Otherwise if PSKTSEL is high and Card is fully inserted in socket one, then nPIOIS16 = S1_nIOIS16. Otherwise nPIOIS16 = high.
nPIOR	IO Read. This signal is an input and used to generate Sx_nIORD signals.
nPIOW	IO Write. This signal is an input and used to create Sx_nIOW signals.
nPOE	Output Enable. This signal is an input and used to generate Sx_nOE signals.

Table 12-2. SA-1110 Interface Signals (Sheet 2 of 2)

Signal Name	Description
nPREG	REG Select. This signal is an input and used to generate Sx_nREG signals.
nPWAIT	Wait signal. This signal is an output and generated as following: IF PSKTSEL is low, and at least one of nPCE<2:1> is asserted, and Card is fully inserted in socket zero and PCCR<4> is set, then nPWAIT = S0_nWAIT; Otherwise if PSKTSEL is high, and at least one of nPCE<2:1> is asserted, and Card is fully inserted in socket one and PCCR<5> is set, then nPWAIT = S1_nWAIT. Otherwise nPWAIT = high.
NPWE	Write Enable. This signal is an input and used to generate Sx_nWE signals.

12.4 Socket 0 Signals

Table 12-3 lists the socket 0 signals.

Table 12-3. Socket 0 Signals (Sheet 1 of 4)

Signal Name	Description
S0_RESET	Reset. This signal is an output and used to reset the card. This signal needs a weak pull-up on board. The signal is active high and generated by the following way. **If** PCCR<2> = 1 **then** {if in sleep mode, {if PCSSR<0> =0, This signal is tri-stated, elseif This signal is forced to low.} else S0_RESET = PCCR<0> **Else** this signal is tristated.
S0_ADDRESS<25:0>	Address Bus. **If** one of the S0_nCD<2:1> signals is high or PCCR<2> is '0' **Then** The bus is tri-stated; **Else if** it is in sleep mode **Then** {if PCSSR<0> =0, This bus is tri-stated, elseif This bus is forced to low.} **Else If** PSKTSEL is high or both of nPCE<2:1> are high **Then** The whole bus is forced to low; **Else** S0_ADDRESS<25:0> = Address<25:0>.

Table 12-3. Socket 0 Signals (Sheet 2 of 4)

Signal Name	Description
S0_Data<15:0>	Data Bus. PC Card bi-directional data bus. Data is transferred from the PC Card socket to the CPU through this 16-bit bus. Only byte and 16 bit operations are supported. **If** PCCR<2> =1 Then {if in sleep mode {if PCSSR<0> = 0, output is tristated else driven low by SA-1111} else if nPIOR or nPOE is asserted and PSKTSEL is low, output is tristated else driven by SA-1111} **Else** output is tristated.
S0_nPREG	Output to PC Card socket nREG signal. This is an output. **If** PCCR<2> = 1 **Then** {if it is in sleep mode, {if PCSSR<0> =0, This signal is tri-stated, elseif This signal is forced to high.} elseif S0_nCD<2:1> =00 and PSKTSEL =0 then S0_nPREG = nPREG Else S0_nPREG =1} **Else** S0_nPREG is tri-stated.
S0_nCE<2:1>	Card Enable: Output to PC Card socket nCE<2:1> signals. **If** PCCR<2> = 1 **Then** {if it is in sleep mode, {if PCSSR<0> =0, This signal is tri-stated, elseif This signal is forced to high.} elseif S0_nCD<2:1> =00 and PSKTSEL =0 then S0_nCE<2:1>= nPCE<2:1> Else S0_nCE<2:1> =11b} **Else** S0_nCE is tri-stated.
S0_nOE	Output Enable. Output to PC Card socket nOE signal. If PCCR<2> = 1 **Then** {if it is in sleep mode, {if PCSSR<0> =0, This signal is tri-stated, elseif This signal is forced to high.} elseif S0_nCD<2:1> =00 and PSKTSEL =0 then S0_nOE = nPOE Else S0_nOE =1} Else S0_nOE is tri-stated.

Table 12-3. Socket 0 Signals (Sheet 3 of 4)

Signal Name	Description
S0_nWE	Write Enable. Output to PC Card socket nWE signal. **If** PCCR<2> = 1 **Then** {if it is in sleep mode, {if PCSSR<0> =0, This signal is tri-stated, elseif This signal is forced to high.} elseif S0_nCD<2:1> =00 and PSKTSEL =0 then S0_nWE= nPWE Else S0_nWE =1} **Else** S0_nWE is tri-stated.
S0_nIORD	IO Read. Output to PC Card socket nIOR signal. **If** PCCR<2> = 1 Then {if it is in sleep mode, {if PCSSR<0> =0, This signal is tri-stated, elseif This signal is forced to high.} elseif S0_nCD<2:1> =00 and PSKTSEL =0 then S0_nIORD = nPIOR Else S0_nIORD =1} **Else** S0_nIORD is tri-stated.
S0_nIOWR	IO Write. Output to PC Card socket nPIOW signal. **If** PCCR<2> = 1 **Then** {if it is in sleep mode, {if PCSSR<0> =0, This signal is tri-stated, elseif This signal is forced to high.} elseif S0_nCD<2:1> =00 and PSKTSEL =0 then S0_nIOWR = nPIOW Else S0_nIOWR =1} **Else** S0_nIOWR is tri-stated.
S0_nWAIT	Wait Signal. Input from PC Card Socket nWAIT. This signal generates the nPWAIT signal.
S0_nIOS16	IO is 16 bit. Input from PC Card Socket nIOIS16. This signal generates the nIOIS16 signal and can be read from Status Register.
S0_READY_nIREQ	Ready/Interrupt Request. Input from PC Card Socket RDY/nIRQ signal. This signal can be accessed through Status Register and used as an interrupt source.
S0_BVD2_nnSPKR	Socket 0 VDD voltage sense signal/audio digital speaker. This signal can be read from Status Register.
S0_BVD1_nSTSCHG	Socket 0 VDD voltage sense signal/card status changed. This signal can be read from Status Register and used as an interrupt source.
S0_nVS<2:1>	Socket 0 VSS voltage sense signals. These signals can be accessed through Status Register.

Table 12-3. Socket 0 Signals (Sheet 4 of 4)

Signal Name	Description
S0_nCD<2:1>	Card Detect. Input from PC Card Socket Card Detect signals nCD<2:1>. These signals are used to generate S0_CDVALID. When both of S0_nCD<2:1> are low, S0_CDVALID= 0; otherwise S0_CDVALID = 1. The S0_CDVALID is used as an interrupt source and can be read from Status Register.
PCMCIA_PWR<3:0>	Voltage Control. This allows the interface to accommodate PCMCIA cards of different operating voltages. This is done through GPIO_A<3:0> – see Chapter 10.

12.5 Socket 1 Signals

Table 12-4 lists the socket 1signals.

Table 12-4. Socket 1 Signals (Sheet 1 of 3)

Signal Name	Description
S1_RESET	Reset. This signal is an output and used to reset the card. This signal needs a weak pull-up on board. The signal is active high and generated as follows. **If** PCCR<3> = 1 **then** {if in sleep mode, {if PCSSR<1> =0, This signal is tristated, elseif This signal is forced to low.} else S1_RESET = PCCR<1> **Else** this signal is tristated.
S1_ADDRESS<10:0>	Address Bus. **If** one of the S1_nCD<2:1> signals is high or PCCR<3> is '0' **Then** The bus is tri-stated; **Elseif** it is in sleep mode **Then** {if PCSSR<1> =0, This bus is tri-stated, elseif This bus is forced to low.} **Else If** PSKTSEL is high or both of nPCE<2:1> are high **Then** The whole bus is forced to low; **Else** S1_ADDRESS<10:0> = Address<10:0>.
S1_Data<15:0>	Data Bus. PC Card bi-directional data bus. Data is transferred from the PC Card socket to the CPU through this 16-bit bus. Only byte and 16 bit operations are supported. **If** PCCR<3> =1 Then {if in sleep mode {if PCSSR<1> = 0, output is tristated else driven low by SA-1111} else if nPIOR or nPOE is asserted and PSKTSEL is low, output is tristated else driven by SA-1111} **Else** output is tristated.

Table 12-4. Socket 1 Signals (Sheet 2 of 3)

Signal Name	Description
S1_nPREG	Output to PC Card socket nREG signal. This is an output. **If** PCCR<3> = 1 **Then** {if it is in sleep mode, {if PCSSR<1> =0, This signal is tri-stated, elseif This signal is forced to high.} elseif S1_nCD<2:1> =00 and PSKTSEL =0 then S1_nPREG = nPREG Else S1_nPREG =1} **Else** S1_nPREG is tri-stated.
S1_nCE<2:1>	Card Enable: Output to PC Card socket nCE<2:1> signals. **If** PCCR<3> = 1 **Then** {if it is in sleep mode, {if PCSSR<1> =0, This signal is tristated, elseif This signal is forced to high.} elseif S1_nCD<2:1> =00 and PSKTSEL =0 then S1_nCE<2:1>= nPCE<2:1> Else S1_nCE<2:1> =11b} **Else** S1_nCE is tristated.
S1_nOE	Output Enable. Output to PC Card socket nOE signal. If PCCR<3> = 1 **Then** {if it is in sleep mode, {if PCSSR<1> =0, This signal is tristated, elseif This signal is forced to high.} elseif S1_nCD<2:1> =00 and PSKTSEL =0 then S1_nOE = nPOE Else S1_nOE =1} Else S1_nOE is tristated.
S1_nWE	Write Enable. Output to PC Card socket nWE signal. **If** PCCR<3> = 1 **Then** {if it is in sleep mode, {if PCSSR<1> =0, This signal is tristated, elseif This signal is forced to high.} elseif S1_nCD<2:1> =00 and PSKTSEL =0 then S1_nWE= nPWE Else S1_nWE =1} **Else** S1_nWE is tristated.

Table 12-4. Socket 1 Signals (Sheet 3 of 3)

Signal Name	Description
S1_nIORD	IO Read. Output to PC Card socket nIOR signal. **If** PCCR<3> = 1 Then {if it is in sleep mode, {if PCSSR<1> =0, This signal is tri-stated, elseif This signal is forced to high.} elseif S1_nCD<2:1> =00 and PSKTSEL =1 then S1_nIORD = nPIOR Else S1_nIORD =1} **Else** S1_nIORD is tri-stated.
S1_IOWR	IO Write. Output to PC Card socket nPIOW signal. **If** PCCR<3> = 1 **Then** {if it is in sleep mode, {if PCSSR<1> =0, This signal is tristated, elseif This signal is forced to high.} elseif S1_nCD<2:1> =00 and PSKTSEL =0 then S1_nIOWR = nPIOW Else S1_nIOWR =1} **Else** S1_nIOWR is tristated.
S1_nWAIT	Wait Signal. Input from PC Card Socket nWAIT. This signal generates the nPWAIT signal.
S1_nIOS16	IO is 16 bit. Input from PC Card Socket nIOIS16. This signal generates the nIOIS16 signal and can be read from Status Register.
S1_READY_nIREQ	Ready/Interrupt Request. Input from PC Card Socket RDY/nIRQ signal. This signal can be accessed through Status Register and used as an interrupt source.
S1_BVD2_nnSPKR	Socket 1 VDD voltage sense signal/audio digital speaker. This signal can be read from Status Register.
S1_BVD1_nSTSCHG	Socket 1 VDD voltage sense signal/card status changed. This signal can be read from Status Register and used as an interrupt source.
S1_nVS<2:1>	Socket 1 VSS voltage sense signals. These signals can be accessed through Status Register.
S1_nCD<2:1>	Card Detect. Input from PC Card Socket Card Detect signalsnCD<2:1>. These signals are used to generate S1_CDVALID. When both of S1_nCD<2:1> are low, S1_CDVALID= 0; otherwise S1_CDVALID = 1. The S1_CDVALID is used as an interrupt source and can be read from Status Register.

12.6 Programmer's Model

This section describes the three registers in the PCMCIA interface.

12.6.1 Status Register (PCSR)

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

Table 12-5. PCSR Bit Descriptions

0x0001808	PCSR	SA-1111 Companion Chip

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																		Socket 1 BVD2	Socket 1 BVD1	Socket 0 BVD2	Socket 0 BVD1	Socket 1 WP	Socket 0 WP	Socket 1 VS2	Socket 1 VS1	Socket 0 VS2	Socket 0 VS1	Socket 1 card detect	Socket 0 card detect	Socket 1 ready	Socket 0 ready
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Type	Description
0	Socket 0 ready	RO	Status of S0_READY_nIREQ
1	Socket 1 ready	RO	Status of S1_READY_nIREQ
2	Socket 0 card detect	RO	0 = Socket 0 card detect 1 and 2 valid
3	Socket 1 card detect	RO	0 = Socket 1 card detect 1 and 2 valid
4	Socket 0 VS1	RO	Socket 0 VS1 - socket 0 voltage sense 1
5	Socket 0 VS2	RO	Socket 0 VS2 - socket 0 voltage sense 2
6	Socket 1 VS1	RO	Socket 1 VS1 - socket 1 voltage sense 1
7	Socket 1 VS2	RO	Socket 1 VS2 - socket 1 voltage sense 2
8	Socket 0 WP	RO	Socket 0 WP (S0_nIOIS16)
9	Socket 1 WP	RO	Socket 1 WP (S1_nIOIS16)
10	Socket 0 BVD1	RO	Socket 0 BVD1_nSTSCHG
11	Socket 0 BVD2	RO	Socket 0 BVD2_nSPKR
12	Socket 1 BVD1	RO	Socket 1 BVD1_nSTSCHG
13	Socket 1 BVD2	RO	Socket 1 BVD2_nSPKR
31:14	—	—	Reserved.

12.6.2 Control Register (PCCR)

This register controls the setting of various output pins in the PCMCIA interface and also allows tristating the control lines. Note that control of the external PCMCIA/Compact Flash power controller is done through GPIO_A<3:0> - see Chapter 10 for details on controlling those pins.

Two bits <5:4> control assertion of nPWAIT on a per-slot basis.

Table 12-6. PCCR Bit Descriptions

0x0001800 PCCR SA-1111 Companion Chip

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																								S0PSE	S0PSE	S1_PWAITEN	S0_PWAITEN	S1_FLT	S0_FLT	S1_RST	S0_RST
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Type	Description
0	S0_RST	RW	Socket 0 reset: 1 = assert reset
1	S1_RST	RW	Socket 1 reset: 1 = assert reset
2	S0_FLT	RW	S0 float 0 = Float all S0 control lines
3	S1_FLT	RW	S1 float 0 = Float all S1 control lines
4	S0_PWAITEN	RW	S0_nPWAIT enable
5	S1_PWAITEN	RW	S1_nPWAIT enable
6	S0PSE	RW	If using a 3 V card, set to 0; if using a 5 V card, set to 1.
7	S0PSE	RW	If using a 3 V card, set to 0; if using a 5 V card, set to 1.
31:8	—	—	Reserved.

12.6.3 Sleep State Register (PCSSR)

This register allows setting the state of PCMCIA/CF output pins during *sleep* mode. Note the four voltage control pins are controlled via the regular GPIO control facilities provided by the GPIO registers.

Table 12-7. PCSSR Bit Descriptions

0x0001804 **PCSSR** **SA-1111 Companion Chip**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																														Socket 1	Socket 0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Type	Description
0	Socket 0	RW	Socket 0 control, data, and address pin states in SLEEP mode For more information on sleep state control, see Table 12-8
1	Socket 1	RW	Socket 1 control, data, and address pin states in SLEEP mode For more information on sleep state control, see Table 12-8
31:2	—	—	Reserved.

In sleep state, PCMCIA or CF signal pins are put into a state dependent on the value in the S0CONT or S1CONT bits, as shown in Table 12-8. Bit PCSSR<0> specifies the sleep state of Socket 0 (PCMCIA) output pins, while bit PCSSR<1> controls Socket 1 (CF) pins.

Table 12-8. S0 (PCMCIA) and S1 (CF) Sleep State Control

PCMCIA Signal (x = 0 or 1)	Socket 0: PCSSR<0> = 0	Socket 0: PCSSR<0> = 1	Socket 1: PCSSR<1> = 0	Socket 1: PCSSR<1> = 1
Sx_DATA	Tristate	0	Tristate	0
Sx_ADDRESS	Tristate	0	Tristate	0
Sx_NOE	Tristate	1	Tristate	1
Sx_nIORD	Tristate	1	Tristate	1
Sx_nIOWR	Tristate	1	Tristate	1
Sx_nCE1	Tristate	1	Tristate	1
Sx_nCE2	Tristate	1	Tristate	1
Sx_nWE	Tristate	1	Tristate	1
Sx_Reset	Tristate	0	Tristate	0

12.6.4 Memory Map

Table 12-9 shows the read and write locations in the PCMCIA interface register memory map. See Section 3.2.1 for the PCMCIA interface base address.

Table 12-9. PCMCIA Interface Register Memory Map

Address	Read Location	Write Location
0x0001800	PCCR register	PCCR register
0x0001804	PCSSR register	PCSSR register
0x0001808	PCSR register	Reserved

Electrical and Timing Specifications 13

This section provides a description of the Intel® StrongARM* SA-1111 Microprocessor Companion Chip (SA-1111) pins and the AC and DC requirements for each pin or pin type.

13.1 AC and DC Signal Requirements

Table 13-1 describes the AC and DC requirements for each pin. The following terms are used to describe the type of pin:

I = Input only

O = Output only

I/O - od = Input/Output - behaves like an open-drain output in PS/2 mode of operation

I/O - Input/Output

Table 13-1. AC and DC Requirements

Name	Type	Description
SA-1110 Processor Interface - Register Access and SDRAM Interfacing		
A<25:0>	I/O	26-bit system address bus. Bits A<24:10> for SDRAM (output): setup 3ns, hold 1ns referenced to SDCLK. Bits A<25>, A<9:0> are functional test only.
D<31:0>	I/O	For SDRAM writes (output): setup 3ns, hold 1ns referenced to SDCLK
DQM<3:0>	I/O	For SDRAM writes (output Byte En): setup 3ns, hold 1ns referenced to SDCLK
MBGNT	I	Input only; for DMA access
MBREQ	O	Output only; Rise and Fall times: min 1ns, max 5ns (30 pF load)
nSDCAS	O	For SDRAM (output): setup 3ns, hold 1ns referenced to SDCLK
nCS	I	Input only; for register access
nSDCS/nRAS	O	Chip Select output for SDRAM: setup 3ns, hold 1ns referenced to SDCLK
nSDRAS	O	For SDRAM (output): setup 3ns, hold 1ns referenced to SDCLK
nWE	I/O	For SDRAM (output): setup 3ns, hold 1ns referenced to SDCLK
RDY	O	Output only; Rise and Fall times: min 1ns, max 5ns (30 pF load)
SDCLK	O	Rising edge is timing reference point for all SDRAM signals Rise and Fall times: min 1ns, max 5ns (40 pF load)
USB Interface		
USB_MINUS	I/O	USB_PLUS and USB_MINUS pins, as outputs, must meet USB rev. 1.1 specification AC and DC requirements under specified load conditions
USB_PLUS	I/O	
AC-link Serial Port for Audio/GPIO_C Bits		
SYS_CLK	O	See I2S standard for rise/fall times and loading
Miscellaneous Signals		
INT	O	Standard CMOS output, point-to-point, max loading 30 pF

13.2 DC Specifications

Table 13-2 specifies the maximum current values for the different operational modes of the SA-1111.

Table 13-2. SA-1111 Power Consumption at 3.3 V VDD

Mode	Maximum Value
Run	<50 mA
Doze	VDD <6.4 mA, PLL_VDD <3.6 mA
Sleep	<20 µA

13.2.1 Power Supply Voltages

Table 13-3 specifies the power supply voltages for the SA-1111.

Table 13-3. SA-1111 Power Supply Voltages

Parameter	Recommended Value
Core and CMOS power supply voltage	3.3 V ± 10%
PLL power supply voltage	3.3 V ± 10%
VDD_PCMCIA	0 (off), 3.3 V or 5V ± 10%
VDD_CF	0 (off), 3.3 V or 5V ± 10%

13.2.2 Absolute Maximum Ratings

Table 13-4 lists the absolute maximum ratings for the SA-1111.

Table 13-4. Absolute Maximum Ratings

Symbol	Parameter	Minimum	Maximum	Note
V_{IP}	Voltage applied to any pin	– 0.5 V	• 3.6 V with PCMCIA at 3.3 V • 5.5 V with PCMCIA at 5.0 V	1
T_S	Storage temperature	–20°C (–4°F)	+125°C (257°F)	1

NOTE: These are stress SA-1111 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.

13.2.3 CMOS Input Voltage Levels

Table 13-5 lists the functional operating dc parameters for the SA-1111.

Table 13-5. SA-1111 CMOS Inputs

Symbol	Parameter	Minimum	Maximum	Note
V_{IHC}	Input high voltage	>2.0 V	3.3 V	1
V_{ILC}	Input low voltage	0.0 V	<0.8 V	—

NOTE:
1. PCMCIA inputs at 3.3 V.

13.2.4 DC Operating Conditions

Table 13-6 lists the functional operating dc parameters for the SA-1111.

Table 13-6. SA-1111 DC Operating Conditions

Symbol	Parameter	Minimum	Nominal	Maximum	Note
V_{OHC}	Output high 3.3V CMOS voltage	2.4 V	—	—	—
V_{OLC}	Output low 3.3V CMOS voltage	—	—	0.4 V	—
I_{OHC}	High level output current	—	—	4 mA	—
T_A	Ambient operating temperature	0°C (32°F)	—	70°C (158°F)	—
I_{IN}	IC input leakage current	—	1 µA	—	—
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.

13.2.5 Compact Flash and PCMCIA 4mA Buffer Input Voltage Levels

The normal operating value for Vcc is 5 V±10% and 3.3 V±10%. Interface signal levels are compatible with standard TTL or CMOS for the operating voltage used. Table 13-7 and Table 13-8 list the functional operating dc parameters for the SA-1111.

Table 13-7. SA-1111 5 V Compact Flash and PCMCIA Buffer Input Level Specifications

Parameter	Minimum	Maximum
V_{CC}	4.50 V	5.50 V
V_{IH}	2.4 V	Vcc +2.5 V
V_{IL}	0.0 V	0.8 V
V_{OH}	2.8 V (TTL load) .9 Vcc (CMOS load)	Vcc +.25 V
V_{OL}	0.0 V	0.5 V (TTL load) .1 Vcc (CMOS Load)

Table 13-8. SA-1111 3.3 V Compact Flash and PCMCIA Buffer Input Level Specifications

Parameter	Minimum	Maximum
V_{CC}	3.0 V	3.6 V
V_{IH}	0.475 Vcc	Vcc +0.5 V
V_{IL}	–0.5 V	0.325 Vcc
V_{OH}	0.9 Vcc	—
V_{OL}	—	.1 Vcc

13.3 AC Characteristics

Operating conditions: V_{3vCore} = 3.3 V ± 10%; Ta = 0°C το 70° C.

13.3.1 I²S and MSB-Justified Interface Timing

Figure 13-1 shows and Table 13-9 lists the specifications for I²S and MSB-Justified interface timing. Figure 13-2 shows and Table 13-10 lists the specifications for the I²S and MSB -Justified interface BIT_CLK, Sync, and Data Timing.

Figure 13-1. I²S and MSB-Justified SYS_CLK Timing Diagram

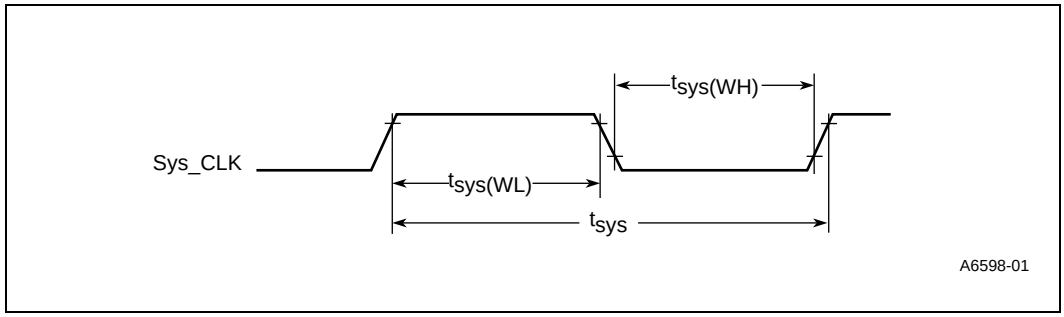


Table 13-9. I²S and MSB-Justified SYS_CLK Timing

Symbol	Parameter	Min.	Typ.	Max.	Units	Conditions
T_{sys}	SYS_CLK period	—	f_{sys}	—	KHz	$f_{sys = 256fs}$
t_{ssWL}	f_{sys} LOW level pulse width	30	—	70	%	% of T_{sys} width
t_{ssWH}	f_{sys} HIGH level pulse width	30	—	70	%	% of T_{sys} width

Figure 13-2. I²S and MSB-Justified Interface Timing Diagram

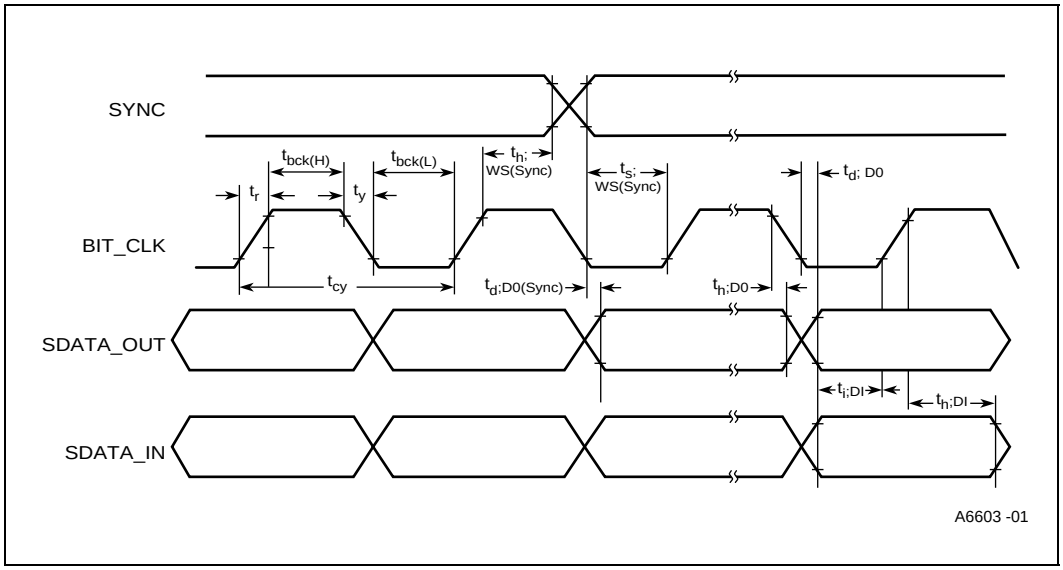


Table 13-10. I²S and MSB-Justified Interface BIT_CLK, SYNC, and Data Timing

Symbol	Parameter	Min.	Typ.	Max.	Units	Conditions
T_{cy}	BIT_CLK period	300	—	—	ns	$f_{BCK} = 64f_s$
$T_{BCK}(H)$	BIT_CLK HIGH time	250	—	—	ns	—
$T_{BCK}(L)$	BIT_CLK LOW time	250	—	—	ns	—
t_r	BIT_CLK rise time	—	—	20	ns	—
t_f	BIT_CLK fall time	—	—	20	ns	—
$t_{s;DI}$	Data input setup time	20	—	—	ns	—
$t_{h;DI}$	Data input hold time	0	—	—	ns	—
$t_{d;DO}$	Data output delay time (from BIT_CLK falling edge)	—	—	80	ns	—
$t_{d;DO(SYNC)}$	Data output delay time (from SYNC edge)	—	—	80	ns	MSB-Justified format SYNC = WS (word select)
$t_{h;DO}$	Data output hold time	0	—	—	ns	—
$t_{s;SYNC}$	Word selection setup time	20	—	—	ns	—
$t_{h;SYNC}$	Word selection setup time	10	—	—	ns	—

13.3.2 L3 Control Bus Interface Timing

Figure 13-4 and Figure 13-4 show and Table 13-11 lists the specifications for the L3 Control Bus interface timing.

Figure 13-3. Timing for L3 Address Mode

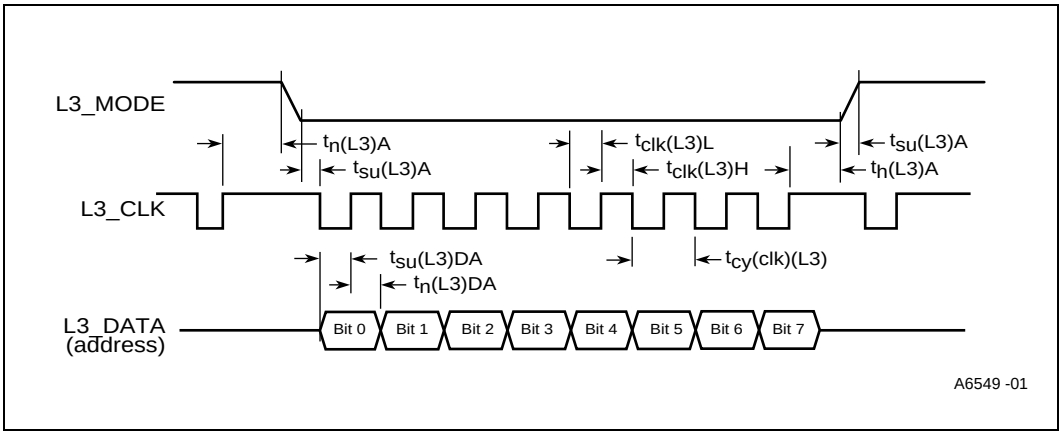


Figure 13-4. Timing for L3 Data Transfer Mode

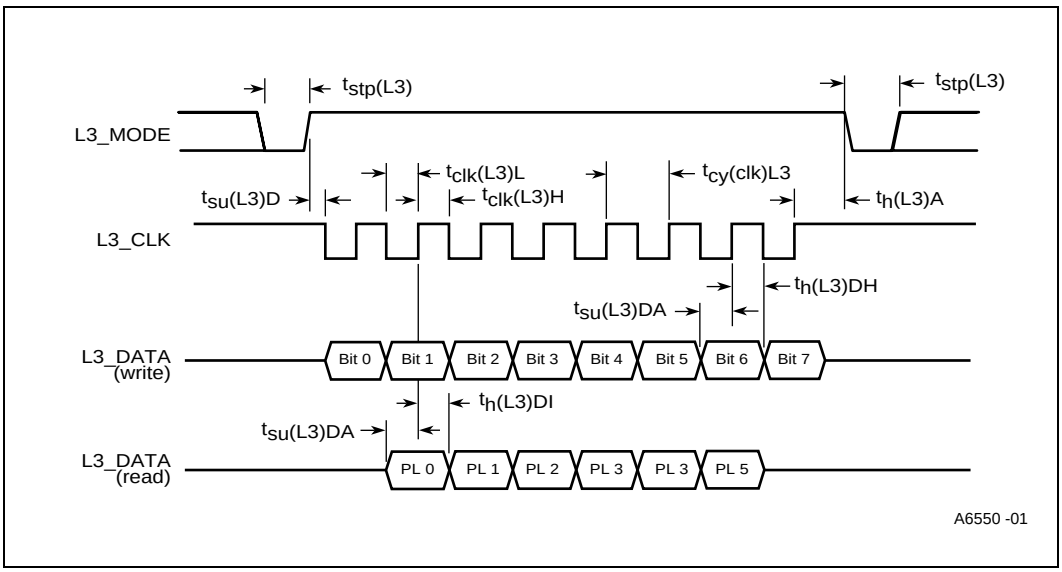


Table 13-11. L3 Control Bus Interface Timing (Sheet 1 of 2)

Symbol	Parameter	Min.	Typ.	Max.	Units	Conditions
Tcy(CLK)(L3)	L3CLK period	500	—	—	ns	—
t_{CLK}(L3)H	L3CLK HIGH period	250	—	—	ns	—
t_{CLK}(L3)L	L3CLK LOW period	250	—	—	ns	—
tsu(L3)A	L3MODE setup time	190	—	—	ns	Addressing mode
th(L3)A	L3MODE hold time	190	—	—	ns	Addressing mode
tsu(L3)D	L3MODE setup time	190	—	—	ns	Data write mode

Table 13-11. L3 Control Bus Interface Timing (Sheet 2 of 2)

Symbol	Parameter	Min.	Typ.	Max.	Units	Conditions
th(L3)D	L3MODE hold time	190	—	—	ns	Data write mode
tsu(L3)DA	L3DATA setup time	190	—	—	ns	Data write and addressing mode
th(L3)DA	L3DATA hold time	30	—	—	ns	Data write and addressing mode
tstp(L3)	L3MODE halt time	190	—	—	ns	—
tsu(L3)DI	L3DATA setup time	30	—	—	ns	Data read mode
th(L3)DI	L3DATA hold time	10	—	—	ns	Data read mode

13.3.3 AC-link Interface Timing

Figure 13-5 shows and Table 13-12 lists the AC-link interface signal rise and fall timing.

Figure 13-5. Signal Rise and Fall Timing Diagram

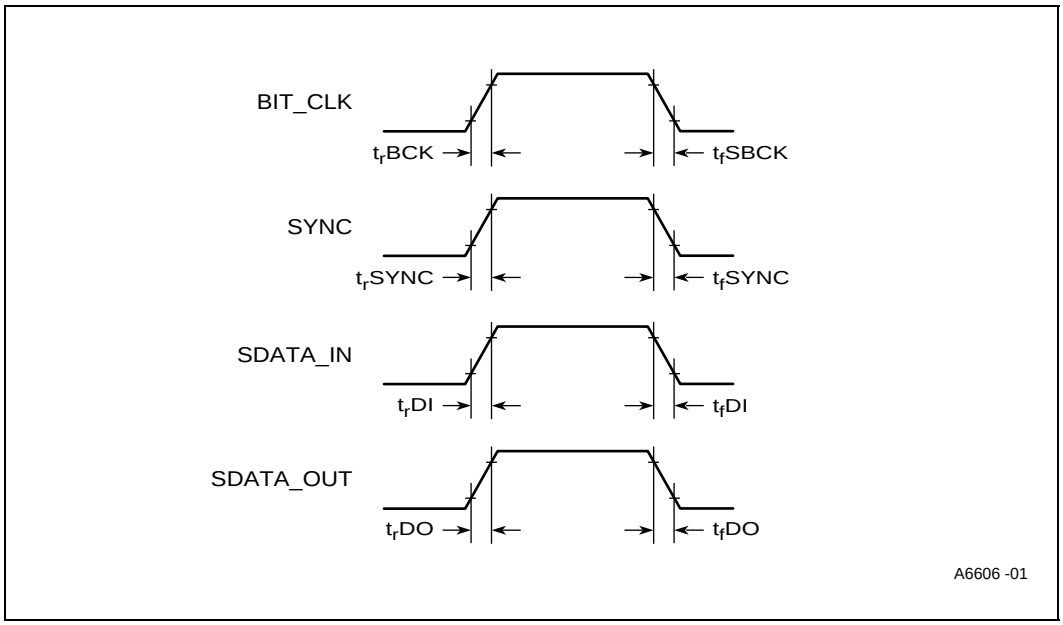


Table 13-12. AC-link Interface Signal Rise and Fall Timing

Symbol	Parameter	Min.	Typ.	Max.	Units
t_{Rbck}	BIT_CLK rise time	—	—	6	ns
t_{fBCK}	BIT_CLK fall time	—	—	6	ns
t_{rSYNC}	SYNC rise time	—	—	6	ns
t_{fSYNC}	SYNC fall time	—	—	6	ns
t_{rDI}	SDATA_IN rise time	—	—	6	ns
t_{fDI}	SDATA_IN fall time	—	—	6	ns
t_{rDO}	SDATA_OUT rise time	—	—	6	ns
t_{fDO}	SDATA_OUT fall time	—	—	6	ns

Figure 13-6 shows and Table 13-13 lists the AC-link interface clock timing.

Figure 13-6. BIT_CLK and SYNC Timing Diagram

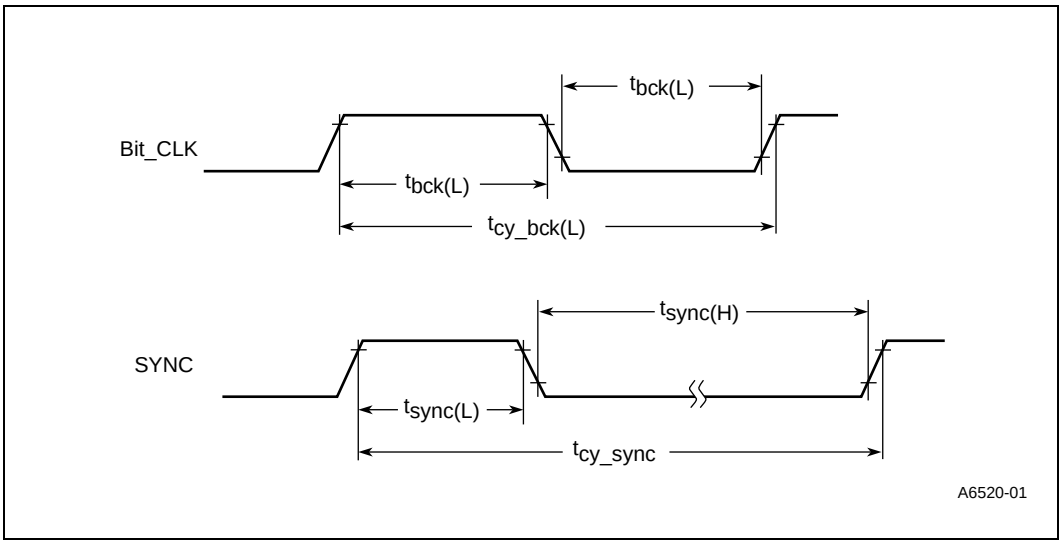


Table 13-13. AC-link Interface Clock Timing

Symbol	Parameter	Min.	Typ.	Max.	Units	Conditions
f_{BCK}	BIT_CLK frequency	—	12.288	—	MHz	f_{BCK} = 256 x 48 KHz
T_{cyBCK}	BIT_CLK period	—	81.4	—	ns	—
$t_{BCK}(H)$	BIT_CLK HIGH time	36	40.7	45	ns	—
$t_{BCK}(L)$	BIT_CLK LOW time	36	40.7	45	ns	—
f_{SYNC}	SYNC frequency	—	48.0	—	KHz	f_s = 48 KHz
T_{cySYNC}	SYNC period	—	81.4	—	µs	—
$t_{SYNC}(H)$	SYNC HIGH time	—	20.8	—	µs	—
$t_{SYNC}(L)$	SYNC LOW time	—	1.3	—	µs	—

Figure 13-7 shows and Table 13-14 lists the AC-link interface data output and input timing.

Figure 13-7. Data Output and Input Timing Diagram

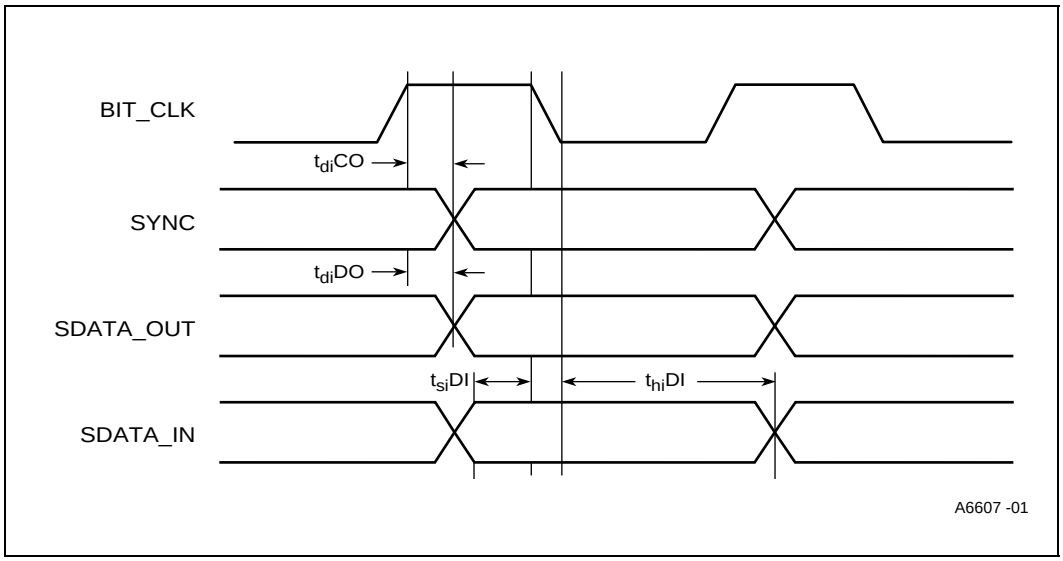


Table 13-14. AC-link Interface Data Output and Input Timing

Symbol	Parameter	Min.	Typ.	Max.	Units	Conditions
$t_{d;DO}$	SDATA_OUT delay time (from BIT_CLK rising edge)	—	—	15	ns	—
$t_{d;SYNC}$	SYNC output delay time (from BIT_CLK rising edge)	—	—	15	ns	—
$t_{s;DI}$	SDATA_IN setup time	10	—	—	ns	—
$t_{h;DI}$	SDATA_IN hold time	10	—	—	ns	—

Figure 13-8 and Figure 13-9 show and Table 13-15 lists the AC-link low power mode and wake up sequence timing.

Figure 13-8. AC-Link Low-Power Mode Timing Diagram

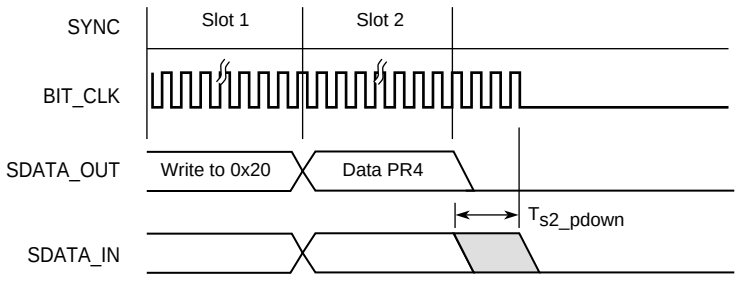


Figure 13-9. Warm Reset Timing Diagram

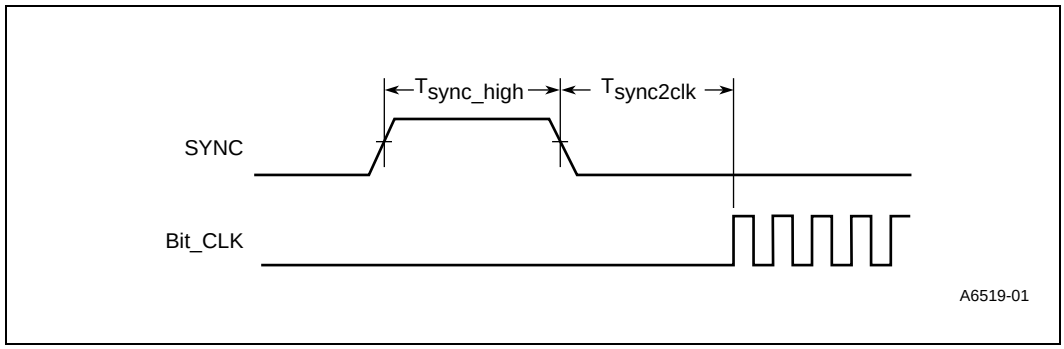


Table 13-15. AC-link Low Power Mode and Wake Up Sequence Timing

Symbol	Parameter	Min.	Typ.	Max.	Units	Conditions
Ts2_pdown	End of slot 2 to BIT_CLK, SDATA_IN low	—	—	1.0	µs	Less than 1.0 µs
Tsync_high	SYNC high active pulse	1.0	—	—	µs	More than 1.0 µs
Tsync2clk	SYNC inactive to BIT_CLK startup delay time	162.8	—	—	ns	More than two BIT_CLK periods

Package and Pinout 14

This chapter defines the package and pinout for the Intel® StrongARM* SA-1111 Microprocessor Companion Chip (SA-1111).

14.1 Package

The SA-1111 is contained in a 256 mini-BGA. Figure 14-1 shows the package details. Table 14-1 lists the SA-1111 pins in numeric order, showing the signal type for each pin.

Figure 14-1. SA-1111 256 mBGA Mechanical Drawing

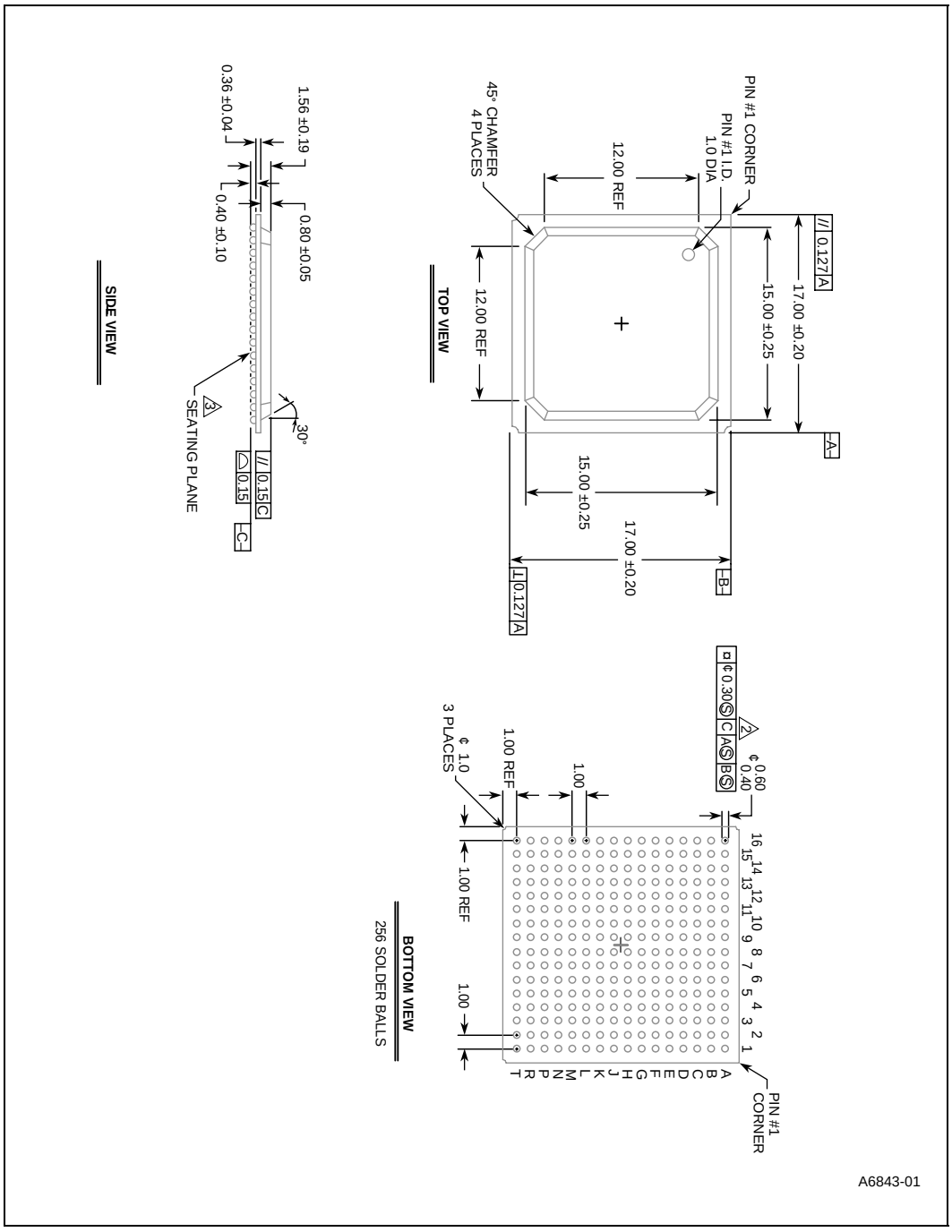


Table 14-1. SA-1111 Pinout — 256 Mini-Ball Grid Array

Pin	Signal	BGA Pad	Pin	Signal	BGA Pad	Pin	Signal	BGA Pad	Pin	Signal	BGA Pad
1	S0_NCE1	B2	65	D31	T1	129	NPIOW	T15	193	S1_DATA14	A16
2	S0_DATA14	B1	66	RVDD4	L7	130	NSDRAS	R14	194	CFVDD	B15
3	PVDD0	C2	67	RVSS4	J8	131	NPREG	P14	195	RVSS13	J6
4	S0_DATA7	C1	68	D30	R2	132	NOE	R15	196	S1_DATA6	B14
5	S0_DATA13	D2	69	D29	R3	133	NPWAIT	T16	197	S1_DATA13	A15
6	RVSS0	H6	70	D5	P4	134	NCAS3	P13	198	S1_DATA5	A14
7	S0_DATA6	D3	71	D27	N4	135	NPCE2	R16	199	S1_DATA12	E13
8	S0_DATA12	D4	72	D14	R4	136	NSDCAS	P16	200	S1_DATA4	B13
9	S0_DATA5	F5	73	D23	T2	137	NPCE1	P15	201	S1_DATA11	C13
10	S0_DATA11	D1	74	D22	T3	138	NCAS0	N14	202	S1_DATA3	D13
11	S0_NVS2	E3	75	D15	T4	139	NCAS1	N15	203	S0_NIOIS16	A13
12	S0_NVS1	E4	76	RVDD5	L10	140	A25	N13	204	S0_DATA10	E12
13	S0_NCD2	G5	77	RVSS5	J9	141	NCAS2	N16	205	S0_DATA2	C12
14	S0_NCD1	E2	78	SDATA_IN	R5	142	A15	M15	206	S0_DATA9	D12
15	GPIO_A3	E1	79	SCLK	P5	143	A18	M14	207	S0_DATA1	E11
16	GPIO_A2	F4	80	SDCLK	T5	144	A8	M16	208	S0_DATA8	B12
17	GPIO_A1	F3	81	SFRM	N5	145	RDY	L14	209	S0_DATA0	A12
18	GPIO_A0	H5	82	TXD	P6	146	A12	L15	210	S0_BVD1	C11
19	TPCLK	F2	83	D11	R6	147	A21	M13	211	S0_A0	D11
20	TPDATA	J5	84	D12	T6	148	A7	M12	212	S0_BVD2	E10
21	MSCLK	F1	85	RXD	N6	149	RVDD10	F9	213	S0_A1	B11
22	MSDATA	G4	86	RVDD6	L11	150	RVSS10	G7	214	S0_NPREG	A11
23	PWM0	G3	87	RVSS6	J10	151	S1_NCD2	L16	215	PVDD	C10
24	PWM1	G2	88	NPWE	M6	152	S1_NCD1	L12	216	RVSS14	J11
25	RVDD1	F6	89	NPIOR	R7	153	S1_NVS2	K15	217	S0_A2	E9
26	RVSS1	K8	90	NCS	P7	154	S1_NVS1	K14	218	S0_A3	B10
27	USB_PLUS	H3	91	PSKTSEL	N7	155	S1_DATA2	L13	219	S0_NWAIT	D10
28	USB_MINUS	G1	92	NPOE	T7	156	S1_DATA9	K16	220	S0_A4	C9
29	D16	H1	93	INT	T8	157	S1_DATA1	K12	221	S0_RESET	A10
30	USB_PWRCNTL	H4	94	NRESET	M7	158	S1_DATA8	K13	222	CVSS3	H11
31	USB_PWR_SENSE	K5	95	NTEST	M8	159	S1_DATA0	J12	223	CVDD3	F7
32	CVDD0	G6	96	CVDD1	K11	160	S1_BVD1	J15	224	S0_A5	A9
33	CVSS0	K9	97	CVSS1	H7	161	S1_A0	J14	225	S0_DATA4	D9
34	D8	H2	98	BAT_FLT	R8	162	CVSS2	G8	226	S0_A6	E8
35	SYS_CLK	J4	99	MBREQ	P8	163	CVDD2	F8	227	S0_A25	B9
36	D17	J1	100	MBGNT	N8	164	S1_BVD2	J16	228	S0_A7	D8
37	D9	J2	101	A23	R9	165	S1_A1	J13	229	S0_A24	A8
38	D0	K4	102	A24	T9	166	CFVDD	H15	230	S0_A12	B8
39	D24	J3	103	A0	P9	167	RVSS11	G9	231	S0_A23	D7
40	BIT_CLK	L5	104	A1	N9	168	S1_NPREG	H16	232	S0_A15	C8
41	D25	K3	105	RVDD7	G11	169	S1_A2	H12	233	RVSS15	L9
42	RVDD2	K6	106	RVSS7	H8	170	S1_NWAIT	H14	234	S0_A22	E7
43	RVSS2	K10	107	A2	M9	171	S1_A3	H13	235	S0_A16	C7
44	D2	K2	108	A13	R10	172	S1_RESET	G13	236	S0_A21	E5
45	SYNC	L8	109	A4	P10	173	S1_A4	G14	237	PVDD	B7
46	D10	K1	110	A14	T10	174	S1_NIOIS16	G15	238	S0_RDY	E6
47	D1	L4	111	A5	N10	175	S1_A5	G12	239	S0_A20	A7
48	D26	L3	112	A3	M10	176	S1_A6	G16	240	S0_NWE	D6
49	D3	L2	113	A20	T11	177	S1_RDY	F13	241	S0_A19	C6
50	D18	M4	114	A17	R11	178	S1_A7	F12	242	S0_A14	B6
51	D19	L1	115	A11	P11	179	CFVDD	F15	243	S0_A18	A6
52	RVDD3	L6	116	A10	N11	180	RVSS12	G10	244	S0_A13	D5
53	SDATA_OUT	M5	117	RVDD8	F11	181	S1_NWE	F16	245	S0_A17	A5
54	RVSS3	J7	118	RVSS8	H9	182	S1_A8	F14	246	S0_A8	B5
55	D20	M2	119	A19	R12	183	S1_NIOWR	E15	247	RVSS16	K7
56	D4	M1	120	A22	T12	184	S1_A9	E16	248	S0_NIOWR	A4
57	D13	N1	121	A6	M11	185	S1_NIORD	D16	249	S0_A9	B4
58	PLL_AVDD	N2	122	NSDCS	T13	186	S1_NOE	D15	250	S0_NIORD	C5
59	CLK	M3	123	NIOIS16	R13	187	S1_DATA10	E14	251	S0_A11	C4
60	PLL_AVSS	N3	124	A16	P12	188	S1_A10	D14	252	S0_DATA3	A3
61	D6	P1	125	A9	N12	189	S1_NCE2	C16	253	S0_NOE	B3
62	D21	P2	126	NWE	T14	190	S1_NCE1	C15	254	S0_NCE2	C3
63	D28	P3	127	RVDD9	F10	191	S1_DATA15	C14	255	S0_A10	A1
64	D7	R1	128	RVSS9	H10	192	S1_DATA7	B16	256	S0_DATA15	A2

Note: Table 14-1 references abbreviated pin names. To view the entire pin's name, see Table 1-2.