SA-110 Microprocessor

Technical Reference Manual

September 1998

Order Number: 278058-001

Contents

Figures

Tables

Introduction 1

The SA-110 Microprocessor (SA-110) is the first member of the StrongARM** family of high-performance, low-power microprocessors. The SA-110 is an implementation of Advanced RISC Machines Ltd. (ARM)** Version 4 architecture and offers significant advances in microprocessor design. The SA-110 has been designed to further extend the ARM family as a leading source of low-power, high-performance RISC processors for embedded consumer markets such as portable products and interactive digital video.

The SA-110 is a general-purpose, 32-bit microprocessor with a 16 Kbyte instruction cache, a 16 Kbyte write-back data cache, an 8-entry write buffer with 16 bytes per entry, and a memory-management unit (MMU) combined in a single chip. The SA-110 is software compatible with the ARM V4 architecture processor family and can be used with ARM support chips such as I/O, memory and video.

1.1 Features

Table 1-1 lists the features of the SA-110.

Table 1-1. SA-110 Device Features

• High performance — 115 Dhrystone 2.1 MIPS @ 100 MHz — 185 Dhrystone 2.1 MIPS @ 160 MHz — 192 Dhrystone 2.1 MIPS @ 166 MHz — 230 Dhrystone 2.1 MIPS @ 200 MHz — 268 Dhrystone 2.1 MIPS @ 233 MHz • Low performance — 130 mW @1.65 V/100 MHz core/27 MHz bus — 170 mW @1.65 V/100 MHz core/33 MHz bus — 230 mW @2.0 V/100 MHz core/33 MHz bus — 330 mW @2.0 V/100 MHz core/50 MHz bus — 420 mW @2.0 V/100 MHz core/66 MHz bus • Internal phase-locked loop (PLL) — 3.68 or 3.56 MHz External Reference Oscillator • Idle and sleep power-down modes • Big and little endian operating modes	• 3.3 V I/O interface • 144-pin thin quad flat pack (TQFP) • 32-way set-associative caches — 16 Kbyte instruction cache — 16 Kbyte write-back data cache • 32-Entry memory-management units — Maps 4 Kbyte, 64 Kbyte, or 1 Mbyte • Write buffer — 8-entry 16 bytes each • Early termination 32-bit multiplier — 32-bit accumulation (2–4 cycles) — 64-bit accumulation (3–5 cycles) • Memory bus Asynchronous or synchronous operation — 0-33 MHz @ 100 MHz — 0-53 MHz @ 160 MHz — 0-53 MHz @ 166 MHz — 0-66 MHz @ 200 MHz — 0-66 MHz @ 230 MHz

The on-chip caches together with the write buffer substantially raise the average execution speed and reduce the average amount of memory bandwidth required by the processor. This allows the external memory to support additional processors or direct memory access (DMA) channels with minimal performance loss.

The instruction set comprises eight basic instruction types:

- Two of these types make use of the onchip arithmetic logic unit, barrel shifter and multiplier to perform high-speed operations on the data in a bank of 16 logical (31 physical) registers, each 32 bits wide.
- Three classes of instructions control data transfer between memory and the registers: one optimized for flexibility of addressing, one for rapid context switching, and one for swapping data.
- Two instructions control the flow and privilege level of execution.
- One class is used to access the privileged state of the machine.

The ARM instruction set is a good target for compilers of many different high-level languages. Where required for critical code segments, assembly code programming is also straightforward, unlike some RISC processors that depend on sophisticated compiler technology to manage complicated instruction interdependencies.

The memory interface has been designed to allow the performance potential to be realized without incurring high costs in the memory system. Speed-critical control signals are pipelined to allow system control functions to be implemented in standard low-power logic, and these control signals permit the exploitation of page mode access offered by industry standard DRAMs.

1.2 Applications

The SA-110 can be used in the following markets:

- Portable products
  - Personal digital assistants (PDAs)
  - Smart phones
  - Digital cameras
  - Organizers
  - Bar-code scanners
- Interactive digital media
  - Digital set-top devices
  - Interactive TV
  - Video game players
- Embedded control
  - Internetworking: routers, bridges, LAN switches
  - Office automation: printers, scanners, copiers
  - Telecommunication: PBX, Cellular Base Station
  - Storage peripherals: drive and RAID controllers
  - PC add-ins: intelligent I/O cards, LAN/WAN
- Network computers
  - Tablet intranet products
  - Internet appliances

1.3 Document Conventions

0x	Marks a hexadecimal quantity.
BOLD	External signals are shown in bold capital letters.
binary	Where it is not clear that a quantity is binary, it is followed by the word binary.

1.4 ARM Architecture

The SA-110 implements the ARM V4 architecture as defined in the *ARM Architecture Reference*, Version A, 7 Feb. 1996, with the following options.

1.4.1 26-Bit Mode

The SA-110 supports 26-bit mode but all exceptions are initiated in 32-bit mode. The P and D bits do not affect the operation of SA-110; they always read as ones and writes to them are ignored.

1.4.2 Coprocessors

The SA-110 only supports MCR and MRC access to coprocessor 15. These instructions are used to access the memory-management, configuration, and cache control registers. All other coprocessor instructions cause an undefined instruction exception. No support for external coprocessors is provided.

1.4.3 Memory Management

Memory-management exceptions preserve the base address registers so no *fixup* code is required. Separate translation lookaside buffers (TLBs) are implemented for the instruction and data streams. Each TLB has 32 entries that can each map a segment, a large page, or a small page. The TLB replacement algorithm is round robin. The data TLBs support both the flush-all and flush-single-entry operations, while the instruction TLBs support only the flush-all operation.

1.4.4 Instruction Cache

The SA-110 has a 16 Kbyte instruction cache (Icache) with 32-byte blocks and 32-way associativity. The cache supports the flush-all-entry function. Replacement is round robin within a set. The Icache can be enabled while memory management is disabled. When memory management is disabled, all of memory is considered cacheable by the Icache.

1.4.5 Data Cache

The SA-110 has a 16 Kbyte data cache (Dcache) with 32-byte blocks and 32-way associativity. The cache supports the flush-all, flush-entry, and copyback-entry functions. The copyback-all function is not supported in hardware. This function can be provided by software. The cache is read allocate with round-robin replacement.

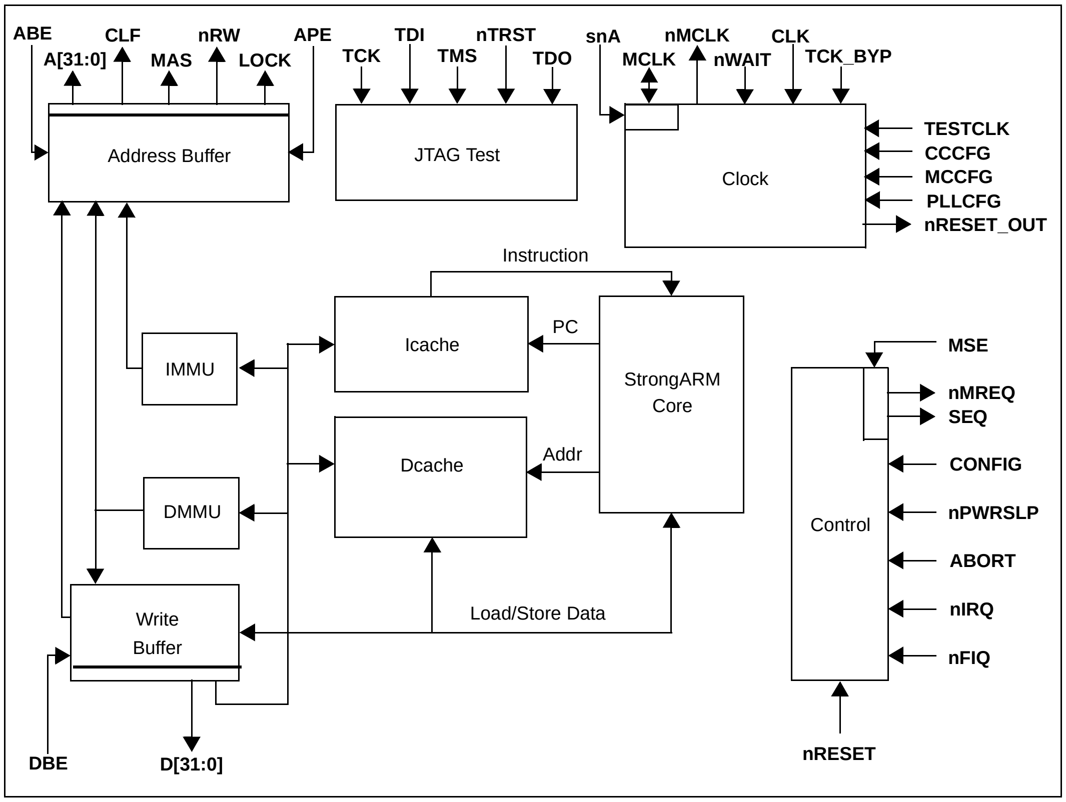
1.4.6 Write Buffer

The SA-110 has an 8-entry write buffer with each entry able to contain 1 to 16 bytes. The drain write buffer operation is supported.

1.5 Block Diagram

Figure 1-1 shows the functional blocks contained in the SA-110.

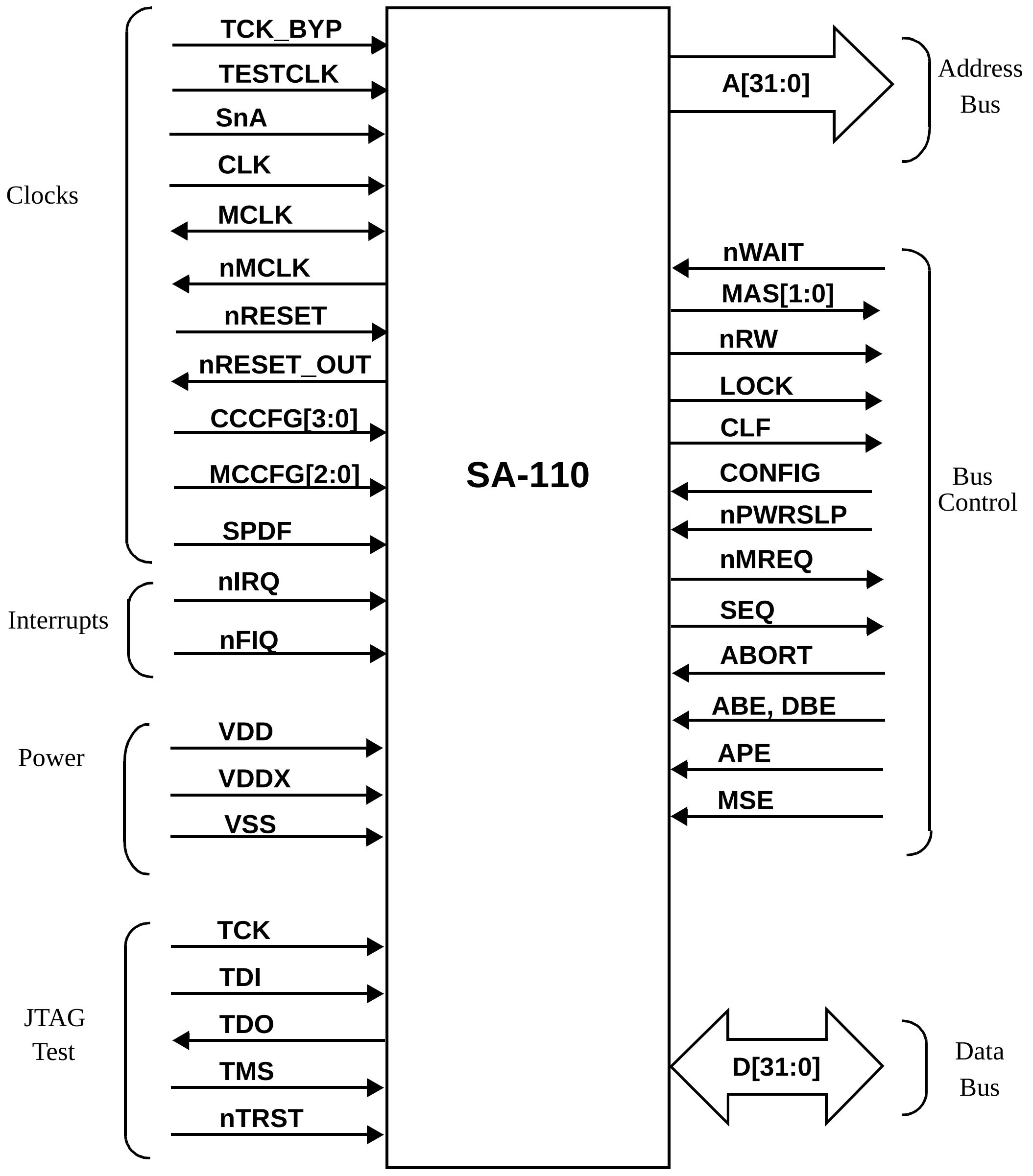
Figure 1-1. SA-110 Block Diagram



1.6 Functional Diagram

Figure 1-2 is a functional diagram of the SA-110.

Figure 1-2. Functional Diagram



Signal Description 2

The following table describes the signals.

Key to Signal Types: **IC** – Input, CMOS threshold

ICOCZ – Input, CMOS threshold, output CMOS levels, tristatable

OCZ – Output, CMOS levels, tristatable

Table 2-1. Signal Pin Description

Name	Type	Description
A[31:0]	OCZ	Address bus. This bus signals the address requested for memory accesses. If **APE** is high, the address changes while **MCLK** is high; if **APE** is low, the address changes during the following low period of **MCLK**. In enhanced bus mode **A[1:0]** contains byte mask bits two and three. The byte mask bits are asserted low on reads to indicate the required data. The byte mask bits are asserted low on writes to signal which bytes of the 32-bit data bus contain data to be accessed. In standard mode **A[1:0]** contains the low-order two bits of the address. In both modes **A[31:2]** contains the upper 30 bits of the byte address.
ABE	IC	Address bus enable. When this input is low, the address bus **A[31:0]**, **nRW**, **MAS[1:0]**, **CLF**, and **LOCK** are put into a high impedance state (Note 1).
ABORT	IC	External abort. Allows the memory system to tell the processor that a requested access has failed. Only monitored when the SA-110 is accessing external memory.
APE	IC	Address pipeline enable. This input is used to control the timing of the latches on the address bus **A[31:0]**, **MAS[1:0]**, **nRW**, **CLF**, and **LOCK**. Normally these signals change during **MCLK** high, but they may be held to the next **MCLK** low by driving **APE** low. **APE** is a static configuration pin and must remain stable at all times.
CCCFG[3:0]	IC	Core clock configuration pins. These four pins configure the core clock speed. They must remain stable at all times.
CLF	OCZ	Cache line fill. An output signal used by the processor to indicate a cache line fill burst read or write. The **CLF** pin is asserted for full 8-word reads and writes. Burst reads are only done for cache line fills. Burst writes are done for castouts of dirty data and data stores that have merged in the write buffer. If **CONFIG** is high, then cache line fetches are wrapped in the four-word subblock and then switched to the other subblock. The **CLF** signal follows address bus timing. Note it is NOT asserted for writes in pass one parts.
CLK	IC	The 3.57/3.68 MHZ crystal oscillator input clock for the core PLL. This is the base clock from which the high speed core clock and the **MCLK** in synchronous mode are generated.
CONFIG	IC	The **CONFIG** input sets the bus mode to standard (low) or enhanced (high). In standard mode all cache line fetches start at word 0 of the cache block and stores are not allowed to merge. If **CONFIG** is high then cache line fetches are wrapped in the 4-word subblock then switch to the other subblock starting at the critical word. The write buffer is allowed to merge stores to the same cache block and will use the **A[1:0]** and **MAS[1:0]** pins to present a byte mask of the bytes being read or written. The **CONFIG** signal must remain stable at all times.
D[31:0]	ICOCZ	Data bus. These are bidirectional pins used for data transfers between the processor and external memory. For read operations (when **nRW** is low), the input data must be valid before the falling edge of **MCLK**. For write operations (when **nRW** is high), the output data will become valid while **MCLK** is low.

Table 2-1. Signal Pin Description

Name	Type	Description
DBE	IC	Data bus enable. When this input is low, the data bus, **D[31:0]**, is put into a high impedance state (Note 1). The drivers will always be high impedance except during write operations, and **DBE** must be driven high in systems that do not require the data bus for DMA or similar activities.
LOCK	OCZ	Locked operation. **LOCK** is driven high to signal a “locked” memory access sequence, and the memory manager should wait until **LOCK** goes low before allowing another device to access the memory. **LOCK** changes while **MCLK** is high and remains high during the locked memory sequence. The **LOCK** signal follows address bus timing.
MAS[1:0]	OCZ	Memory access size. An output signal used by the processor in standard bus mode to indicate to the external memory system the number of bytes being transferred. **MAS[1:0]** is 2 for word transfers, 1 for halfword transfers, and 0 for byte transfers, and is valid for both read and write operations. In enhanced mode, the **MAS** pins contain bits zero and one of the byte mask. The byte mask bits are asserted low on reads to indicate the required data. The byte masks bits are asserted low on writes to signal which bytes of the 32-bit data bus contain data to be accessed. The **MAS** signals follow address bus timing.
MCCFG[2:0]	IC	Memory clock configuration pins. These pins configure the system clock speed when **SnA** is asserted.They must remain stable at all times and may not be dynamically changed. When **SnA** is asserted **MCLK** is generated by dividing the core clock by the value of **MCCFG** plus two (2..9) for **MCCFG** in the range 0..7.
MCLK	ICOCZ	Memory clock input or output. When **SnA** is deasserted **MCLK** is an input clock, when **SnA** is asserted **MCLK** is an output. This clock times all SA-110 memory accesses. The low or high period of **MCLK** may be stretched for slow peripherals; alternatively, the **nWAIT** input may be used with a free-running **MCLK** to achieve similar effects.
MSE	IC	Memory request/sequential enable. When this input is low, the **nMREQ** and **SEQ** outputs are put into a high impedance state (Note 1).
nFIQ	IC	Not fast interrupt request. If **FIQ**s are enabled, the processor will respond to a low-level on this input by taking the **FIQ** interrupt exception. This is an asynchronous, level-sensitive input, and must be held low until a suitable response is received from the processor.
nIRQ	IC	Not interrupt request. As **nFIQ**, but with lower priority. This is an asynchronous, level-sensitive input, and must be held low until a suitable response is received from the processor.
nMCLK	OCZ	Not memory clock output. When **SnA** is high, **nMCLK** is the inverse of **MCLK**. If **SnA** is low then **nMCLK** is held low. This output can also be disabled by an MCR instruction to save power if **nMCLK** is not used.
nMREQ	OCZ	Not memory request. A pipelined signal that changes while **MCLK** is low to indicate whether or not, in the following cycle, the processor will be accessing external memory. When **nMREQ** is low, the processor will be accessing external memory.
nPWRSLP	IC	Power sleep. When low this puts the SA-110 I/O pins into sleep mode. In sleep mode all outputs are driven low except **nMREQ** which is driven high.
nRESET	IC	Not reset. This is a level-sensitive input which is used to start the processor from a known address. A low level will cause the current instruction to terminate abnormally, and the on-chip caches, MMU, and write buffer to be disabled. When **nRESET** is driven high, the processor will restart from address 0. **nRESET** must remain low for at least two full **MCLK** cycles. While **nRESET** is low the processor will perform idle cycles.
nRESET_OUT	OCZ	Not reset out. This signal is asserted when **nRESET** is asserted and deasserts when the processor has completed resetting. This signal remains asserted until the PLL is stable.
nRW	OCZ	Not read/write. When high this signal indicates a processor write operation; when low, a read. The **nRW** signal follows address bus timing.

Table 2-1. Signal Pin Description

Name	Type	Description
nTRST	IC	Test interface reset. Note this pin does NOT have an internal pull-up resistor. This pin must be pulsed or driven low to achieve normal device operation, in addition to the normal device reset (**nRESET**).
nWAIT	IC	Not wait. When low this allows extra **MCLK** cycles to be inserted in memory accesses. It must be asserted low before the rising edge of **MCLK** to extend the **MCLK** cycle and is latched with the rising edge of **MCLK**. Note this is different than previous ARM** processors that required **nWAIT** to be asserted for the entire **MCLK** high time and did not latch **nWAIT**.
PARITY[3:0]	OCZ	Data bus parity. This output signal is the odd byte parity for **D[31:0]**. For write operations (when **nRW** is high), the output data will become valid while **MCLK** is low. For read operations (when **nRW** is low), these pins are UNPREDICTABLE. Note that these pins are not available in the 144-pin package.
PLLCFG	IC	PLL configuration. This pin selects between two sets of divide ratios for the PLL. It is needed to maintain compatibility with the SA-110, while still allowing for higher core clock frequencies. This pin uses one of the n/c pins on the SA-110. This pin has an internal pull-up resistor. If this pin is not driven externally, it will be pulled high, causing the PLL to use the slower SA-110 divide ratios.
SEQ	OCZ	Sequential address. This signal is the inverse of **nMREQ**, and is provided for compatibility with existing ARM memory systems. The signal changes while **MCLK** is high.
SnA	IC	Synchronous/not asynchronous. When **SnA** is low, MCLK is an input and **nMCLK** is driven low. When **SnA** is high, **nMCLK** and **MCLK** are output clocks with the frequency selected by the **MCCFG** and **CCCFG** pins. **SnA** must remain stable at all times and may not be dynamically changed.
SPDF	IC	**SPDF**. The **SPDF** pin should be tied low.
TCK	IC	Test interface reference clock. This times all the transfers on the JTAG test interface.
TCK_BYP	IC	Test clock PLL bypass. When **TCK_BYP** is high, the **TESTCLK** is used as the core clock in place of the PLL clock; when low, the internal PLL output is used.
TDI	IC	Test interface data input. Note this pin does NOT have an internal pull-up resistor.
TDO	OCZ	Test interface data output. Note this pin does NOT have an internal pull-up resistor.
TESTCLK	IC	Test clock. **TESTCLK** is used to provide the core clock when **TCK_BYP** is high. It should be tied low if **TCK_BYP** is low. This clock should never be driven higher than $\mathbf{V_{DD}}$.
TMS	IC	Test interface mode select. Note this pin does NOT have an internal pull-up resistor.
VDD	IC	Positive supply for the core. Eight pins are allocated to **VDD**.
VDDX	IC	Positive supply for the I/O pins. Nine pins are allocated to **VDDX**.
VSS	IC	Ground supply. Eighteen pins are allocated to **VSS**. The ground plane on the board should be the same for these pins.

NOTES:

1. When output pads are placed in the high impedance state for long periods, care must be taken to ensure that they do not float to an undefined logic level, as this can dissipate power, especially in the pads.
2. Unless all inputs are driven to the **VSS** or **VDDX**, the input circuits will consume power.

ARM Implementation Options 3

The following sections describe ARM** architecture options that are implemented by the SA-110.

3.1 Big and Little Endian

The big endian bit in the control register sets whether the SA-110 treats words in memory as being stored in big endian or little endian format. Memory is viewed as a linear collection of bytes numbered upwards from 0. Bytes 0 to 3 hold the first stored word, bytes 4 to 7 the second, and so on.

In the little endian scheme, the lowest numbered byte in a word is considered to be the least significant byte of the word and the highest numbered byte is the most significant. Byte 0 of the memory system should be connected to data lines 7 through 0 (**D[7:0]**) in this scheme.

In the big endian scheme the most significant byte of a word is stored at the lowest numbered byte and the least significant byte is stored at the highest numbered byte. Byte 0 of the memory system should therefore be connected to data lines 31 through 24 (**D[31:24]**).

The state of the big endian bit only changes the location of the bytes within a 32-bit word. The accessed bytes are changed for the load byte, store byte, load halfword, and store halfword instructions only. Instruction fetches and word load and stores are not changed by the state of the big endian bit.

3.2 Exceptions

Exceptions arise whenever there is a need for the normal flow of program execution to be broken; for example, so that the processor can be diverted to handle an interrupt from a peripheral. The processor state just prior to handling the exception must be preserved so that the original program can be resumed when the exception routine has completed. Many exceptions may arise at the same time.

The SA-110 handles exceptions by making use of banked registers to save state. The old PC and CPSR contents are copied into the appropriate R14 and SPSR and the PC and mode bits in the CPSR bits are forced to a value that depends on the exception. Interrupt disable flags are set where required to prevent otherwise unmanageable nestings of exceptions. In the case of a reentrant interrupt handler, R14 and the SPSR should be saved onto a stack in main memory before reenabling the interrupt; when transferring the SPSR register to and from a stack, it is important to transfer the whole 32-bit value, and not just the flag or control fields.

When multiple exceptions arise simultaneously, a fixed priority determines the order in which they are handled. The priorities are listed later in this chapter. Most exceptions are fully defined in the *ARM Architecture Reference*. The following sections specify the exceptions where the SA-110 implementation differs from the *ARM Architecture Reference*.

The SA-110 initiates all exceptions in 32-bit mode. When an exception occurs while the SA-110 is running in 26-bit mode, the SA-110 saves only the PC and CPSR. The PC is saved in R14 and the CPSR is saved in the SPSR of the exception mode. The 32-bit handler must merge the condition codes, the interrupt enables, and the mode from the SPSR into R14 if a handler wants to run in 26-bit mode.

3.2.1 Reset

When the **nRESET** signal goes low, the SA-110 stops executing instructions, asserts the **nRESET_OUT** pin, and then performs idle cycles on the bus.

When **nRESET** goes high again, the SA-110 does the following:

- Overwrites R14_svc and SPSR_svc by copying the current values of the PC and CPSR into them. The values of the saved PC and CPSR are not defined.
- Forces M[4:0]=10011 (32-bit supervisor mode) and sets the I and F bits in the CPSR.
- Forces the PC to fetch the next instruction from address 0x0.

At the end of the reset sequence, the MMU, Icache, Dcache, and write buffer are disabled. Alignment faults are also disabled, and little endian mode is enabled.

3.2.2 Abort

An abort can be signalled by either the internal memory-management unit or from the external abort input pin. An abort indicates that the current memory access cannot be completed. For instance, in a virtual memory system, the data corresponding to the current address may have been moved out of memory onto a disk, and considerable processor activity may be required to recover the data before the access can be performed successfully. The SA-110 checks for an abort during memory access cycles. When aborted, the SA-110 responds in one of two ways:

- If the abort occurred during an instruction prefetch (a *prefetch abort*), the prefetched instruction is marked as invalid but the abort exception does not occur immediately. If the instruction is not executed, for example, as a result of a branch being taken while it is in the pipeline, no abort will occur. An abort will take place if the instruction reaches the head of the pipeline and is about to be executed.
- If the abort occurred during a data access (a *data abort*), the action depends on the instruction type.
  - — Single data transfer instructions (LDR, STR) will abort with no registers modified.
  - — The swap instruction (SWP) is aborted as though it had not executed, though externally the read access may take place.
  - — Block data transfer instructions (LDM, STM) abort on the first access that cannot complete. If write-back is set, the base is **NOT** updated. If the instruction would normally have overwritten the base with data (for example, an LDM instruction with the base in the transfer list), the original value in the base register is restored.

When either a prefetch or data abort occurs, the SA-110 performs the following operations:

- Saves the address of the aborted instruction plus 4 (for prefetch aborts) or 8 (for data aborts) in R14_abt; saves CPSR in SPSR_abt.
- Forces M[4:0]=10111 (abort mode) and sets the I bit in the CPSR.
- Forces the PC to fetch the next instruction from either address 0x0C (prefetch abort) or address 0x10 (data abort).

To return after fixing the reason for the abort, use SUBS PC, R14_abt,#4 (for a prefetch abort) or SUBS PC,R14_abt,#8 (for a data abort). This will restore both the PC and the CPSR and retry the aborted instruction.

The abort mechanism allows a *demand paged virtual memory system* to be implemented when suitable memory-management software is available. The processor is allowed to generate arbitrary addresses, and when the data at an address is unavailable the MMU signals an abort. The processor traps the aborted instruction into system software which must work out the cause of the abort, make the requested data available, and retry the aborted instruction. The application program needs no knowledge of the amount of memory available to it, nor is its state in any way affected by the abort.

3.2.3 Vector Summary

Table 3-1 lists byte addresses, which normally contain a branch instruction pointing to the relevant routines.

Table 3-1. Vector Summary

Address	Exception	Mode on Entry
0x00000000	Reset	Supervisor
0x00000004	Undefined instruction	Undefined
0x00000008	Software interrupt	Supervisor
0x0000000C	Abort (prefetch)	Abort
0x00000010	Abort (data)	Abort
0x00000014	Not used	Not used
0x00000018	IRQ	IRQ
0x0000001C	FIQ	FIQ

3.2.4 Exception Priorities

When multiple exceptions arise at the same time, a fixed priority system determines the order in which they will be handled:

1. Reset (highest priority)
2. Data abort
3. FIQ
4. IRQ
5. Prefetch abort
6. Undefined instruction, software interrupt (lowest priority)

Note: Not all exceptions can occur at once. Undefined instruction and software interrupt are mutually exclusive because they each correspond to particular (nonoverlapping) decodings of the current instruction.

If a data abort occurs at the same time as a FIQ, and FIQs are enabled (that is, the F flag in the CPSR is clear), the SA-110 will enter the data abort handler and then immediately proceed to the FIQ vector. A normal return from FIQ will cause the data abort handler to resume execution. Placing data abort at a higher priority than FIQ is necessary to ensure that the transfer error does not escape detection; the time for this exception entry should be added to worst-case FIQ latency calculations.

3.2.5 Interrupt Latencies

Calculating the worst-case interrupt latency for the SA-110 is quite complex (due to the cache, MMU, and write buffer) and is dependent on the configuration of the whole system.

3.3 Coprocessors

The SA-110 has no external coprocessor bus, so it is not possible to add external coprocessors to this device.

SA-110 uses the internal coprocessor 15 for control of the on-chip MMU, caches, and clocks. If a coprocessor other than coprocessor 15 is accessed, then the CPU will take the undefined instruction trap. The coprocessor load, store, and data operation instructions also take the undefined instruction trap.

Instruction Set 4

This chapter describes the instruction timings for the SA-110. The SA-110 implements the ARM** V4 architecture as defined in the *ARM Architecture Reference*, Version A, dated 7 Feb 1996, with previously noted options.

4.1 Instruction Timings

Table 4-1 lists the instruction timings for the SA-110. The result delay is the number of cycles the next sequential instruction would stall if it used the result as an input. The issue cycles is the number of cycles this instruction takes to issue. For most instructions, the result delay is zero and the issue cycles is one. For load and stores the timing is for cache hits.

Table 4-1. Instruction Timings

Instruction Group	Result Delay	Issue Cycles
Data processing shift amount literal	0	1
Data processing shift amount from register	0	2
Mul or Mul/Add giving 32-bit result	1..3	1
Mul or Mul/Add giving 64-bit result	1..3	2
Load single – write-back of base	0	1
Load single – load data zero extended	1	1
Load single – load data sign extended	2	1
Store single – write-back of base	0	1
Load multiple (delay for last register)	1	MAX (2, number of registers loaded)
Store multiple – write-back of base	0	MAX (2, number of registers loaded)
Branch or branch and link	0	1
MCR	2	1
MRC	1	1
MSR to control	0	3
MRS	0	1
Swap	2	2

Configuration 5

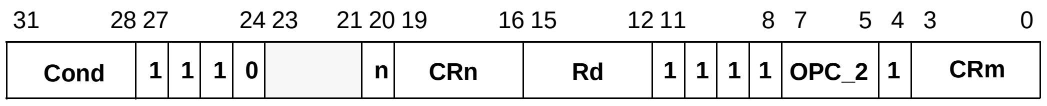
The operation and configuration of the SA-110 is controlled with coprocessor instructions, configuration pins, and with the memory-management page tables. The coprocessor instructions manipulate on-chip registers that control the configuration of the cache, write buffer, MMU and a number of other configuration options.

Note: The shaded areas in the register and translation diagrams are reserved and should be programmed 0 for future compatibility.

5.1 Internal Coprocessor Instructions

The on-chip registers may be read using MRC instructions and written using MCR instructions. These operations are allowed only in nonuser modes and the undefined instruction trap will be taken if accesses are attempted in user mode. Figure 5-1shows the format of the internal coprocessor instructions MRC and MCR.

Figure 5-1. Format of Internal Coprocessor Instructions MRC and MCR



Cond	– ARM** condition codes
n	– 1 MRC register read 0 MCR register write
CRn	– SA-110 register
Rd	– ARM register
OPC_2	– Function bits for some MRC/MCR instructions
CRm	– Function bits for some MRC/MCR instructions

5.2 Registers

The SA-110 contains registers that control the cache and MMU operation. These registers are accessed using CPRT instructions to coprocessor 15 with the processor in any privileged mode. Only some of registers 0..15 are valid; the result of an access to an invalid register is UNPREDICTABLE. Table 5-1 shows the SA-110 registers.

Table 5-1. SA-110 Registers

Register	Register Reads	Register Writes
0	ID register	Reserved
1	Control	Control
2	Translation table base	Translation table base
3	Domain access control	Domain access control
4	Reserved	Reserved
5	Fault status	Fault status
6	Fault address	Fault address
7	Reserved	Cache operations
8	Reserved	TLB operations
9..14	Reserved	Reserved
15	Reserved	Test, clock, and idle

5.2.1 Register 0 – ID

Register 0 is a read-only register that returns the code for this chip: 0x4401A10x. The low-order four bits of the register are the chip revision number.

31 24	23 16	15 4	3 0
44	**01**	**A10**	**Revision**

5.2.2 Register 1 – Control

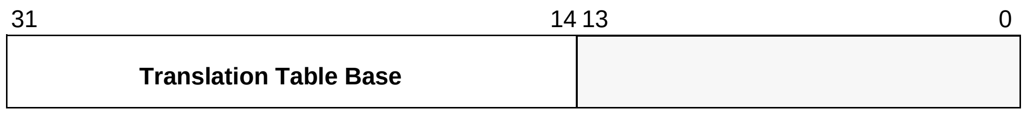
Register 1 is read/write register and contains control bits. All writable bits in this register are forced low by reset. The shaded bits (also labeled r) are reserved and are not readable or writable.

31	30																		12			9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	I	0	0	R	S	B	1	1	1	W	C	A	M

Bit	Name	Description
0	M	Enable/Disable 0 – On-chip memory-management unit disabled. 1 – On-chip memory-management unit enabled.
1	A	Address Fault Enable/Disable 0 – Alignment fault disabled. 1 – Alignment fault enabled.
2	C	Data Cache Enable/Disable 0 – Data cache disabled. 1 – Data cache enabled.
3	W	Write Buffer Enable/Disable 0 – Write buffer disabled. 1 – Write buffer enabled.
7	B	Big/Little Endian 0 – Little endian operation. 1 – Big endian operation.
8	S	System This selects the access checks performed by the memory-management unit. See the *ARM Architecture Reference* for more information.
9	R	ROM This selects the access checks performed by the memory-management unit. See the *ARM Architecture Reference* for more information.
12	I	Instruction Cache Enable/Disable 0 – Instruction cache disabled. 1 – Instruction cache enabled.
13..31	—	Unused. Undefined on read. Writes ignored.

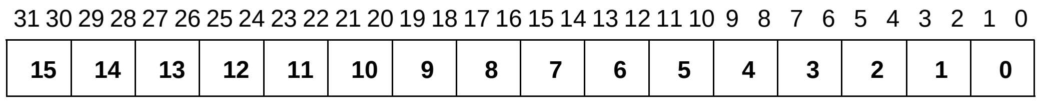
5.2.3 Register 2 – Translation Table Base

Register 2 is a read/write register that holds the base of the currently active level 1 page table. Bits [13:0] are undefined on read, ignored on write.



5.2.4 Register 3 – Domain Access Control

Register 3 is a read/write register that holds the current access control for domains 0 to 15.

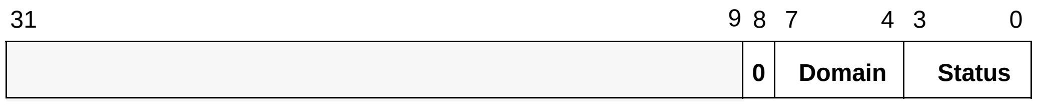


5.2.5 Register 4 – RESERVED

Register 4 is reserved. Accessing this register has no effect.

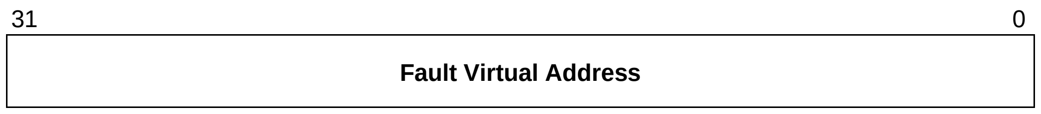
5.2.6 Register 5 – Fault Status

Register 5 is a read/write register. Reading this register returns the current contents of the fault status register (FSR). The FSR is written when a data memory fault occurs or can be written by an MCR to the FSR. It is not updated for a prefetch fault. See the Chapter 7, “Memory-Management Unit (MMU)” for more details. Bits [31:9] are undefined on read, ignored on write. Bit 8 is ignored on write and is always returned as zero.



5.2.7 Register 6 – Fault Address

Register 6 is a read/write register. Reading this register returns the current contents of the fault address register (FAR). The FAR is written when a data memory fault occurs with the address of the data fault or can be written by an MCR to the FAR.



5.2.8 Register 7 – Cache Control Operations

Register 7 is a write-only register. The CRm and OPC_2 fields are used to encode the cache control operations. All other values for OPC_2 and CRm are UNPREDICTABLE.

Table 5-2. Cache Control Operations

Function	OPC_2	CRm	Data
Flush I+D	0b000	0b0111	Ignored
Flush I	0b000	0b0101	Ignored
Flush D	0b000	0b0110	Ignored
Flush D single entry	0b001	0b0110	Virtual address
Clean D cache entry	0b001	0b1010	Virtual address
Drain write buffer	0b100	0b1010	Ignored

5.2.9 Register 8 – TLB Operations

Register 8 is a write-only register. The CRm and OPC_2 fields are used to encode the following TLB flush operations. All other values for OPC_2 and CRm are UNPREDICTABLE.

Table 5-3. TLB Control Operations

Function	OPC_2	CRm	Data
Flush I+D	0b000	0b0111	Ignored
Flush I	0b000	0b0101	Ignored
Flush D	0b000	0b0110	Ignored
Flush D single entry	0b001	0b0110	Virtual address

5.2.10 Registers 9..14 – RESERVED

The results of accessing any of these registers is UNPREDICTABLE.

5.2.11 Register 15 – Test, Clock, and Idle Control

Register 15 is a write-only register. The CRm and OPC_2 fields are used to encode the following control operations. All other values for OPC_2 and CRm are UNPREDICTABLE.

Table 5-4. Test, Clock, and Idle Controls

Function	OPC_2	CRm
Enable odd word loading of Icache LFSR.	0b001	0b0001
Enable even word loading of Icache LFSR.	0b001	0b0010
Clear Icache LFSR.	0b001	0b0100
Move LFSR to R14. Abort.	0b001	0b1000
Enable clock switching.	0b010	0b0001
Disable clock switching.	0b010	0b0010
Disable nMCLK output.	0b010	0b0100
Wait for interrupt.	0b010	0b1000

5.2.11.1 Icache LFSR Controls

The OPC_2=1 functions are used to control the Icache linear feedback shift register (LFSR) that is used for manufacturing tests.

5.2.11.2 Clock Controls

The four OPC_2=2 functions are used for enabling and disabling **DCLK** switching, disabling the **nMCLK** output, disabling **MCLK** output, and waiting for an interrupt. See the Chapter 8, “Clocks” for information.

Caches and Write Buffer 6

To reduce effective memory access time, the SA-110 has an instruction cache, a data cache, and a write buffer. In general, these are transparent to program execution. The following sections describe each of these units and give all necessary programming information.

6.1 Instruction Cache (Icache)

The SA-110 contains a 16 Kbyte instruction cache (Icache). The Icache has 512 lines of 32 bytes (8 words), arranged as a 32-way set-associative cache, and uses the virtual addresses generated by the processor core. The Icache is always reloaded a line at a time (8 words). It may be enabled or disabled via the SA-110 control register and is disabled by the **nRESET** pin. The operation of the cache, when memory management is enabled, is further controlled by the *cacheable* or C bit stored in the memory-management page table (see the Chapter 7, “Memory-Management Unit (MMU)”). If memory management is disabled, all addresses are marked as cacheable (C=1). When memory management is enabled the C bit in each page table entry can disable caching for an area of virtual memory.

6.1.1 Icache Operation

In the SA-110 the Icache will be searched regardless of the state of the C bit; only reads that miss the Icache are affected. If, on an Icache miss, the C bit is a one or the MMU is disabled, a cache line fill of 8 words is performed and it is placed in an Icache bank with a round-robin replacement algorithm. If, on a miss, the MMU is enabled and the C bit is a zero for the given virtual address, an external memory access for a single word is performed and the Icache will not be written. The Icache should be enabled as soon as possible after reset for best performance.

6.1.2 Icache Validity

The Icache operates with virtual addresses, so care must be taken to ensure that its contents remain consistent with the virtual-to-physical mappings performed by the memory-management unit. If the memory mappings are changed, the Icache validity must be ensured. The Icache is not coherent with stores to memory, so programs that write cacheable instruction locations must ensure the Icache validity. Instruction fetches do not check the write buffer, so data must not only be pushed out of the Icache but the write buffer must also be drained.

6.1.2.1 Software Icache Flush

The entire Icache can be invalidated by writing to the SA-110 cache operations register (register 7). The cache will be flushed immediately when the register is written, but note that the following instruction fetches may come from the cache before the register is written.

6.1.3 Icache Enable/Disable and Reset

The Icache is automatically disabled and flushed on **nRESET**. Once enabled, cacheable read accesses cause lines to be placed in the Icache. If the Icache is subsequently disabled, no new lines will be placed in the Icache, but the Icache is still searched and if the data is found, it will be used by the processor. If the data in the Icache must not be used, then the Icache must be flushed.

6.1.3.1 Enabling the Icache

To enable the Icache, set bit 12 in the control register. The MMU and Icache may be enabled simultaneously with a single control register write.

6.1.3.2 Disabling the Icache

To disable the Icache, clear bit 12 in the control register.

6.2 Data Cache (Dcache)

The SA-110 contains a 16 Kbyte write-back data cache (Dcache). The Dcache has 512 lines of 32 bytes (eight words), arranged as a 32-way set-associative cache, and uses the virtual addresses generated by the processor. A line also contains the physical address the block was fetched from and two dirty bits. There is a dirty bit associated with both the first and second half of the block. When a store hits in the Dcache, the dirty bit associated with it is set. When a block is evicted from the Dcache, the dirty bits are used to decide if all, half, or none of the block will be written back to memory using the physical address stored with the block. The Dcache is always reloaded a line at a time (eight words). It can be enabled or disabled via the SA-110 control register and is disabled on **nRESET**. The operation of the Dcache is further controlled by the *cacheable* or C bit and the *bufferable* or B bit stored in the memory-management page table. See Chapter 7, "Memory-Management Unit (MMU)". For this reason, in order to use the Dcache, the MMU must be enabled. However, the two functions can be enabled simultaneously with a single write to the control register.

6.2.1 Cacheable Bit – C

The cacheable bit determines whether data being read can be placed in the Dcache and used for subsequent read operations. Typically main memory is marked as cacheable to improve system performance, and I/O space as noncacheable to stop the data from being stored in SA-110's Dcache. For example, if the processor is polling a hardware flag in I/O space, it is important that the processor is forced to read data from the external peripheral, and not a copy of initial data held in the cache.

6.2.2 Bufferable Bit – B

The bufferable bit does not affect writes that hit in the Dcache. If a store hits in the Dcache, the store is assumed to be bufferable. Write-backs of dirty lines are treated as bufferable writes. See Section 6.3, "Write Buffer (WB)" on page 6-4 for more information on the B bit.

6.2.3 Dcache Operation

The Dcache is searched regardless of the state of the C bit; only reads that miss the Dcache check the C bit. The C bit controls loading the Dcache on a miss not checking the Dcache on an access.

6.2.3.1 Cacheable Reads – C=1

A cache line fill of of eight words is performed and it is placed in a Dcache bank with a round-robin replacement algorithm.

6.2.3.2 Noncacheable Reads – C=0

An external memory access is performed and the Dcache is not written to.

Note: Load multiples to C = 0 space do not perform a burst read.

6.2.4 Dcache Validity

The Dcache operates with virtual addresses, so care must be taken to ensure that its contents remain consistent with the virtual-to-physical mappings performed by the memory management unit. If the memory mappings are changed, the Dcache validity must be ensured.

6.2.4.1 Software Data Cache Flush

SA-110 supports the flush and clean operations on single entries of the Dcache by writes to the cache operations registers. The flush whole cache is also supported. Note that since this is a write-back cache, in order to prevent the loss of data, a flush whole must be preceded by a sequence of loads to cause the cache to write back any dirty entries.

.The following code causes the main data cache to flush all dirty entries.

```
;+
;Call:
;   R0 points to the start of a 16384 byte region of readable data used
;   only for this cache flushing routine. If this area is to be used
;   by other code, then 32 K must be loaded and the flush MCR is not needed
;   bl    writeBackDcache
;Return:
;   R0, R1, R2 trashed
;   Data cache is clean
;-
writeBackDcache
    add r1, r0, #16384
l1
    ldr r2, [r0], #32
    teq r1, r0
    bne l1
    mcr p15, 0, r0, c7, c6, 0
    mov pc, r14
```

6.2.4.2 Doubly-Mapped Space

Since the caches work with virtual addresses, it is assumed that every virtual address maps to a different physical address. If the same physical location is accessed by more than one virtual address, the cache cannot maintain consistency, since each virtual address has a separate entry in the cache, and only one entry is updated on a processor write operation. To avoid any cache inconsistencies, doubly mapped virtual addresses should be marked as noncacheable.

6.2.5 Dcache Enable/Disable and Reset

The Dcache is automatically disabled and flushed on the assertion of **nRESET**. Once enabled, cacheable read accesses place lines in the cache. If subsequently disabled, no new lines are placed in the cache; it is searched and if data is found, it is used by the processor. Write operations continue to update the cache, thus maintaining consistency with the external memory. If the data in the cache must not be used, then the cache must be flushed.

6.2.5.1 Enabling the Dcache

To enable the Dcache, set bit 0 in the control register to enable the MMU and set bit 2 in the control register to enable the Dcache. The MMU and Dcache can be enabled simultaneously with a single control register write.

6.2.5.2 Disabling the Dcache

To disable the data caches, clear bit 2 in the control register.

6.3 Write Buffer (WB)

The SA-110 write buffer is provided to improve system performance. It can buffer up to eight blocks of data of 1 to 16 bytes, at independent addresses. It can be enabled or disabled via the W bit (bit 3) in the SA-110 control register and the buffer is disabled and all entries are marked empty on reset. The operation of the write buffer is further controlled by the *cacheable* or C bit and the *bufferable* or B bit, which are stored in the memory-management page tables. For this reason, in order to use the write buffer, the MMU must be enabled. The two functions can be enabled simultaneously with a single write to the control register. For a write to use the write buffer, both the W bit in the control register and the B bit in the corresponding page table must be set. It is not possible to abort buffered writes externally; the abort pin will be ignored.

6.3.1 Bufferable Bit

This bit controls whether a write operation can use the write buffer. Typically, main memory is bufferable and I/O space unbufferable.

6.3.2 Write Buffer Operation

When the CPU performs a store, the Dcache is first checked. If the Dcache hits on the store and the protection for the location and mode of the store allows the write, then the write completes in the cache and the write buffer is not used. If the location misses in the Dcache, then the translation entry for that address is inspected and the state of the B and C bits determines which of the three following actions are performed. If the write buffer is disabled via the SA-110 control register, writes are treated as if the B it is a zero.

6.3.2.1 Writes to a Bufferable and Cacheable Location (B=1, C=1)

If the write buffer is enabled, and the **CONFIG** pin is asserted, and the processor performs a write to a bufferable and cacheable location, and the data is in the Dcache, then the data is written to the Dcache and the Dcache line is marked dirty. If a write to a bufferable area misses in the data cache, the data is placed in the write buffer and the CPU continues execution. When placing the write data into the write buffer, if the data is being written to the same 16 byte aligned area of the previous write, then the current write is merged into the same entry in the write buffer. The write buffer performs the external write some time later. If a write is done and the write buffer is full, then the processor is stalled until there is sufficient space in the buffer. If the **CONFIG** pin is not asserted, then no merging is allowed.

6.3.2.2 Writes to Bufferable and Noncacheable Location (B=1, C=0)

If the write buffer is enabled and the processor performs a write to a bufferable but noncacheable location and misses in the data cache, the data is placed in the write buffer and the CPU continues execution. When placing the write data into the write buffer no merging is allowed and each write takes it own write buffer entry. The write buffer performs the external write some time later.

Note: The case of a data cache hit for a bufferable but noncacheable location is a system programming bug. In this case, the data is written to the Dcache and the Dcache line is marked dirty.

6.3.2.3 Unbufferable Writes (B=0)

If the write buffer is disabled or the CPU performs a write to an unbufferable area, the processor is stalled until the write buffer empties and the write completes externally. This will require several external clock cycles.

6.3.3 Enabling the Write Buffer

To enable the write buffer, ensure the MMU is enabled by setting bit 0 in the control register, then enable the write buffer by setting bit 3 in the control register. The MMU and write buffer can be enabled simultaneously with a single write to the control register.

6.3.4 Disabling the Write Buffer

To disable the write buffer, clear bit 3 in the control register. Note that any writes already in the write buffer will complete normally, but a drain write buffer needs to be done to force all writes out to memory. Note the write buffer will be used for copybacks from the Dcache even when it is disabled.

Memory-Management Unit (MMU) 7

The SA-110 implements the standard ARM** memory-management functions using two 32-entry fully associative TLBs. One is used for instruction accesses and the other for data accesses. On a TLB miss, the translation table hardware is invoked to retrieve the translation and access permission information. Once retrieved, if the entry maps to a valid page or section, then the information is placed into the TLB. The replacement algorithm in the TLB is round robin. For an invalid page or section, an abort is generated and the entry is not placed in the TLB.

7.1 MMU Registers

See the Section 5.2, "Registers" on page 5-2 for a description of the MMU coprocessor 15 registers supported by the SA-110.

7.2 MMU Faults and CPU Aborts

The MMU generates four faults:

- Alignment fault
- Translation fault
- Domain fault
- Permission fault

Alignment faults are generated by word loads or stores with the low-order two address bits not zero, and by load or store half words when the low-order address bit is a one. Translation faults are generated by access to pages marked invalid by the memory-management page tables. Domain faults and permission faults are generated by accesses to memory disallowed by the current mode, domain, and page protection. See the *ARM Architecture Reference* for more information.

7.3 External Aborts

In addition to the MMU-generated aborts, the SA-110 has an external abort pin that can be used to flag an error on an external memory access. However, some accesses aborted in this way are not restartable. Not all accesses can be aborted in this way, so this pin must be used with great care. Writes to memory areas marked as bufferable ignore the external abort pin.

The following accesses can be aborted and restarted safely. If any of the following accesses are aborted, the external access will cease on the next cycle. In the case of a read-lock-write sequence in which the read aborts, the write will not happen.

- Reads
- Unbuffered writes
- Level one descriptor fetch
- Level two descriptor fetch
- Read-lock-write sequence

7.3.1 Cacheable Reads (Cache Line Fills)

A cache line fill can be safely aborted on any word in the transfer. If an abort occurs during the cache line fill, the cache is purged so it will not contain invalid data. It the abort happens before the word that was requested by the access is returned, the load is aborted. It the abort happens after the word that was requested by the access is returned, the load completes and the fill is aborted (but no exception is generated).

7.3.2 Buffered Writes

Buffered writes cannot be externally aborted. Therefore, the system should be configured such that it does not do buffered writes to areas of memory that are capable of flagging an external abort.

7.4 Interaction of the MMU, Icache, Dcache and Write Buffer

The MMU, Icache, Dcache, and write buffer can be enabled or disabled independently. The Icache can be enabled with the MMU enabled or disabled. However, the Dcache and write buffer can only be enabled when the MMU is enabled. Because the write buffer is used to hold dirty copy-back cached lines from the Dcache, it must be enabled along with the Dcache. Table 7-1 lists the only four of the eight combinations of the MMU, Dcache, and write buffer enables that are valid. There are no hardware interlocks on these restrictions, so invalid combinations will cause undefined results.

Table 7-1. Valid MMU, Dcache and Write Buffer Combinations

MMU	Dcache	Write Buffer
Off	Off	Off
On	Off	Off
On	Off	On
On	On	On

To enable the MMU:

1. Program the translation table base and domain access control registers.
2. Program level 1 and level 2 page tables as required.
3. Enable the MMU by setting bit 0 in the control register.

Note: Care must be taken if the translated address differs from the untranslated address because the three instructions following the enabling of the MMU may have been fetched using “flat translation”, and enabling the MMU may be considered as a branch with delayed execution. A similar situation occurs when the MMU is disabled. Consider the following code sequence:

```
MOV                         R1, #0x1
MCR                         15,0,R1,0,0; Enable MMU
Fetch Flat
Fetch Flat
Fetch Flat
Fetch Translated
```

To disable the MMU:

1. Disable the write buffer by clearing bit 3 in the control register.
2. Disable the Dcache by clearing bit 2 in the control register.
3. Disable the Icache by clearing bit 12 in the control register.
4. Disable the MMU by clearing bit 0 in the control register.

Note: If the MMU is enabled, then disabled and subsequently reenabled, the contents of the TLB are preserved. If these settings are invalid, the TLB should be flushed before reenabling the MMU.

All three functions can be disabled simultaneously.

Clocks 8

This chapter describes the SA-110 clocks.

8.1 SA-110 Operating Modes

The SA-110 supports three operating modes: normal, idle, and sleep. The SA-110 can be switched between the three operating modes to minimize power consumption. In normal mode, the chip executes instructions. In idle mode, no instructions are executed but the internal phase-locked loop (PLL) continues to run. In sleep mode, the core power (**VDD**) is switched off; the chip does not execute instructions, and the PLL does not run. The I/O power, **VDDX**, is used to hold all of the output pins that are enabled at a low state except **nMREQ**, which is held at a high state.

8.2 SA-110 Clocking

The SA-110 receives a 3.57/3.68-MHz clock (**CLK**) from a crystal-based clock generator on the **CLK** pin and uses an internal PLL to lock to the input clock and multiply the frequency by a variable multiplier to produce a high-speed core clock (**CCLK**). The 3.57/3.68-MHz oscillator and PLL run constantly in normal and idle modes. There are two clocking domains in the SA-110:

- Core logic domain clocked by **DCLK**
- Bus interface domain clocked by **MCLK**

The core clock, **DCLK**, switches between being driven by the high-speed core clock, **CCLK** and the bus clock, **MCLK**. **CCLK** is used except when SA-110 is waiting for fills to complete after a cache miss.

At RESET, clock switching is disabled and the **DCLK** is driven from **MCLK**. The time from the deassertion of the **nRESET** (and the **nPWRSLP**) pin until the chip comes out of reset (and **nRESET_OUT** deasserts) is about 150 µsecs and is asynchronous (to **MCLK**). The clock switching can also be disabled by writing to the coprocessor 15 register 15 with OPC_2=2 and CRm=2. Clock switching is enabled by writing to the coprocessor 15 register 15 with OPC_2=2 and CRm=1. Disabling switching only disables switching for **DCLK**; it does not force the **DCLK** to **MCLK**. To force **DCLK** to be driven by **MCLK** after disabling switching, force an instruction or data cache miss.

The SA-110 can be configured to either source or sink the **MCLK** pin. If **SnA** is deasserted, **MCLK** is an input to the SA-110. If **SnA** is asserted, the SA-110 drives the **MCLK** pin with clock generated by dividing **CCLK** by 2 to 9 as specified by the **MCCFG** pins. The SA-110 synchronizes signals, which cross between the **CCLK** and **MCLK** domains. The **SnA** pin should be programmed by connecting it to **VDDX** or **VSS** and should not change. With **SnA** asserted **nMCLK**, an inverted copy of **MCLK** is also provided. It is enabled at reset and can be disabled by writing to the coprocessor 15 register 15 with OPC_2=2 and CRm=4 to save power if it is not used. If **SnA** is a 1, then **MCLK** and **nMCLK** start clocking about 10 µsecs before the **nRESET_OUT** deasserts. If **SnA** is deasserted, the **nMCLK** pin is held high. The only difference between running with **SnA** asserted or deasserted is that **MCLK** is driven into or out of the SA-110. The **SnA** mode is intended to provide a low-cost system clock. For the highest performance systems, an external clock generator may be a better choice.

8.2.1 Switching to Idle Mode

With **SnA** deasserted, idle mode is entered by disabling clock switching and then doing a load from a noncacheable location (C=0). The external driver of **MCLK** should then stop **MCLK** to stop the SA-110 from executing instructions. To resume normal operation, **MCLK** is restarted and clock switching is enabled.

With **SnA** asserted, idle mode is entered by disabling clock switching, then doing a load from a noncacheable location (C=0), and then doing a wait for interrupt instruction. The SA-110 stops driving **MCLK** when it is low and holds **MCLK** low. Asserting either **nFIQ** or **nIRQ** will cause **MCLK** to start clocking again. To resume normal operation, reenable clock switching.

8.2.2 Switching to Sleep Mode

To enter sleep mode, first assert **nRESET** for at least 20 ns and then assert **nPWRSLP**, and then switch off the **VDD** power (drive **VDD** to 0 V). To exit sleep mode, restore the **VDD** supply and deassert **nRESET** and **nPWRSLP**. The order of deassertion of **nRESET** and **nPWRSLP** does not matter. Note there is no difference between initial reset and leaving sleep mode.

8.2.3 Core Clock Configuration (CCCFG)

The high-speed core clock frequency is configured at reset by the four core clock configuration pins (**CCCFG[3:0]**). These pins should be programmed by connecting them to **VDDX** or **VSS** and should not change.

Table 8-1. CCLK Configurations

CCCFG	CCLK Frequency in MHz (3.57-MHz Input)	CCLK Frequency in MHz (3.68-MHz Input)
0	85.7	88.3
1	92.7	95.6
2	96.4	99.4
3	103.5	106.7
4	139.2	143.5
5	146.4	150.9
6	157.1	161.9
7	164.2	169.3
8	185.6	191.3
9	196.3	202.4
10	207	213.4
11	221.3	228.1
12	235.5	242.8
13	249.9	257.6
14	267.8	276
15	278.4	287

8.2.4 Memory Clock Configuration (MCCFG)

If **SnA** is asserted, then the **MCLK** frequency is selected by the three memory clock configuration pins (**MCCFG[2:0]**). These pins should be programmed by connecting them to **VDDX** or **VSS** and should not change. Table 8-2 lists the **MCLK** divisors as a function of the **MCCFG** pins. With **SnA** asserted, **MCLK** is a 50% duty cycle clock. The nominal frequency is the **CCLK** frequency divided by the selected divisor. With **SnA** asserted, the low and high period of **MCLK** is generated at ±5% of the nominal frequency short term and at the error of the crystal oscillator long term.

Note: Some combinations of **MCCFG** and **CCCFG** settings will produce an **MCLK** output that exceeds the maximum supported **MCLK** frequency. For pass one parts, an odd divisor *n* with a CCLK cycle time of Tcycle will produce an **MCLK** with a high time of Tcycle*(*n*-1)/2 and a low time of Tcycle*(*n*+2)/2.

Table 8-2. MCLK Configurations

MCCFG	CCLK Divisor
0	2
1	3
2	4
3	5
4	6
5	7
6	8
7	9

8.2.5 Tester and Debug Clocks

If **TCK_BYP** is high, then the PLL output is not used and the high-speed core clock is supplied externally on the **TESTCLK** pin. This mode is for testing purposes only and is not supported for standard operation. There are important restrictions on the use of **TESTCLK** that must be observed to avoid permanent damage to the chip. Casual use of **TESTCLK** is strongly discouraged.

Bus Interface 9

This section describes the external bus interface for the SA-110.

9.1 Bus Modes

The SA-110 bus has two modes: standard and enhanced. When in enhanced mode, data cache line fills are wrapped to provide the critical word first and the write buffer allows more generalized merging. The **CONFIG** signal is asserted (=1) for enhanced mode and deasserted (=0) for standard mode. The **CONFIG** signal can only change while **nRESET** is asserted.

9.1.1 Standard and Enhanced Mode

In enhanced mode, line fills are performed with the critical word first. The burst read is wrapped at the end of the 4-word cache subblock, then the other subblock is accessed in the same order. In standard mode, line fills are started on cache block word 0 so no wrap is needed. The byte mask is presented on the **A[1:0]** and **MAS[1:0]** pins and the byte mask bits are asserted low on reads to indicate the required data.

Note: This matches current SRAMs, burst mode EDO DRAMs, and SDRAMs.

Table 9-1. Enhanced Mode Wrapping Order

Miss Address Bits [4:2]	Fill Order
0	0, 1, 2, 3, 4, 5, 6, 7
1	1, 2, 3, 0, 5, 6, 7, 4
2	2, 3, 0, 1, 6, 7, 4, 5
3	3, 0, 1, 2, 7, 4, 5, 6
4	4, 5, 6, 7, 0, 1, 2, 3
5	5, 6, 7, 4, 1, 2, 3, 0
6	6, 7, 4, 5, 2, 3, 0, 1
7	7, 4, 5, 6, 3, 0, 1, 2

In enhanced mode, the write buffer is allowed to merge random byte, half word, and word stores. On each bus cycle the word address is specified on the **A[31:2]** pins and the bytes to be read/written are specified by a byte mask. The byte mask bits [3:0] specify that bytes [3:0] of the data bus are being transferred. The byte mask is presented on the **A[1:0]** and **MAS[1:0]** pins and is asserted low. (See Table 9-2.)

Table 9-2. Byte Mask

Byte Mask Pins	Dbus Bytes
A[1]	D[31:24]
A[0]	D[23:16
MAS[1]	D[15:8]
MAS[0]	D[7:0]

In standard mode, the write buffer will only merge stores from store multiple instructions and castouts. In standard mode, a **MAS[1:0]** of 2 is for a word, a 1 is for a halfword, and a 0 is for a byte. The full byte address of the transfer is given by **A[31:0]**.

Read data must always be returned on the correct byte of the data bus **D[31:0].** The chip provides write data on the data bus bytes specified by the byte mask in enhanced mode or by the low-order two address bits and the memory access size in standard mode.

Note: Unlike previous ARM** chips, the SA-110 does not replicate byte stores to all four bytes of the data bus.

9.2 SA-110 Bus Stalls

The bus interface is controlled by **MCLK**, and all timing parameters are referenced with respect to this clock. The way the bus is stalled depends on the mode of the bus. If the system is providing **MCLK** (**SnA**=0), the bus is stalled in one of two ways.

- The low or high phases of the clock may be stretched.
- **nWAIT** can be used to insert entire **MCLK** cycles into the access. When low, this signal maintains the low phase of the cycle by gating out **MCLK** internally. The signal **nWAIT** is asserted before the rising edge of **MCLK** and latch by the SA-110.

If the SA-110 is providing **MCLK** (**SnA**=1), only the **nWAIT** signal can be used to stall the bus with the same timing as with **SnA**=0.

The abort input is not sampled during clock cycles where **nWAIT** is stalling the bus.

9.3 Cycle Types

There are two cycle types performed by the SA-110. These are *idle* cycles and *sequential* cycles. Idle and sequential cycles are combined to perform memory accesses. The two cycle types are differentiated by **nMREQ**. The **SEQ** signal is the inverse of **nMREQ** and is provided for compatibility with earlier memory controllers. Because **nMREQ** and **SEQ** are pipelined, their value determines what type the following cycle will be. **nMREQ** and **SEQ** become valid during the low phase of the cycle before the one to which they refer. Table 9-3 shows the encoding used to generate the idle and sequential cycles.

Table 9-3. Cycle Type Encodings

nMREQ	Cycle Type
0	Sequential
1	Idle

The address from the SA-110 becomes valid during the high phase of **MCLK,** but can be delayed by using **APE**. It is also pipelined, and its value refers to the following memory access.

9.4 Memory Access

There are two types of memory access. These are *nonsequential and sequential*. The nonsequential cycles occur when a new memory access takes place. Sequential cycles occur when the cycle is of the same type as, and the address is 1 word (4 bytes) greater than, the previous access or during cache line fills. Cache line fill cycles occur in response to a cache miss and the address sequence is given above. Burst reads are only done for cache line fills. Burst writes are done for castouts of dirty data and data stores that have merged in the write buffer. Cache line fills have the **CLF** high and use the defined wrapping order but otherwise are the same as sequential cycles. **CLF** has the same timing as the address.

Nonsequential accesses consist of an idle cycle followed by a sequential cycle; sequential accesses consist simply of a sequential cycle.

9.5 Read/Write

Memory accesses may be read or write, differentiated by the signal **nRW**. This signal has the same timing as the address. In the case of a write, the SA-110 outputs data on the data bus during the memory cycle. They become valid during **MCLK** low, and are held until the end of the cycle. In the case of a read, the data is sampled at the end of the memory cycle. **nRW** may not change during a sequential or cache line fill access, so if a read from address A is followed immediately by a write to address (A+4), then the write to address (A+4) would be a nonsequential access.

9.6 Address Pipeline Enable (APE)

APE is used to control the timing of the address signals including **A[31:0], MAS[1:0**], **nRW, CLF**, and **LOCK**. Normally these signals change during **MCLK** high, but they may be held to the next **MCLK** low if **APE** is low. **APE** is a static configuration pin and must remain stable at all times. The remaining sections in this chapter assume **APE** is high. If **APE** is low the address will be held a **MCLK** phase longer and be driven an **MCLK** phase later.

9.7 Memory Access Types

The following timing diagrams assume **APE** is high and are only included to demonstrate timing relationships.

Figure 9-1 shows a one-word read or write memory access. Figure 9-2 shows a two-word sequential access. Figure 9-3 shows two one-word accesses back-to-back. Notice that an idle cycle followed by a sequential cycle distinguishes a nonsequential access.

Figure 9-1. **Read or Write (One Word)**

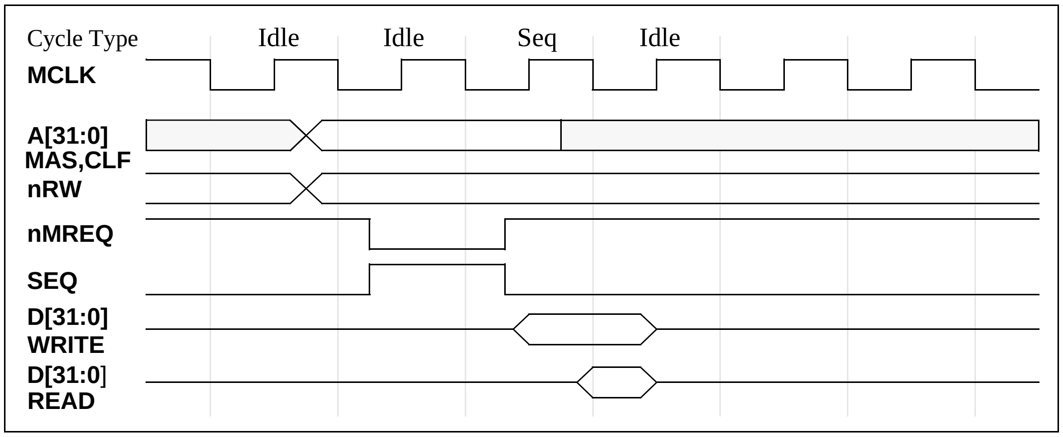


Figure 9-2. **Sequential Read or Write (Two Word)**

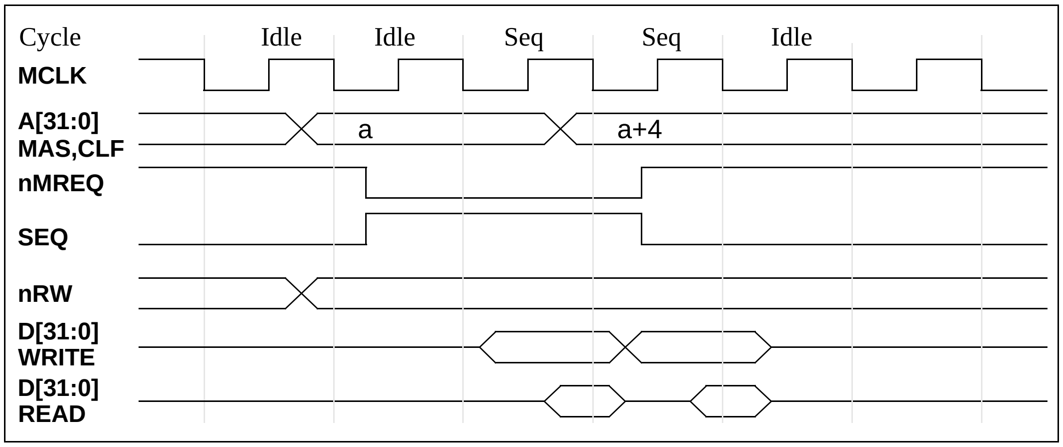
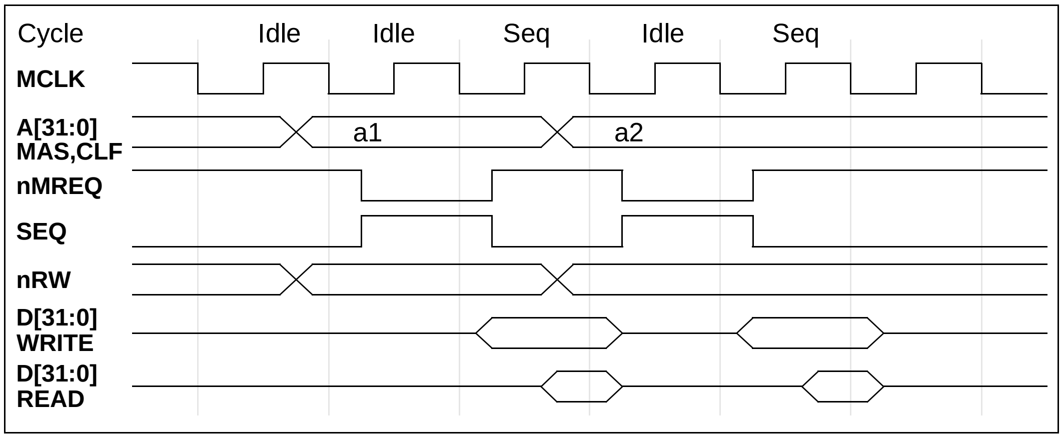


Figure 9-3. **Two 1-Word Nonsequential Reads or Writes Back-to-Back**



Note: The data bus is piped one cycle after the address bus. This means that for a read following a write, the data bus is not tristated by the processor until the falling edge of **MCLK** in the SEQ cycle of the READ. Notice that the asserting **nWAIT** will extend the time the processor is driving the data bus.

9.8 External Accesses

The SA-110 performs bus access for many different operations. All are constructed by combinations of nonsequential or sequential accesses. There may be any number of idle cycles between two other memory accesses. If a memory access is followed by an idle period on the bus (as opposed to another nonsequential access), then the address **A[31:0]**, and the signals **nRW** and **DL[1:0]** will remain at their previous value in order to avoid unnecessary bus transitions.

The accesses performed by a SA-110 are:

Unbuffered write	Level 1 translation fetch
Uncached read	Level 2 translation fetch
Buffered write	Cache line copy-back
Cache line fill	Read-lock-write sequence

9.8.1 Unbuffered Writes/Noncacheable Reads

Unbuffered writes and noncacheable reads are the most basic access types. Each may consist of a single access. The cycles always reflect the type (read or write) of the instruction requesting the cycle. Level 1 and level 2 translation fetches also generate noncacheable reads.

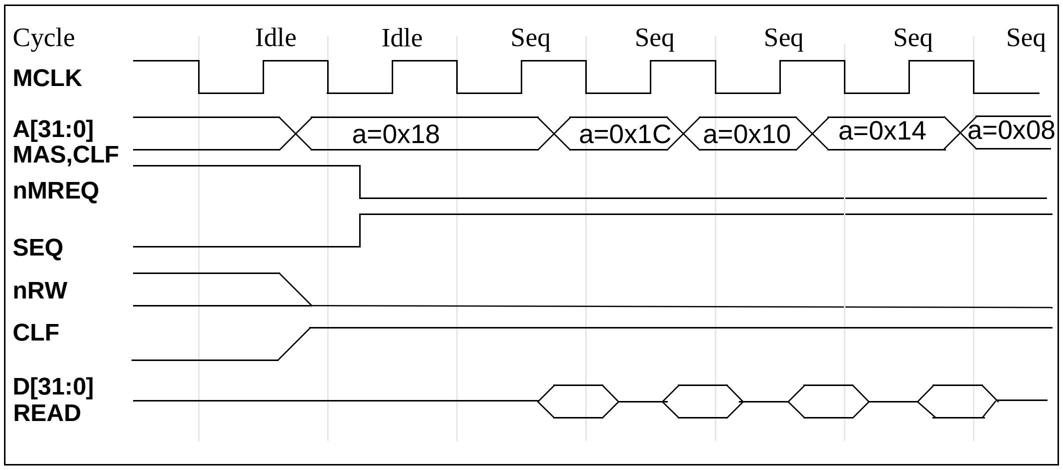
9.9 Buffered Write

Buffered writes can consist of sequential or nonsequential accesses. Note, if in enhanced mode, that if several write accesses are stored concurrently within the write buffer, then each access on the bus will start with a nonsequential access. If several write accesses occur to the same cache subblock, the SA-110 will merge these in the write buffer and write to memory using several sequential accesses with the byte masks indicating the valid byte lanes. Burst writes are done for castouts of dirty data and data stores that have merged in the write buffer. Cache line copy-backs are also buffered writes.

9.10 Cache Line Fill

This access appears on the bus as a nonsequential access followed by sequential accesses. Cache line fills are not required to start on an 8-word boundary and will be 8 words long. The first access will fetch the critical word if the bus is in enhanced mode or word 0 if in standard mode. CLF will be high for line fetches.Figure 9-4 is a cache line fill in enhanced mode. If it was in standard mode the fill would have started at location 0x00 rather than location 0x18.

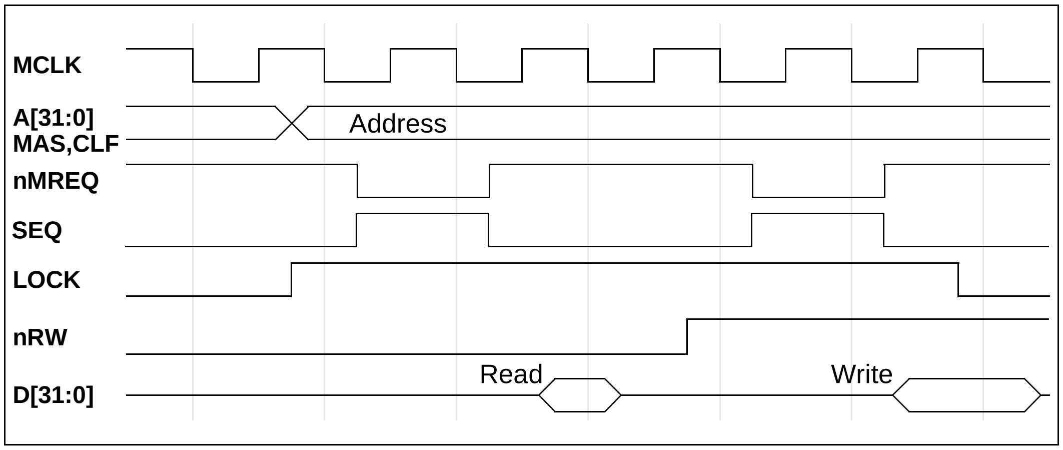
Figure 9-4. Cache Line Fill



9.11 Read-Lock-Write

The read-lock-write sequence is generated by an SWP instruction to a noncacheable/nonbufferable location. On the bus it consists of a single word read access followed by a single word write access to the same address, and both are treated as nonsequential accesses. The cycle is differentiated by the **LOCK** signal. **LOCK** has the timing of address; that is, it goes high in the high phase of **MCLK** at the start of the read access. However, it always goes low at the end of the write access, even if the following cycle is an idle cycle (unless of course the following access was a read-lock-write sequence). There may be several idle cycles between the read and the write. Note that for a cacheable/bufferable access, the two accesses are done in the data cache preceeded by a cache line fill, if needed.

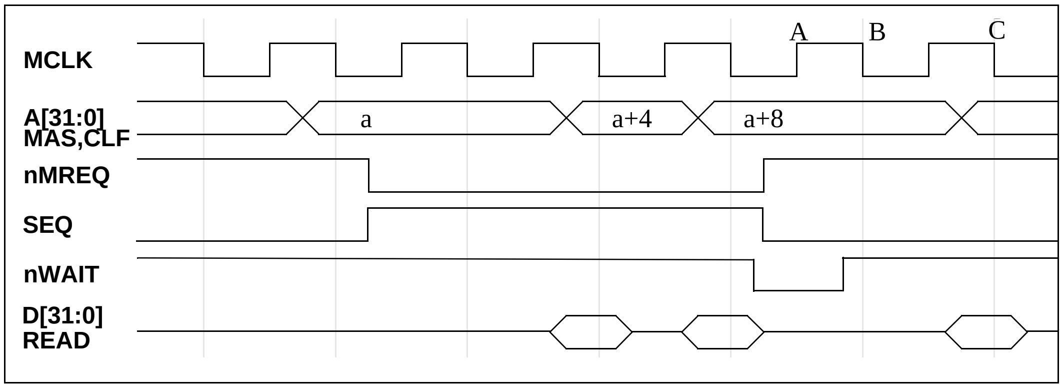
Figure 9-5. Read-Lock-Write



9.12 Stalling the Bus

The SA-110 can be stalled by **nWAIT**, as shown for a load in Figure 9-6. The following diagram shows part of a load request that has been stalled for one cycle waiting for return data on the third word.

Figure 9-6. Using nWAIT to Stop the SA-110 for One MCLK Cycle



The SA-110 samples the **nWAIT** signal pin at time A. The signal **nWAIT** must meet setup and hold requirements with respect to the rising edge of **MCLK** at A. Without the stall cycle, the data would have been sampled at time B. Inserting a single stall cycle causes the data to be sampled at time C.

9.13 Summary of Transactions

The SA-110 will generate only a subset of all possible transactions on the bus. The SA-110 will generate only single nonsequential transactions and bursts described in the following sections.

9.13.1 Read Bursts

The only actions that cause read bursts are cache line fills. All other reads will be single nonsequential accesses. All cache line fills are 8 words long.

9.13.2 Write Bursts

In standard mode, write bursts are caused by STM to B=1 locations and castouts of the data cache. Because the data cache keeps a dirty bit for each 16-byte subblock in standard mode, write bursts are either 4 or 8 words long. In enhanced mode, the write buffer will merge random writes. The write burst sizes in enhanced mode are 2, 3, 4, or 8 words. The **CLF** bit is set on stores. The **CLF** signal is asserted on stores for all 32-byte write bursts. Note that pass one parts never assert **CLF** on writes. When **CLF** is asserted, the external memory system does not need to wait until the byte mask information is valid during write bursts. Optimization is aided because all byte lanes will be valid.

9.13.3 Transaction Summary

Table 9-4 lists all the transactions that the SA-110 can generate. No burst will cross an aligned 32-byte boundary.

Table 9-4. SA-110 Transactions

Bus Operation	CONFIG	Burst Size	Starting Address Bits [4:2]	Notes
Read single	0 or 1	1	Any	
Read burst	0	8	0	Generated by cache line fills. **CLF** is asserted.
Read burst	1	8	Any	Generated by cache line fills. **CLF** is asserted.
Write single	0	1	Any	
Write single	1	1	Any	1..4 bytes are written as specified by the byte mask.
Write burst	0	2	0, 1, 2, 4, 5, 6	All 8 bytes are written.
Write burst	0	3	0, 1, 4, 5	All 12 bytes are written.
Write burst	0	4	0, 4	All 16 bytes are written.
Write burst	1	2	0, 1, 2, 4, 5, 6	Each word can have one to four bytes written.
Write burst	1	3	0, 1, 4, 5	The first and last word can have one to four bytes written. The middle word can have zero to four bytes written.
Write burst	1	4	0, 4	The first and last word can have one to four bytes written. The middle words can have zero to four bytes written.
Write burst	0 or 1	8	0	**CLF** is asserted and all 32 bytes are written.

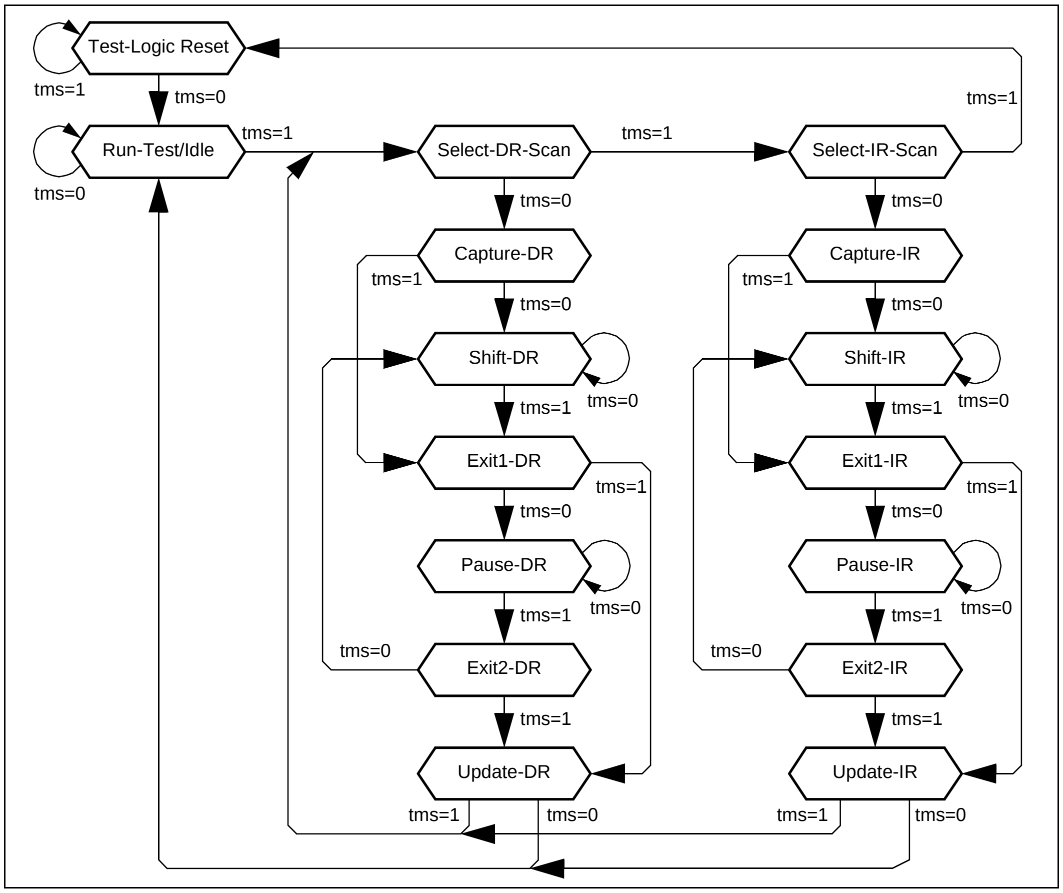
Boundary-Scan Test Interface 10

The boundary-scan interface conforms to the IEEE standard 1149.1 – 1990, *Standard Test Access Port and Boundary-Scan Architecture*. Refer to this standard for an explanation of the terms used in this section and for a description of the test access port (TAP) controller states. The SA-110 supports only JTAG continuity testing.

10.1 Overview

The boundary-scan interface provides a means of driving and sampling all the external pins of the device irrespective of the core state. This function permits testing of both the device's electrical connections to the circuit board, and (in conjunction with other devices on the circuit board having a similar interface) testing the integrity of the circuit board connections between devices. The interface intercepts all external connections within the device, and each such "cell" is then connected together to form a serial register (the boundary-scan register). The whole interface is controlled via five dedicated pins: **TDI**, **TMS**, **TCK**, **nTRST** and **TDO**. Figure 10-1 shows the state transitions that occur in the TAP controller.

Figure 10-1. Test Access Port (TAP) Controller State Transitions



10.2 Reset

The boundary-scan interface includes a state-machine controller (the TAP controller). In order to force the TAP controller into the correct state after power up of the device, a reset pulse must be applied to the **nTRST** pin. If the boundary-scan interface is to be used, then **nTRST** must be driven low, and then high again. If the boundary-scan interface is not to be used, then the **nTRST** pin may be tied permanently low. Note that a clock on **TCK** is not necessary to reset the device.

The action of reset (either a pulse or a dc level) is as follows:

System mode is selected.
The boundary-scan chain does NOT intercept any of the signals passing between the pads and the core.

IDcode mode is selected.
If **TCK** is pulsed, the contents of the ID register will be clocked out of **TDO**.

10.3 Pull-Up Resistors

The IEEE 1149.1 standard effectively requires that **TDI**, **nTRST** and **TMS** should have internal pull-up resistors. In order to minimize static current draw, **nTRST** has an internal pull-down resistor. These four pins can be left unconnected for normal operation and overdriven to use the JTAG features.

10.4 nPWRSLP

The **nPWRSLP** input pin is not sampled by the JTAG interface and must be deasserted to use the JTAG interface. The **TDO** pin is tristated when **nPWRSLP** is asserted.

10.5 Instruction Register

The instruction register is five bits in length. There is no parity bit. The fixed value loaded into the instruction register during the CAPTURE-IR controller state is: 00001.

10.6 Public Instructions

Table 10-1 lists the supported public instructions.

Table 10-1. Public Instructions

Instruction	Binary Code
EXTEST	00000
SAMPLE/PRELOAD	00001
CLAMP	00100
HIGHZ	00101
IDCODE	00110
BYPASS	11111
Private	00010, 00011, 00111, 01000-01111, 10000-11110

In the descriptions that follow, **TDI** and **TMS** are sampled on the rising edge of **TCK** and all output transitions on **TDO** occur as a result of the falling edge of **TCK**.

10.6.1 EXTEST (00000)

The boundary-scan (BS) register is placed in test mode by the EXTEST instruction. The EXTEST instruction connects the BS register between **TDI** and **TDO**. When the instruction register is loaded with the EXTEST instruction, all the boundary-scan cells are placed in their test mode of operation.

In the CAPTURE-DR state, inputs from the system pins and outputs from the boundary-scan output cells to the system pins are captured by the boundary-scan cells. In the SHIFT-DR state, the previously captured test data is shifted out of the BS register via the **TDO** pin, while new test data is shifted in via the **TDI** pin to the BS register parallel input latch. In the UPDATE-DR state, the new test data is transferred into the BS register parallel output latch. Note that this data is applied immediately to the system logic and system pins.

10.6.2 SAMPLE/PRELOAD (00001)

The BS register is placed in normal (system) mode by the SAMPLE/PRELOAD instruction. The SAMPLE/PRELOAD instruction connects the BS register between **TDI** and **TDO**. When the instruction register is loaded with the SAMPLE/PRELOAD instruction, all the boundary-scan cells are placed in their normal system mode of operation.

In the CAPTURE-DR state, a snapshot of the signals at the boundary-scan cells is taken on the rising edge of **TCK**. Normal system operation is unaffected. In the SHIFT-DR state, the sampled test data is shifted out of the BS register via the **TDO** pin, while new data is shifted in via the **TDI** pin to preload the BS register parallel input latch. In the UPDATE-DR state, the preloaded data is transferred into the BS register parallel output latch. Note that this data is not applied to the system logic or system pins while the SAMPLE/PRELOAD instruction is active. This instruction should be used to preload the boundary-scan register with known data prior to selecting EXTEST instructions. See Table 10-2 for appropriate guard values to be used for each boundary-scan cell.

10.6.3 CLAMP (00100)

The CLAMP instruction connects a 1-bit shift register (the BYPASS register) between **TDI** and **TDO**.

When the CLAMP instruction is loaded into the instruction register, the state of all output signals is defined by the values previously loaded into the boundary-scan register. A guarding pattern (specified for this device at the end of this section) should be preloaded into the boundary-scan register using the SAMPLE/PRELOAD instruction prior to selecting the CLAMP instruction.

In the CAPTURE-DR state, a logic 0 is captured by the bypass register. In the SHIFT-DR state, test data is shifted into the bypass register via **TDI** and out via **TDO** after a delay of one **TCK** cycle. Note that the first bit shifted out will be a zero. The bypass register is not affected in the UPDATE-DR state.

10.6.4 HIGHZ (00101)

The HIGHZ instruction connects a 1-bit shift register (the BYPASS register) between **TDI** and **TDO**. When the HIGHZ instruction is loaded into the instruction register, all outputs are placed in an inactive drive state.

In the CAPTURE-DR state, a logic 0 is captured by the bypass register. In the SHIFT-DR state, test data is shifted into the bypass register via **TDI** and out via **TDO** after a delay of one **TCK** cycle. Note that the first bit shifted out will be a zero. The bypass register is not affected in the UPDATE-DR state.

10.6.5 IDCODE (00110)

The IDCODE instruction connects the device identification register (or ID register) between **TDI** and **TDO**. The ID register is a 32-bit register that allows the manufacturer, part number and version of a component to be determined through the TAP. When the instruction register is loaded with the IDCODE instruction, all the boundary-scan cells are placed in their normal (system) mode of operation.

In the CAPTURE-DR state, the device identification code (specified at the end of this section) is captured by the ID register. In the SHIFT-DR state, the previously captured device identification code is shifted out of the ID register via the **TDO** pin, while data is shifted in via the **TDI** pin into the ID register. In the UPDATE-DR state, the ID register is unaffected.

10.6.6 BYPASS (11111)

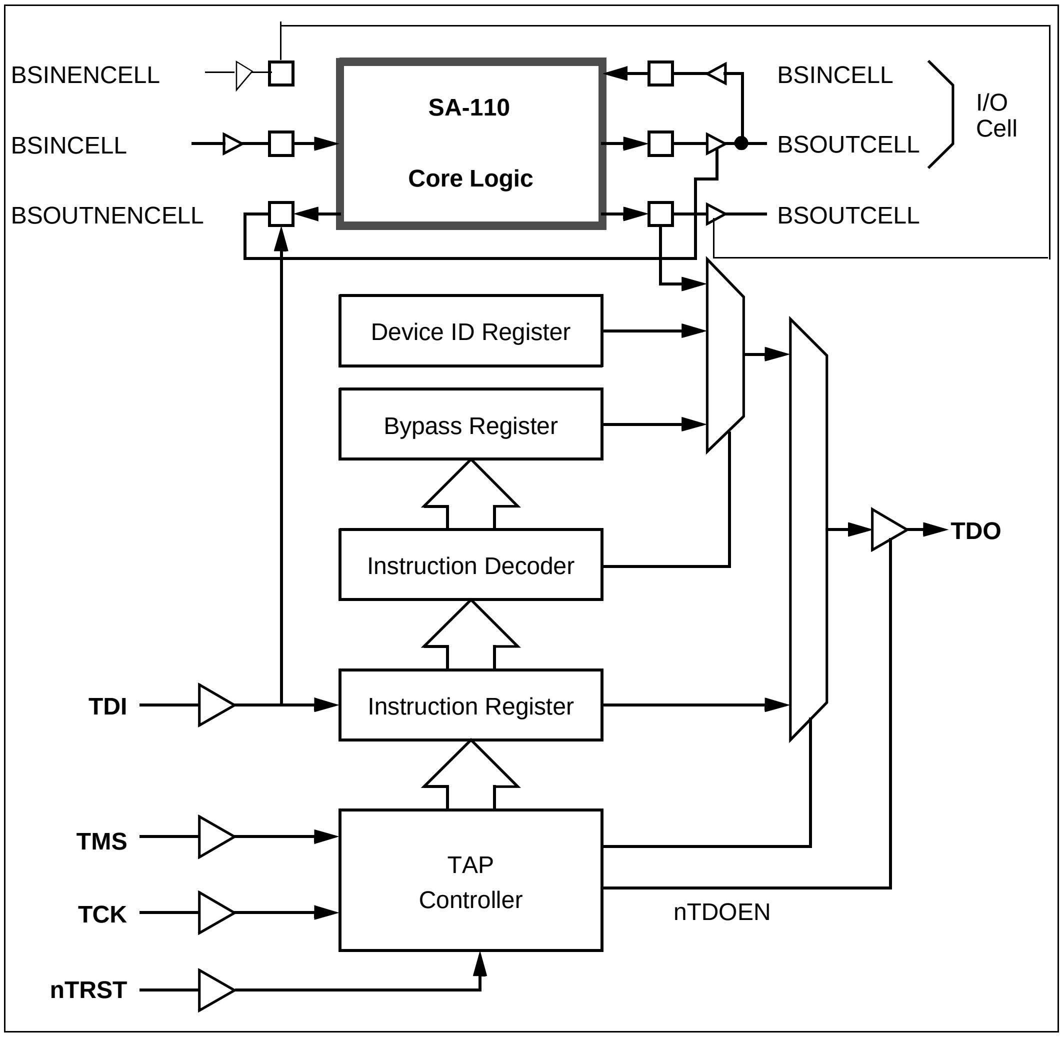
The BYPASS instruction connects a 1-bit shift register (the BYPASS register) between **TDI** and **TDO**. When the BYPASS instruction is loaded into the instruction register, all the boundary-scan cells are placed in their normal (system) mode of operation. This instruction has no effect on the system pins.

In the CAPTURE-DR state, a logic 0 is captured by the bypass register. In the SHIFT-DR state, test data is shifted into the bypass register via **TDI** and out via **TDO** after a delay of one **TCK** cycle. Note that the first bit shifted out will be a zero. The bypass register is not affected in the UPDATE-DR state.

10.7 Test Data Registers

Figure 10-2 illustrates the structure of the boundary-scan logic.

Figure 10-2. Boundary-Scan Block Diagram



10.7.1 Bypass Register

The bypass register is a single bit register that can be selected as the path between **TDI** and **TDO** to allow the device to be bypassed during boundary-scan testing.

Length: 1 bit

Operating Mode: When the BYPASS instruction is the current instruction in the instruction register, serial data is transferred from **TDI** to **TDO** in the SHIFT-DR state with a delay of one **TCK** cycle.

There is no parallel output from the bypass register. A logic 0 is loaded from the parallel input of the bypass register in the CAPTURE-DR state.

10.7.2 Device Identification (ID) Code Register

The ID register is used to read the 32-bit device identification code. No programmable supplementary identification code is provided.

Length: 32 bits

Operating Mode: When the IDCODE instruction is current, the ID register is selected as the serial path between **TDI** and **TDO**.

The format of the ID register is as follows:

31 28	27 12	11 0
Version	**Part Number**	**JEDEC Code**

The high-order four bits of the ID register are the version and bits 27..0 are 0x102C06B.

There is no parallel output from the ID register. The 32-bit device identification code is loaded into the ID register from its parallel inputs during the CAPTURE-DR state.

10.7.3 Boundary-Scan (BS) Register

The BS register consists of a serially connected set of cells around the periphery of the device, at the interface between the core logic and the system input/output pads. This register can be used to isolate the pins from the core logic and then drive or monitor the system pins.

Operating modes: The BS register is selected as the register to be connected between **TDI** and **TDO** only during the SAMPLE/PRELOAD and EXTEST instructions. Values in the BS register are used, but are not changed, during the CLAMP instruction.

In the normal (system) mode of operation, straight-through connections between the core logic and pins are maintained and normal system operation is unaffected.

In TEST mode (when EXTEST is the currently selected instruction), values can be applied to the output pins independently of the actual values on the input pins and core logic outputs. On the SA-110 all of the boundary-scan cells include an update register and thus all of the pins can be controlled in the above manner. An additional boundary-scan cell is interposed in the scan chain in order to control the enabling of the data bus.

The correspondence between boundary-scan cells and system pins, system direction controls and system output enables is as shown in Table 10-3. The cells are listed in the order in which they are connected in the boundary-scan register, starting with the cell closest to **TDI**.

The EXTEST guard values specified in Table 10-3 should be clocked into the boundary-scan register (using the SAMPLE/PRELOAD instruction) before the EXTEST instruction is selected, to ensure that known data is applied to the core logic during the test. This guard value should also be used when new EXTEST vectors are clocked into the boundary-scan register.

The values stored in the BS register after power-up are not defined. Similarly, the values previously clocked into the BS register are not guaranteed to be maintained across a boundary-scan reset (from forcing **nTRST** low or entering the test logic reset state).

10.8 Boundary-Scan Interface Signals

Figure 10-3, Figure 10-4, and Figure 10-5 show the typical timing for the BS register.

Figure 10-3. Boundary-Scan General Timing

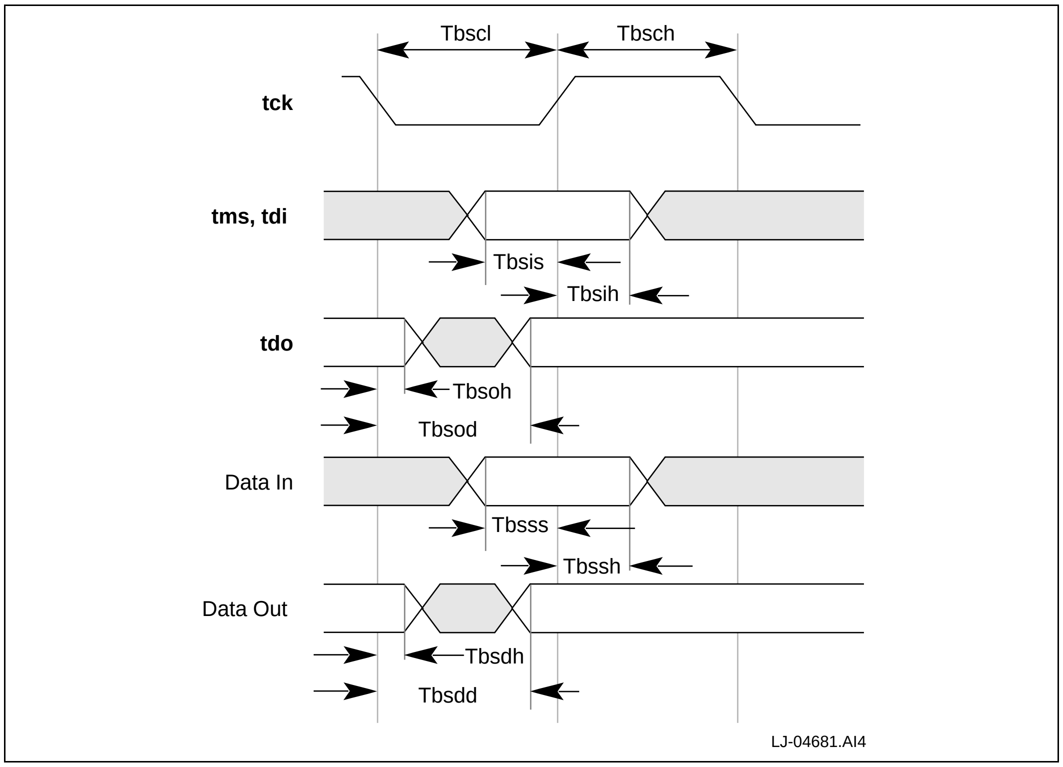


Figure 10-4. Boundary-Scan Tristate Timing

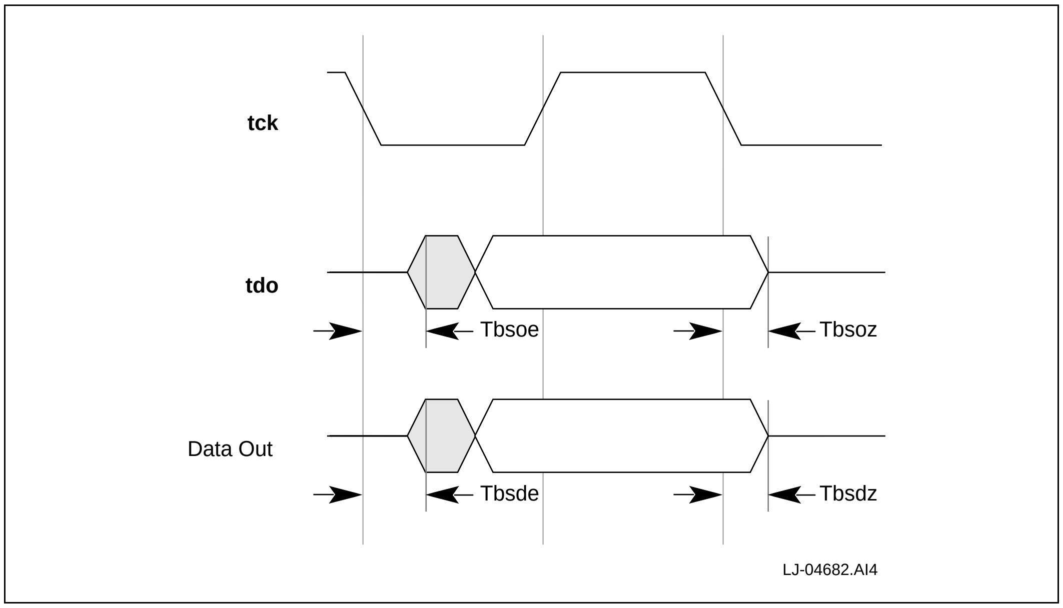


Figure 10-5. Boundary-Scan Reset Timing

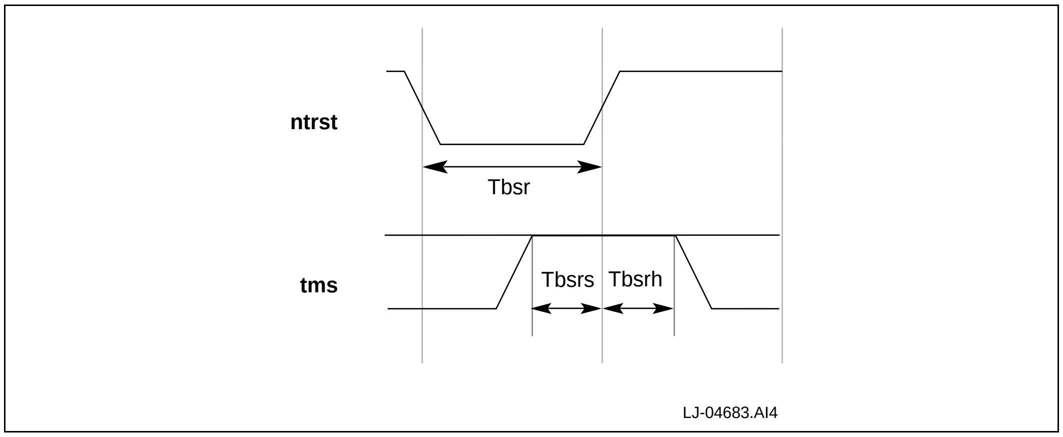


Table 10-2 shows the SA-110 boundary-scan interface timing specifications.

Table 10-2. SA-110 Boundary-Scan Interface Timing

Symbol	Parameter	Min	Max	Units	Notes
Tbscl	**TCK** low period	50	—	ns	9
Tbsch	**TCK** high period	50	—	ns	9
Tbsis	**TDI**,**TMS** setup to [TCr]	15	—	ns	—
Tbsih	**TDI**,**TMS** hold from [TCr]	25	—	ns	—
Tbsoh	**TDO** hold time	1	—	ns	1
Tbsod	TCf to **TDO** valid	—	40	ns	1
Tbsss	I/O signal setup to [TCr]	15	—	ns	4
Tbssh	I/O signal hold from [TCr]	25	—	ns	4
Tbsdh	Data output hold time	1	—	ns	5
Tbsdd	TCf to data output valid	—	40	ns	—
Tbsoe	**TDO** enable time	5	—	ns	1, 2
Tbsoz	**TDO** disable time	—	40	ns	1, 3
Tbsde	Data output enable time	5	—	ns	5, 6
Tbsdz	Data output disable time	—	40	ns	5, 7
Tbsr	Reset period	30	—	ns	—
Tbsrs	Tms setup to [TRr]	15	—	ns	9
Tbsrh	Tms hold from [TRr]	25	—	ns	9

NOTES:
1. Assumes a 25-pF load on **TDO**. Output timing derates at 0.072 ns/pF of extra load applied.
2. **TDO** enable time applies when the TAP controller enters the Shift-DR or Shift-IR states.
3. **TDO** disable time applies when the TAP controller leaves the Shift-DR or Shift-IR states.
4. For correct data latching, the I/O signals (from the core and the pads) must be set up and held with respect to the rising edge of **TCK** in the CAPTURE-DR state of the SAMPLE/PRELOAD and EXTEST instructions.
5. Assumes that the data outputs are loaded with the ac test loads (see the ac parameter specification).
6. Data output enable time applies when the boundary-scan logic is used to enable the output drivers.
7. Data output disable time applies when the boundary-scan is used to disable the output drivers.
8. **TMS** must be held high as **nTRST** is taken high at the end of the boundary-scan reset sequence.
9. **TCK** may be stopped indefinitely in either the low or high phase.

Table 10-3 lists the SA-110 boundary-scan signals and pins.

Table 10-3. Boundary-Scan Signals and Pins (Sheet 1 of 2)

No.	Cell Name	Pin	Type	Output Enable BS Cell	Guard Value	EX	No.	Cell Name	Pin	Type	Output Enable BS Cell	Guard Value	EX
from tdi							47	a29	A[29]	OUT	abe		
1	mse	MSE	IN	—			48	a30	A[30]	OUT	abe		
2	config	CONFIG	IN	—			49	a31	A[31]	OUT	abe		
3	nmreq	nMREQ	OUT	mse			50	ape	APE	IN	—		
4	seq	SEQ	OUT	mse			51	abort	ABORT	IN	—		
5	clf	CLF	OUT	abe			52	mclkin	MCLK	IN	—		
6	lock	LOCK	OUT	abe			53	mclkout	MCLK	OUT	sna		
7	nrw	nRW	OUT	abe			54	nmclk	NMCLK	OUT	sna		
8	spdf	SPDF	IN	—			55	nwait	nWAIT	IN	—		
9	mccfg0	MCCFG[0]	IN	—			56	clk	CLK	IN	—		
10	mccfg1	MCCFG[1]	IN	—			57	test_byp	TEST_BYP	IN	—		
11	mccfg2	MCCFG[2]	IN	—			58	testclk	TESTCLK	IN	—		
12	cccfg2	CCCFG[2]	IN	—			59	cccfg1	CCCFG[1]	IN	—		
13	cccfg3	CCCFG[3]	IN	—			60	cccfg0	CCCFG[0]	IN	—		
14	nresetout	nRESET_OUT	OUT	—			61	nreset	nRESET	IN	—		
15	abe	ABE	IN	abe			62	sna	SnA	IN	—		
16	mas0	MAS[0]	OUT	abe			63	fiq	FIQ	IN	—		
17	mas1	MAS[1]	OUT	abe			64	irq	IRQ	IN	—		
18	a0	A[0]	OUT	abe			65	din0	D[0]	IN	—		
19	a1	A[1]	OUT	abe			66	dout0	D[0]	OUT	dbe		
20	a2	A[2]	OUT	abe			67	din1	D[1]	IN	—		
21	a3	A[3]	OUT	abe			68	dout1	D[1]	OUT	dbe		
22	a4	A[4]	OUT	abe			69	din2	D[2]	IN	—		
23	a5	A[5]	OUT	abe			70	dout2	D[2]	OUT	dbe		
24	a6	A[6]	OUT	abe			71	din3	D[3]	IN	—		
25	a7	A[7]	OUT	abe			72	dout3	D[3]	OUT	dbe		
26	a8	A[8]	OUT	abe			73	din4	D[4]	IN	—		
27	a9	A[9]	OUT	abe			74	dout4	D[4]	OUT	dbe		
28	a10	A[10]	OUT	abe			75	din5	D[5]	IN	—		
29	a11	A[11]	OUT	abe			76	dout5	D[5]	OUT	dbe		
30	a12	A[12]	OUT	abe			77	din6	D[6]	IN	—		
31	a13	A[13]	OUT	abe			78	dout6	D[6]	OUT	dbe		
32	a14	A[14]	OUT	abe			79	din7	D[7]	IN	—		
33	a15	A[15]	OUT	abe			80	dout7	D[7]	OUT	dbe		
34	a16	A[16]	OUT	abe			81	din8	D[8]	IN	—		
35	a17	A[17]	OUT	abe			82	dout8	D[8]	OUT	dbe		
36	a18	A[18]	OUT	abe			83	din9	D[9]	IN	—		
37	a19	A[19]	OUT	abe			84	dout9	D[9]	OUT	dbe		
38	a20	A[20]	OUT	abe			85	din10	D[10]	IN	—		
39	a21	A[21]	OUT	abe			86	dout10	D[10]	OUT	dbe		
40	a22	A[22]	OUT	abe			87	din11	D[11]	IN	—		
41	a23	A[23]	OUT	abe			88	dout11	D[11]	OUT	dbe		
42	a24	A[24]	OUT	abe			89	din12	D[12]	IN	—		
43	a25	A[25]	OUT	abe			90	dout12	D[12]	OUT	dbe		
44	a26	A[26]	OUT	abe			91	din13	D[13]	IN	—		
45	a27	A[27]	OUT	abe			92	dout13	D[13]	OUT	dbe		
46	a28	A[28]	OUT	abe			93	din14	D[14]	IN	—		

Table 10-3. Boundary-Scan Signals and Pins (Sheet 2 of 2)

No.	Cell Name	Pin	Type	Output Enable BS Cell	Guard Value	EX		No.	Cell Name	Pin	Type	Output Enable BS Cell	Guard Value	EX
94	dout14	D[14]	OUT	dbe				120	dout26	D[26]	OUT	dbe		
95	din15	D[15]	IN	—				121	din27	D[27]	IN	—		
96	dout15	D[15]	OUT	dbe				122	dout27	D[27]	OUT	dbe		
97	din16	D[16]	IN	—				123	din28	D[28]	IN	—		
98	dout16	D[16]	OUT	dbe				124	dout28	D[28]	OUT	dbe		
99	din17	D[17]	IN	—				125	din29	D[29]	IN	—		
100	dout17	D[17]	OUT	dbe				126	dout29	D[29]	OUT	dbe		
101	din18	D[18]	IN	—				127	din30	D[30]	IN	—		
102	dout18	D[18]	OUT	dbe				128	dout30	D[30]	OUT	dbe		
103	din19	D[19]	IN	—				129	din31	D[31]	IN	—		
104	dout19	D[19]	OUT	dbe				130	dout31	D[31]	OUT	dbe		
105	din20	D[20]	IN	—										
106	dout20	D[20]	OUT	dbe										
107	din21	D[21]	IN	—										
108	dout21	D[21]	OUT	dbe										
109	din22	D[22]	IN	—										
110	dout22	D[22]	OUT	dbe										
111	dbe	DBE	IN	—										
112	dbein	DBE	IN	—										
113	din23	D[23]	IN	—										
114	dinout23	D[23]	OUT	dbe										
115	din24	D[24]	IN	—										
116	dout24	D[24]	OUT	dbe										
117	din25	D[25]	IN	—										
118	dout25	D[25]	OUT	dbe										
119	din26	D[26]	IN	—										
										to TDO				

Key:

IN	Input pad
OUT	Output pad

DC Parameters 11

This chapter defines the dc parameters for the SA-110.

11.1 Absolute Maximum Ratings

Table 11-1 lists the absolute maximum ratings for the SA-110.

Table 11-1. SA-110 DC Maximum Ratings

Symbol	Parameter	Minimum	Maximum	Unit	Note
VDD	Core supply voltage	**VSS** – 0.5	**VSS** + 2.2	V	1
VDDX	I/O voltage	MIN (**VSS** – 0.5, **VDD** – 0.3)	**VSS** + 3.6	V	1
Vip	Voltage applied to any pin	**VSS** – 0.3	MIN (**VDDX** + 0.3, **VSS** + 3.6)	V	1
Ts	Short-term storage temperature	–20	125	°C	1

NOTE:
1. These are stress SA-110 ratings only. Exceeding the absolute maximum ratings may permanently damage the device.

11.2 DC Operating Conditions

Table 11-2 lists the functional operating dc parameters for the SA-110.

Table 11-2. SA-110 DC Operating Conditions

Symbol	Parameter	Minimum	Maximum	Unit	Note
Vihc	IC input high voltage	0.8 * **VDDX**	**VDDX**	V	1, 2
Vilc	IC input low voltage	0.0	0.2 * **VDDX**	V	1, 2
Vohc	OCZ output high voltage at –2 mA	0.8 * **VDDX**	**VDDX**	V	1, 2, 3
Volc	OCZ output low voltage at 2 mA	0.0	0.2 * **VDDX**	V	1, 2, 3
Iohc	High-level output current	—	–2 mA	mA	3
Iolc	Low-level output current	—	2 mA	mA	3
Ta	Ambient operating temperature	0	70	°C	

NOTES:
1. Voltages measured with respect to **VSS**.
2. IC - CMOS-level inputs (includes IC and ICOCZ pin types).
3. In sleep mode, the levels on the output and I/O pins are maintained by weak leakers only. Systems that require the SA-110 to sink or source dc current may not use sleep mode.

11.3 DC Characteristics

Table 11-3 lists the dc characteristics for the SA-110.

Table 11-3. SA-110 DC Characteristics

Symbol	Parameter	Nominal	Units
Iin	IC input leakage current	100	µA
Its	Tristate leakage current	100	µA
Cin	Input pin capacitance	5	pF
Cio	Input /output pin capacitance	12	pF
ESD	HBM human body model	1	KV

11.4 Power Supply Voltages and Currents

Table 11-4 specifies the power supply voltages and currents for the SA-110.

Table 11-4. SA-110 Power Supply Voltages and Currents

Parameter	AA	BA	CA	DA	EA	Units	Notes
Maximum normal mode power	450	300	900	700	1000	mW	1
Maximum idle power	20	20	n/a	n/a	n/a	mW	2
Maximum sleep mode current	50	50	n/a	n/a	n/a	µA	2
VDD							
Minimum internal power supply voltage	1.5	1.5	1.8	1.8	1.9	V	
Nominal internal power supply voltage	1.65	1.65	2.1	2.0	2.0	V	
Maximum internal power supply voltage	1.8	1.8	2.2	2.2	2.1	V	
VDDX							
Minimum external power supply voltage	3.0	3.0	3.0	3.0	3.0	V	
Nominal external power supply voltage	3.3	3.3	3.3	3.3	3.3	V	
Maximum external power supply voltage	3.6	3.6	3.6	3.6	3.6	V	

NOTE:
1. Running Dhrytones 2.1 at 25°C, nominal power supply voltages, and 20-pF load on output and I/O pins.
2. Measured at 25°C junction temperature. Since the power levels for these modes are low, the junction temperature and ambient temperature are nearly equal during sleep and idle modes.

AC Parameters 12

This chapter defines the ac parameters for the SA-110.

12.1 Test Conditions

The ac timing diagrams presented in this section assume that the outputs of the SA-110 have been loaded with a 50-pF capacitive load on output signals and a 15-pF load on clock outputs. The output pads of the SA-110 are CMOS drivers, which exhibit a propagation delay that increases with the increase in load capacitance. An output derating figure is given for each output pad, showing the approximate rate of increase of delay with increasing or decreasing load capacitance for a typical process at room temperature.

Table 12-1. SA-110 Output Derating

Output Signal	Load for Nominal Value	Output Derating (ns/pF) VDD=1.65 Rising Edge	Output Derating (ns/pF) VDD=1.65 Falling Edge	Output Derating (ns/pF) VDD=2.0 Rising Edge	Output Derating (ns/pF) VDD=2.0 Falling Edge
A[31:0], nR/W, LOCK, nMREQ, SEQ, D[31:0]	50 pf	0.07	0.09	0.07	0.07
nMCLK, MCLK	15 pf	0.03	0.04	0.03	0.03

12.2 Module Considerations

The edge rates for the SA-110 are such that the lumped load model presented above can only be used for etch lengths up to one inch. Over one inch of etch, the signal is a transmission line and needs to be modeled as such.

12.3 Main Bus Signals

The **MCLK** signal is the reference for all SA-110 timing. The **nMCLK** signal is an inverted copy of the **MCLK** signal when **SnA** is high. The **nMCLK** signal is low when **SnA** is low. The SA-110 will stall its internal clock for one cycle if **nWAIT** is low when **MCLK** goes high. Figure 12-1, Figure 12-2, and Figure 12-3 show the ac timing definitions.

Figure 12-1. SA-110 Main Bus Timing

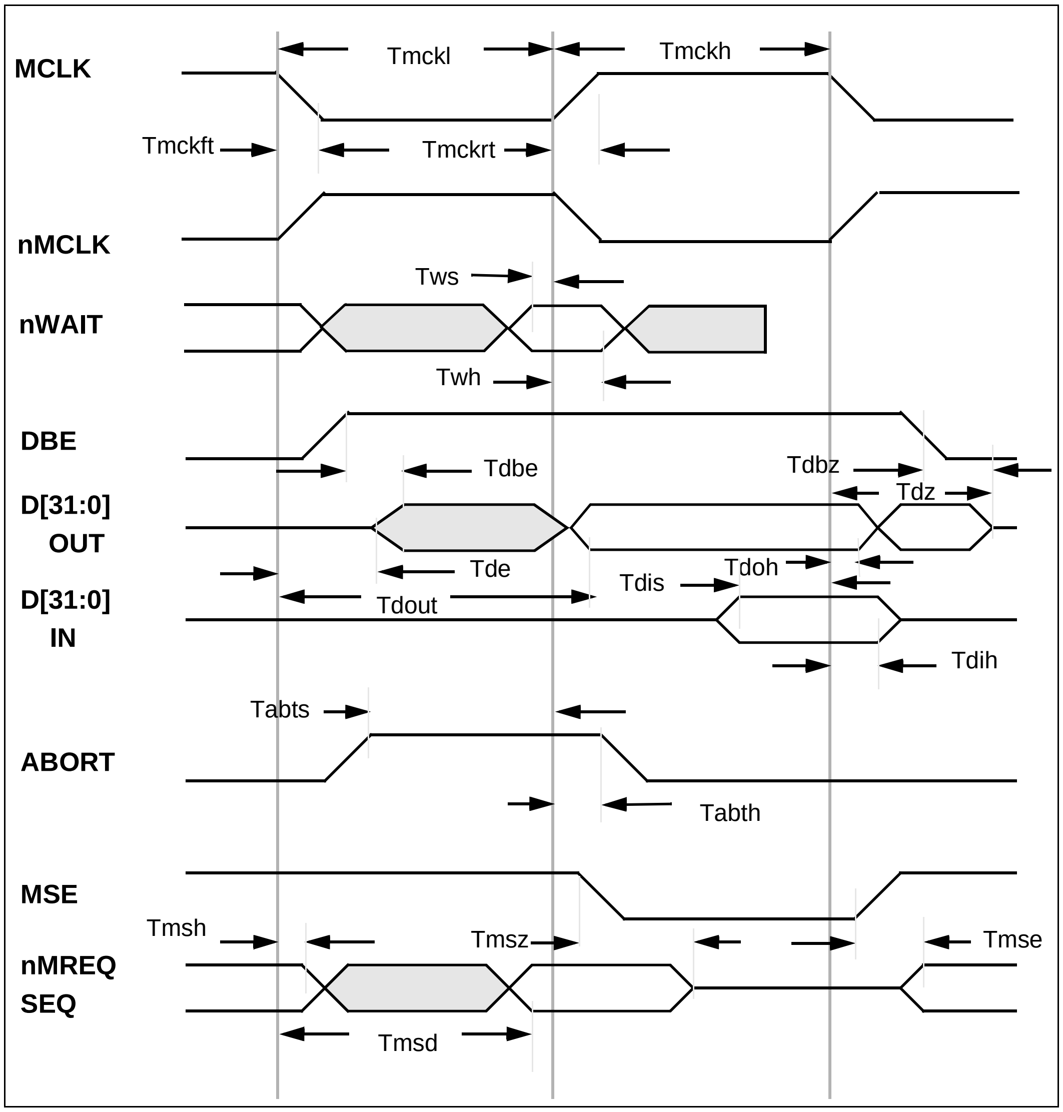


Figure 12-2. SA-110 Address Timing with APE=Low

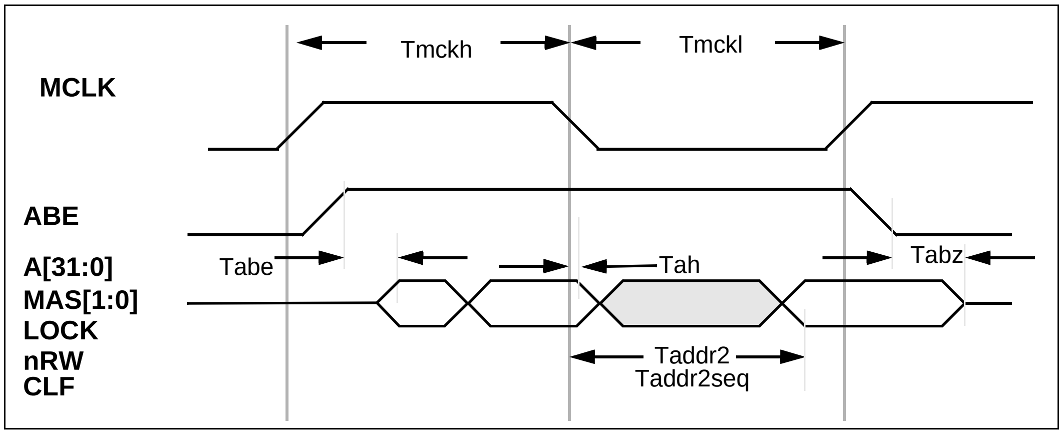
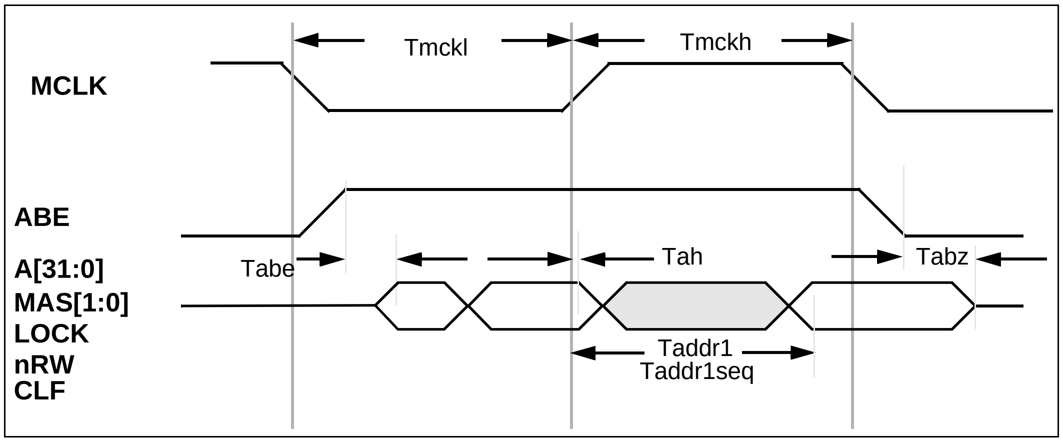


Figure 12-3. SA-110 Address Timing with APE=High



12.4 SA-110 AC Parameters

The ac parameters specified in the following tables are valid for the full range of voltage, temperature, and process defined for each SA-110 part in Table 11-4.

The following notes are used in Table 12-2 through Table 12-6.

1. **MCLK** and **nMCLK** timings measured between clock edges at 50% of **VDDX** driving 15 pF load.
2. The timings of these signals are measured to 80%/20% for outputs and 90%/10% for inputs.
3. The Taddrx times are for the first address of a sequential cycle. The Taddrxseq times are for the subsequent addresses of a sequential cycle.
4. The **MCCFG** setting with **SnA** asserted should be selected so the nominal **MCLK** speed does not exceed 1/(2*Tmckh).
5. Tdoh is guaranteed to be greater than a minimum Tdz.

Table 12-2. SA-110 AC Parameters for -AA Parts

Symbol	Parameter	Min	Max	Unit	Note
Tmckl	**MCLK** LOW time	10		ns	1
Tmckh	**MCLK** HIGH time	10		ns	1
Tmckrt	**MCLK**, **nMCLK** rise time **SnA**=1	0.75	2	ns	1
Tmckft	**MCLK**, **nMCLK** fall time **SnA**=1	1	3	ns	1
CCCFG	**CCCFG** setting	0	6	value	4
Tmckr	**MCLK** rise to **nMCLK** fall **SnA**=1	–0.5	1.5	ns	1
Tmckf	**MCLK** fall to **nMCLK** rise **SnA**=1	–1.5	0.5	ns	1
Tws	**nWAIT** setup to **MCLK**	1		ns	2
Twh	**nWAIT** hold from **MCLK**	2		ns	2
Tabe	Address bus enable		11	ns	2
Tabz	Address bus disable		4	ns	2
Taddr1	**MCLK** to address delay **APE** high		15	ns	2,3
Taddr2	**MCLK** to address delay **APE** low		10	ns	2,3
Taddr1seq	**MCLK** to address delay **APE** high		10	ns	2,3
Taddr2seq	**MCLK** to address delay **APE** low		10	ns	2,3
Tah	Address hold time	2		ns	2
Tdbe	**DBE** to data enable		10	ns	2
Tde	**MCLK** to data enable	3		ns	2
Tdbz	**DBE** to data disable		4	ns	2
Tdz	**MCLK** to data disable		8	ns	2,5
Tdout	Data out delay		11	ns	2
Tdoh	Data out hold	2		ns	2
Tdis	Data in setup	0		ns	2
Tdih	Data in hold	2		ns	2
Tabts	**ABORT** setup time	1		ns	2
Tabth	**ABORT** hold time	2		ns	2
Tmse	**nMREQ** & **SEQ** enable		9	ns	2
Tmsz	**nMREQ** & **SEQ** disable		3	ns	2
Tmsd	**nMREQ** & **SEQ** delay		11	ns	2
Tmsh	**nMREQ** & **SEQ** hold	2		ns	2

Table 12-3. SA-110 AC Parameters for -BA Parts

Symbol	Parameter	Min	Max	Unit	Note
Tmckl	**MCLK** low time	15		ns	1
Tmckh	**MCLK** HIGH time	15		ns	1
Tmckrt	**MCLK**, **nMCLK** rise time **SnA**=1	0.75	3	ns	1
Tmckft	**MCLK**, **nMCLK** fall time **SnA**=1	1	3.5	ns	1
CCCFG	**CCCFG** setting	0	2	value	4
Tmckr	**MCLK** rise to **nMCLK** fall **SnA**=1	–1	2	ns	1
Tmckf	**MCLK** fall to **nMCLK** rise **SnA**=1	–1	.5	ns	1
Tws	**nWAIT** setup to **MCLK**	–2		ns	2
Twh	**nWAIT** hold from **MCLK**	2		ns	2
Tabe	Address bus enable	3	15	ns	2
Tabz	Address bus disable		6	ns	2
Taddr1	**MCLK** to address delay **APE** high		23	ns	2,3
Taddr2	**MCLK** to address delay **APE** low		13	ns	2,3
Taddr1seq	**MCLK** to address delay **APE** high		15	ns	2,3
Taddr2seq	**MCLK** to address delay **APE** low		13	ns	2,3
Tah	Address hold time	2		ns	2
Tdbe	**DBE** to data enable		14	ns	2
Tde	**MCLK** to data enable	3		ns	2
Tdbz	**DBE** to data disable		5	ns	2
Tdz	**MCLK** to data disable		11	ns	2,5
Tdout	Data out delay		15	ns	2
Tdoh	Data out hold	2		ns	2
Tdis	Data in setup	2		ns	2
Tdih	Data in hold	3		ns	2
Tabts	**ABORT** setup time	2		ns	2
Tabth	**ABORT** hold time	3		ns	2
Tmse	**nMREQ** & **SEQ** enable		13	ns	2
Tmsz	**nMREQ** & **SEQ** disable		4	ns	2
Tmsd	**nMREQ** & **SEQ** delay		15	ns	2
Tmsh	**nMREQ** & **SEQ** hold	2		ns	2

Table 12-4. SA-110 AC Parameters for -CA Parts

Symbol	Parameter	Min	Max	Unit	Note
Tmckl	**MCLK** LOW time	7.5		ns	1
Tmckh	**MCLK** HIGH time	7.5		ns	1
Tmckrt	**MCLK**, **nMCLK** rise time **SnA**=1	0.75	2	ns	1
Tmckft	**MCLK**, **nMCLK** fall time **SnA**=1	0.75	2.5	ns	1
CCCFG	**CCCFG** setting	0	9	value	4
Tmckr	**MCLK** rise to **nMCLK** fall **SnA**=1	–1	1	ns	1
Tmckf	**MCLK** fall to **nMCLK** rise **SnA**=1	–1	1	ns	1
Tws	**nWAIT** setup to **MCLK**	1		ns	2
Twh	**nWAIT** hold from **MCLK**	2		ns	2
Tabe	Address bus enable		10	ns	2
Tabz	Address bus disable		4	ns	2
Taddr1	**MCLK** to address delay **APE** high		13	ns	2,3
Taddr2	**MCLK** to address delay **APE** low		9	ns	2,3
Taddr1seq	**MCLK** to address delay **APE** high		9	ns	2,3
Taddr2seq	**MCLK** to address delay **APE** low		9	ns	2,3
Tah	Address hold time	2		ns	2
Tdbe	**DBE** to data enable		9	ns	2
Tde	**MCLK** to data enable	3		ns	2
Tdbz	**DBE** to data disable		3	ns	2
Tdz	**MCLK** to data disable		7	ns	2,5
Tdout	Data out delay		10	ns	2
Tdoh	Data out hold	2		ns	2
Tdis	Data in setup	0		ns	2
Tdih	Data in hold	2		ns	2
Tabts	**ABORT** setup time	1		ns	2
Tabth	**ABORT** hold time	2		ns	2
Tmse	**nMREQ** & **SEQ** enable		8	ns	2
Tmsz	**nMREQ** & **SEQ** disable		3	ns	2
Tmsd	**nMREQ** & **SEQ** delay		10	ns	2
Tmsh	**nMREQ** & **SEQ** hold	2		ns	2

Table 12-5. SA-110 AC Parameters for -DA Parts

Symbol	Parameter	Min	Max	Unit	Note
Tmckl	**MCLK** low time	5		ns	1
Tmckh	**MCLK** HIGH time	5		ns	1
Tmckrt	**MCLK**, **nMCLK** rise time **SnA**=1	0.75	3	ns	1
Tmckft	**MCLK**, **nMCLK** fall time **SnA**=1	0.75	2.5	ns	1
CCCFG	**CCCFG** setting	0	7	value	4
Tmckr	**MCLK** rise to **nMCLK** fall **SnA**=1	–1.5	1	ns	1
Tmckf	**MCLK** fall to **nMCLK** rise **SnA**=1	–1	1.5	ns	1
Tws	**nWAIT** setup to **MCLK**	2		ns	2
Twh	**nWAIT** hold from **MCLK**	3		ns	2
Tabe	Address bus enable		13	ns	2
Tabz	Address bus disable		5	ns	2
Taddr1	**MCLK** to address delay **APE** high		19	ns	2,3
Taddr2	**MCLK** to address delay **APE** low		12	ns	2,3
Taddr1seq	**MCLK** to address delay **APE** high		13	ns	2,3
Taddr2seq	**MCLK** to address delay **APE** low		12	ns	2,3
Tah	Address hold time	2		ns	2
Tdbe	**DBE** to data enable		13	ns	2
Tde	**MCLK** to data enable	3		ns	2
Tdbz	**DBE** to data disable		5	ns	2
Tdz	**MCLK** to data disable		9	ns	2,5
Tdout	Data out delay		14	ns	2
Tdoh	Data out hold	2		ns	2
Tdis	Data in setup	2		ns	2
Tdih	Data in hold	3		ns	2
Tabts	**ABORT** setup time	2		ns	2
Tabth	**ABORT** hold time	3		ns	2
Tmse	**nMREQ** & **SEQ** enable		12	ns	2
Tmsz	**nMREQ** & **SEQ** disable		3	ns	2
Tmsd	**nMREQ** & **SEQ** delay		14	ns	2
Tmsh	**nMREQ** & **SEQ** hold	2		ns	2

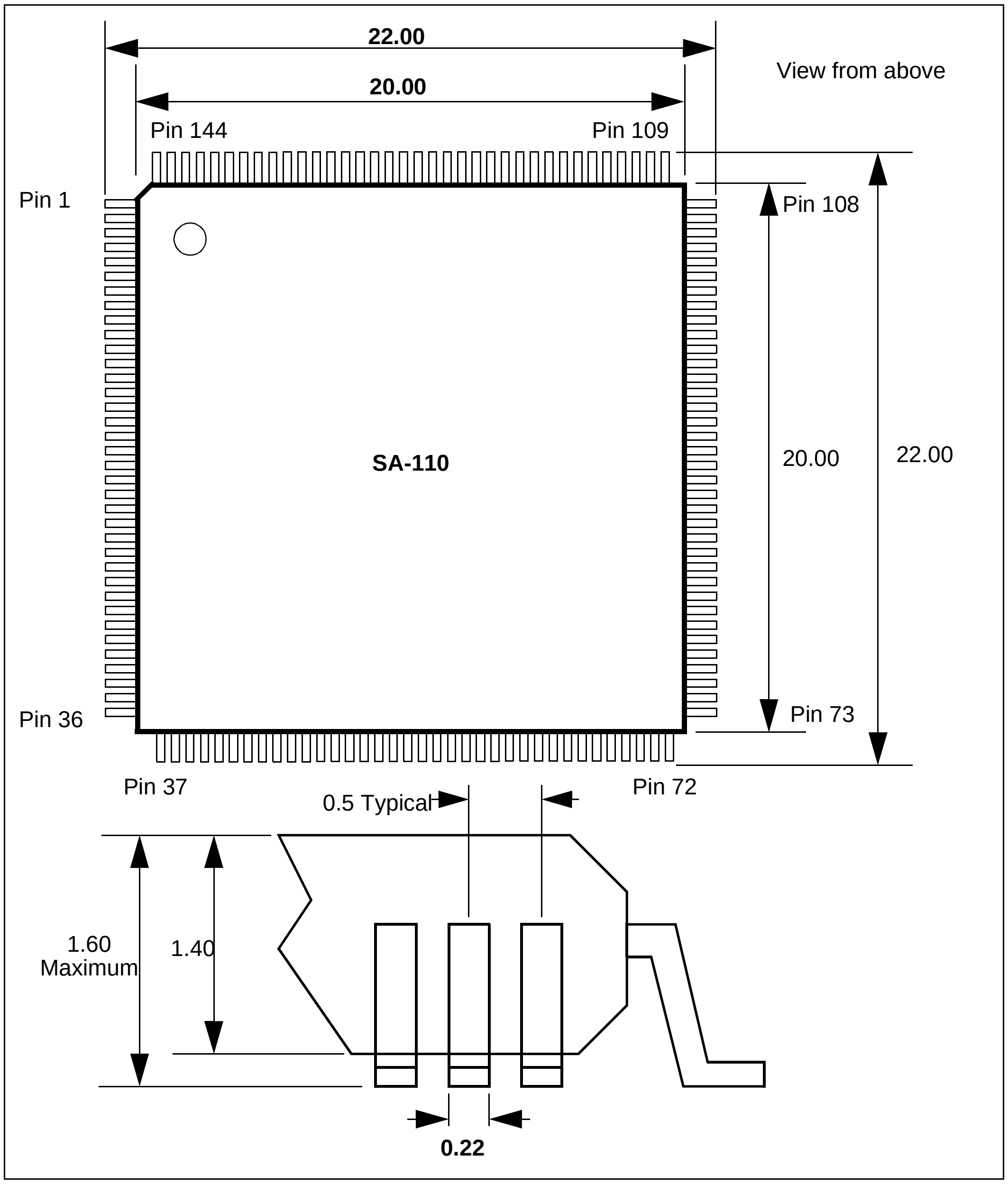
Table 12-6. SA-110 AC Parameters for -EA Parts

Symbol	Parameter	Min	Max	Unit	Note
Tmckl	**MCLK** LOW time	7.5		ns	1
Tmckh	**MCLK** HIGH time	7.5		ns	1
Tmckrt	**MCLK**, **nMCLK** rise time **SnA**=1	0.75	2	ns	1
Tmckft	**MCLK**, **nMCLK** fall time **SnA**=1	0.75	2.5	ns	1
CCCFG	**CCCFG** setting	0	12	value	4
Tmckr	**MCLK** rise to **nMCLK** fall **SnA**=1	–1	1	ns	1
Tmckf	**MCLK** fall to **nMCLK** rise **SnA**=1	–1	1	ns	1
Tws	**nWAIT** setup to **MCLK**	1		ns	2
Twh	**nWAIT** hold from **MCLK**	2		ns	2
Tabe	Address bus enable		10	ns	2
Tabz	Address bus disable		4	ns	2
Taddr1	**MCLK** to address delay **APE** high		13	ns	2,3
Taddr2	**MCLK** to address delay **APE** low		9	ns	2,3
Taddr1seq	**MCLK** to address delay **APE** high		9	ns	2,3
Taddr2seq	**MCLK** to address delay **APE** low		9	ns	2,3
Tah	Address hold time	2		ns	2
Tdbe	**DBE** to data enable		9	ns	2
Tde	**MCLK** to data enable	3		ns	2
Tdbz	**DBE** to data disable		3	ns	2
Tdz	**MCLK** to data disable		7	ns	2,5
Tdout	Data out delay		10	ns	2
Tdoh	Data out hold	2		ns	2
Tdis	Data in setup	0		ns	2
Tdih	Data in hold	2		ns	2
Tabts	**ABORT** setup time	1		ns	2
Tabth	**ABORT** hold time	2		ns	2
Tmse	**nMREQ** & **SEQ** enable		8	ns	2
Tmsz	**nMREQ** & **SEQ** disable		3	ns	2
Tmsd	**nMREQ** & **SEQ** delay		10	ns	2
Tmsh	**nMREQ** & **SEQ** hold	2		ns	2

Package and Pinout 13

This chapter describes the SA-110 pins. Figure 13-1 shows the SA-110 144-pin TQPF mechanical dimensions in millimeters.

Figure 13-1. SA-110 144-Pin TQFP Mechanical Dimensions



13.1 TQFP Pinout

Table 13-1 lists the SA-110 pins for the 144-TQPF pinout in numeric order, showing the signal type for each pin.

Table 13-1. SA-110 Pinout – 144-pin Thin Quad Flat Pack

Pin	Signal	Type	Pin	Signal	Type	Pin	Signal	Type	Pin	Signal	Type
1	D[0]	I/O	37	D[23]	I/O	73	MAS[0]	O	109	A[22]	O
2	D[1]	I/O	38	D[24]	I/O	74	MAS[1]	O	110	A[23]	O
3	D[2]	I/O	39	D[25]	I/O	75	A[0]	O	111	A[24]	O
4	D[3]	I/O	40	D[26]	I/O	76	A[1]	O	112	A[25]	O
5	VDD	–	41	D[27]	I/O	77	A[2]	O	113	VSS	–
6	VSS	–	42	VSS	–	78	VDDX2	–	114	VDDX2	–
7	VSS	–	43	VDDX1	–	79	VSS	–	115	A[26]	O
8	VDDX1	–	44	D[28]	I/O	80	VSS	–	116	A[27]	O
9	D[4]	I/O	45	D[29]	I/O	81	VDD	–	117	A[28]	O
10	D[5]	I/O	46	D[30]	I/O	82	A[3]	O	118	A[29]	O
11	D[6]	I/O	47	D[31]	I/O	83	A[4]	O	119	A[30]	O
12	D[7]	I/O	48	TDO	O	84	A[5]	O	120	A[31]	O
13	D[8]	I/O	49	TDI	I	85	A[6]	O	121	APE	I
14	D[9]	I/O	50	nTRST	I	86	A[7]	O	122	ABORT	I
15	D[10]	I/O	51	TMS	I	87	A[8]	O	123	MCLK	I/O
16	D[11]	I/O	52	TCK	I	88	A[9]	O	124	nMCLK	O
17	VDDX1	–	53	n/c		89	A[10]	O	125	VSS	–
18	VSS	–	54	VSS	–	90	VDD	–	126	VDDX2	–
19	VSS	–	55	VDD	–	91	VSS	–	127	nWAIT	I
20	VDD	–	56	MSE	I	92	VSS	–	128	CLK	I
21	D[12]	I/O	57	CONFIG	I	93	VDDX2	–	129	VDD	–
22	D[13]	I/O	58	nMREQ	O	94	A[11]	O	130	TCK_BYP	I
23	D[14]	I/O	59	SEQ	O	95	A[12]	O	131	TESTCLK	I
24	D[15]	I/O	60	CLF	O	96	A[13]	O	132	VSS	–
25	D[16]	I/O	61	LOCK	O	97	A[14]	O	133	VDD	–
26	D[17]	I/O	62	nRW	O	98	A[15]	O	134	n/c	
27	D[18]	I/O	63	VSS	–	99	A[16]	O	135	n/c	
28	D[19]	I/O	64	VDDX1	–	100	A[17]	O	136	n/c	
29	VDDX1	–	65	SPDF	I	101	A[18]	O	137	PLLCFG	
30	VSS	–	66	MCCFG[0]	I	102	VDDX2	–	138	CCCFG[1]	I
31	VSS	–	67	MCCFG[1]	I	103	VSS	–	139	CCCFG[0]	I
32	VDD	–	68	MCCFG[2]	I	104	VSS	–	140	nPWRSLP	I
33	D[20]	I/O	69	CCCFG[2]	I	105	VDD	–	141	nRESET	I
34	D[21]	I/O	70	CCCFG[3]	I	106	A[19]	O	142	SnA	I
35	D[22]	I/O	71	nRESET_OUT	O	107	A[20]	O	143	nFIQ	I
36	DBE	I	72	ABE	I	108	A[21]	O	144	nIRQ	I

Note: **VDDX1** and **VDDX2** should be tied to the **VDDX** power plane of the system, but are not internally connected.

Support, Products, and Documentation

If you need technical support, a *Product Catalog*, or help deciding which documentation best meets your needs, visit the Intel World Wide Web Internet site:

http://www.intel.com

Copies of documents that have an ordering number and are referenced in this document, or other Intel literature may be obtained by calling **1-800-332-2717** or by visiting Intel's website for developers at:

http://developer.intel.com

You can also contact the Intel Massachusetts Information Line or the Intel Massachusetts Customer Technology Center. Please use the following information lines for support:

For documentation and general information:	
Intel Massachusetts Information Line	
United States:	1–800–332–2717
Outside United States:	1–303-675-2148
Electronic mail address:	techdoc@intel.com

For technical support:	
Intel Massachusetts Customer Technology Center	
Phone (U.S. and international):	1–978–568–7474
Fax:	1–978–568–6698
Electronic mail address:	techsup@intel.com

StrongARM™