Preliminary Data Sheet
March 1993

AT&T
Microelectronics

DSP1610 Signal Coding Digital Signal Processor

1.0 Features

- 25 ns or 33 ns instruction cycle
- 512 word boot ROM, and 4 Kword or 8 Kword downloadable dual-port RAM (on-chip)
- Low-power (15 mW/MIPS typical) 0.9 µm CMOS technology

  Fully static design with sleep mode
- 16 × 16-bit multiplication and 36-bit accumulation in one instruction cycle
- Bit manipulation unit with barrel shifter Special instructions for high signal coding efficiency
- Two 36-bit accumulators

  Two 36-bit alternate accumulators
- Instruction cache for high-speed, program-efficient, zero-overhead looping
- Four external vectored interrupts and trap
- Two 64 Kword address spaces with software wait-states for external accesses
- 20 Mbits/s serial I/O port with multiprocessor capability—16-bit data channel, 8-bit protocol channel
- Parallel I/O port, 8- or 16-bit bidirectional host interface
- Second serial I/O port
- 256 memory-mapped I/O ports, four internally decoded for glueless device interface
- 8-bit control I/O bus provides eight flexible status or control pins
- 12 boot routines for flexible downloading
- Full-speed in-circuit emulation with hardware development system on-chip
- IEEE P1149.1 test port (JTAG)
- Interrupt timer
- Object code upward compatible with *WE*®DSP16 and DSP16A Digital Signal Processors
- Supported by DSP1610-ST Support Software Tools and DSP1600-DS Digital Signal Processor Development System

2.0 Description

The DSP1610 Signal Coding DSP is a 16-bit, high-speed, programmable integrated circuit fabricated in low-power 0.9 µm CMOS technology and packaged in a 132-pin plastic quad flat pack. While the DSP1610 is a general-purpose processor that can be programmed to perform a wide variety of signal processing functions, it includes features to support efficient implementation of signal coding algorithms like speech compression, error correction, and encryption. The DSP1610 achieves high throughput without programming restrictions or latencies, due to its parallel pipelined architecture. The processor has an arithmetic unit capable of a 16 x 16-bit multiplication and 36-bit accumulation or a 32-bit ALU operation in one instruction cycle. Data is supplied by two independent addressing units. A high-performance bit manipulation unit provides single-cycle 36-bit barrel shifting, normalization, bit-field extraction, and bit-field insertion. The DSP1610 device can function in a stand-alone manner, requiring only an external clock.

The DSP1610 contains 512 words of internal ROM and either 4 Kwords or 8 Kwords of downloadable internal dual-port RAM (DPRAM). The DSP1610 is packaged in a 100-pin PQFP and is available with 33 ns and 25 ns instruction cycle speeds. The DSP1610 is object code upward compatible with the DSP16/A/C/J.

Table of Contents

Contents **Page**

3.0 Pin Information

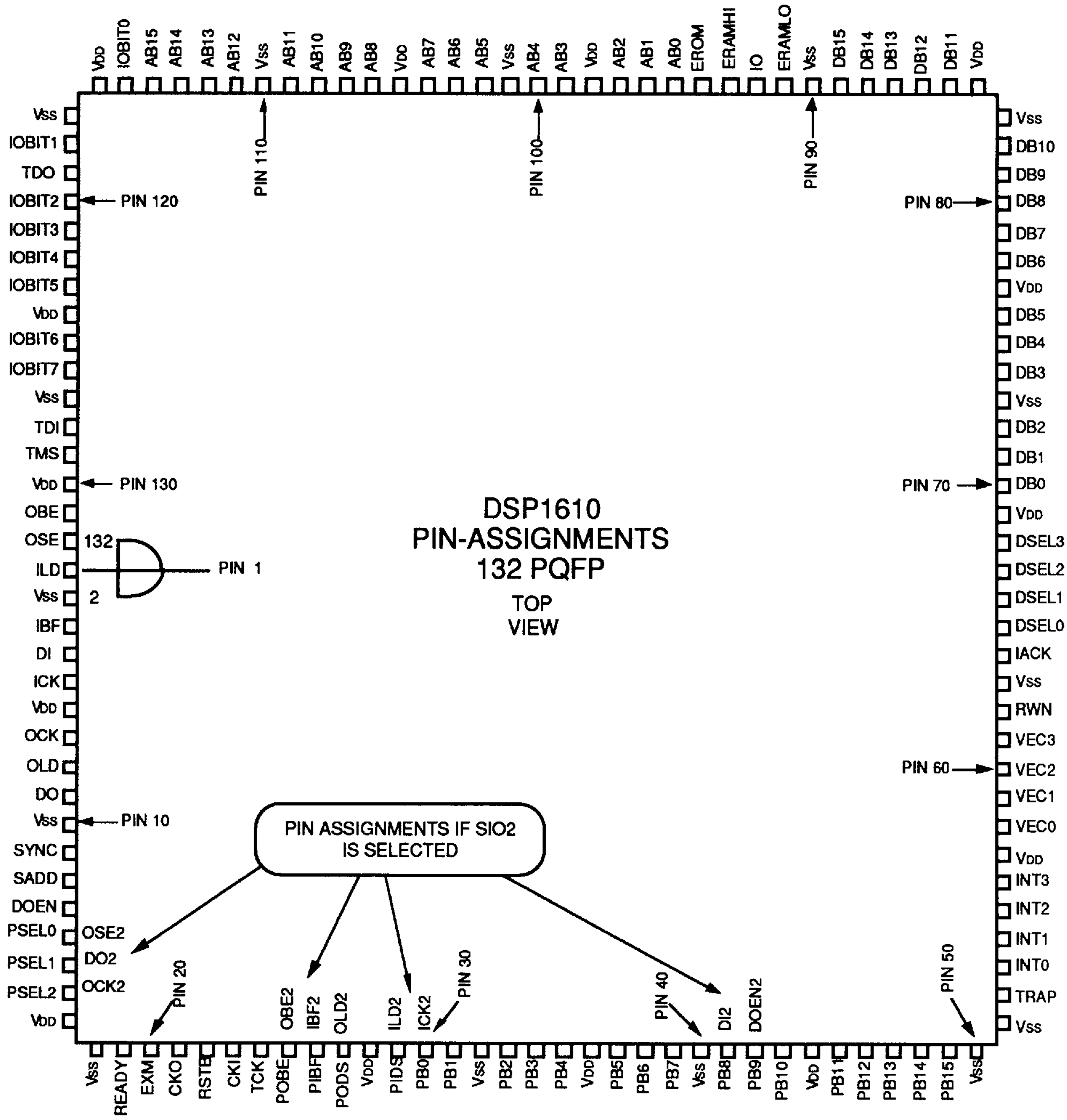


Figure 1. Pin Diagram

Pin Information (continued)

Functional descriptions of pins 1—132 are found in Section 6, Signal Descriptions. All unused input or I/O pins should be tied through a 47 kΩ to 100 kΩ resistor to VDD (when the pin is active low) or Vss (when the pin is active high).

Table 1. Pin Descriptions

Pin	Symbol	Type	Name/Function
1	ILD	I/O*	**SIO1 Input Load**
3	IBF	O*	**SIO1 Input Buffer Full**
4	DI	I	**SIO1 Data Input**
5	ICK	I/O*	**SIO1 Input Clock**
7	OCK	I/O*	**SIO1 Output Clock**
8	OLD	I/O*	**SIO1 Output Load**
9	DO	O*	**SIO1 Data Output**
11	SYNC	I/O*	**SIO1 Multiprocessor Synchronization**
12	SADD	I/O*	**SIO1 Multiprocessor Address**
13	DOEN	I/O*	**SIO1 Data Output Enable**
14	PSEL0/OSE2	O*	**Peripheral Select0 / SIO2 Output Shift Register Empty**
15	PSEL1/DO2	I/O*	**Peripheral Select1 / SIO2 Data Output**
16	PSEL2/OCK2	I/O*	**Peripheral Select2 / SIO2 Output Clock**
19	READY	I	**Processing Enable** (not functional in versions F12 and earlier)
20	EXM	I	**External ROM Enable**
21	CKO	O†	**Processor Clock Output**
22	RSTB	I	**Reset Bar**
23	CKI	I	**Processor 2x Clock Input**
24	TCK	I	**JTAG Test Clock**
25	POBE/OBE2	O*	**PIO Output Buffer Empty / SIO2 Output Buffer Empty**
26	PIBF/IBF2	O*	**PIO Input Buffer Full / SIO2 Input Buffer Full**
27	PODS/OLD2	I/O*	**PIO Output Data Strobe / SIO2 Output Load**
29	PIDS/ILD2	I/O*	**PIO Input Data Strobe / SIO2 Input Load**
30	PB0/ICK2	I/O*	**PIO Data Bus Bit 0 / SIO2 Input Clock**
31, 33, 34, 35, 37, 38, 39	PB1, PB2, PB3, PB4, PB5, PB6, PB7	I/O*	**PIO Data Bus Bits 1—7**
41	PB8/DI2	I/O*	**PIO Data Bus Bit 8 / SIO2 Data Input**
42	PB9/DOEN2	I/O*	**PIO Data Bus Bit 9 / SIO2 Data Output Enable**
43, 45, 46, 47, 48, 49	PB10, PB11, PB12, PB13, PB14, PB15	I/O*	**PIO Data Bus Bits 10—15**

* 3-states when RSTB = 0, or by JTAG.

† 3-states when RSTB = 0 and INT3 = 1, or by JTAG. Output = 1 when RSTB = 0 and INT3 = 0.

Pin Information (continued)

Table 1. Pin Descriptions (continued)

Pin	Symbol	Type	Name/Function
52	TRAP	I/O*	**Nonmaskable Program Trap/Breakpoint Indication**
53	INT0	I	**Vectored Interrupt 0**
54	INT1	I	**Vectored Interrupt 1**
55	INT2	I	**Vectored Interrupt 2**
56	INT3	I	**Vectored Interrupt 3**
58	VEC0	O*	**Vectored Interrupt Indication 0**
59	VEC1	O*	**Vectored Interrupt Indication 1**
60	VEC2	O*	**Vectored Interrupt Indication 2**
61	VEC3	O*	**Vectored Interrupt Indication 3**
62	RWN	O†	**Read/Write Not**
64	IACK	O*	**Interrupt Acknowledge**
65, 66, 67, 68	DSEL[3:0]	O†	**IO Enables 0—3 for Data Address 0x7F0—0x7F3**
70, 71, 72, 74, 75, 76, 78, 79, 80, 81, 82, 85, 86, 87, 88, 89	DB[15:0]	I/O*	**External Memory Data Bus 0—15**
91	ERAMLO	O†	**Data Address 0x2000 to 0x7EFF External RAM Enable**
92	IO	O†	**Data Address 0x7F00 to 0x7FFF IO Enable**
93	ERAMHI	O†	**Data Address 0x8000 to 0xFFFF External RAM Enable**
94	EROM	O†	**Program Address External ROM Enable**
95, 96, 97, 99, 100, 102, 103, 104, 106, 107, 108, 109, 111, 112, 113, 114	AB[15:0]	O*	**External Memory Address Bus 0—15**

* 3-states when RSTB = 0, or by JTAG.

† 3-states when RSTB = 0 and INT3 = 1, or by JTAG. Output = 1 when RSTB = 0 and INT3 = 0.

Pin Information (continued)

Table 1. Pin Descriptions (continued)

Pin	Symbol	Type	Name/Function
115, 118, 120, 121, 122, 123, 125, 126	IOBIT[7:0]	I/O*	**Status/Control Bits 0—7**
119	TDO	O‡	**JTAG Test Data Output**
128	TDI	I§	**JTAG Test Data Input**
129	TMS	I§	**JTAG Test Mode Select**
131	OBE	O*	**SIO1 Output Buffer Empty**
132	OSE	O*	**SIO1 Output Shift Register Empty**
2, 10, 18, 32, 40, 50, 51, 63, 73, 83, 90, 101, 110, 117, 127	Vss	P	**Ground**
6, 17, 28, 36, 44, 57, 69, 77, 84, 98, 105, 116, 124, 130	VDD	P	**5 V Supply**

* 3-states when RSTB = 0, or by JTAG.
† 3-states when RSTB = 0 and INT3 = 1, or by JTAG. Output = 1 when RSTB = 0 and INT3 = 0.
‡ 3-states by JTAG only.
§ Only inputs with pull-up resistors.

4.0 Hardware Architecture

The DSP1610 Signal Coding Processor is a 16-bit fixed-point digital signal processor (DSP) that is upward object code compatible with the DSP16(A). The DSP1610 consists of a DSP1600 core together with on-chip memory and peripherals. Many added architectural features give the DSP1610 high programming efficiency and flexibility for signal coding applications.

4.1 DSP1610 Architectural Overview

Figure 2 shows a block diagram of the DSP1610 which consists of a number of modules described below. The DSP1610 has a pair of internal buses (address bus and data bus) for program/coefficient memory (X memory space) and a second independent pair of internal buses for data memory (Y memory space).

DSP1600 Core

The DSP1600 core is the heart of the DSP1616 chip. The core contains data and address arithmetic units, and its instruction set has been enhanced over that of the DSP16/A/C/J. The core provides support for external memory wait-states and on-chip dual-port RAM and features vectored interrupts and a trap mechanism.

Dual-Port RAM (DPRAM)

This module contains four or eight banks of zero wait-state memory. Each bank consists of 1K 16-bit words and can be accessed from the instruction/coefficient (X) or data (Y) memory spaces. A program can reference the memory in either memory space at any time, transparently and without restriction. The DSP1600 core automatically performs the multiplexing. In the event that references to both ports of a single bank are made simultaneously, the DSP1600 core automatically inserts a wait-state and performs the data access first, followed by the instruction/coefficient access.

A program can be downloaded from slow off-chip memory into DPRAM, and then executed without wait-states. DPRAM is also useful for improving convolution performance in cases where the coefficients are adaptive. Since the DPRAM can be downloaded through the JTAG port, full-speed remote in-circuit emulation is possible. DPRAM can also be used for downloading self-test code via the JTAG port.

Read-Only Memory (ROM)

The DSP1610 contains 512 16-bit words of zero wait-state read-only memory (ROM). The ROM contains code for downloading programs from off-chip into the on-chip dual-port RAM and code to support the hardware development system.

External Memory Multiplexer (EMUX)

The EMUX connects the DSP1610 to external memory and I/O devices. It supports read/write operations from/to instruction/coefficient memory (X memory space) and data memory (Y memory space). The DSP1600 core automatically controls the EMUX. Instructions can transparently reference external memory from either set of internal buses. An instruction cannot reference external memory from both sets of internal buses (i.e., instruction/coefficient and data) simultaneously.

Bit Manipulation Unit (BMU)

The BMU adds signal coding extensions to the DSP1600 core instruction set to provide more efficient bit operations on accumulators. The BMU contains logic for barrel shifting, normalization, and bit-field insertion/extraction. The unit also contains a set of 36-bit alternate accumulators. The data in the alternate accumulators can be swapped with the data in the main accumulators. Flags returned by the BMU mesh seamlessly with the DSP1600 core conditional instructions.

Timer

The timer may be used to provide an interrupt at the expiration of a programmed interval. The interrupt may be a single interrupt or a repetitive interrupt. It provides more than nine orders of magnitude in interval selection. The timer may be stopped and restarted at any time.

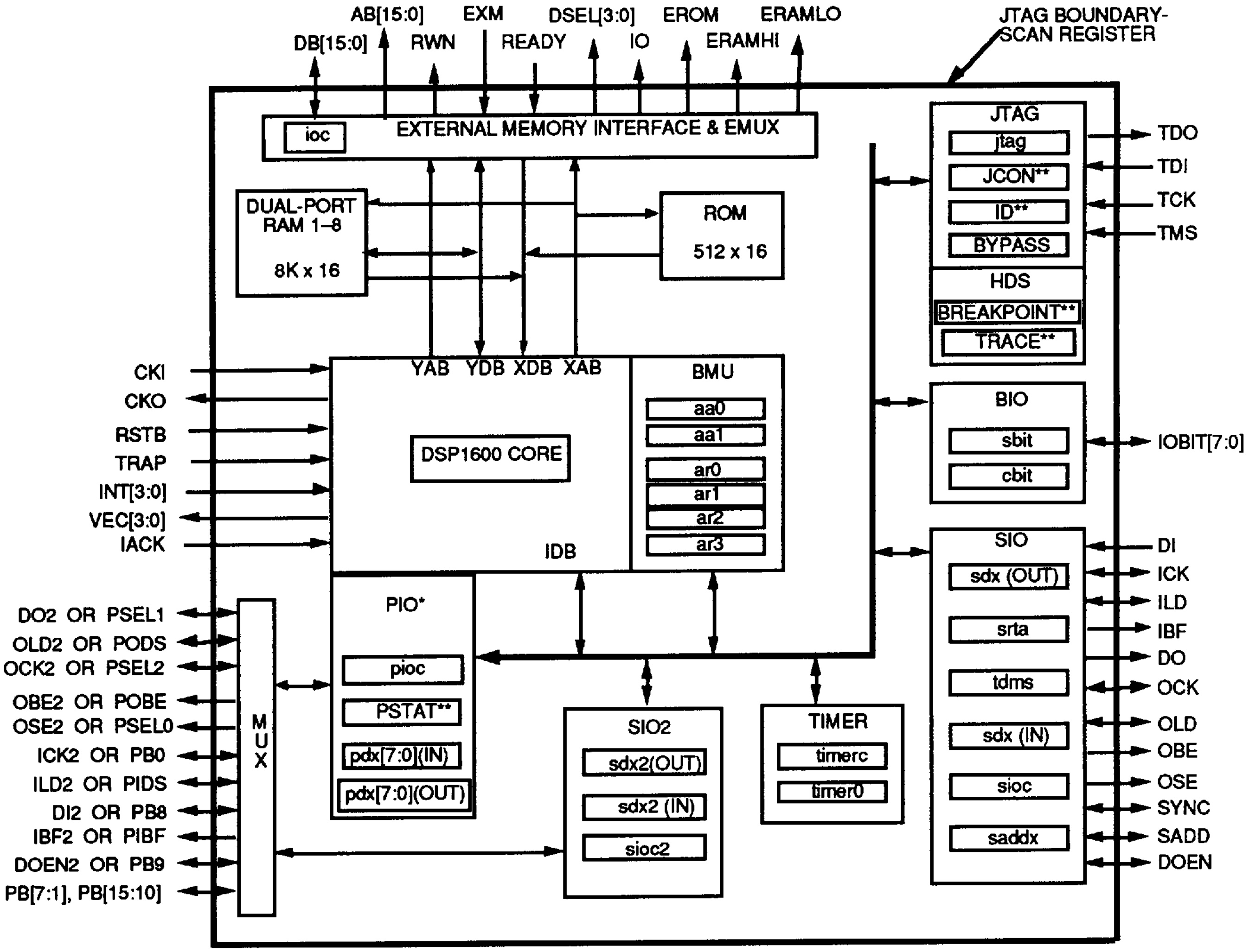
Bit Input/Output (BIO)

The BIO provides convenient and efficient monitor and control of eight individually configurable pins. The BIO pins may be individually configured as either inputs or outputs. As outputs, the bits may be individually set, cleared, or toggled. As inputs, individual pins or combinations of input pins can be tested for patterns. Flags returned by the BIO mesh seamlessly with the DSP1600 core conditional instructions.

JTAG

The JTAG section contains logic that implements the JTAG/IEEE P1149.1 standard four-signal test port, and an input/output JTAG boundary-scan register. The JTAG port provides a mechanism for the DSP1600 core to communicate with remote test equipment or a remote hardware development system (HDS). The JTAG port also communicates with the on-chip HDS module and supports program upload/download, execution start/stop, and memory and register upload/download.

Hardware Architecture (continued)



* The PIO is tightly coupled to the CORE to provide DSP16/A/C/J compatibility.

** These registers are only accessible through the pins.

Figure 2. DSP1610 Block Diagram

Hardware Architecture (continued)

Table 2. DSP1610 Block Diagram Legend

Symbol	Name
aa0—aa1	Alternate Accumulators
ar0—ar3	Auxiliary BMU Registers
BIO	Bit Input/Output Unit
BMU	Bit Manipulation Unit
BREAKPOINT	Four Instruction Breakpoint Registers
BYPASS	JTAG Bypass Register
cbit	Control Register for BIO
EMUX	External Memory Multiplexer
HDS	Hardware Development System
ID	JTAG Device Identification Register
IDB	Internal Data Bus
ioc	I/O Configuration Register
JCON	JTAG Configuration Register
jtag	16-bit Serial/Parallel Register
pdx[7:0](in)	Parallel I/O Data Transmit Input Registers 7—0
pdx[7:0](out)	Parallel I/O Data Transmit Output Registers 7—0
PIO	Parallel Input/Output Unit
pioc	Parallel I/O Control Register
PSTAT	Parallel I/O Status Register
ROM	Internal ROM 512 Words
saddx	Multiprocessor Protocol Register
sbit	Status Register for BIO
sdx(in)	Serial Data Transmit Input Register
sdx2(in)	Serial Data Transmit Input Register for SIO2
sdx(out)	Serial Data Transmit Output Register
sdx2(out)	Serial Data Transmit Output Register for SIO2
SIO	Serial Input/Output Unit
SIO2	Serial Input/Output Unit #2
sioc	Serial I/O Control Register
sioc2	Serial I/O Control Register for SIO2
srta	Serial Receive/Transmit Address Register
tdms	Serial I/O Time-division Multiplex Signal Control Register
TIMER	Programmable Timer
timer0	Timer Running Count Register
timerc	Timer Control Register
TRACE	Program Discontinuity Trace Buffer
XAB	X Memory Space Address Bus
XDB	X Memory Space Data Bus
YAB	Y Memory Space Address Bus
YDB	Y Memory Space Data Bus

Hardware Architecture (continued)

Serial Input/Output Units (SIO and SIO2)

The SIO is an asynchronous, full-duplex, double-buffered channel that operates at up to 16 Mbits/s (for 33 ns part) or 20 Mbits/s (for 25 ns part) and easily interfaces with other AT&T fixed-point DSPs in a multiple-processor environment. Commercially available codecs and time-division multiplex (TDM) channels can be interfaced to the SIO with few, if any, additional components. SIO2 is a second serial port without multiprocessor capability. The SIO2 shares device pins with the PIO.

The SIO has been enhanced over the DSP16/A/C/J by adding an 8-bit serial protocol channel to the SIO multiprocessor mode. This feature uses the SADD pin and saddx register to transmit an 8-bit software-definable field in addition to the address of the called processor. This feature is useful for transmitting high-level framing information or for error detection and correction.

Parallel Input/Output Unit (PIO)

The PIO is a parallel I/O unit which can interface to a 16-bit bus containing other AT&T DSPs (e.g., DSP16/A/C/J, DSP1610, DSP1616), microprocessors, or peripheral I/O devices. The port data rate depends upon the instruction cycle rate. For a 33 ns instruction cycle, the PIO supports data rates up to 30 Mbytes/s. For a 25 ns instruction cycle, data rates up to 40 Mbytes/s are obtainable through this port.

The PIO is accessed in two basic modes, active or passive. Input or output can be configured in either of these modes independently. In active mode, the DSP1610 supports eight logical ports. The logical port number is output on PSEL[2:0]. In passive mode, PSEL[2:1] become inputs that allow for a glueless host interface to microprocessors.

Hardware Development System (HDS) Module

The on-chip HDS performs instruction breakpointing and branch tracing at full speed without additional off-chip hardware. The breakpointing is set up and the trace history read back remotely, through the JTAG port. The port works in conjunction with HDS code in the on-chip ROM and the hardware and software in a remote computer.

Four hardware breakpoints can be set on instruction addresses. A counter can be preset with the number of breakpoints to receive before trapping the core. Breakpoints can be set in interrupt service routines. Alternately, the counter can be preset with the number of cache instructions to execute before trapping the core.

Every time the program branches instead of executing the next sequential instruction, the addresses of the instructions executed before and after the branch are caught in circular memory. The memory contains the last four pairs of such program discontinuities for hardware tracing.

In systems with multiple processors, the processors may be configured such that any processor reaching a breakpoint will cause all the other processors to be trapped. (See Section 4.3, Interrupts and Trap.)

4.2 DSP1600 Core Architectural Overview

Figure 3 shows a block diagram of the DSP1600 core. It consists of the following sections.

System Cache and Control Section (SYS)

This section controls the instruction sequencing. It handles vectored interrupts and traps, contains a 15-word cache memory, and provides decoding for registers outside of the DSP1600 core. For external accesses, the processor cycle is stretched if wait-states are required (wait-states can be programmed by external segment). SYS sequences downloading via JTAG of self-test programs to the on-chip dual-port RAM.

SYS also contains a 15-word cache memory. Instructions can be loaded into the cache and executed multiple times. The cache loop iteration count can be specified at run time under program control as well as at assembly time.

Data Arithmetic Unit (DAU)

The data arithmetic unit (DAU) contains a 16 x 16-bit parallel multiplier that generates a full 32-bit product in one instruction cycle. The product can be accumulated with one of two 36-bit accumulators. The data in these accumulators can be directly loaded from or stored to memory in two 16-bit words with automatic saturation on overflow. The arithmetic logical unit (ALU) supports a full set of arithmetic and logical operations on either 16- or 32-bit data. A standard set of flags can be tested for conditional ALU operations, branches, and subroutine calls. This procedure allows the processor to perform as a powerful 16- or 32-bit microprocessor for logical and control applications. The available instruction set has been enhanced over that in the DSP16/A/C/J and is fully compatible with the DSP1616 instruction set.

The user also has access to two additional DAU registers. The psw register contains status information from the DAU (see Table 27, Processor Status Word Register). The arithmetic control register, auc, is used to configure some of the features of the DAU (see Table 28, Arithmetic Unit Control Register). The auc register is not cleared by reset and must be set up before using the DAU.

Hardware Architecture (continued)

The counters, c0 to c2, are signed, 8 bits wide and may be used to count events such as the number of times the program has executed a sequence of code. They are controlled by the conditional instructions and provide a convenient method of program looping.

Y Space Address Arithmetic Unit (YAAU)

The YAAU supports high-speed, register-indirect, compound, and direct addressing of data (Y) memory. Four general-purpose 16-bit registers, r0 to r3, are available in the YAAU. These registers can be used to supply the read or write addresses for Y space data. The YAAU also decodes the 16-bit data memory address and outputs individual memory enables for the data access. The YAAU can address the 4K or 8K of on-chip DPRAM and three external data memory segments. Up to 56 Kwords of off-chip RAM are addressable. Four individual addresses in an external data memory segment (the IO segment) also have individually decoded outputs.

Two 16-bit registers, rb and re, allow zero-overhead modulo addressing of data for efficient filter implementations. Two 16-bit signed registers, j and k, are used to hold user-defined postmodification increments. Fixed increments of +1, –1, and +2 are also available. Four compound-addressing modes are provided to make read/write operations more efficient.

The YAAU allows direct (or indexed) addressing of data memory. In direct addressing, the 16-bit base register, ybase, supplies the 11 most significant bits of the address. The direct data instruction supplies the remaining 5 bits to form an address to Y Space memory and also specifies one of 16 registers for the source or destination.

X Space Address Arithmetic Unit (XAAU)

This section supports high-speed, register-indirect, instruction/coefficient memory addressing with postmodification of the register. The 16-bit pt register is for addressing coefficients. The signed register i is used to hold a user-defined postincrement. A fixed postincrement of +1 is also available. Register pc is the program counter. Registers pr and pi hold the return address for subroutine calls and interrupts, respectively.

The adder and i register in the XAAU of the DSP1610 have been increased to 16 bits from the 12 bits used in the DSP16/A/C/J.

The XAAU decodes the 16-bit instruction/coefficient address and produces enable signals for the appropriate X memory segment. The addressable X segments are internal ROM (up to 12 Kwords), four or eight 1K banks of dual-port RAM, and external ROM.

The locations of these memory segments depend on which of the four memory maps is selected (see Table 5, Instruction/Coefficient Memory Maps). A security mode can be selected by mask option. This prevents unauthorized access to the contents of on-chip memory.

4.3 Interrupts and Trap

The DSP1610 supports vectored interrupts and a trap. The DSP has nine internal hardware sources of program interrupt and four external interrupt pins. Additionally, there is a trap pin and a trap signal from the HDS. A software interrupt is available through the icall instruction. Each of these sources of interrupt and trap has a unique vector address assigned to it. In addition, for compatibility with the DSP16/A/C/J, four of the internal sources and one external pin can cause a jump on interrupt to memory address (vector) 0x1, if enabled in the pioc register (see Table 29, Parallel I/O Control Register).

Vectored interrupts are enabled in the inc register (see Table 33, Interrupt Control Register) and monitored in the ins register (see Table 34, Interrupt Status (ins) Register).

A nonmaskable trap input is provided. The trap, which also has a unique interrupt vector, gains control of the DSP1610 at the end of the executing instruction even if the instruction is noninterruptible. When the 1610 goes into an interrupt or trap service routine, the IACK pin is asserted. Pins VEC[3:0] encode which interrupt/trap is being serviced. Table 4 details the encoding used for VEC[3:0].

Hardware Architecture (continued)

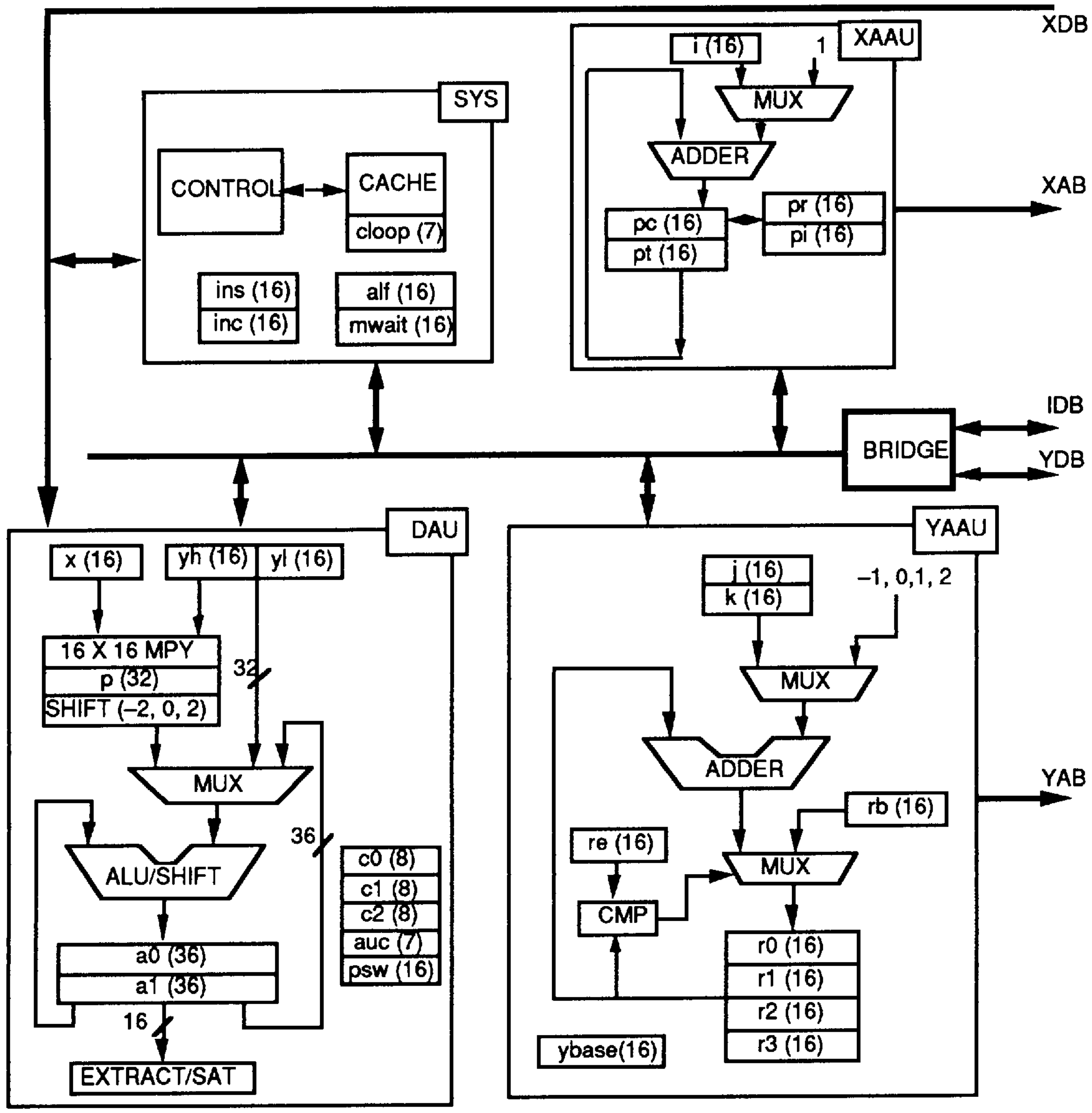


Figure 3. DSP1600 Core Block Diagram

Hardware Architecture (continued)

Table 3. DSP1600 Core Block Diagram Legend

Symbol	Name
16 x 16 MPY	16-Bit x 16-Bit Multiplier
a1—a0	Accumulators 1 and 0 (16-bit halves specified as a0, a0l and a1, a1l)*
alf	Await, Lowpr, Flags
ALU/SHIFT	Arithmetic Logic Unit/Shifter
auc	Arithmetic Unit Control
c2—c0	Counters 2—0
cloop	Cache Loop Count
CMP	Compare
DAU	Digital Arithmetic Unit
i	Increment Register
IDB	Internal Data Bus
inc	Interrupt Control
ins	Interrupt Status
j	Increment Register
k	Increment Register
MUX	Multiplexer
mwait	External Memory Wait-states
p	Product Register (16-bit halves specified as p, pl)
pc	Program Counter
pi	Program Interrupt Return Register
pr	Program Return Register
psw	Processor Status Word
pt	X Address Space Pointer
r3—r0	Y Address Space Pointers
rb	Modulo Addressing Register (Begin Address)
re	Modulo Addressing Register (End Address)
SYS	System Cache and Control Section
x	Multiplier Input Register
XAAU	X Space Address Arithmetic Unit
XAB	X Space Address Bus
XDB	X Space Data Bus
YAAU	Y Space Address Arithmetic Unit
YAB	Y Space Address Bus
YDB	Y Space Data Bus
ybase	Direct Addressing Base Register
y	y(High) DAU Register (16-bit halves specified as y, yl)

* F3 ALU instructions with immediates require specifying the high half of the accumulators as a0h and a1h.

Hardware Architecture (continued)

Interruptibility

Vectored interrupts are serviced only after an interruptible instruction or the completion of a prior interrupt service routine. If more than one vectored interrupt is asserted at the same time, the interrupts are serviced sequentially according to their assigned priorities. See Table 4 for the priorities assigned to the vectored interrupts. Interrupt service routines, branch and conditional branch instructions, cache loops, and instructions which only decrement one of the RAM pointers, r0 to r3 (e.g., *r3--), are not interruptible.

A trap is similar to an interrupt, but it gains control of the processor by branching to the trap service routine, even when the current instruction is noninterruptible. It may not be possible to return to normal instruction execution from the trap service routine, since the state of the machine cannot always be saved. In particular, program execution cannot be continued from a trapped cache loop or interrupt service routine. While in a trap service routine, another trap is ignored.

When set to 1, the status bits in the ins register indicate that an interrupt has occurred. The processor must reach an interruptible state (completion of an interruptible instruction or a prior interrupt service routine) before an enabled vectored interrupt will be acted on. An interrupt will not be serviced if it is not enabled. Polled interrupt service can be implemented by disabling the interrupt in the inc register, and then polling the ins register for the expected event.

Compatibility Mode

In the DSP16/A, the pioc register contains the enable and status bits for two PIO interrupts (PIDS, PODS), two SIO interrupts (IBF, OBE), and a hardware interrupt (INT0). For compatibility with DSP16/A programs, the 1610 pioc register contains enables and status bits for the INT0 pin and the PIDS, PODS, IBF, and OBE interrupts. See Table 29, Parallel I/O Control (pioc) Register. As in the DSP16/A, when enabled in the pioc register, all of these interrupts vector to a common address (0x1). These interrupts may also be enabled in the inc register. When enabled in the inc register, program control jumps to a different vector location for each. If they are enabled from both the inc and pioc registers, they are serviced as if they were enabled only in inc.

Vectored Interrupts

Tables 33 and 34 show the inc and ins registers. A logic 1 written to any bit of inc enables (or unmasks) the associated interrupt. If the bit is cleared to a logic 0, the interrupt is masked. The occurrence of an interrupt which is not masked will cause the program execution to transfer to the memory location pointed to by that interrupt's vector address, assuming no other interrupt is being serviced. (See Table 4, Interrupt Vector Table.) The occurrence of an interrupt that is masked causes no automatic processor action, but will set the corresponding status bit in the ins register. An interrupt that occurs while masked is latched and will cause an interrupt when unmasked. The status of the 13 interrupt sources is readable in the ins register, even if the interrupt is masked in the inc register. See the *DSP1610 Digital Signal Processor Information Manual* for a more detailed description of the interrupts.

Signaling Interrupt Service Status

Five pins of DSP1610 are devoted to signaling interrupt service status. The IACK pin goes high while any interrupt or trap is being serviced and goes low when the ireturn instruction from the service routine is issued. Four pins, VEC[3:0], carry a code indicating which of the interrupts or trap is being serviced. If no interrupt or trap is being serviced, VEC[3:0] are all 0. Table 4 contains the encodings.

Clearing Interrupts

The PIO interrupts (PIDS and PODS) are cleared, as in the DSP16/A, by reading or writing the parallel I/O data transmit registers pdx[in] and pdx[out], respectively. The SIO and SIO2 interrupts (IBF, IBF2, OBE, and OBE2) are cleared, as in the DSP16/A, by reading or writing, as appropriate, the serial data registers sdx[in], sdx2[in], sdx[out], and sdx2[out]. The JTAG interrupt (JINT) is cleared by reading the jtag register. Writing a 1 to the INT[0:3], TIME, or EMUXBOTH bits in the ins will cause the corresponding interrupt status bit to be cleared to a logic 0. Int[0:3] are also cleared when the ireturn is issued.

If interrupts occur concurrently, all interrupts are serviced according to their priority. The status bit for the vectored interrupt being serviced is cleared when the ireturn instruction is issued, leaving set any other vectored interrupts that are pending.

Trap Mechanism

The TRAP pin of the DSP1610 is a bidirectional signal. At reset, it is configured as an input to the processor. Asserting the TRAP pin will force a user trap. The trap mechanism is used for two purposes. It can be used by an application to rapidly gain control of the processor for asynchronous time-critical event handling (typically for catastrophic error recovery). It is also used by the HDS for breakpointing and gaining control of the processor. Separate vectors are provided for the user trap (0x46) and the HDS trap (0x3). A trap is not maskable.

Hardware Architecture (continued)

Table 4. Interrupt Vector Table

Source	Vector	Priority	VEC[3:0]	Issued by
No Interrupt	—	—	0x0	—
Software Interrupt	0x2	1	0x1	icall
IBF from pioc	0x1	1	0x1	SIO in
OBE from pioc	0x1	1	0x1	SIO out
PIDS from pioc	0x1	1	0x1	PIO in
PODS from pioc	0x1	1	0x1	PIO out
INT0	0x1	2	0x2	pin
JINT	0x42	3	0x8	jtag in
INT1	0x4	4	0x9	pin
INT2	0x8	5	0xa	pin
INT3	0xc	6	0xb	pin
TIMEOUT	0x10	7	0xc	timer
IBF2	0x14	8	0xd	SIO2 in
OBE2	0x18	9	0xe	SIO2 out
EMUXBOTH	0x1c	10	0x0	external access collision
IBF from inc	0x2c	14	0x3	SIO in
OBE from inc	0x30	15	0x4	SIO out
PIDS from inc	0x34	16	0x5	PIO in
PODS from inc	0x38	17	0x6	PIO out
TRAP from HDS	0x3	18	0xf	breakpoint, jtag, or pin
TRAP from User	0x46	19 = highest	0x7	pin

A trap has four cycles of latency. At most, two instructions will execute from the time the trap is received at the pin to when it gains control. An instruction that is executing when the trap occurs is allowed to complete before the trap service routine is entered. (Note that the instruction could be lengthened by wait-states.) During normal program execution, the pi register contains either the address of the next instruction (two-cycle instruction executing) or the address following the next instruction (one-cycle instruction executing). In an interrupt service routine, pi contains the interrupt return address. When a trap occurs during an interrupt service routine, the value of the pi register is overwritten. The result is that it is not possible to return to an interrupt service routine from the trap service routine. Continuing program execution when a trap occurs during a cache loop is also impossible.

The HDS trap causes circuitry to force the program memory map to MAP1 (with on-chip ROM starting at address 0x0) when the trap is taken. The previous memory map is restored when the trap service routine exits by issuing an ireturn. The map is forced to MAP1 because the HDS code resides in the on-chip ROM.

Using the AT&T development tools, the TRAP pin may be configured to be an output, or an input vectoring to address 0x3. In a multiprocessor environment, the TRAP pins of all the DSPs present can be tied together. During HDS operations, one DSP is selected by the host software to be the master. The master processor's TRAP pin is configured to be an output. The TRAP pins of the slave processors are configured as inputs. When the master processor reaches a breakpoint, the master's TRAP pin is asserted. The slave processors will respond to their TRAP input by beginning to execute the HDS code. For additional details, see the *DSP1600 Support Tools Manual.*

Hardware Architecture (continued)

AWAIT Interrupt (Standby Power Down)

Setting the AWAIT bit (bit 15) of the alf register (alf = 0x8000) causes the processor to go into a power-saving standby mode. Only the minimum circuitry on the chip required to process an incoming interrupt remains active. After the AWAIT bit is set, one additional instruction will be executed before the standby power saving mode is entered. A PIO or SIO word transfer will complete if already in progress. The AWAIT bit is reset when the first interrupt occurs. The chip then wakes up and continues executing.

Two NOPs should follow the instruction that sets the AWAIT bit. The first NOP (one cycle) will be executed before sleeping, the second will be executed after the interrupt signal awakens the DSP and before the interrupt service routine is executed.

For maximum power savings, in addition to setting alf = 0x8000, set ioc = 0x0600 and timerc = 0x0040. This will hold the CKO pin low and stop the timer clocks (see Tables 30, 35, 36 and 38).

4.4 Memory Maps and Wait-States

The DSP1610 implements a modified Harvard architecture that has separate on-chip 16-bit address and data buses for the instruction/coefficient (X) and data (Y) memory spaces. The DSP1610 contains 512 words of ROM (IROM) and 4K or 8K of dual-port RAM (DPRAM). It also provides a multiplexed external bus which accesses external RAM (ERAM) and ROM (EROM). Programmable wait-states are provided for external memory accesses. The instruction/coefficient memory map is configurable to provide application flexibility. Table 5 shows the four instruction/ coefficient memory maps available. Table 6 shows the data memory map, which is fixed.

Instruction/Coefficient Memory Map Selection

In determining which memory map to use, the processor evaluates the state of two parameters. The first is the LOWPR bit (bit 14) of the alf register. The LOWPR bit of the alf register is initialized to 0 automatically at reset. LOWPR controls the address in memory assigned to the 4K or 8K banks of dual-port RAM (i.e., RAM1, RAM2, etc). If LOWPR is low, internal dual-port RAM begins at address 0x3000. If LOWPR is high, internal dual-port RAM begins at address 0x0. LOWPR also moves the start of IROM from 0x0 in MAP1 to 0x2000 in MAP3 and the start of EROM from 0x0 in MAP2 to 0x2000 in MAP4.

The second parameter is the value at reset of the EXM pin (pin 20). EXM determines whether the internal 512-word ROM (IROM) will be addressable in the memory map.

The AT&T development system tools, together with the on-chip HDS circuitry and the jtag port, can independently set the memory map. Specifically, during an HDS trap, the memory map is forced to MAP1. The user's map selection is restored when the trap service routine has completed execution.

MAP1 has the 512-word IROM starting at 0x0, and the 4 or 8 Kword banks of DPRAM starting at 0x3000. External ROM (EROM) is accessed for addresses above 0x5000. This map is used if EXM is low at reset and the LOWPR parameter is programmed to zero. MAP1 is also used during an HDS trap.

MAP2 differs from MAP1 in that the lowest 12 Kwords reference external ROM. MAP2 is used if EXM is high at reset, the LOWPR parameter is programmed to zero, and an HDS trap is not in progress.

MAP3 has the 4 or 8 Kword banks of DPRAM located starting at 0x0. The 512-word IROM starts at address 0x2000. This map is used if EXM is low at reset, the LOWPR bit is programmed to one, and an HDS trap is not in progress. Note that this map is not available if the secure mask-programmable option has been ordered.

MAP4 differs from MAP3 in that addresses above 0x2000 reference external ROM. This map is used if the LOWPR bit is programmed to 1, an HDS trap is not in progress, and, either EXM is high during reset or the secure mask programmable option has been ordered.

Whenever the chip is reset using the RSTB pin, the default memory map will be MAP1 or MAP2, depending upon the state of the EXM pin as RSTB goes high. A reset through the HDS will not reinitialize the alf register so the previous memory map is retained.

Boot from External ROM

After RSTB goes from low to high, the DSP comes out of reset and fetches an instruction from address zero of the instruction/coefficient space. The physical location of address zero is determined by the memory map in effect. If EXM is high at the rising edge of RSTB, MAP2 is selected. MAP2 has EROM at location zero, thus program execution begins from external memory. If INT1 is low when RSTB rises, the mwait register defaults to 15 wait-states for all external memory segments. If INT1 is high, the mwait register defaults to 0 wait-states.

Hardware Architecture (continued)

Table 5. Instruction/Coefficient Memory Maps

Decimal Address	Address in pc, pt, pi, pr	MAP1* EXM = 0 LOWPR† = 0	MAP2 EXM = 1 LOWPR = 0	MAP3‡ EXM = 0 LOWPR = 1	MAP4 EXM = 1 LOWPR = 1
0	0x0000	IROM Reserved	EROM	RAM1	RAM1
1K	0x0400			RAM2	RAM2
2K	0x0800			RAM3	RAM3
	0x0C00			RAM4	RAM4
	0x1000			RAM5§	RAM5§
	0x1400			RAM6§	RAM6§
	0x1800			RAM7§	RAM7§
	0x1C00			RAM8§	RAM8§
8K	0x2000 0x2400 0x2800 0x2C00			IROM Reserved	EROM
12K	0x3000	RAM1	RAM1		
13K	0x3400	RAM2	RAM2		
14K	0x3800	RAM3	RAM3		
15K	0x3C00	RAM4	RAM4		
16K	0x4000	RAM5§	RAM5§		
17K	0x4400	RAM6§	RAM6§		
18K	0x4800	RAM7§	RAM7§		
19K	0x4C00	RAM8§	RAM8§		
20K 64K – 1	0x5000 • • • up to 0xFFFF	EROM	EROM	EROM	

* MAP1 is set automatically during an HDS trap. The user-selected map is restored at the end of the HDS trap service routine.
† LOWPR is an alf register bit. The AT&T development system tools can independently set the memory map.
‡ MAP3 is not available if the secure mask-programmable option is selected.
§ For the 4K RAM version of the DSP1610, these addresses are reserved.

Hardware Architecture (continued)

Table 6. Data Memory Map

Decimal Address	Address in r0, r1, r2, r3	Segment
0	0x0000	RAM1
1K	0x0400	RAM2
2K	0x0800	RAM3
3K	0x0C00	RAM4
4K	0x1000	RAM5*
5K	0x1400	RAM6*
6K	0x1800	RAM7*
7K	0x1C00 0x1FFF	RAM8*
8K 32K – 257	0x2000 0x7EFF	ERAMLO
32K – 256 32K – 1	0x7F00 0x7FFF	IO
32K 64K – 1	0x8000 0xFFFF	ERAMHI

* For the 4K RAM version of the DSP1610, these addresses are reserved.

On the data address side (Table 6), the four or eight banks of 1K DPRAM are located starting at 0. Addresses from 0x2000 to 0x7EFF reference low external data RAM (ERAMLO). Addresses from 0x7F00 to 0x7FFF reference a 256-word memory mapped I/O segment (IO). Addresses above 0x8000 reference high external data RAM (ERAMHI).

Wait-States

The number of wait-states (from 0 to 15) used when accessing each of the four external memory segments (ERAMLO, IO, ERAMHI, and EROM) is programmable in the mwait register (see Table 37). When the program references memory in one of the four external segments, the internal multiplexer is automatically switched to the appropriate set of internal buses and the associated external enable of ERAMLO, IO, ERAMHI, or EROM, is issued. The external memory cycle is automatically stretched by the number of wait-states configured in the appropriate field of the mwait register.

Due to the multiplexing of two internal buses to a single external bus, a reference to external data is not allowed in the same cycle that a reference is made to an external instruction/coefficient. If this condition occurs, the internal multiplexer defaults to the instruction/coefficient access, and the data access will not take place. If bit 11, EMUXBOTH, of the inc register is a logic 1, a vectored interrupt would then become pending. Refer to Technical Bulletin #232 for further discussion.

4.5 External Memory Interface (EMI)

The external memory interface supports read/write operations from instruction/coefficient memory, data memory, and memory-mapped I/O devices. The DSP1610 provides a 16-bit external address bus, AB[15:0], and a 16-bit external data bus, DB[15:0]. These buses are multiplexed between the internal buses for the instruction/coefficient memory and the data memory. Four external memory enables, ERAMLO, IO, ERAMHI, and EROM, select the external memory segment to be addressed.

If a data memory location with an address between 0x2000 and 0x7EFF is addressed, ERAMLO is asserted low.

If one of the 256 external data memory locations with an address greater than or equal to 0x7F00 and less than or equal to 0x7FFF is addressed, IO is asserted low. IO is intended for memory-mapped I/O. If one of the first four addresses (0x7F00—0x7F03) of these 256 external memory locations is addressed, the associated predecoded enable DSEL[3:0] is also asserted. The assertion level of DSEL[3:0] is programmable in the ioc register (see Tables 38 and 39).

If a data memory location with an address greater than or equal to 0x8000 is addressed, ERAMHI is asserted low. When a location in EROM in the X memory space is addressed, EROM is asserted low.

The ioc register (see Tables 38 and 39) provides the means for configuring a number of input and output parameters. Each of the eight enables can be individually programmed to delay the enable from the beginning of the external cycle. If bits 7 to 0 of the ioc register are set, the assertion of the associated enable is delayed by 1/2 CKO period.

When external data memory is written, the RWN signal goes low for the external cycle. The DB[15:0] is driven by DSP1610 starting halfway through the cycle. The data driven on DB[15:0] is automatically held after the cycle unless an external read cycle immediately follows.

Hardware Architecture (continued)

When an access to internal memory is made, the AB[15:0] bus holds the last valid external memory address. Asserting the RSTB pin low 3-states the AB[15:0] bus. After reset, the AB[15:0] value is undefined. The default definition of the segment and device enable pins is inactive-high, active-low. If bit 9 of the ioc register is set, the definition of the device enable pins will be inactive-low, active-high. Bits 10 and 11 of the ioc register select whether the CKO pin provides the free-running unstretched cycle clock (CKI/2), the wait-stated clock, CKI/(2[1+w]), a high level, or a low level (see Tables 39 and 40). The high-to-low transition of the wait-stated clock is synchronized with the high-to-low transition of the free-running CKO clock.

The flexibility provided by the programmable options of the external memory interface (see Table 37, the mwait Register and Table 38, the ioc Register) allows the DSP1610 to interface gluelessly with a variety of commercial memory chips.

4.6 Bit Manipulation Unit (BMU)

The BMU interfaces directly to the main accumulators in the DAU providing the following features:

- Barrel shifting — logical and arithmetic, left and right shift
- Normalization and extraction of exponent
- Bit-field extraction and insertion

These features increase the efficiency of the DSP in applications such as control or data encoding and decoding. For example, data packing and unpacking, in which short data words are packed into one 16-bit word for more efficient memory storage, is very easy.

In addition, the BMU provides two auxiliary accumulators (aa0 and aa1). 36-bit data can be shuffled, or swapped, between one of the main accumulators and one of the alternate accumulators in one instruction cycle. The ar[0:3] registers are 16-bit registers which control the operations of the BMU. They store a value which determines the amount of shift, or the width and offset for bit extraction or insertion. Certain operations in the BMU set flags in the DAU psw register and the alf register. See Table 27, Processor Status Word (psw) Register and Table 35, alf Register. The ar[0:3] registers can also be used as general-purpose registers.

The BMU instructions are detailed in Section 5.1. For a more complete description of the BMU see the *DSP1610 Digital Signal Processor Information Manual.*

4.7 Serial I/O Units

The serial I/O ports on the DSP1610 device provide a serial interface to many codecs and signal processors with little, if any, external hardware required. Each high-speed, double-buffered port (sdx and sdx2) supports back-to-back transmissions of data. The output buffer empty (OBE and OBE2) and input buffer full (IBF and IBF2) flags facilitate the reading and/or writing of each serial I/O port by program- or interrupt-driven I/O. The OBE flag indicates that the content of the output buffer has been transferred to the output shift register. OSE indicates that the data has been shifted out of the output shift register. It can be used by external hardware to latch a shift register receiving the output data.

The SIO port clocks may be active (outputs driven by the DSP1610) or passive (inputs driven by external logic). ILD and OLD can also be configured as active or passive. See the *DSP1610 Digital Signal Processor Information Manual* for more information on active and passive modes of operation.

There are four selectable active clock speeds. A bit-reversal mode provides compatibility with either the most significant bit (MSB) first or least significant bit (LSB) first serial I/O formats. See Table 23, Serial I/O Control Registers (sioc and sioc2). A multiprocessor I/O configuration is supported by the primary SIO. This feature allows up to eight DSP1610 devices to be connected together without requiring external glue logic.

The serial data may be internally looped back by setting the SIO loopback control bit, SIOLBC, of the ioc register. SIOLBC affects both the SIO and SIO2. The data output signals are wrapped around internally from the output to the input [DO to DI]. For proper operation, the SIO clocks and loads (ICK, OCK, ILD and OLD) should either all be in the active mode, 16-bit condition, or each pair (ICK/OCK, ILD/OLD) should be driven from one external source in passive mode. During loopback, pins DO, DO2, ICK, ICK2, OCK, OCK2, ILD, ILD2, OLD, OLD2, SADD, SYNC, and DOEN are 3-stated.

Hardware Architecture (continued)

Programmable Modes

Programmable modes of operation for the SIO and SIO2 are controlled by the serial I/O control (sioc and sioc2) registers. These registers, shown in Table 23, are used to set the ports into various configurations. Both input and output operations can be independently configured as either active or passive. When active, the DSP1610 generates load and clock signals. When passive, load and clock signal pins are inputs.

Since input and output can be independently configured, each SIO has four different modes of operation. Each of the sioc registers is also used to select the frequency of active clocks for that SIO. Finally, these registers are used to configure the serial I/O data formats. The data can be 8 or 16 bits long, and can also be input/output MSB first or LSB first. Both input and output data formats can be independently configured.

Multiprocessor Mode

The SIO port, but not the SIO2 port, may be used in a multiprocessor mode. The multiprocessor mode allows up to eight processors (DSP1610, DSP16/A/C/J, or DSP1616) to be connected together to provide data transmission among any of the individual DSPs in the system. The multiprocessor interface is a four-wire interface, consisting of a data channel, an address/protocol channel, a transmit/receive clock, and a sync signal (see Figure 4). The DI and DO pins of all the DSPs are connected to transmit and receive the data channel. The SADD pins of all the DSPs are connected to transmit and receive the address/protocol channel. ICK and OCK should be tied together and driven from one source. The SYNC pins of all the DSPs are connected.

In the configuration shown in Figure 4, the master DSP (DSP0) generates active SYNC and OCK signals while the slave DSPs use the SYNC and OCK signals in passive mode to synchronize operations. In addition, all DSPs must have their ILD and OLD signals in active mode. While ILD and OLD are not required externally for multiprocessor operation, they are used internally in the DSP's SIO. Setting the LD field of the master's sioc register to a logic level 1 will ensure that the active generation of SYNC, ILD, and OLD is derived from OCK (see Table 23). With this configuration, all DSPs should use ICK (tied to OCK) in passive mode to avoid conflicts on the clock line. See the *DSP1610 Digital Signal Processor Information Manual* for more information.

Four registers are used to configure the multiprocessor mode. They are the time-division multiplexed slot register (tdms), the serial receive and transmit address register (srta), the serial data transmit register (sdx), and the multiprocessor serial address/protocol register (saddx). Since the SIO2 unit does not support multiprocessor mode, it does not have analogous registers.

Multiprocessor mode requires no external logic and uses a TDM interface with eight 16-bit time slots per frame. The transmission in any time slot consists of 16 bits of serial data in the data channel and 16 bits of address and protocol information in the address/protocol channel. The address information consists of the transmit address field of the srta register of the transmitting device. The address information is transmitted concurrently with the transmission of the first 8 bits of data. The protocol information consists of the transmit protocol field written to the saddx register and is transmitted concurrently with the last 8 bits of data (see Table 26, Saddx Multiprocessor Protocol Register). Data is received or recognized by other DSP(s) whose receive address matches the address in the address/protocol channel. Each SIO port has a user-programmable receive address and transmit address associated with it. The transmit and receive addresses are programmed in the srta register.

In multiprocessor mode, each device can send data in a unique time slot designated by the tdms register transmit slot field (bits 7—0). The tdms register has a fully decoded transmit slot field in order to allow one DSP1610 device to transmit in more than one time slot. This procedure is useful for multiprocessor systems with less than eight DSP1610 devices when a higher bandwidth is necessary between certain devices in that system. The DSP (the master), operating during time slot 0, drives the SYNC.

Each SIO also has a fully decoded transmitting address specified by the srta register transmit address field (bits 7—0). This is used to transmit information regarding the destination(s) of the data. The fully decoded receive address specified by the srta register receive address field (bits 15—8) determines which data will be received.

The SIO protocol channel data is controlled via the saddx register. When the saddx register is written, the lower 8 bits contain the 8-bit protocol field. On a read, the high-order 8 bits read from saddx are the most recently received protocol field sent from the transmitting DSP's saddx output register. The low-order 8 bits are read as zeros.

An example use of the protocol channel is to use the top 3 bits of the saddx value as an encoded source address for the DSPs on the multiprocessor bus. This leaves the remaining 5 bits available to convey additional control information, such as whether the associated field is an opcode or data, or whether it is the last word in a transfer, etc. These bits can also be used to transfer parity information about the data. Alternatively, the entire field can be used for data transmission, boosting the bandwidth of the port by 50%.

Hardware Architecture (continued)

SIO2

Upon reset, the second serial unit, SIO2, is disconnected. Setting bit 13, ESIO2, of the ioc register connects SIO2 to the pins and disables the PIO. SIO2 does not support multiprocessing, but is otherwise identical to SIO.

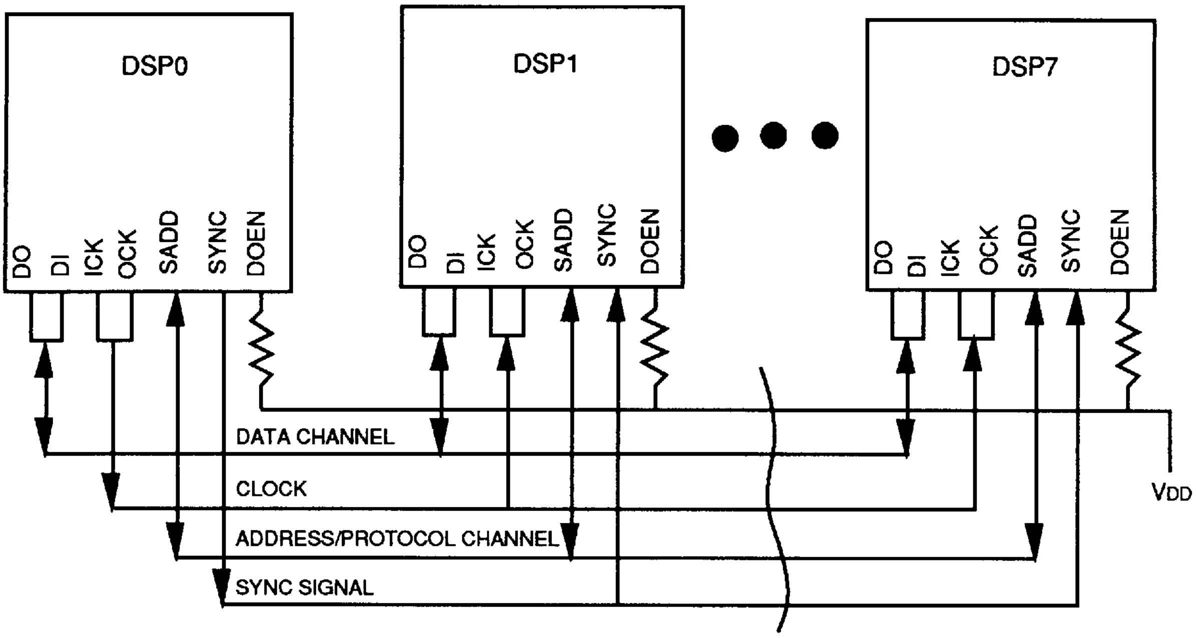


Figure 4. Multiprocessor Communication and Connections

Hardware Architecture (continued)

4.8 Parallel I/O Unit (PIO)

The DSP1610 has a 16-bit parallel I/O interface for rapid transfer of data with external devices. The PIO can operate in the active mode (data strobes provided by the DSP) or in the passive mode (data strobes provided by an external device). As a passive port, the PIO acts as a flexible host interface, requiring little or no glue logic to interface to other devices (e.g., microcontrollers, microprocessors, or another DSP). Five maskable interrupts are included in the PIO unit. Although there is only one physical PIO port, there are eight logical PIO ports: pdx0 through pdx7. The eight logical ports are distinguished by the state of the peripheral select pins (PSEL[2:0]).

The data path of the PIO is comprised of a 16-bit input buffer, pdx(in), and a 16-bit output buffer, pdx(out). There are two pins which indicate the state of these buffers, parallel input buffer full (PIBF) and parallel output buffer empty (POBE), in passive mode. The pdx(in) register is shadowed in some modes to allow the PIO to accept data in an interrupt service routine without disrupting its normal operation. In addition, there are two registers used to control and monitor the PIO's operation, the Parallel I/O Control (pioc, see Table 29) register, and the PIO Status (PSTAT, see Table 8) register. The PSTAT register can only be read by an external device, and it reflects the condition of the PIO. The pioc contains information about interrupts and can be used to set the PIO in a variety of modes. Access times are programmable via the strobe field in the pioc. The PIO is accessed in two basic modes, active or passive. Input or output can be configured in either of these modes independently.

If the PIO loopback control (PIOLBC) of the ioc register is set, the PIO register is wrapped around internally from output to input. Setting PODS in active mode and PIDS in passive mode with PIOLBC set 3-states the PB[7:0] pins as well as PODS and PIDS, where PODS now drives PIDS internally. Thus, the PIO loopback feature allows a PIO write to automatically generate an input data strobe (PIDS). If the pioc register is configured for active output and passive input with PIOLBC set, pdx[7:0] can then be used as a temporary storage register.

Active Mode Operation

In active mode, parallel input data strobe (PIDS) is an output which indicates when data can be put on the bus during a read. Likewise, parallel output data strobe (PODS) is an output which indicates when data is available on the bus during a write. The STROBE field of the pioc configures the width of PIDS and PODS when used as active signals, allowing the port to be used with a variety of peripherals with different access times.

If both input and output are configured for active mode, the three pins PSEL[2:0] are outputs of the DSP1610 indicating which of the eight channels are being accessed. If either input or output are passive, some of these pins become inputs, and serve different purposes.

Passive Mode Operation

In passive mode, the DSP1610 device can be used as a peripheral to other devices such as a microprocessor. Bits 12 (PODS) and 11 (PIDS) of the pioc register are used to configure the passive mode. Providing that their respective interrupt enable bits are set in the pioc or inc registers, the assertion of PIDS and PODS by an external device causes the DSP1610 device to recognize an interrupt. This provides functional synchronization between the DSP1610 device and an external device. The PIO interrupts can be configured in vectored mode (via the inc register) or DSP16/A compatibility mode (via the pioc register).

The function of the three PSEL pins changes depending upon the selected mode (active or passive) of the PIDS and PODS bits in the pioc register. Table 7 shows the effects of various modes on the PSEL[2:0] bits.

When PODS is active and PIDS is passive, PSEL1 and PSEL0 are outputs with the encoded pdx[3:0] channel number (i.e., pdx[7:4] will alias into pdx[3:0]). PSEL2 is an input which, when low, enables PIDS.

When PODS is passive and PIDS is active, PSEL0 is an output indicating if the encoded channel number is odd or even (i.e., pdx 0, 2, 4, and 6 alias into 0, while pdx1, 3, 5, 7 alias into 1). PSEL2 is an input that, when low, enables PODS. PSEL1 is an input which, when high, causes PSTAT (port status) instead of pdx (data) to be output to PB[15:0]. This register is shown in Table 8. It is not readable or writable from the DSP program.

When PIDS and PODS are both passive, PSEL2 is an input which, when low, enables PODS and PIDS. PSEL1 is an input which, when high, causes PSTAT instead of data to be output to PB[15:0]. PSEL0 is an output which is the logical OR of the signals PIBF and POBE. This output can be useful if the user wants to have one signal that tells an external device when the DSP1610 is ready for a PIO access.

If passive mode is enabled for either PIDS or PODS, PSEL2 becomes an active-low enable or chip select input. While PSEL2 is high, the DSP1610 ignores any activity of the passive PIDS and/or PODS. If a DSP1610 using passive strobes is intended to be continuously enabled, PSEL2 should be grounded through a 10 kΩ resistor.

Hardware Architecture (continued)

Table 7. PSEL Function

PODS	PIDS	PSEL2	PSEL1	PSEL0
active	active	output (psel2)	output (psel1)	output (psel0)
active	passive	input (enablebar)	output (psel1)	output (psel0)
passive	active	input (enablebar)	input (status/data)	output (psel0)
passive	passive	input (enablebar)	input (status/data)	output (pibf or pobe)

Table 8. PSTAT Register as Seen on PB[15:0]

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	RESERVED													LPIDS	PIBF	POBE

Notes:
LPIDS = 1 indicates active mode input.
PIBF = 1 indicates input buffer full.
POBE = 1 indicates output buffer empty.

4.9 Bit Input/Output Unit (BIO)

The BIO individually controls the directions of eight bidirectional control I/O pins, IOBIT[7:0]. If a pin is configured as an output, it can be individually set, cleared, or toggled. If a pin is configured as an input, it can be read and/or tested.

The lower half of the sbit register (see Table 31) contains the current values (VALUE[7:0]) of the eight bidirectional pins, IOBIT[7:0]. The upper half of sbit register (DIREC[7:0]) controls the direction of each pin. A logic 1 configures the corresponding pin to an output; a logic 0 configures it as an input. The upper half of sbit is cleared upon reset.

The cbit register (see Table 32) contains two 8-bit-fields, MODE/MASK[7:0] and DATA/PAT[7:0]. The values of DATA[7:0] are cleared upon reset. The meaning of a bit in either field depends on whether it has been configured as an input or an output in sbit. If a bit has been configured to be an output, the meanings are MODE and DATA. For input, the meanings are MASK and PAT(tern). Table 9 shows the functionality of the MODE/MASK and DATA/PAT bits based on the direction selected for the associated IOBIT pin.

Those bits that have been configured as inputs can be individually tested for 1 or 0. For those inputs which are being tested, there are four flags produced: allt (all true), allf (all false), somet (some true), and somef (some false). These flags can be used for conditional branch or special instructions. The state of these flags can be saved and restored by reading and writing bits 0 to 3 of the alf register (see Table 35).

Table 9. BIO Operations

Direction	MODE/ MASK	DATA/ PAT	Action
1 (Output)	0	0	Clear
1 (Output)	0	1	Set
1 (Output)	1	0	No Change
1 (Output)	1	1	Toggle
0 (Input)	0	0	No Test
0 (Input)	0	1	No Test
0 (Input)	1	0	Test for Zero
0 (Input)	1	1	Test for One

If a BIO pin is switched from being an output to being an input and then back to being an output, the pin remembers the previous output value.

Hardware Architecture (continued)

4.10 Timer

The interrupt timer is composed of three main blocks: the timer control register, the prescaler, and the timer itself. The timer control register (see Table 30, timerc Register) sets up the operational state of the timer and prescaler. The timer0 register is used to hold the timer reload value, and to set the initial value of the timer.

The prescaler slows down the DSP1610 clock by a number of possible divisors to allow for a wide range of interrupt delay periods.

The timer is a 16-bit down counter which can be loaded with an arbitrary number from software. It then counts down to 0 at the clock rate provided by the prescaler. Upon reaching 0 count, an interrupt is issued to the DSP1610 through a vectored interrupt (bit 8 of inc and ins registers). The timer will then either wait in a quiescent state for another command from software, or will automatically repeat the last interrupting period, depending upon the state of the RELOAD bit in the timerc register.

The timer interrupt can be individually enabled or disabled through the inc register. The timer can be stopped and started by software, and can be reloaded with a new delay at any time. Its current value can also be read by software. When the DSP1610 is reset, the timer is in an inactive state. The RELOAD bit of the timerc register (see Table 30) selects one of two operating modes for the interrupt timer.

When RELOAD is 0, the timer counts down from a specified value to 0, interrupts the DSP1610, and then stops, awaiting the load of a new value. When RELOAD is 1, the timer counts down from a specified value to 0, interrupts the DSP1610, automatically reloads the specified initial value into the timer, and repeats indefinitely. This provides for either a single timed interrupt event or a regular interrupt clock of arbitrary period.

The T0EN bit of the timerc register enables the clock to the timer. When T0EN is a 1, the timer counts down towards 0. When T0EN is a 0, the timer holds its current count.

The PRESCALE field of the timerc register selects one of 16 possible clock speeds for the timer input clock. Table 10 describes the options.

Setting the DISABLE bit of the timerc register to a logic 1 shuts down the timer and the prescaler for power savings.

Writing to the timer0 register causes both the timer itself and the reloadable period register to be written with the specified 16-bit number. The timer (when enabled with T0EN) then starts counting down from this number to 0, at the clock rate specified by the PRESCALE field, one count per clock cycle. When the timer reaches 0, the DSP1610 is interrupted, vectoring to 0x10, assuming the interrupt is enabled.

Upon reaching a count of 0, the timer either remains quiescent until another value is written to the timer0 register (RELOAD = 0), or automatically reloads the previous starting value from the period register into the timer register and begins counting down (RELOAD = 1). At any time in the sequence, a new value can be written by the software into the timer and period registers (e.g., timer0 = 0x0300). The timer then starts counting down from this new value.

The timer0 register can also be read at any time. The timer value at the time of the read is the value returned.

When the DSP1610 is reset, the bottom 6 bits of the timerc register and the timer itself are initialized to 0. This sets the prescaler to CKI/4, turns off the reload feature, disables timer counting, and initializes the timer value to its inactive state. The act of resetting the chip does not cause the timer interrupt. Note that the period register is not initialized on reset.

Table 10. PRESCALE Field

PRESCALE	Frequency	Frequency at CKI = 60 MHz
0000	CKI/4	15 MHz
0001	CKI/8	7.5 MHz
0010	CKI/16	3.25 MHz
0011	CKI/32	1.875 MHz
0100	CKI/64	937.5 kHz
0101	CKI/128	468.75 kHz
0110	CKI/256	234.38 kHz
0111	CKI/512	117.19 kHz
1000	CKI/1024	58.59 kHz
1001	CKI/2048	29.30 kHz
1010	CKI/4096	14.65 kHz
1011	CKI/8192	7.32 kHz
1100	CKI/16384	3.66 kHz
1101	CKI/32768	1.83 kHz
1110	CKI/65536	915.5 kHz
1111	CKI/131072	457.8 Hz

Hardware Architecture (continued)

4.11 JTAG Test Port

The DSP1610 incorporates a JTAG/IEEE 1149.1 standard four-wire test port. This port allows standard board environment testing. An I/O scan path is provided with bypass capability and the DSP1610 implements the optional device identification register.

Boundary-Scan Register

All of the chip inputs and outputs are incorporated in a JTAG scan path that follows. The meanings of the values in the TYPE column are as follows:

I = input
O = 3-state output cell
B = bidirectional (I/O) cell
OE = 3-state controller
DC = bidirectional controller cell

Table 11. JTAG Scan Register

Note: The direction of shifting is from TDI to cell 124 to cell 123 . . . to cell 0 to TDO.

Cell	Type	Signal Name
0	O	POBE*
1	O	PIBF*
2	B	PODS*
3	DC	controls cell 2
4	DC	controls cell 5
5	B	PIDS*
6—13	B	PB[0:7]* (cell #6 is PB0, etc.)
14—21	B	PB[8:15]*
22	B	TRAP
23	DC	controls cell 22
24	DC	controls cells 6—13
25	DC	controls cells 14—21
26—29	I	INT[0:3]
30—33	O	VEC[3:0]
34	O	RWN
35	O	IACK
36—39	O	DSEL[3:0]
40—55	B	DB[15:0]
56	DC	controls cells 40—55
57—60	O	ERAMLO, IO, ERAMHI, EROM
61—72	O	AB[0:11]
73	DC	controls cell 78
74—77	O	AB[12:15]
78	B	I/OBIT0
79	B	I/OBIT1
80	DC	controls cell 79
81	DC	controls cell 82
82	B	I/OBIT2
83	B	I/OBIT3
84	B	I/OBIT4
85	B	I/OBIT5
86	DC	controls cell 83
87	DC	controls cell 84
88	B	I/OBIT6
89	B	I/OBIT7
90	DC	controls cell 85
91	DC	controls cell 88
92	DC	controls cell 89
93	DC	controls cell 96
94	O	OBE
95	O	OSE
96	B	ILD
97	DC	controls cell 101
98	DC	controls cell 104
99	O	IBF
100	I	DI
101	B	ICK
102	DC	controls cell 105
103	DC	controls cell 110
104	B	OCK
105	B	OLD
106	O	DO
107	OE	controls cell 106
108	DC	controls cell 111
109	DC	controls cell 112
110	B	SYNC
111	B	SADD
112	B	DOEN
113	O	PSEL0*
114	B	PSEL1*
115	B	PSEL2*
116	DC	controls cell 114
117	DC	controls cell 115
118	OE	controls cells 34, 36—39, 57—60, 122
119	OE	controls cells 0, 1, 30—33, 35, 61—72, 74—77, 94, 95, 99, 113
120	I	READY
121	I	EXM
122	O	CKO
123	I	RSTB
124	I	CKI

* For multiplexed SIO2 signal names, see Table 23.

Hardware Architecture (continued)

Other JTAG Features

The JTAG ID for different codes of the DSP1610 are listed in Table 12 below.

Table 12. JTAG ID

Code	JTAG ID
F10	0x0EA0203B
F11	0x1EA0203B
F12	0x2EA0203B
F12P*	0x3EA0203B
F13	0x4EA0203B

* The "P" in F12P does not appear on the package label, but will appear in the simulator message when the JTAG ID is interrogated.

For future products, the four most significant bits [31:28] will change.

The instruction register is 4 bits long. The instruction for accessing the device ID is 0xE (1110).

There is no separate TRST input pin.

Behavior of the instruction register is summarized in Table 13. In the following, cell 0 is LSB (closest to TDO).

Table 13. JTAG Instruction Register

IR cell #:	3	2	1	0
parallel input?	Y	Y	N	N
always logic 1?	N	N	N	Y
always logic 0?	N	N	Y	N

The first line shows the cells in the IR that capture from a parallel input in the capture-IR controller state. The second line shows the cells that always load a logic 1 in the capture-IR controller state. The third line shows the cells that always load a logic 0 in the capture-IR controller state. Cell 3 (MSB of IR) is tied to status signal PINT, and cell 2 is tied to status signal JINT. The state of these signals can therefore be captured during capture-IR and shifted out during SHIFT-IR controller states.

5.0 Software Architecture

5.1 Instruction Set

The DSP1610 processor has seven types of instructions: multiply/ALU, special function, control, F3 ALU, BMU, cache, and data move. The multiply/ALU instructions are the primary instructions used to implement signal processing algorithms. Statements from this group can be combined to generate multiply/accumulate, logical, and other ALU functions and to transfer data between memory and registers in the data arithmetic unit. The special function instructions can be conditionally executed based on flags from the previous ALU or BMU operation, the condition of one of the counters, or the value of a pseudorandom bit in the DSP1610 device. Special function instructions perform shift, round, and complement functions. The F3 ALU instructions enrich the operations available on accumulators. The BMU instructions provide high-performance bit manipulation. The control instructions implement the goto and call commands. Control instructions can also be executed conditionally. Cache instructions are used to implement low-overhead loops, conserve program memory, and decrease the execution time of certain multiply/ALU instructions. Data move instructions are used to transfer data between memory and registers or between accumulators and registers. See the *DSP1610 Digital Signal Processor Information Manual* for a detailed description of the instruction set.

The following operators are used in describing the instruction set:

*	16 x 16-bit –> 32-bit multiplication **or** register-indirect addressing when used as a prefix to an address register **or** denotes direct addressing when used as a prefix to an immediate
+	36-bit addition[†]
–	36-bit subtraction[†]
>>	Arithmetic right shift
>>>	Logical right shift
<<	Arithmetic left shift
<<<	Logical left shift
|	36-bit bitwise OR[†]
&	36-bit bitwise AND[†]
^	36-bit bitwise EXCLUSIVE OR[†]
:	Compound address swapping, accumulator shuffling
~	One's complement

[†] These are 36-bit operations. One operand is 36-bit data in an accumulator; the other operand may be 16, 32, or 36 bits.

Software Architecture (continued)

Multiply/ALU Instructions

Note that the function statements and transfer statements in Table 14 are chosen independently. Any function statement (F1) can be combined with any transfer statement to form a valid multiply/ALU instruction. If either statement is not required, a single statement from either column constitutes a valid instruction. The number of cycles to execute the instruction is a function of the transfer column. (An instruction with no transfer statement executes in one instruction cycle.) Whenever pc, pt, or rM is used in the instruction and points to external memory, the programmed number of wait-states must be added to the instruction cycle count. All multiply/ALU instructions require one word of program memory. The no-operation (nop) instruction is a special case encoding of a multiply/ALU instruction and executes in one cycle. The assembly language representation of a nop is either nop or a single semicolon.

Table 14. Multiply/ALU Instructions

F1 Function Statement	Transfer Statement*	Transfer Statement Cycles‡	
		Cycles† Not Using Cache	Cycles† Using Cache
p=x * y	y=Y x=X	2	1
aD=p p=x * y	y=aT x=X	2	1
aD=aS+p p=x * y	y[l]=Y	1	1
aD=aS–p p=x * y	aT[l]=Y	1	1
aD=p	x=Y	1	1
aD=aS+p	Y	1	1
aD=aS–p	Y=y[l]	2	2
aD=y	Y=aT[l]	2	2
aD=aS+y	Z:y x=X	2	2
aD=aS–y	Z:y[l]	2	2
aD=aS&y	Z:aT[l]	2	2
aD=aS|y	(none)	1	1
aD=aS^y			
aS–y			
aS&y			

* The l in [] is an optional argument that specifies the low 16 bits of aT or y. For transfer statements that load the upper half of an accumulator, the lower half is cleared if the corresponding CLR bit in the auc register is zero. auc is not cleared by reset. See Section 5.2, Register Settings.

† A instruction cycle is twice the period of the input clock, CKI.

‡ Add cycles for the following cases:

1. When an external memory access is made in X or Y space and wait-states are programmed, add the number of wait-states.
2. If an X space access and a Y space access are made to the same bank of DPRAM in one instruction, add one cycle.

Table 15. Replacement Table for Multiply/ALU Instructions

Replace	Value	Meaning
aD, aS, aT	a0, a1	One of two DAU accumulators.
X	*pt++, *pt++i	X Space memory location pointed to by pt. pt is postmodified by +1 and i, respectively.
Y	*rM, *rM++, *rM--, rM++j	RAM location pointed to by rM (M = 0, 1, 2, 3). rM is postmodified by 0, +1, –1, or j, respectively.
Z	*rMzp, *rMpz, *rMm2, *rMjk	Read/Write compound addressing. rM (M = 0, 1, 2, 3) is used twice. First, postmodified by 0, +1, –1, or j, respectively; and, second, postmodified by +1, 0, +2, or k, respectively.

Software Architecture (continued)

Special Function Instructions

All forms of the special function require one word of program memory and execute in one instruction cycle. (If pc points to external memory, add programmed wait-states.)

aD=aS>>1		Arithmetic right shift (sign preserved) of 36-bit accumulators
aD=aS>>4		
aD=aS>>8		
aD=aS>>16		
aD=aS	—	load destination accumulator from source accumulator
aD=—aS	—	2's complement
aD=~aS*	—	1's complement
aD=rnd(aS)	—	Round upper 20 bits of accumulator
aDh=aSh+1	—	Increment upper half of accumulator (lower half cleared)
aD=aS+1	—	Increment accumulator
aD=y	—	load accumulator with 32-bit y register value with sign extend
aD=p	—	load accumulator with 32-bit p register value with sign extend
aD=aS<<1		Arithmetic left shift (sign not preserved) of the lower 32 bits of accumulators (upper 4 bits are sign-bit-extended from bit 31 at the completion of the shift)
aD=aS<<4		
aD=aS<<8		
aD=aS<<16		

* This function is not available for the DSP16/A/C/J.

The above special functions can be conditionally executed, as in:

```
    if CON instruction
```

and with an event counter

```
    ifc CON instruction
```

which means:

```
    if CON is true then
        c1=c1+1
        instruction
        c2=c1
    else
        c1=c1+1
```

The above special function statements can be executed unconditionally by writing them directly, e.g., a0=a1.

Table 16. Replacement Table for Special Function Instructions

Replace	Value	Meaning
aD aS	a0, a1	One of two DAU accumulators
CON	mi, pl, eq, ne, gt, le, lvs, lvc, mvs, mvc, c0ge, c0lt, c1ge, c1lt, heads, tails, true, false, allt, allf, somet, somef, oddp, evenp, mns1, nmns1, npint, njint	See Table 18 for definitions of mnemonics

Software Architecture (continued)

Control Instructions

All control instructions executed unconditionally execute in two cycles, except icall which takes three cycles. Control instructions executed conditionally execute in three instruction cycles. (If pc, pt, or pr point to external memory, add programmed wait-states.) Control instructions executed unconditionally require one word of program memory, while control instructions executed conditionally require two words. Control instructions cannot be executed from the cache.

```
goto JA*
goto pt
call JA*
call pt
icall
return     (goto pr)
ireturn    (goto pi)
```

* The goto JA and call JA instructions should not be placed in the last or next-to-last instruction before the boundary of a 4 Kword page. If the goto or call is placed there, the program counter will have incremented to the next page and the jump will be to the next page, rather than to the desired current page.

The above control instructions, with the exception of ireturn and icall, can be conditionally executed. For example:

```
if le goto 0x0345
```

Table 17. Replacement Table for Control Instructions

Replace	Value	Meaning
CON	mi, pl, eq, ne, gt, le, nlvs, lvc, mvs, mvc, c0ge, c0lt, c1ge, c1lt, heads, tails, true, false, allt, allf, somet, somef, oddp, evenp, mns1, nmns1, npint, njint	See Table 18 for definitions of mnemonics
JA	12-bit value	Least significant 12 bits of absolute address within the same 4 Kword memory section

Software Architecture (continued)

Conditional Mnemonics (Flags)

Table 18 lists mnemonics used in conditional execution of special function and control instructions.

Table 18. DSP1610 Conditional Mnemonics

Test	Meaning	Test	Meaning
pl	Result is nonnegative (sign bit is bit 35) ≥ 0	mi	Result is negative <0
eq	Result is equal to 0 $=0$	ne	Result is not equal to 0 $\neq 0$
gt	Result is greater than 0 >0	le	Result is less than or equal to 0 ≤ 0
lvs	Logical overflow set[*]	lvc	Logical overflow clear
mvs	Mathematical overflow set[†]	mvc	Mathematical overflow clear
c0ge	Counter 0 greater than or equal to 0	c0lt	Counter 0 less than 0
c1ge	Counter 1 greater than or equal to 0	c1lt	Counter 1 less than 0
heads	Pseudorandom sequence bit set	tails	Pseudorandom sequence bit clear
true	The condition is always satisfied in an if instruction	false	The condition is never satisfied in an if instruction
allt	All True, all BIO input bits tested compared successfully	allf	All False, no BIO input bits tested compared successfully
somet	Some True, some BIO input bits tested compared successfully	somef	Some False, some BIO input bits tested did not compare successfully
oddp	Odd Parity, from BMU operation	evenp	Even Parity, from BMU operation
mns1	Minus 1, result of BMU operation	nmns1	Not Minus 1, result of BMU operation
npint	Not PINT — used by hardware development system	njint	Not JINT — used by hardware development system

[*] Result is not representable in the 36-bit accumulators (36-bit overflow).

[†] Bits 35–31 are not the same (32-bit overflow).

Notes:

Testing the state of the counters (c0 or c1) automatically increments the counter by one.

The 10-bit pseudorandom sequence is reset by writing the pi register when **not** in an interrupt service routine. Writing to the pi register will not affect its contents **except** during interrupt service routines.

Software Architecture (continued)

F3 ALU Instructions

These instructions are implemented in the DSP1600 core. They allow accumulator two-operand operations with either another accumulator, the p register, or a 16-bit immediate operand. The result is placed in a destination accumulator that can be independently specified. All operations are done with the full 36 bits. For the accumulator with accumulator operations, both inputs are 36 bits. For the accumulator with p register operations, the p register is sign-extended into bits 35—32 before the operation. For the accumulator high with immediate operations, the immediate is sign extended into bits 35—32 and the lower bits 15—0 are filled with zeros, except for the AND operation, for which they are filled with ones. These conventions allow the user to do operations with 32-bit immediates by programming two consecutive 16-bit immediate operations. The F3 ALU instructions are shown below.

Table 19. F3 ALU Instructions

F3 ALU Instructions*	
Cacheable (one-cycle)	Not Cacheable (two-cycle)†
aD = aS + aT	aD = aS**h** + Imm
aD = aS – aT	aD = aS**h** – Imm
aD = aS & aT	aD = aS**h** & Imm
aD = aS | aT	aD = aS**h** | Imm
aD = aS ^ aT	aD = aS**h** ^ Imm
aS – aT	aS**h** – Imm
aS & aT	aS**h** & Imm
aD = aS + p	aD = aS**l** + Imm
aD = aS – p	aD = aS**l** – Imm
aD = aS & p	aD = aS**l** & Imm
aD = aS | p	aD = aS**l** | Imm
aD = aS ^ p	aD = aS**l** ^ Imm
aS – p	aS**l** – Imm
aS & p	aS**l** & Imm

* If pc points to external memory, add programmed wait-states.

† The **h** and **l** are required notation in these instructions.

Note: The F3 ALU instructions that do not have a destination accumulator are used to set flags for conditional operations, i.e., bit test operations.

F4 BMU Instructions

The bit manipulation unit in the DSP1610 provides a set of efficient bit manipulation operations on accumulators. It contains four auxiliary registers: ar0—ar3 (arN, N = 0, 1, 2, 3), two alternate accumulators (aa0—aa1), which can be shuffled with the working set, and four flags (oddp, evenp, mns1, and nmns1). The flags are testable by conditional instructions and can be read and written via bits 4 to 7 of the alf register. The BMU also sets the LMI, LEQ, LLV, and LMV flags in the psw register:

LMI = 1 if negative (i.e., bit 35 = 1)
LEQ = 1 if zero (i.e., bits 35—0 are 0)
LMV = 1 if bits 31—35 are not the same (32-bit overflow)
LLV = 1 if (a) 36-bit overflow, or if (b) illegal shift on field width/offset condition

The BMU instructions and cycle times follow. (If pc points to external memory, add programmed wait-states.) All BMU instructions require 1 word of program memory unless otherwise noted. Please refer to the *DSP1610 Digital Signal Processor Information Manual* for further discussion of the BMU instructions.

Software Architecture (continued)

- **Barrel Shifter**

aD = aS >> Imm arithmetic right shift by immediate (36-bit, sign filled in); 2-cycle, 2-word.
aD = aS >> arN arithmetic right shift by arN (36-bit, sign filled in); 1-cycle.
aD = a$\overline{S}$ >> aS arithmetic right shift by aS (36-bit, sign filled in); 2-cycle.

aD = aS >>> Imm logical right shift by immediate (32-bit shift, 0s filled in); 2-cycle, 2-word.
aD = aS >>> arN logical right shift by arN (32-bit shift, 0s filled in); 1-cycle.
aD = a$\overline{S}$ >>> aS logical right shift by aS (32-bit shift, 0s filled in); 2-cycle.

aD = aS << Imm arithmetic left shift* by immediate (36-bit shift, 0s filled in); 2-cycle, 2-word.
aD = aS << arN arithmetic left shift* by arN (36-bit shift, 0s filled in); 1-cycle.
aD = a$\overline{S}$ << aS arithmetic left shift* by aS (36-bit shift, 0s filled in); 2-cycle.

aD = aS <<< Imm logical left shift by immediate (36-bit shift, 0s filled in); 2-cycle, 2-word.
aD = aS <<< arN logical left shift by arN (36-bit shift, 0s filled in); 1-cycle.
aD = a$\overline{S}$ <<< aS logical left shift by aS (36-bit shift, 0s filled in); 2-cycle.

* Not the same as the special function arithmetic left shift. Here, the guard bits in the destination accumulator are shifted into, not sign-extended.

- **Normalization and Exponent Computation**

aD = exp(aS) Detect the number of redundant sign bits in accumulator; 1-cycle.
aD = norm(aS, arN) Normalize aS with respect to bit 31, with exponent in arN; 1-cycle.

- **Bit Field Extraction and Insertion**

aD = extracts(aS, Imm) extraction with sign extension, field specified as immediate; 2-cycle, 2-word.
aD = extracts(aS, arN) extraction with sign extension, field specified in arN; 1-cycle.

aD = extractz(aS, Imm) extraction with zero extension, field specified as immediate; 2-cycle, 2-word.
aD = extractz(aS, arN) extraction with zero extension, field specified in arN; 1-cycle.

aD = insert(aS, Imm) bit field insertion, field specified as immediate; 2-cycle, 2-word.
aD = insert(aS, arN) bit field insertion, field specified in arN; 2-cycle.

Note: The bit field to be inserted or extracted is specified as follows. The width (in bits) of the field is the upper byte of the operand (immediate or arN), and the offset from the LSB is in the lower byte.

- **Alternate Accumulator Set**

aD=aS:aa0 shuffle accumulators with alternate accumulator 0 (aa0); 1-cycle.
aD=aS:aa1 shuffle accumulators with alternate accumulator 1 (aa1); 1-cycle.

Note: The alternate accumulator gets what was in aS. aD gets what was in the alternate accumulator.

Table 20. Replacement Table for F3 ALU Instructions and F4 BMU Instructions

Replace	Value	Meaning
aD, aT, aS	a0 or a1	One of the two accumulators
Imm	immediate	16-bit data, sign, zero, or one-extended as appropriate
arN	ar[3:0]	One of the auxiliary BMU registers

Software Architecture (continued)

Cache Instructions

Cache instructions require one word of program memory. The do instruction executes in one instruction cycle; the redo instruction executes in two instruction cycles. (If pc points to external memory, add programmed wait-states.) Control instructions and long immediate values cannot be stored inside the cache. The instruction formats are as follows:

```
do K {
    instr1
    instr2
    .
    .
    .
    instrNI
    }

redo K
```

Table 21. Replacement Table for Cache Instructions

Replace	Instruction Encoding	Meaning
K	cloop*	Number of times the instructions are to be executed taken from bits zero through six of the cloop register
	1 to 127	Number of times the instructions are to be executed, encoded in instruction
NI	1 to 15	1 to 15 instructions can be included

* The assembly language statement: do cloop (or redo cloop) is used to specify that the number of iterations is to be taken from the cloop register. K is encoded as 0 in the instruction encoding to select cloop.

When the cache is used to execute a block of instructions, the cycle timing of the instructions is as follows:

1. In the first pass, the instructions are fetched from program memory and the cycle times are the normal out-of-cache values except the last instruction in the block of NI instructions. This instruction executes in two cycles.
2. During pass 2 through pass K–1, each instruction is fetched from cache and the in-cache timing applies.
3. During the last (Kth) pass, the block of instructions is fetched from cache and the in-cache timing applies, except that the timing of the last instruction is the same as if it were out-of-cache.
4. If any of the instructions access external memory, programmed wait-states must be added to the cycle counts.

The **redo** instruction treats the instructions currently in the cache memory as another loop to be executed K times. Using the redo instruction, instructions are re-executed from the cache without reloading the cache.

The number of iterations, K, for a do or redo can be set at run time by first moving the number of iterations into the cloop register (7 bits unsigned), and then issuing the do cloop or redo cloop. At the completion of the loop, the value of cloop is decremented to 0; hence, cloop needs to be written before each **do cloop** or **redo cloop**.

Software Architecture (continued)

Data Move Instructions

Data move instructions normally execute in two instruction cycles. (If pc or rM point to external memory, any programmed wait-states must be added. In addition, if pc and rM point to the same bank of DPRAM, then one cycle must be added.) Immediate data move instructions require two words of program memory; all other data move instructions require only one word. The only exception to these statements is a special case immediate load (short immediate) instruction. If a YAAU register is loaded with a 9-bit short immediate value, the instruction requires only one word of memory and executes in one instruction cycle. All data move instructions except those doing long immediate loads can be executed from within the cache. A direct data addressing mode has been added to the DSP1600 core. The data move instructions are given below:

R = N
aT[l] = R
SR = IM
Y = R
R = Y
Z : R
R = aS[l]
DR = *(O)
*(O) = DR

Table 22. Replacement Table for Data Move Instructions

Replace	Value‡‡	Meaning†‡
R	Any of the registers in Table 51§	
DR	r[0:3], a0[l], a1[l], y[l], p[l], x, pt, pr, psw	Subset of registers accessible with direct addressing
aS, aT	a0, a1	High half of accumulator
Y	*rM, *rM++, *rM--, *rM++j	Same as in multiply/ALU instructions
Z	*rMzp, *rMpz, *rMm2, *rMjk	Same as in multiply/ALU instructions
N	16-bit value	Long immediate data
IM	9-bit value	Short immediate data for YAAU registers
O	5-bit value from instruction 11-bit value in base register	Value in bits 15—5 of ybase register form the eleven most significant bits of the base address. The 5-bit offset is concatenated to this to form a 16-bit address
SR	r[0:3], rb, re, j, k	Subset of registers for short immediate

† When signed registers less than 16 bits wide (c0, c1, c2) are read, their contents are sign-extended to 16 bits. When unsigned registers less than 16 bits wide are read, their contents are zero-extended to 16 bits.
‡ Loading an accumulator with a data move instruction does not affect the flags.
§ Register codes for a0, a0l, a1, and a1l have been added to the DSP1600 core to allow data moves to and from a0, a0l, a1, and a1l.
‡‡ sioc, sioc2, tdms, and srta registers are not readable.

Software Architecture (continued)

5.2 Register Settings

Tables 23 through 40 describe the programmable registers of the DSP1610 device.

Note that the following abbreviations are used in the tables:

x = don't care
R = read only
W = read/write

Table 23. Serial I/O-Control (sioc) Registers

Bit	9	8	7	6	5	4	3	2	1	0
Field	LD	CLK		MSB	OLD	ILD	OCK	ICK	OLEN	ILEN

Field	Value	Description
LD	0 1	Active ILD/OLD = ICK / 16, active SYNC = ICK / [128/256*] Active ILD/OLD = OCK /16, active SYNC = OCK / [128/256*]
CLK	00 01 10 11	Active clock = CKI / 4 Active clock = CKI / 12 Active clock = CKI / 16 Active clock = CKI / 20
MSB	0 1	LSB first MSB first
OLD	0 1	OLD is an input (passive mode) OLD is an output (active mode)
ILD	0 1	ILD is an input (passive mode) ILD is an output (active mode)
OCK	0 1	OCK is an input (passive mode) OCK is an output (active mode)
ICK	0 1	ICK is an input (passive mode) ICK is an output (active mode)
OLEN	0 1	16-bit output 8-bit output
ILEN	0 1	16-bit input 8-bit input

* See tdms register, SYNC field.

sioc2

Bit	9	8	7	6	5	4	3	2	1	0
Field	LD2	CLK2		MSB2	OLD2	ILD2	OCK2	ICK2	OLEN2	ILEN2

Software Architecture (continued)

Table 24. Time-Division Multiplex Slot (tdms) Register

Bit	9	8	7	6	5	4	3	2	1	0
Field	SYNCSP	MODE	TRANSMIT SLOT							SYNC

Field	Value	Description

SYNCSP	0
	1
		SYNC = ICK/OCK† / 128*
		SYNC = ICK/OCK† / 256

MODE	0 1	Multiprocessor mode off, DOEN is an input (passive mode) Multiprocessor mode on, DOEN is an output (active mode)
TRANSMIT SLOT	1xxxxxx	Transmit slot 7
	x1xxxxx	Transmit slot 6
	xx1xxxx	Transmit slot 5
	xxx1xxx	Transmit slot 4
	xxxx1xx	Transmit slot 3
	xxxxx1x	Transmit slot 2
	xxxxxx1	Transmit slot 1
SYNC	1 0	Transmit slot 0, SYNC is an output (active mode) SYNC is an input (passive mode)

* Select this mode when in multiprocessor mode.

† See sioc register, LD field.

Table 25. Serial Receive/Transmit Address (srta) Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	RECEIVE ADDRESS								TRANSMIT ADDRESS							

Field	Value	Description
RECEIVE ADDRESS	1xxxxxxx	Receive address 7
	x1xxxxxx	Receive address 6
	xx1xxxxx	Receive address 5
	xxx1xxxx	Receive address 4
	xxxx1xxx	Receive address 3
	xxxxx1xx	Receive address 2
	xxxxxx1x	Receive address 1
	xxxxxxx1	Receive address 0
TRANSMIT ADDRESS	1xxxxxxx	Transmit address 7
	x1xxxxxx	Transmit address 6
	xx1xxxxx	Transmit address 5
	xxx1xxxx	Transmit address 4
	xxxx1xxx	Transmit address 3
	xxxxx1xx	Transmit address 2
	xxxxxx1x	Transmit address 1
	xxxxxxx1	Transmit address 0

Software Architecture (continued)

Table 26. saddx Multiprocessor Protocol Register

Bit Field	15—8	7—0
write	X	Write Protocol Field[7:0]
read	Read Protocol Field[7:0]	0

Table 27. Processor Status Word (psw) Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	DAU Flags				X	X	a1[V]	a1[35:32]				a0[V]	a0[35:32]			

Field	Value	Description
DAU Flags*	Wxxx	LMI — logical minus when set (bit 35 = 1)
	xWxx	LEQ — logical equal when set (bit 35—0 = 0)
	xxWx	LLV — logical overflow when set
	xxxW	LMV — mathematical overflow when set
a1[V]	W	Accumulator 1 (a1) math overflow when set
a1[35:32]	Wxxx	Accumulator 1 (a1) bit 35
	xWxx	Accumulator 1 (a1) bit 34
	xxWx	Accumulator 1 (a1) bit 33
	xxxW	Accumulator 1 (a1) bit 32
a0[V]	W	Accumulator 0 (a0) math overflow when set
a0[35:32]	Wxxx	Accumulator 0 (a0) bit 35
	xWxx	Accumulator 0 (a0) bit 34
	xxWx	Accumulator 0 (a0) bit 33
	xxxW	Accumulator 0 (a0) bit 32

* The DAU flags can be set by either BMU or DAU operations.

Table 28. Arithmetic Unit Control (auc) Register

Bit	6	5	4	3	2	1	0
Field	CLR			SAT		ALIGN	

Field	Value	Description
CLR	1xx	Clearing yl is disabled (enabled when 0)
	x1x	Clearing a1l is disabled (enabled when 0)
	xx1	Clearing a0l is disabled (enabled when 0)
SAT	1x x1	a1 saturation on overflow is disabled (enabled when 0) a0 saturation on overflow is disabled (enabled when 0)
ALIGN	00	a0, a1 ← p
	01	a0, a1 ← p/4
	10	a0, a1 ← p*4
	11	Reserved

Note: The auc register is not cleared by reset and must be set before using the DAU.
The auc is less than 16 bits and is zero-extended when read.

Software Architecture (continued)

Table 29. Parallel I/O Control (pioc) Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	IBF	STROBE		PODS	PIDS	S/C	INTERRUPTS					STATUS				

Field	Value	Description
IBF	R	IBF interrupt status bit (same as bit 4)
STROBE	 00 01 10 11	Strobe width of: PODS PIDS T* T 2T 2T 3T 3T 4T 4T
PODS	0 1	PODS is an input (passive mode) PODS is an output (active mode)
PIDS	0 1	PIDS is an input (passive mode) PIDS is an output (active mode)
S/C	0 1	Not status/control mode Status/control mode
INTERRUPTS	1xxxx	IBF interrupt enabled (disabled when 0)†
	x1xxx	OBE interrupt enabled (disabled when 0)†
	xx1xx	PIDS interrupt enabled (disabled when 0)
	xxx1x	PODS interrupt enabled (disabled when 0)
	xxxx1	INT0 interrupt enabled (disabled when 0)
STATUS	Rxxxx	IBF status bit
	xRxxx	OBE status bit
	xxRxx	PIDS status bit
	xxxRx	PODS status bit
	xxxxR	INT0 status bit

* T = 2 x t1. See timing diagrams.

† The enable and status bits in the pioc affect only the SIO, not the SIO2.

In active mode, reading or writing pdx[7:0] produces an encoded channel number from 111 to 000 to appear on pins PSEL[2:0]. Note that PSEL0 then provides the same information as the DSP16/A PSEL pin, for compatibility. In passive modes, the PSEL pins take on other functions shown in Table 7.

Software Architecture (continued)

Table 30. timerc Register

Bit	15—7	6	5	4	3—0
Field	Reserved	DISABLE	RELOAD	T0EN	PRESCALE

Field	Value	Description
DISABLE	0 1	Timer enabled. Timer clocks off for power saving. The period register and timer0 are not reset.
RELOAD	0 1	Timer stops after counting down to 0. Timer automatically reloads and repeats indefinitely.
T0EN	0 1	Timer holds current count. Timer counts down to 0.
PRESCALE		See table below.

PRESCALE Field

PRESCALE	Frequency	Frequency at CKI = 60 MHz
0000	CKI/4	15 MHz
0001	CKI/8	7.5 MHz
0010	CKI/16	3.25 MHz
0011	CKI/32	1.875 MHz
0100	CKI/64	937.5 kHz
0101	CKI/128	468.75 kHz
0110	CKI/256	234.38 kHz
0111	CKI/512	117.19 kHz
1000	CKI/1024	58.59 kHz
1001	CKI/2048	29.30 kHz
1010	CKI/4096	14.65 kHz
1011	CKI/8192	7.32 kHz
1100	CKI/16384	3.66 kHz
1101	CKI/32768	1.83 kHz
1110	CKI/65536	915.5 kHz
1111	CKI/131072	457.8 Hz

Software Architecture (continued)

Table 31. sbit Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	DIREC[7:0]								VALUE[7:0]							

Field	Value	Description
DIREC	1xxxxxxxx	IOBIT7 is an output (input when 0)
	x1xxxxxxx	IOBIT6 is an output (input when 0)
	xx1xxxxxx	IOBIT5 is an output (input when 0)
	xxx1xxxxx	IOBIT4 is an output (input when 0)
	xxxx1xxx	IOBIT3 is an output (input when 0)
	xxxxx1xx	IOBIT2 is an output (input when 0)
	xxxxxx1x	IOBIT1 is an output (input when 0)
	xxxxxxx1	IOBIT0 is an output (input when 0)
VALUE	Rxxxxxxx	Reads the current value of IOBIT7
	xRxxxxxx	Reads the current value of IOBIT6
	xxRxxxxx	Reads the current value of IOBIT5
	xxxRxxxx	Reads the current value of IOBIT4
	xxxxRxxx	Reads the current value of IOBIT3
	xxxxxRxx	Reads the current value of IOBIT2
	xxxxxxRx	Reads the current value of IOBIT1
	xxxxxxxR	Reads the current value of IOBIT0

Table 32. cbit Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	MODE/MASK[7:0]								DATA/PAT[7:0]							

DIREC[n]*	MODE/ MASK[n]	DATA/ PAT[n]	Action
1 (Output)	0	0	Clear
1 (Output)	0	1	Set
1 (Output)	1	0	No Change
1 (Output)	1	1	Toggle
0 (Input)	0	0	No Test
0 (Input)	0	1	No Test
0 (Input)	1	0	Test for Zero
0 (Input)	1	1	Test for One

* 0 ≤ n ≤ 7

Software Architecture (continued)

Table 33. Interrupt Control (inc) Register

Bit	15	14—12	11	10	9	8	7—4	3	2	1	0
Field	JINT*	Reserved	EMUXBOTH	OBE2	IBF2	TIME	INT[0:3]	PIDS	PODS	OBE	IBF

* JINT is a JTAG interrupt and is controlled by the HDS. It may be made unmaskable by the AT&T development system tools.

Encoding: A 0 disables an interrupt; a 1 enables an interrupt.

Table 34. Interrupt Status (ins) Register

Bit	15	14—12	11	10	9	8	7—4	3	2	1	0
Field	JINT	Reserved	EMUXBOTH	OBE2	IBF2	TIME	INT[0:3]	PIDS	PODS	OBE	IBF

Encoding: A 0 indicates no interrupt. A 1 indicates an interrupt has been recognized and is pending or being serviced. If a 1 is written to bits 8, 7, 6, 5, 4, or 11 of ins, the corresponding interrupt is cleared in ins.

Table 35. alf Register

Bit	15	14	13—0
Field	AWAIT	LOWPR	FLAGS

Field	Value	Action
AWAIT	1 0	Power-saving standby mode enabled. Normal operation.
LOWPR	1 0	The internal DPRAM is addressed beginning at 0x3000 in X space. The internal DPRAM is addressed beginning at 0x0000 in X space.
FLAGS		See Table 36 below.

Table 36. alf Flags

Bit	Flag	Use
13—8	Reserved	—
7	nmns1	NOT-MINUS-ONE from BMU
6	mns1	MINUS-ONE from BMU
5	evenp	EVEN PARITY from BMU
4	oddp	ODD PARITY from BMU
3	somef	SOME FALSE from BIO
2	somet	SOME TRUE from BIO
1	allf	ALL FALSE from BIO
0	allt	ALL TRUE from BIO

Table 37. mwait Register

Bit	15—12	11—8	7—4	3—0
Field	EROM[3:0]	ERAMHI[3:0]	IO[3:0]	ERAMLO[3:0]

If the EXM pin is high and INT1 low upon reset, the mwait register is initialized to all 1s (15 wait-states for all external memory). Otherwise, the mwait register is initialized to all 0s (0 wait-states) upon reset.

Software Architecture (continued)

Table 38. ioc Register

Bit	15—14	13	12	11—10	9	8	7—4	3—0
Field	RESERVED	ESIO2	SIOLBC	CKO[1:0]	DSELH	PIOLBC	DDSEL[3:0]	DENB[3:0]

Table 39. ioc Fields

ioc Field	Effect
ESIO2	If 1, enables SIO2 and disables PIO from pins
SIOLBC	If 1, signals DO and DO2 looped back to DI and DI2
CKO[1:0]	CKO configuration (See Table 40.)
DSELH	If 1, pins DSEL[3:0] active high
PIOLBC	If 1, signals PB[15:0] and PODS to PIDS internally looped back
DDSEL3	If 1, delay pin DSEL3
DDSEL2	If 1, delay pin DSEL2
DDSEL1	If 1, delay pin DSEL1
DDSEL0	If 1, delay pin DSEL0
DENB3	If 1, delay pin EROM
DENB2	If 1, delay pin ERAMHI
DENB1	If 1, delay pin IO
DENB0	If 1, delay pin ERAMLO

Table 40. CKO Options

CKO1	CKO0	CKO OUTPUT	Description
0	0	CKI/2	Free-running clock
0	1	CKI/(2[1+w])	Wait-stated clock
1	0	1	Held high
1	1	0	Held low

Note: The phase of CKI/2 is synchronized by the rising edge of RSTB.

Software Architecture (continued)

5.3 Instruction Set Formats

This section defines the hardware-level encoding of the DSP1610 device instructions.

Multiply/ALU Instructions

Format 1: Multiply/ALU Read/Write Group:

Field	T					D	S	F1				X	Y			
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Format 1a: Multiply/ALU Read/Write Group:

Field	T					$\overline{aT}$	S	F1				X	Y			
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Format 2: Multiply/ALU Read/Write Group:

Field	T					D	S	F1				X	Y			
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Format 2a: Multiply/ALU Read/Write Group:

Field	T					$\overline{aT}$	S	F1				X	Y			
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Special Function Instructions

Format 3: F2 ALU Special Functions:

Field	T					D	S	F2				CON				
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Format 3a: F3 ALU Operations:

Field	T					D	S	F3				SRC2		aT	0	1
	Immediate Operand (N)															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Format 3b: BMU Operations:

Field	T					D	S	F4[3—1]			0	F4[0]	AR[3—0]			
	Immediate Operand (N)															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Software Architecture (continued)

Control Instructions

Format 4: Branch Direct Group:

Field	T				JA											
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Format 5: Branch Indirect Group:

Field	T					B			Reserved							0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Format 6: Conditional Branch Qualifier/Software Interrupt (icall):

Note that a branch instruction immediately follows except for a software interrupt (icall).

Field	T					SI	Reserved				CON					
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Data Move Instructions

Format 7: Data Move Group:

Field	T					$\overline{a}$T	R						Y/Z			
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Format 8: Data Move (immediate operand — 2 words)

Field	T					D	R						Reserved			
	Immediate Operand (N)															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Format 9: Short Immediate Group:

Field	T					I		Short Immediate Operand (IM)								
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Format 9a: Direct Addressing:

Field	T					R/W	DR[3—0]				1	OFFSET				
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Cache Instructions

Format 10: Do-Redo

Field	T					NI				K						
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Software Architecture (continued)

Field Descriptions

Table 41. T Field

Specifies the type of instruction.

T	Operation		Format
0000x	goto JA		4
00010	Short imm j, k, rb, re		9
00011	Short imm r0, r1, r2, r3		9
00100	Y=a1[l]	F1	1
00101	Z : aT[l]	F1	2a
00110	Y	F1	1
00111	aT[l]=Y	F1	1a
01000	Bit0=0, aT=R		7
01000	Bit0=1, aTl=R*		7
01001	Bit10=0, R=a0		7
01001	Bit10=1, R=a0l*		7
01010	R=N		8
01011	Bit10=0, R=a1		7
01011	Bit10=1, R=a1l*		7
01100	Y=R		7
01101	Z : R		7
01110	Do, redo		10
01111	R=Y		7
1000x	call JA		4
10010	ifc CON	F2	3
10011	if CON	F2	3
10100	Y=y[l]	F1	1
10101	Z : y[l]	F1	2
10110	x=Y	F1	1
10111	y[l]=Y	F1	1
11000	Bit0=0, Branch indirect		5
11000	Bit0=1, F3 ALU*		3a
11001	y=a0 x=X	F1	1
11010	Conditional branch qualifier		6
11011	y=a1 x=X	F1	1
11100	Y=a0[l]	F1	1
11101	Z : y x=X	F1	2
11110	Bit5=0, F4 ALU (BMU)*		3b
11110	Bit5=1, Direct Addressing*		9a
11111	y=Y x=X	F1	1

* These instructions are new to the DSP1600 core. The rest of the instructions in the table are available in DSP16(A).

Table 42. D Field

Specifies a destination accumulator.

D	Register
0	Accumulator 0
1	Accumulator 1

Table 43. $\overline{\mathrm{aT}}$ Field

Specifies transfer accumulator.

$\overline{\mathrm{aT}}$	Register
0	Accumulator 1
1	Accumulator 0

Table 44. S Field

Specifies a source accumulator.

S	Register
0	Accumulator 0
1	Accumulator 1

Table 45. F1 Field

Specifies the multiply/ALU function.

F1	Operation	
0000	aD = p	p = x * y
0001	aD = aS+p	p = x * y
0010		p = x * y
0011	aD = aS − p	p = x * y
0100	aD = p	
0101	aD = aS + p	
0110	NOP	
0111	aD = aS − p	
1000	aD = aS|y	
1001	aD = aS^y	
1010	aS & y	
1011	aS − y	
1100	aD = y	
1101	aD = aS+y	
1110	aD = aS&y	
1111	aD = aS − y	

Table 46. X Field

Specifies the addressing of ROM data in two-operand multiply/ALU instructions. Specifies the high or low half of an accumulator or the y register in one-operand multiply/ALU instructions.

X	Operation
Two-Operand Multiply/ALU	
0	*pt++
1	*pt++i
One-Operand Multiply/ALU	
0	aTl, yl
1	aTh, yh

Software Architecture (continued)

Table 47. Y Field

Specifies the form of register indirect addressing with postmodification.

Y	Operation
0000	*r0
0001	*r0++
0010	*r0--
0011	*r0++j
0100	*r1
0101	*r1++
0110	*r1--
0111	*r1++j
1000	*r2
1001	*r2++
1010	*r2--
1011	*r2++j
1100	*r3
1101	*r3++
1110	*r3--
1111	*r3++j

Table 48. Z Field

Specifies the form of register indirect compound addressing with postmodification.

Z	Operation
0000	*r0zp
0001	*r0pz
0010	*r0m2
0011	*r0jk
0100	*r1zp
0101	*r1pz
0110	*r1m2
0111	*r1jk
1000	*r2zp
1001	*r2pz
1010	*r2m2
1011	*r2jk
1100	*r3zp
1101	*r3pz
1110	*r3m2
1111	*r3jk

Table 49. F2 Field

Specifies the special function to be performed.

F2	Operation
0000	aD = aS >> 1
0001	aD = aS << 1
0010	aD = aS >> 4
0011	aD = aS << 4
0100	aD = aS >> 8
0101	aD = aS << 8
0110	aD = aS >> 16
0111	aD = aS << 16
1000	aD = p
1001	aDh = aSh + 1
1010	aD = ~aS*
1011	aD = rnd(aS)
1100	aD = y
1101	aD = aS + 1
1110	aD = aS
1111	aD = – aS

* This operation is new to the DSP1600 core. The rest of the operations in the table are available in the DSP16(A).

Table 50. CON Field

Specifies the condition for special functions and conditional control instructions.

CON	Condition	CON	Condition
00000	mi	01110	true
00001	pl	01111	false
00010	eq	10000	gt
00011	ne	10001	le
00100	lvs	10010	allt*
00101	lvc	10011	allf*
00110	mvs	10100	somet*
00111	mvc	10101	somef*
01000	heads	10110	oddp*
01001	tails	10111	evenp*
01010	c0ge	11000	mns1*
01011	c0lt	11001	nmns1*
01100	c1ge	11010	npint*
01101	c1lt	11011	njint*
Other codes	Reserved	—	—

* These condition codes are new to DSP1610. All other codes in the table are available in DSP16(A).

Software Architecture (continued)

Table 51. R Field

Specifies the register for data move instructions.

R	Register	R	Register
000000	r0	100000	inc*
000001	r1	100001	ins*
000010	r2	100010	sdx2†
000011	r3	100011	saddx†
000100	j	100100	cloop*
000101	k	100101	mwait*
000110	rb	100110	Reserved
000111	re	100111	sioc2†
001000	pt	101000	cbit†
001001	pr	101001	sbit†
001010	pi	101010	ioc†
001011	i	101011	jtag†
001100	p§	101100	pdx4*
001101	pl§	101101	pdx5*
001110	pdx2*	101110	pdx6*
001111	pdx3*	101111	pdx7*
010000	x	110000	a0*
010001	y	110001	a0l*
010010	yl	110010	a1*
010011	auc	110011	a1l*
010100	psw	110100	timerc†
010101	c0	110101	timer0†
010110	c1	110110	Reserved
010111	c2	110111	Reserved
011000	sioc‡	111000	Reserved
011001	srta‡	111001	Reserved
011010	sdx‡	111010	ar0†
011011	tdms‡	111011	ar1†
011100	pioc	111100	ar2†
011101	pdx0	111101	ar3†
011110	pdx1	111110	Reserved
011111	ybase*	111111	alf*†

* These registers are new in the DSP1600 core.

† These registers are new in the DSP1610.

‡ These registers are located outside the DSP1600 core.

§ These registers were not in DSP16.

Note: Registers sioc, srta, and tdms are not readable.

Table 52. B Field

Specifies the type of branch instruction (except software interrupt).

B	Operation
000	return
001	ireturn
010	goto pt
011	call pt
1xx	Reserved

Table 53. DR Field

DR	Register
0000	r0
0001	r1
0010	r2
0011	r3
0100	a0
0101	a0l
0110	a1
0111	a1l
1000	y
1001	yl
1010	p
1011	pl
1100	x
1101	pt
1110	pr
1111	psw

Table 54. I Field

Specifies a register for short immediate data move instructions.

I	Register
00	r0/j
01	r1/k
10	r2/rb
11	r3/re

Table 55. SI Field

Specifies when the conditional branch qualifier instruction should be interpreted as a software interrupt instruction.

SI	Operation
0	Not a software interrupt
1	Software interrupt

Software Architecture (continued)

NI Field. Number of instructions to be loaded into the cache. Zero implies redo operation.

K Field. Number of times the NI instructions in cache are to be executed. Zero specifies use of value in cloop register.

JA Field. 12-bit jump address.

Table 56. F3 Field

Specifies the operation in an F3 ALU instruction.

F3	Operation		
1000	aD = aS[h, l]	|	{aT, Imm, p}
1001	aD = aS[h, l]	^	{aT, Imm, p}
1010	aS[h, l]	&	{aT, Imm, p}
1011	aS[h, l]	–	{aT, Imm, p}
1101	aD = aS[h, l]	+	{aT, Imm, p}
1110	aD = aS[h, l]	&	{aT, Imm, p}
1111	aD = aS[h, l]	–	{aT, Imm, p}

Table 57. SRC2 Field

Specifies operands in an F3 ALU instruction.

SRC2	Operands
00	aSl, Imm
10	aSh, Imm
01	aS, aT
11	aS, p

Table 58. BMU Encodings

F4	AR	Operation
0000	00xx	aD = aS >> arn
0001	00xx	aD = aS << arn
0000	10xx	aD = aS >>> arn
0001	10xx	aD = aS <<< arn
1000	0000	aD = a$\overline{S}$ >> aS
1001	0000	aD = a$\overline{S}$ << aS
1000	1000	aD = a$\overline{S}$ >>> aS
1001	1000	aD = a$\overline{S}$ <<< aS
1100	0000	aD = aS >> Imm
1101	0000	aD = aS << Imm
1100	1000	aD = aS >>> Imm
1101	1000	aD = aS <<< Imm
0000	1100	aD = exp(aS)
0001	11xx	aD = norm(aS, arn)
1110	0000	aD = extracts(aS, Imm)
0010	00xx	aD = extracts(aS, arn)
1110	0100	aD = extractz(aS, Imm)
0010	01xx	aD = extractz(aS, arn)
1110	1000	aD = insert(aS, Imm)
1010	10xx	aD = insert(aS, arn)
0111	0000	aD = aS:aa0
0111	0001	aD = aS:aa1

Note: xx encodes the auxilliary register to be used. 00(ar0), 01(ar1), 10(ar2), or 11(ar3).

6.0 Signal Descriptions

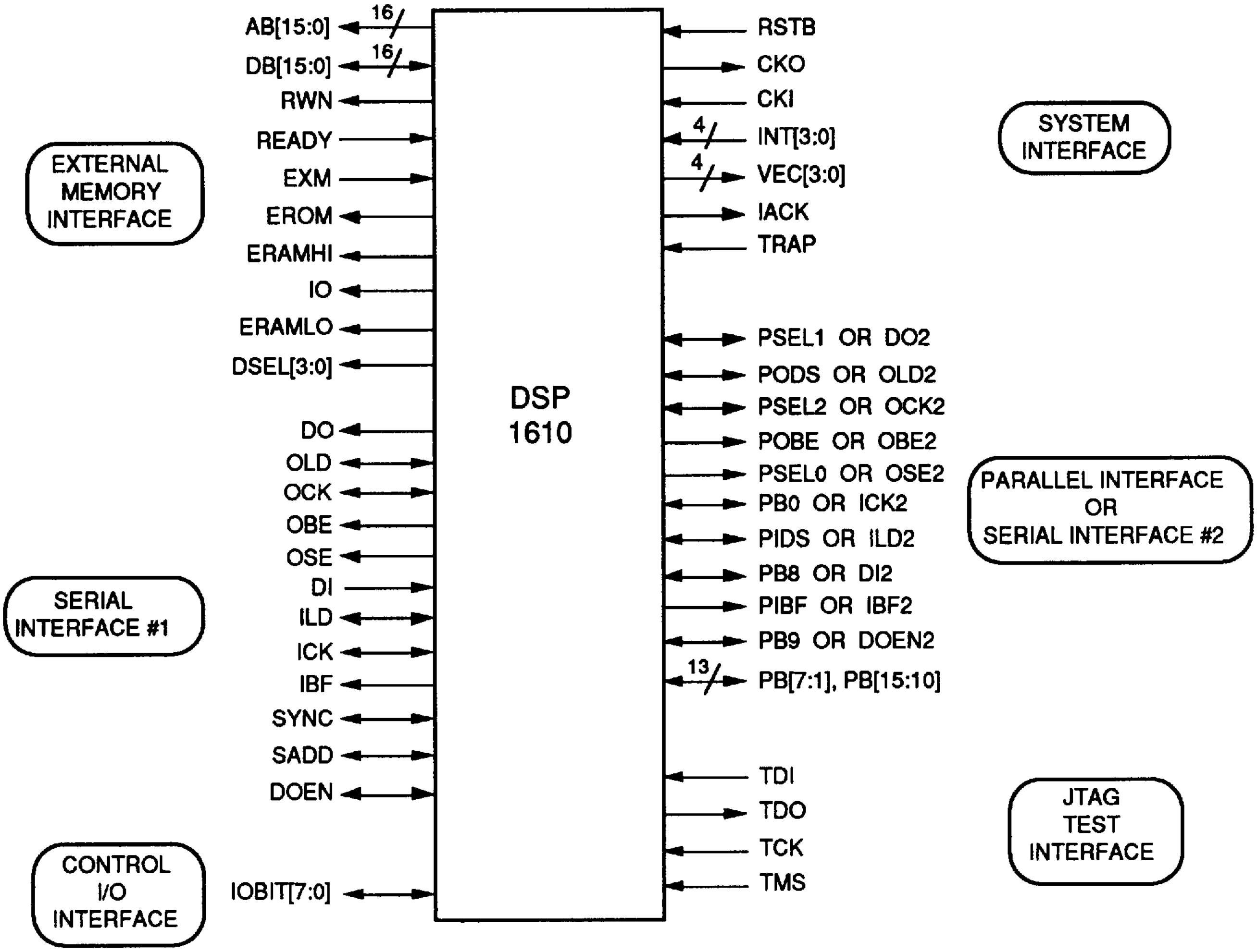


Figure 5. DSP1610 Pinout by Interface

Signal Descriptions (continued)

Figure 5 shows the pinout for the DSP1610. The signals can be separated into six interfaces as shown. These interfaces and the signals that comprise them are described below.

6.1 System Interface

The system interface consists of the clock, interrupt, and reset signals for the processor.

RSTB

Reset. Negative assertion. A high-to-low transition causes the processor to enter the reset state. The sioc, sioc2, tdms, timerc (lower byte), timer0, sbit (upper byte), cbit (lower byte), inc, alf (upper two bits, AWAIT and LOWPR), ioc, rb, and re registers are cleared. The pioc and ins registers are cleared except for the OBE and OBE2 bits which are set. The mwait register is initialized to all 0s (zero wait-states) unless the EXM pin is high and the INT1 pin is low. In that case, the mwait register is initialized to all 1s (15 wait-states). Reset clears external signals IACK, IBF, and IBF2, while it sets OSE and OSE2. IOBIT[7:0] are initialized as inputs. If any of the IOBIT pins are switched to outputs (by writing the sbit register), their initial value will be logic 0. The DAU condition flags and the auc register are not affected by reset. See Table 59, Register States after Reset.

Upon negation of the RSTB signal, the processor begins execution at location 0x0000 in the active memory map. See Section 4.4, Memory Maps and Wait-States.

CKI

Clock In. Input clock at twice the frequency of internal operations.

CKO

Clock Out. Buffered output clock with options programmable via the ioc register. (See Table 38.) The selectable CKO options (see Table 40) are:

- Free-running output clock at half the frequency of CKI
- A wait-stated clock during external memory accesses
- A logic 0
- A logic 1

INT[0:3]

Processor Interrupts 0 through 3. Positive assertion. Hardware interrupts to the DSP1610. Each is enabled via the inc register and, when enabled, each cause the processor to jump to the memory location described in Table 4. INT0 can be enabled in the pioc for DSP16/A compatibility. INT1 is used in conjunction with the EXM signal to select the desired reset initialization of the mwait register (see footnote, Table 37). **When both INT3 and RSTB are asserted, all output and bidirectional pins are put in a 3-state condition unless a JTAG controller has sent an overriding sequence. An exception is the TDO pin which is 3-stated only by power up or by the JTAG controller.**

VEC[3:0]

Interrupt Output Vector. These four pins indicate which interrupt is currently being serviced by the device. Table 4 shows the code associated with each interrupt condition.

IACK

Interrupt Acknowledge. Positive assertion. Interrupt Acknowledge signals when an interrupt is being serviced by the DSP1610. IACK remains asserted while in an interrupt service routine and is cleared when the ireturn instruction is executed.

TRAP

Trap signal. Positive assertion. When asserted, the processor is put into the trap condition, which normally causes a branch to the location 0x0046. The hardware development system (HDS) can configure the trap pin to cause an HDS trap which causes a branch to location 0x0003. Although normally an input, the pin can be configured as an output by the HDS. As an output, the signal is used to signal an HDS breakpoint in a multiple processor environment.

Signal Descriptions (continued)

Table 59. Register States after Reset

A • indicates this bit is unknown on power up reset and unaffected on subsequent reset. An S indicates this bit shadows the PC. P indicates the value on an input pin, i.e., the bit in the register reflects the value on the corresponding input pin.

Register	Bits 15—0	Register	Bits 15—0
r0	••••••••••••••••	pdx0	••••••••••••••••
r1	••••••••••••••••	pdx1	••••••••••••••••
r2	••••••••••••••••	ybase	••••••••••••••••
r3	••••••••••••••••	inc	0000000000000000
j	••••••••••••••••	ins	0111010000000010
k	••••••••••••••••	sdx2	••••••••••••••••
rb	0000000000000000	saddx	••••••••••••••••
re	0000000000000000	cloop	000000000•••••••
pt	••••••••••••••••	mwait	0000000000000000†
pr	••••••••••••••••	sioc2	••••••0000000000
pi	SSSSSSSSSSSSSSSS	cbit	••••••••••••••••
i	••••••••••••••••	sbit	00000000PPPPPPPP
p	••••••••••••••••	ioc	••00000000000000
p1	••••••••••••••••	jtag	••••••••••••••••
pdx2	••••••••••••••••	pdx4	••••••••••••••••
pdx3	••••••••••••••••	pdx5	••••••••••••••••
x	••••••••••••••••	pdx6	••••••••••••••••
y	••••••••••••••••	pdx7	••••••••••••••••
y1	••••••••••••••••	a0	••••••••••••••••
auc	000000000•••••••	a01	••••••••••••••••
psw	••••00••••••••••	a1	••••••••••••••••
c0	••••••••••••••••	a11	••••••••••••••••
c1	••••••••••••••••	timerc	••••••••00000000
c2	••••••••••••••••	timer0	0000000000000000
sioc	••••••0000000000	ar0	••••••••••••••••
srta	••••••••••••••••	ar1	••••••••••••••••
sdx	••••••••••••••••	ar2	••••••••••••••••
tdms	••••••0000000000	ar3	••••••••••••••••
pioc	0000000000001000	alf	00000000••••••••

† If EXM is high and INT1 is low when RSTB goes high, mwait will contain all ones instead of all zeros.

Signal Descriptions (continued)

6.2 External Memory Interface

The external memory interface is used to interface the DSP1610 to external memory and I/O devices. It supports read operations from program and data memory spaces and read/write operations from/to data memory space. The interface supports four external memory segments. Each external memory segment can have an independent number of software programmable wait-states. Four hardware addresses are decoded, and enable lines provided, to allow glueless I/O interfacing.

AB[15:0]

External Memory Address Bus. Output only. This 16-bit bus supplies the address for read or write operations to external memory or I/O.

DB[15:0]

External Memory Data Bus. This 16-bit bidirectional data bus is used for read or write operations to external memory or I/O.

RWN

Read/Write Not. Output only. When a logic 1, the pin indicates that the memory access is a read operation; when a logic 0, the memory access is a write operation.

EXM

External Memory. Input only. This signal is latched into the device on the rising edge of RSTB. The value latched determines whether the internal ROM is addressable. If EXM is low, internal ROM is addressable. If EXM is high, only external ROM is addressable in the instruction/coefficient memory map. See Table 5, Instruction/Coefficient Memory Maps. EXM chooses between MAP1 and MAP2, and between MAP3 and MAP4.

EROM

External ROM Enable signal. Output only. Negative assertion. When asserted, the signal indicates an access to external program memory. (See Table 5, Instruction/Coefficient Memory Maps). This signal's leading edge can be delayed via the ioc register (see Table 39).

ERAMHI

External RAM High Enable signal. Output only. Negative assertion. When asserted, the signal indicates an access to external data memory addresses 0x8000 through 0xFFFF. See Table 6, Data Memory Map. This signal's leading edge can be delayed via the ioc register (see Table 39).

IO

External IO Enable signal. Output only. Negative assertion. When asserted, the signal indicates an access to external data memory addresses 0x7F00 through 0x7FFF. This memory segment is intended for I/O devices. This signal's leading edge can be delayed via the ioc register (see Table 39).

ERAMLO

External RAM Low Enable signal. Output only. Negative assertion. When asserted, the signal indicates an access to external data memory addresses 0x2000 through 0x7EFF. See Table 6, Data Memory Map. This signal's leading edge can be delayed via the ioc register (see Table 39).

DSEL[3:0]

Device Select Lines 0—3. Output only. Default negative assertion (positive assertion is selectable via the ioc register, see Table 39). These signals pre-decode four specific memory addresses in the IO external memory segment. Accesses to locations 0x7F00—0x7F03 assert DSEL[3:0] respectively.

Signal Descriptions (continued)

6.3 Serial Interface #1

The serial interface pins implement a full-featured synchronous/asynchronous serial I/O channel. In addition, several pins offer a glueless TDM interface for multiprocessing communication applications. See Figure 4, Multiprocessor Communication and Connections.

DI

Data Input. Serial data latched on the rising edge of ICK, either LSB or MSB first, according to the sioc register MSB field (see Table 23, Serial I/O-Control Registers).

ICK

Input Clock. Clock for serial input data. In active mode, ICK is an output; in passive mode, ICK is an input, according to the sioc register ICK field. (see Table 23, Serial I/O-Control Registers).

ILD

Input Load. The strobe for loading the input buffer sdx[in] from the input shift register. A falling edge of ILD indicates the beginning of a serial input word. In active mode, ILD is an output. In passive mode, ILD is an input and should only be asserted once per serial input word. Passive or active mode is set via the sioc register ILD field, (see Table 23, Serial I/O-Control Registers).

IBF

Input Buffer Full. Output only. Positive assertion. IBF is asserted when the input buffer, sdx[in], is filled and negated by a read of the buffer, as in a0=sdx. IBF is also negated by asserting RSTB.

DO

Data Output. Serial data output from the output shift register, either LSB or MSB first (according to the sioc register MSB field, see Table 23). DO changes on the rising edges of OCK. DO is 3-stated when DOEN is high.

DOEN

Data Output Enable. Negative assertion. An input when not in the multiprocessor mode. DO and SADD are enabled only if DOEN is low. DOEN is bidirectional when in the multiprocessor mode (tdms register MODE field set, see Table 24).

OCK

Output Clock. Clock for serial output data. In active mode, OCK is an output; in passive mode, OCK is an input, according to the sioc register OCK field (see Table 23, Serial I/O-Control Registers).

OLD

Output Load. The strobe for loading the output shift register from the output buffer sdx[out]. A falling edge of OLD initiates the beginning of a serial output word on the next rising edge of OCK. In active mode, OLD is an output. In passive, OLD is an input and should only be asserted once per serial output word. Passive or active mode is set via the sioc register OLD field (see Table 23, Serial I/O-Control Registers).

OSE

Output Shift Register Empty. Output only. Positive assertion. The high state of OSE indicates the output shift register is currently empty. The falling edge of OSE occurs on the third rising edge of OCK after OLD falls and indicates that a serial transmission has begun.

OSE is set (makes a low-to-high transition) when the serial output has been completed (8 or 16 bits later) or RSTB is asserted. If no new word is written to the serial output buffer, OSE remains high, regardless of activity on OLD.

OBE

Output Buffer Empty. Output only. Positive Assertion. OBE is asserted when the output buffer, sdx[out], is emptied (moved to the output shift register for transmission). It is cleared with a write to the buffer, as in sdx=a0. OBE is also set by asserting RSTB.

SADD

Serial Address. Negative assertion. A 16-bit serial bit stream typically used for addressing or transmission of protocol information during multiprocessor communication between multiple DSP1610 devices. In multiprocessor mode, SADD is an output when the tdms time slot dictates a serial transmission; otherwise, it is an input. Both the source and destination DSP can be identified in the transmission. SADD is always an output when not in multiprocessor mode and can be used as a second 16-bit serial output. See the *DSP1610 Digital Signal Processor Information Manual* for additional information. SADD is 3-stated when DOEN is high.

Signal Descriptions (continued)

SYNC

Multiprocessor Synchronization. Typically used in the multiprocessor mode, a falling edge of SYNC indicates the first word of a TDM I/O stream and causes the resynchronization of the active ILD and OLD generators. SYNC is an output when the tdms register SYNC field is set (i.e., selects the master DSP and uses time slot 0 for transmit). As an input, SYNC must be tied low unless part of a TDM interface. When used as an output, SYNC = [ILD/OLD]/8 or 16, depending on the setting of the SYNCSP field of the tdms register (see Table 24). When configured as described above, SYNC can be used to generate a slow clock for SIO operation.

6.4 Parallel Interface or Serial Interface #2

The parallel I/O interface and a second serial I/O interface are multiplexed onto one set of pins. Both full interfaces cannot be used simultaneously. The selection is made by writing the ESIO2 bit in the ioc register (see Table 38). The functionality of the signals for SIO #2 correspond pin for pin with those in SIO #1, with the exception that the multiprocessing mode is not supported in SIO #2. Because multiprocessing mode is not supported, no SADD or SYNC pins are provided and DOEN2 is an input only. The pin descriptions below will only discuss the PIO pins.

PB[15:0]

Parallel I/O Data Bus. This 16-bit bidirectional bus is used to input data to, or output data from, the PIO.

PSEL[2:0]

Peripheral Select 2—0. When the PIO configuration for both PIDS and PODS are in active mode, this 3-bit field is an output. The 3-bit field can be decoded to determine which of the eight logical channels (pdx[7:0]) is active.

If the PIO is configured with either PIDS or PODS passive, PSEL2 becomes an input, which acts as a chip select. In this capacity, the chip is selected if PSEL2 is low.

When PODS is configured in active mode and PIDS in passive mode, PSEL1 and PSEL0 form a 2-bit-field selecting between four channels (pdx[7:4] alias into pdx[3:0]).

When PODS is passive, PSEL1 becomes an input. If driven high, the PIO will output the contents of the PSTAT register on PB[15:0]. If PSEL1 is low, PIO will output the contents of pdx[out]. PSEL0 is always an output.

As long as either PIDS or PODS is configured for active mode, PSEL0 indicates if the channel being written is odd or even (i.e., pdx7, 5, and 3 alias into pdx1, while pdx6, 4, and 2 alias into pdx0).
When both PIDS and PODS are in passive mode, PSEL0 becomes the logical OR of PIBF and POBE.

PIDS

Parallel Input Data Strobe. Negative assertion. In active mode, PIDS is an output. When PIDS is asserted, data can be placed onto the PB bus by an external device. PIDS is asserted by the DSP1610 during active mode read transactions.

In passive mode, PIDS is an input. The external device pulls PIDS low, places the data on the PB, and raises PIDS to latch the data into the DSP.

In both passive and active modes, the DSP latches data on the PB bus on the rising edge (low-to-high transition) of PIDS.

PODS

Parallel Output Data Strobe. Negative assertion. In active mode, PODS is an output. A falling edge of PODS indicates that data is available on the PB bus. PODS is asserted by the DSP1610 during an active mode write transaction.

In passive mode, PODS is an input. When PODS is pulled low by an external device, the DSP1610 places the contents of the parallel output register (pdx[7:0][out]) onto the PB bus.

PIBF

Parallel Input Buffer Full. Output only. Positive assertion. When PIDS is placed in active mode, this flag is cleared. It is also cleared after reset.

PIBF can only be set when PIDS is passive. In this case, it is set one cycle after the rising edge of PIDS, indicating that data has been latched into the parallel input register (pdx0—pdx7[in]). When the DSP1610 reads the contents of this register, emptying the buffer, PIBF is cleared.

POBE

Parallel Output Buffer Empty. Output only. Positive assertion. When PODS is placed in active mode, this flag is cleared. It is also cleared after reset.

POBE can only be set when PODS is passive. In this case, it is set one cycle after the rising edge of PODS, indicating that the data in the pdx parallel output register has been driven onto the PB bus. When the DSP1610 writes to this register, filling the buffer, POBE is cleared.

Signal Descriptions (continued)

6.5 Control I/O Interface

This interface is used for status and control operations provided by the bit I/O unit of the DSP1610.

IOBIT[7:0]

I/O Bits 7—0. Each of these bits can be independently configured as either an input or an output. As outputs, they can be independently set, toggled, or cleared. As inputs, they can be tested independently or in combinations for various data patterns. See Tables 31 and 32.

6.6 JTAG Test Interface

The JTAG test interface implements the IEEE 1149.1 standard test access port (TAP). In addition to supporting the complete standard, the interface has optional features that allow programs to be downloaded into the DSP via these four pins. This provides extensive test and diagnostic capability. In addition, internal circuitry allows the device to be controlled through the JTAG port to allow for on-chip in-circuit emulation. AT&T provides hardware and software tools to interface to the on-chip HDS via the JTAG port. See the *DSP1610 Digital Signal Processor Information Manual* for additional information on the JTAG test interface.

TDI

Test Data Input. JTAG serial input signal. All serial scanned data and instructions are input on this pin. This pin has an internal pull-up resistor.

TDO

Test Data Output. JTAG serial output signal. All serial-scanned data and status bits are output on this pin.

TMS

Test Mode Select. JTAG mode control signal that, when combined with TCK, controls the scan operations. This pin has an internal pull-up resistor.

TCK

Test Clock. JTAG serial shift clock. This signal clocks all data into the port through TDI, out of the port through TDO, and controls the port by latching the TMS signal inside the state-machine controller.

7.0 ROM Boot Programs

There are several ways to download an external program into on-chip RAM on the DSP1610. The 512 words of on-chip ROM contain boot programs for this purpose. The boot program to be used is selected through the PIO.

When the DSP1610 is reset (RSTB low) and EXM is low, the program begins executing from location 0 of on-chip ROM (see Table 5, Instruction/Coefficient Memory Maps). The program in ROM configures the PIO for active PIDS with maximum strobe width and proceeds to read PB[15:0]. (It actually reads the word twice because active PIO reads are latent.) The boot routine is selected by the initial value read on PB[13:10] (see Table 60). For those lines whose PB[15:14] field is marked w, t, or c, these two digits are used to configure the number of wait-states for the external reads, the PIDS strobe width, or the active clock (CKI) width, respectively. A blank entry indicates no parameter is specified for this boot routine. The specific encoding of the PB[15:14] bits and its meaning for each parameter type is shown in Table 61.

Each of the twelve boot routines will load the entire content of dual-port RAM unless interrupted by INT0. Each download program ends by setting the LOWPR bit of the alf register and starting the downloaded program from dual-port RAM at address 0x0000 of X memory space.

The 8-bit boot programs transfer one byte at a time. A 16-bit word is transferred as follows: the least significant byte first, and then the most significant byte. The four SIO routines transfer serial words with the least significant bit first. For those 8-bit transfers that are parallel, the information is transferred on the least significant byte of the bus. All active reads of the PIO use pdx0.

Downloading Self-Test Code from JTAG

In-circuit self-test of DSP1610 utilizes downloading of programs through the JTAG port. AT&T provides the hardware and software tools for downloading via this method.

Table 60. ROM Boot Programs

PB[15:14]	PB[13:10]	Boot
w	0x0	16-bit wide EROM starting at address 0x8000, 4w wait-states
w	0x1	8-bit wide EROM starting at address 0x8000, 4w wait-states
	0x2	PIO 16-bit with passive PIDS*
t	0x3	PIO 16-bit with active PIDS, strobe width t†
	0x4	PIO 8-bit with passive PIDS*
t	0x5	PIO 8-bit with active PIDS, strobe width t†
	0x6	SIO 16-bit with passive ILD,ICK
c	0x7	SIO 16-bit with active ILD,ICK, active clock c
	0x8	SIO 8-bit with passive ILD,ICK
c	0x9	SIO 8-bit with active ILD,ICK, active clock c
w	0xa	16-bit device at 0x7F00 (DSEL0), 4w wait-states
w	0xb	8-bit device at 0x7F00 (DSEL0), 4w wait-states

* For the boot routines that use PIO in passive mode, the mode is switched from active to passive after the word selecting the boot mode is read twice in active mode.

† For the boot routines that use PIO in active mode, the second word read is the first data to be downloaded.

Table 61. ROM Boot Program Parameters

PB[15:14] Value	Parameter		
	w External Memory Wait-states	t PIDS Strobe Width	c ICK Clock Width
00	0	T*	CKI ÷ 4
01	4	2T	CKI ÷ 12
10	8	3T	CKI ÷ 16
11	12	4T	CKI ÷ 20

* T is the period of CKO; CKI is in units of frequency.

8.0 Absolute Maximum Ratings

Stresses in excess of the Absolute Maximum Ratings can cause permanent damage to the device. These are absolute stress ratings only. Functional operation of the device is not implied at these or any other conditions in excess of those given in the operational sections of the data sheet. Exposure to Absolute Maximum Ratings for extended periods can adversely affect device reliability.

External leads can be bonded and soldered safely at temperatures of up to 300 °C.

Voltage Range on any Pin with Respect to Ground	–0.5 V to +6 V
Power Dissipation	1 W
Ambient Temperature Range	–40 °C to +85 °C
Storage Temperature Range	–65 °C to +150 °C

8.1 Handling Precautions

All MOS devices must be handled with certain precautions to avoid damage due to the accumulation of static charge. Although input protection circuitry has been incorporated into the devices to minimize the effect of this static build-up, proper precautions should be taken to avoid exposure to electrostatic discharge during handling and mounting. AT&T employs a human-body model for ESD susceptibility testing. Since the failure voltage of electronic devices is dependent on the current, voltage, and, hence, the resistance and capacitance, it is important that standard values be employed to establish a reference by which to compare test data. Values of 100 pF and 1500 Ω are the most common and are the values used in the AT&T human-body model test circuit. The breakdown voltage for the DSP1610 is greater than 1000 V.

8.2 Recommended Operating Conditions

Device Speed	Temperature Class	Supply Voltage V_{DD} (V)		Ambient Temperature T_A (°C)	
		Min	Max	Min	Max
25 ns	Commercial	4.75	5.25	0	70
	Industrial	4.75	5.25	–40	85
33 ns	Commercial	4.5	5.5	0	70
	Industrial	4.75	5.25	–40	85

Package Thermal Considerations

The recommended operating temperature specified above is based on the maximum power, package type, and maximum junction temperature. The following equations describes the relationship between these parameters. Certain applications' maximum power may be less than the worst-case value and can use this relationship to determine the maximum ambient temperature allowed.

$$T_A = T_J - P \times \Theta_{JA}$$

Maximum Junction Temperature (T_J)	+125 °C
132-pin PQFP Maximum Thermal Resistance in Still-air-ambient (Θ_{JA})	+42 °C/W

9.0 Electrical Characteristics

The following electrical characteristics are preliminary and are subject to change. The parameters below are valid for the following conditions: V_{DD} = 5 V ± 10% (please see Recommended Operating Conditions for exceptions).

Table 62. Electrical Characteristics and Requirements*

Parameter	Sym	T_A = 0 °C to 70 °C		T_A = –40 °C to +85 °C		Unit
		Min	Max	Min	Max	
Input Voltage (nonclocks):						
Low	V_{IL}	—	0.8	—	0.8	V
High	V_{IH}	2.0	—	2.0	—	V
Clock Input Voltage: (pins: CKI, ICK, OCK, PIDS, SYNC)						
Low	V_{IL}	—	0.8	—	0.8	V
High	V_{IH}	2.0	—	2.4	—	V
Input Current (except TMS, TDI):						
Low (V_{IL} = 0 V, V_{DD} = 5.5 V)	I_{IL}	-5	—	-5	—	µA
High (V_{IH} = 5.5 V; V_{DD} = 5.5 V)	I_{IH}	—	5	—	5	µA
Input Current (TMS, TDI):						
Low (V_{IL} = 0 V, V_{DD} = 5.5 V)	I_{IL}	–100	—	–100	—	µA
High (V_{IH} = 5.5 V; V_{DD} = 5.5 V)	I_{IH}	—	5	—	5	µA
Output Low Voltage :						
Low (I_{OL} = 2.0 mA)	V_{OL}	—	0.4	—	0.4	V
Low (I_{OL} = 50 µA)	V_{OL}	—	0.2	—	0.2	V
Output High Voltage:						
High (I_{OH} = –2.0 mA)	V_{OH}	V_{DD} – 0.7	—	V_{DD} – 0.7	—	V
High (I_{OH} = –50 µA)	V_{OH}	V_{DD} – 0.2	—	V_{DD} – 0.2	—	V
Output 3-State Current:						
Low (V_{DD} = 5.5 V, V_{IL} = 0 V)	I_{OZL}	–10	—	–10	—	µA
High (V_{DD} = 5.5 V, V_{IH} = 5.5 V)	I_{OZH}	—	10	—	10	µA
Input Capacitance	C_I	—	10	—	10	pF

* Characteristics are properties of the DSP1610. Requirements are restrictions on the external device connected to the DSP1610.

Electrical Characteristics (continued)

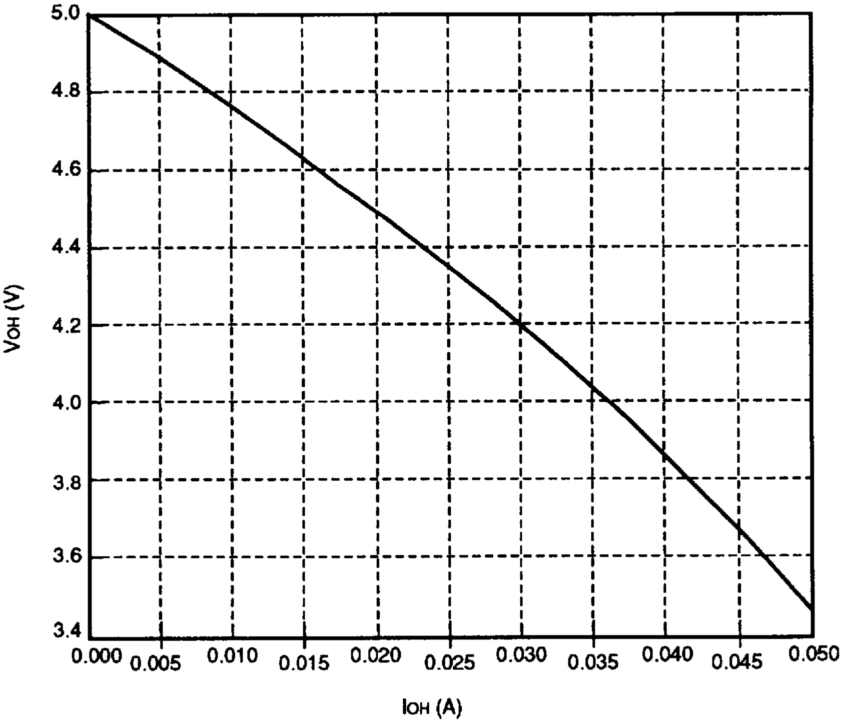


Figure 6. Plot of VOH vs. IOH Under Typical Operating Conditions

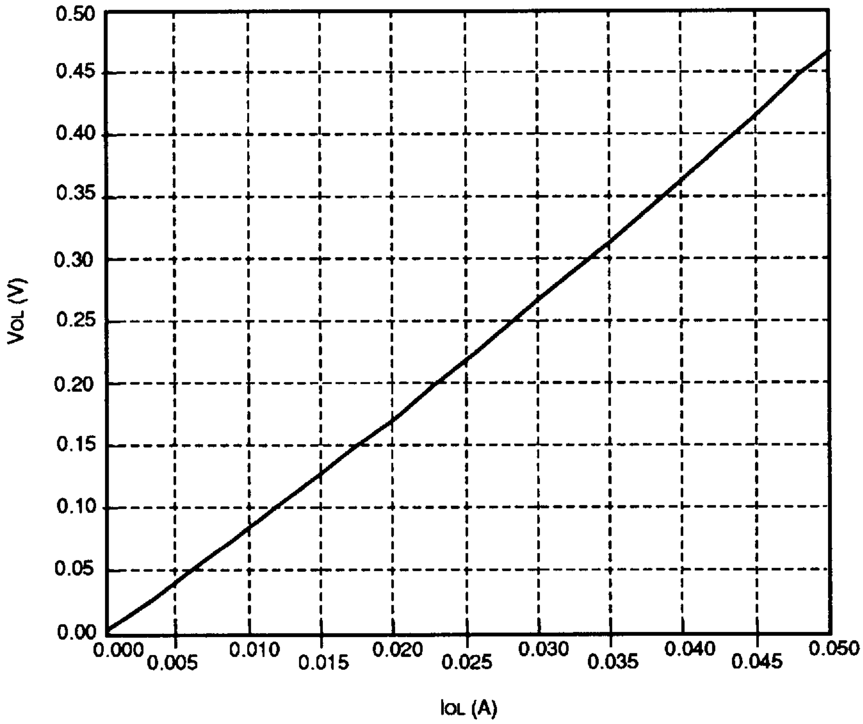


Figure 7. Plot of VOL vs. IOL Under Typical Operating Conditions

Electrical Characteristics (continued)

9.1 Power Dissipation

Power dissipation is highly dependent on program activity and the frequency of operation. The typical power dissipation listed is for a selected application. The following electrical characteristics are preliminary and are subject to change.

Table 63. Power Dissipation

Parameter	Symbol	Typical	Unit
Power Dissipation (inputs at VDD, ioc = 0xc00):			
VDD = 5.0 V, CKI = 60 MHz	PD30	450	mW
VDD = 5.0 V, CKI = 0 MHz	PD0	6	mW

Power Dissipation (sleep mode):
(ioc = 0x0c00, alf = 0x8000, CKI = 60 MHz,
timerc = 0x0040), VDD = 5.0 V
	PDA30	60	mW

Total power dissipation can be calculated on the basis of the application by adding C x V_{DD}^2 x f for each output, where C is the additional load capacitance and f is the output frequency.

Power dissipation due to the input and I/O buffers is highly dependent on the input voltage level. At full CMOS levels, essentially no dc current is drawn; but, for levels near the threshold of 1.4 V, high and unstable levels can flow. Therefore, all unused input pins should be tied inactive to VDD or Vss, and all unused I/O pins should be tied inactive through a 10 kΩ resistor to VDD or Vss. Table 64 shows the input buffer power dissipation for 43 inputs biased at dc level, VIN, with VDD at 5.0 V.

Table 64. Input Buffer Power Dissipation

VIN	5.0	3.6	2.8	2.4	2.0	1.4	0.8	0.4	0
PD (mW)	<1.0	13.5	140	180	180	High and unstable	80	7.5	<1.0

WARNING: **The device needs to be clocked (CKI) for at least six cycles during reset after powerup. Otherwise, high and unstable current may flow.**

10.0 Timing Characteristics

Device Timing Characteristics and Requirements

Characteristics are properties of the DSP1610. Requirements are restrictions on the external device connected to the DSP1610.

The characteristics listed are valid under the following conditions:

$V_{SS} = 0$ V, $T_A = -40$ °C to +85 °C and 0 °C to 70 °C, $C_{LOAD} = 50$ pF. (See Recommended Operating Conditions.)

Output characteristics can be derated as a function of load capacitance (C_L). All outputs: $dt/dC_L \le 0.06$ ns/pF for $0 \le C_L \le 100$ pF at 2.0 V for rising edge, $dt/dC_L \le 0.05$ ns/pF for $0 \le C_L < 100$ pF at 0.8 V for falling edge. For example, the derating for a time delay that includes a rising edge with an external load of 20 pF is

$$\Delta t = (C_L - C_{LOAD})\ dt/dC_L = (20 - 50)\ .06 = -0.18 \text{ ns}$$

Test conditions for inputs:

- Rise and fall times of 4 ns or less
- Timing reference levels for delays = V_{IH}, V_{IL}

Test conditions for outputs:

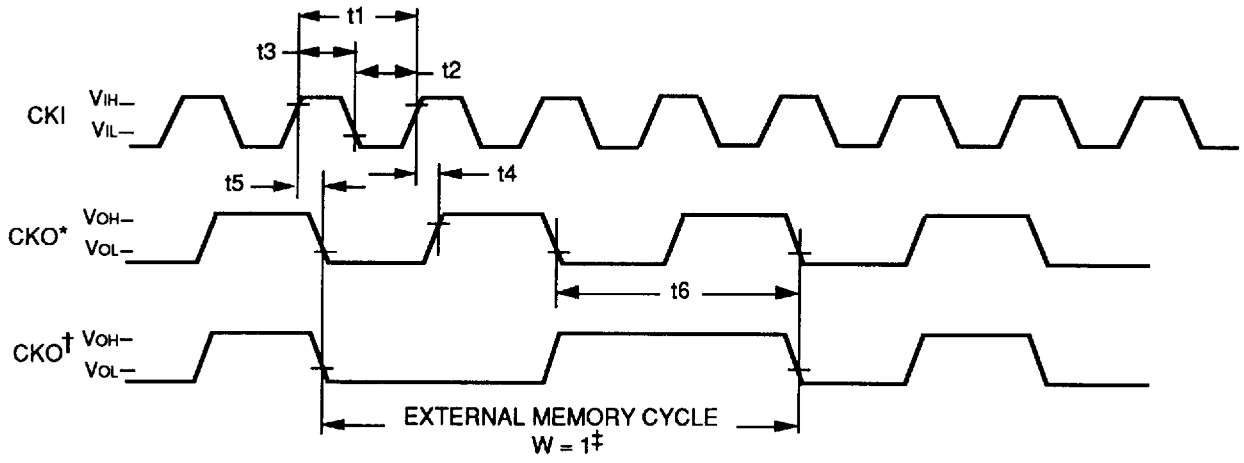
- $C_{LOAD} = 50$ pF (some are noted as 100 pF)
- Timing reference levels for delays = 2.0 V and 0.8 V

For the timing diagrams: $V_{OL} = V_{IL} = 0.8$ V and $V_{OH} = V_{IH} = 2.0$ V. Outputs and inputs can be distinguished in the timing diagrams by the V_{OH}, V_{OL}, or V_{IH}, V_{IL} labels.

For timing diagrams that show the function of the device in a broader sense, see the corresponding diagrams in the DSP1610 Information Manual. The following timing diagrams show the detailed timing at the nanosecond level.

Timing Characteristics (continued)

10.1 DSP Clock Generation



* Free-running clock.
† Wait-stated clock (see Table 40).
‡ W = Number of wait-states.

Figure 8. I/O Clock Timing Diagram

Table 65. Timing Requirements for Input Clock and Output Clock

Abbreviated Reference	Parameter	33 ns		25 ns		Unit
		Min	Max	Min	Max	
t1	Clock in Period (high to high)	16.5	—*	12.5	—*	ns
t2	Clock in Low Time (low to high)	5	—	4	—	ns
t3	Clock in High Time (high to low)	5	—	4	—	ns

* Device is fully static, t1 is tested at 125 ns, and memory hold time is tested at 0.1 s.

Table 66. Timing Characteristics for Input Clock and Output Clock

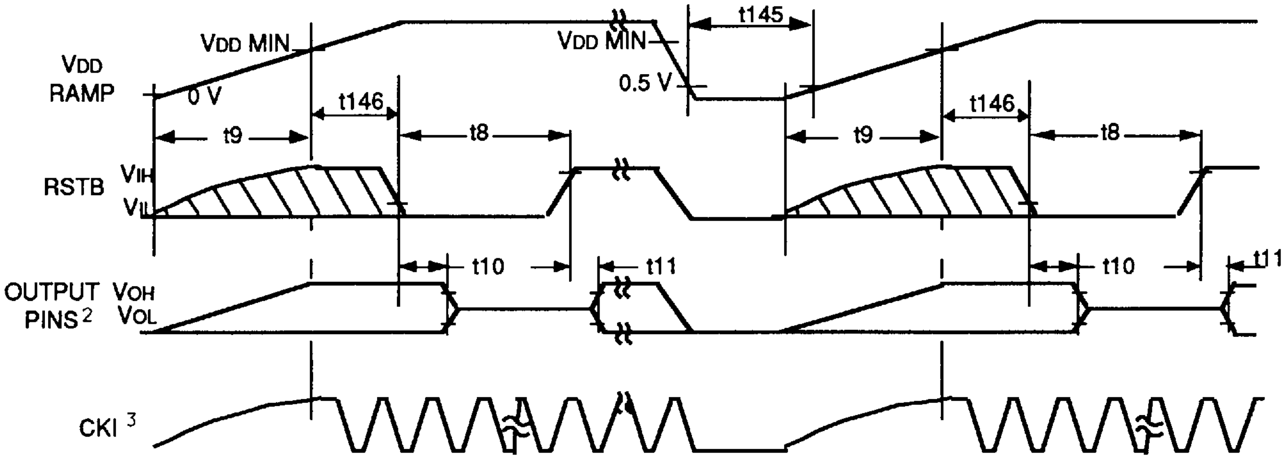
Abbreviated Reference	Parameter	33 ns		25 ns		Unit
		Min	Max	Min	Max	
t4	Clock out High Delay (high to high)	—	22.5	—	22.5	ns
t5	Clock out Low Delay (high to low)	—	22.5	—	22.5	ns
t6	Clock out Period (low to low)	33	—	25	—	ns

Timing Characteristics (continued)

10.2 Reset Circuit

The DSP1610 has a power up reset circuit that automatically clears the JTAG controller upon power up. If the supply voltage falls below VDD MIN[1] and a reset is required, the JTAG controller must be reset with another power up reset, followed by the usual RSTB and CKI reset sequence. Figure 9 shows two separate events: an initial power up and a power up following a drop in the power supply.

[1] See Table 61, Recommended Operating Conditions.



2. When both INT0 and RSTB are asserted, all output and bidirectional pins (except TDO, which 3-states by JTAG control) are put in a 3-state condition. With RSTB asserted and INT0 not asserted, the EROM, ERAMHI, ERAMLO, IO, DSEL, and RWN outputs remain high, and CKO remains a free-running clock.
3. See Table 62 for electrical requirements.

Figure 9. Power Up Reset and Chip Reset Timing Diagram

Table 67. Timing Requirements for Power Up Reset and Chip Reset

Abbreviated Reference	Parameter	Min	Max	Unit
t8	Reset Pulse (low to high)	6T*	—	ns
t9	VDD Ramp	—	40	ms
t145	VDD Low†	3		s
t146	VDD MIN to RSTB Low	0		ns

* T = 2 x t1 for 2X input clock.

† Time below 0.5 V required to activate the on-chip power up reset circuit that resets the JTAG controller. Following a reset of the JTAG controller, the chip must also be reset with the usual RSTB and CKI sequence.

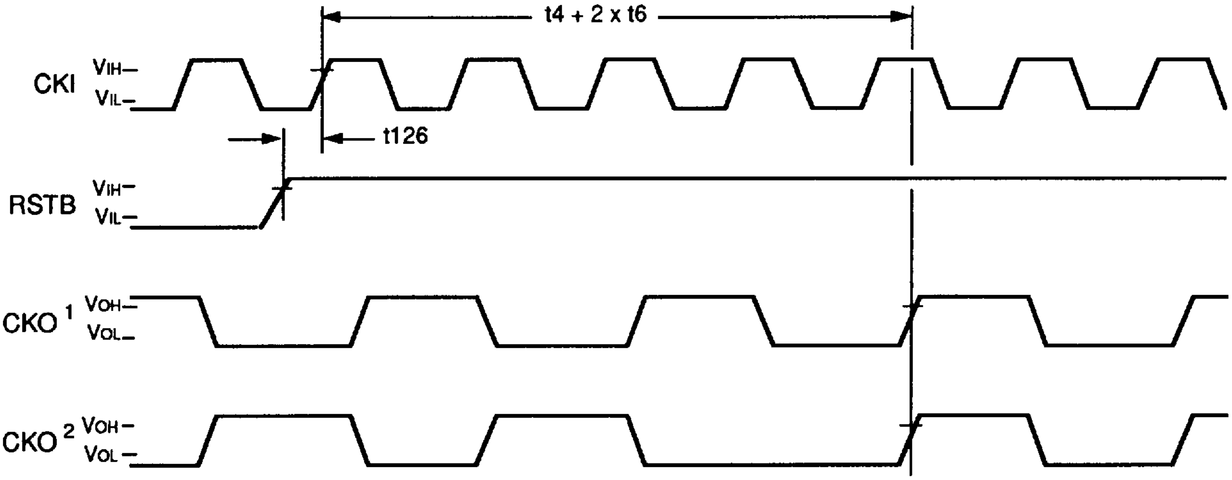
Table 68. Timing Characteristics for Power Up Reset and Chip Reset

Abbreviated Reference	Parameter	Min	Max	Unit
t10	RSTB Disable Time (low to 3-state)	—	100	ns
t11	RSTB Enable Time (high to valid)	—	100	ns

Note: The device needs to be clocked for at least 12 CKI cycles during reset after power up. Otherwise, high and unstable current may flow.

Timing Characteristics (continued)

10.3 Reset Synchronization



Note: CKO[1] and CKO[2] are two possible CKO states before reset. CKO is free-running.

Figure 10. Reset Synchronization Timing

Table 69. Timing Requirements for Reset Synchronization Timing

Abbreviated Reference	Parameter	Min	Max	Unit
t126	Reset Setup (high to high)	10	—	ns

Timing Characteristics (continued)

10.4 JTAG I/O Specifications

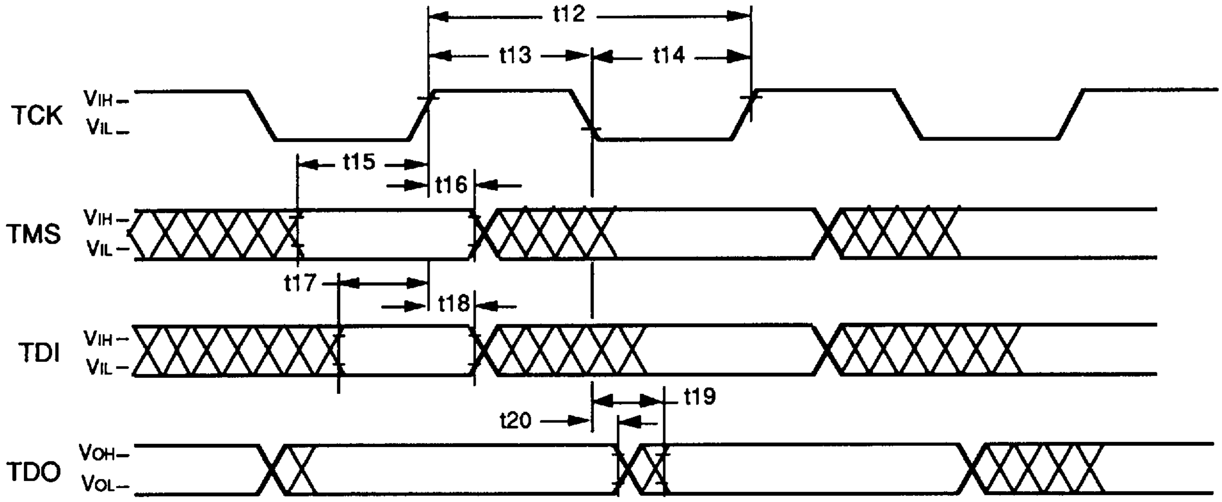


Figure 11. JTAG Timing Diagram

Table 70. Timing Requirements for JTAG Input/Output

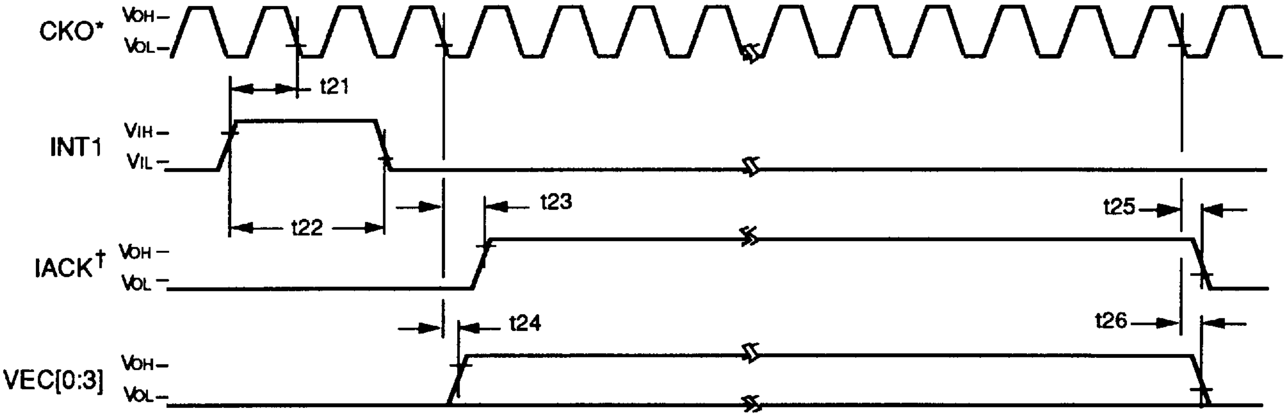
Abbreviated Reference	Parameter	33 ns, 25 ns		Unit
		Min	Max	
t12	TCK Period (high to high)	66	—	ns
t13	TCK High Time (high to low)	30	—	ns
t14	TCK Low Time (low to high)	30	—	ns
t15	TMS Setup Time (valid to high)	10	—	ns
t16	TMS Hold Time (high to invalid)	0	—	ns
t17	TDI Setup Time (valid to high)	10	—	ns
t18	TDI Hold Time (high to invalid)	0	—	ns

Table 71. Timing Characteristics for JTAG Input/Output

Abbreviated Reference	Parameter	Min	Max	Unit
t19	TDO Delay (low to valid)	—	25	ns
t20	TDO Hold (low to invalid)	0	—	ns

Timing Characteristics (continued)

10.5 Interrupt



* CKO is free-running.

† IACK is always guaranteed to be enclosed by VEC[3:0].

Figure 12. Interrupt Timing Diagram

Table 72. Timing Requirements for Interrupt

Note: Interrupt is asserted during an interruptible instruction and no other pending interrupts.

Abbreviated Reference	Parameter	Min	Max	Unit
t21	Interrupt Setup (high to low)	19	—	ns
t22	INT Assertion Time (high to low)	2T*	—	ns

* T= free-running CKO period (i.e., T = 2 x t1 where t1 = CKI period, see Table 65).

Table 73. Timing Characteristics for Interrupt

Note: Interrupt is asserted during an interruptible instruction and no other pending interrupts.

Abbreviated Reference	Parameter	Min	Max	Unit
t23	IACK Assertion Time (low to high)	—	T*/2 + 10	ns
t24	VEC Assertion Time (low to high)	—	12.5	ns
t25	IACK Invalid Time (low to low)	—	10	ns
t26	VEC Invalid Time (low to low)	—	12.5	ns

* T= free-running CKO period (i.e., T = 2 x t1 where t1 = CKI period, see Table 65)

Timing Characteristics (continued)

10.6 Bit Input/Output (BIO)

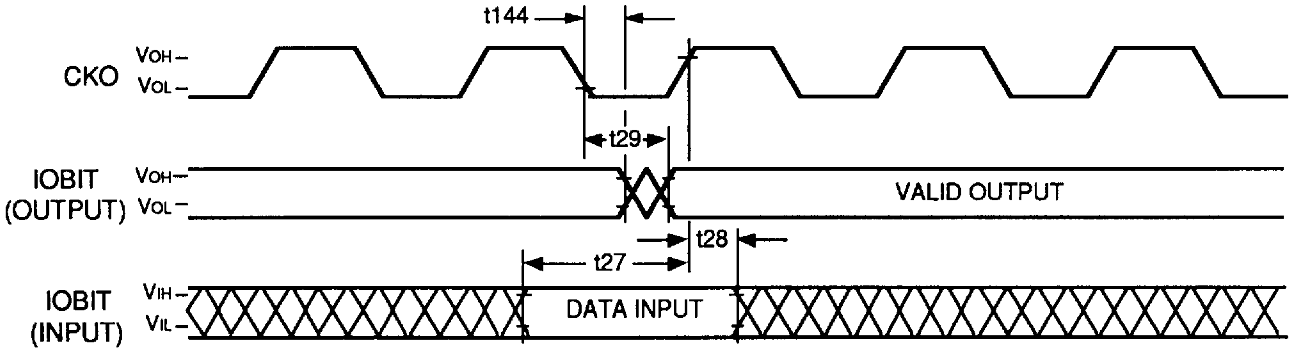


Figure 13. Write Outputs Followed by Read Inputs (cbit = Immediate a1 = sbit)

Table 74. Timing Requirements for BIO Input Read

Abbreviated Reference	Parameter	Min	Max	Unit
t27	IOBIT Input Setup Time (valid to high)	20	—	ns
t28	IOBIT Input Hold Time (high to invalid)	0	—	ns

Table 75. Timing Characteristics for BIO Output

Abbreviated Reference	Parameter	Min	Max	Unit
t29	IOBIT Output Valid Time (low to valid)	—	5	ns
t144	IOBIT Output Hold Time (low to invalid)	–5	—	ns

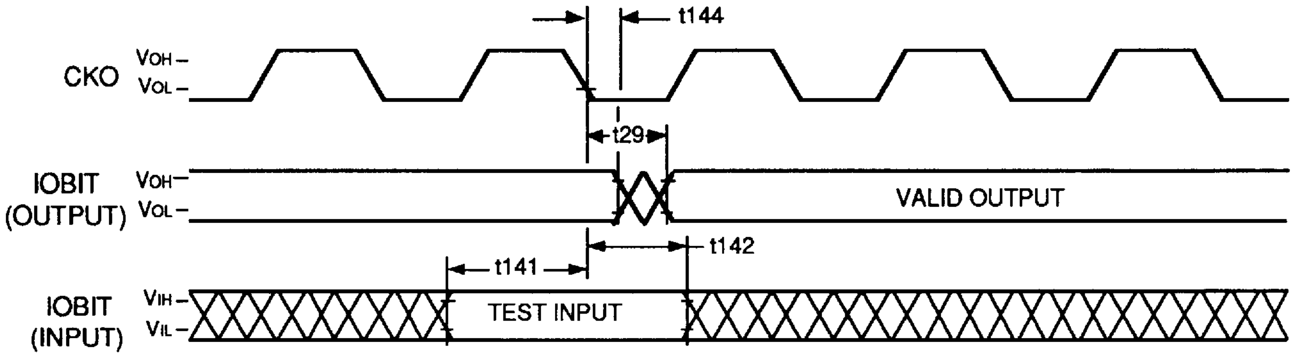


Figure 14. Write Outputs and Test Inputs (cbit = Immediate)

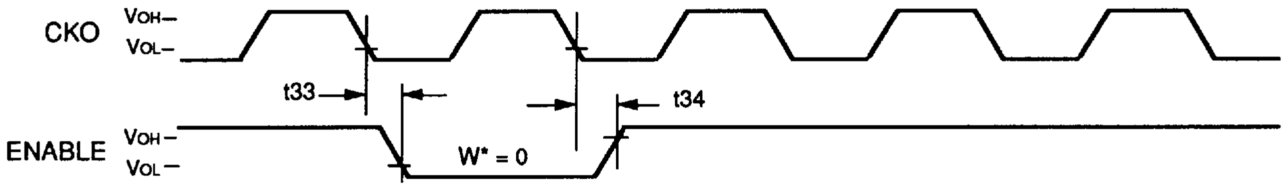
Table 76. Timing Requirements for BIO Input Test

Abbreviated Reference	Parameter	Min	Max	Unit
t141	IOBIT Input Setup Time (valid to low)	20	—	ns
t142	IOBIT Input Hold Time (low to invalid)	0	—	ns

Timing Characteristics (continued)

10.7 External Memory Interface

The following timing diagrams, characteristics, and requirements do not apply to interactions with delayed external memory enables unless so stated. See the *DSP1610 Digital Signal Processor Information Manual* for a more detailed description of the external memory interface including other functional diagrams.



* W = number of wait-states.

Figure 15. Enable Transition Timing

Table 77. Timing Characteristics for External Memory Enables (EROM, ERAMHI, IO, ERAMLO)

Abbreviated Reference	Parameter	Min	Max	Unit
t33	Enable Assert (low to low)	0	5	ns
t34	Enable Deassert (low to high)	0	5	ns

Table 78. Timing Characteristics for Device Enables (DSEL[3:0])

Abbreviated Reference	Parameter	Min	Max	Unit
t33	Enable Assert Time (low to low)	0	6	ns
t34	Enable Deassert Time (low to high)	0	6	ns

Table 79. Timing Characteristics for Delayed External Memory Enables (ioc = 0x000F)

Abbreviated Reference	Parameter	Min	Max	Unit
t33	Delayed Enable Assert Time	T*/2	T*/2 + 9	ns

* T= free-running CKO period (i.e., T = 2 x t1 where t1 = CKI period, see Table 65)

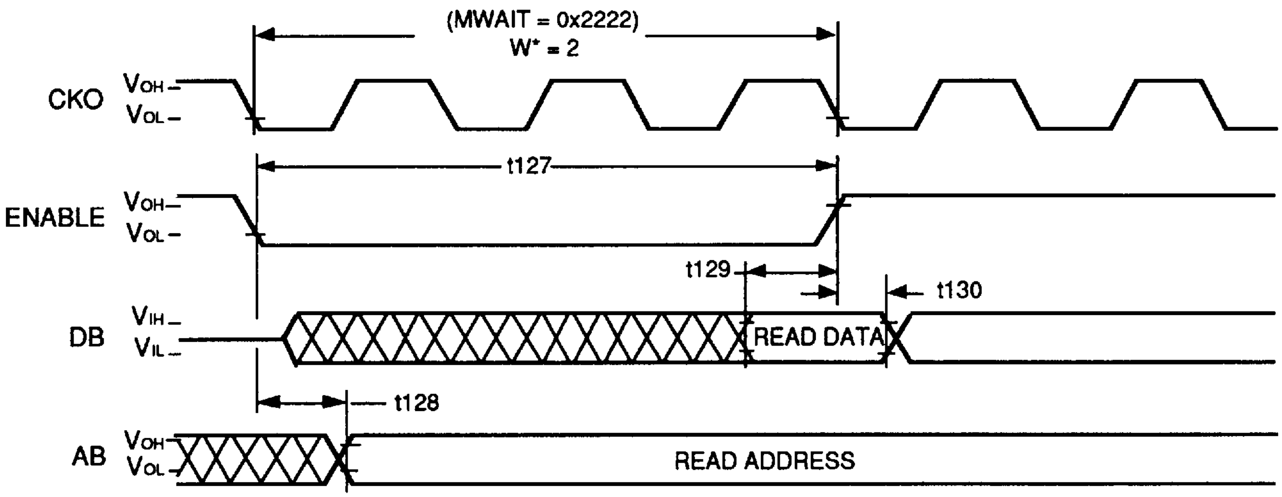
Table 80. Timing Characteristics for Delayed Device Enables (ioc = 0x00F0)

Abbreviated Reference	Parameter	Min	Max	Unit
t33	Delayed Enable Assert Time	T*/2	T*/2 + 10	ns

* T= free-running CKO period (i.e., T = 2 x t1 where t1 = CKI period, see Table 65)

Timing Characteristics (continued)

External Memory Interface



* W = number of wait-states.

Figure 16. External Memory Data Read Timing Diagram

Table 81. Timing Characteristics for External Memory Access

Abbreviated Reference	Parameter	Min	Max	Unit
t127	Enable Width (low to high)	T*(1 + w) − 1	—	ns
t128	Address Valid (enable low to valid)	—	0	ns

* T= free-running CKO period (i.e., T = 2 x t1 where t1 = CKI period, see Table 65).

Table 82. Timing Requirements for External Memory Read (EROM, ERAMHI, IO, ERAMLO)

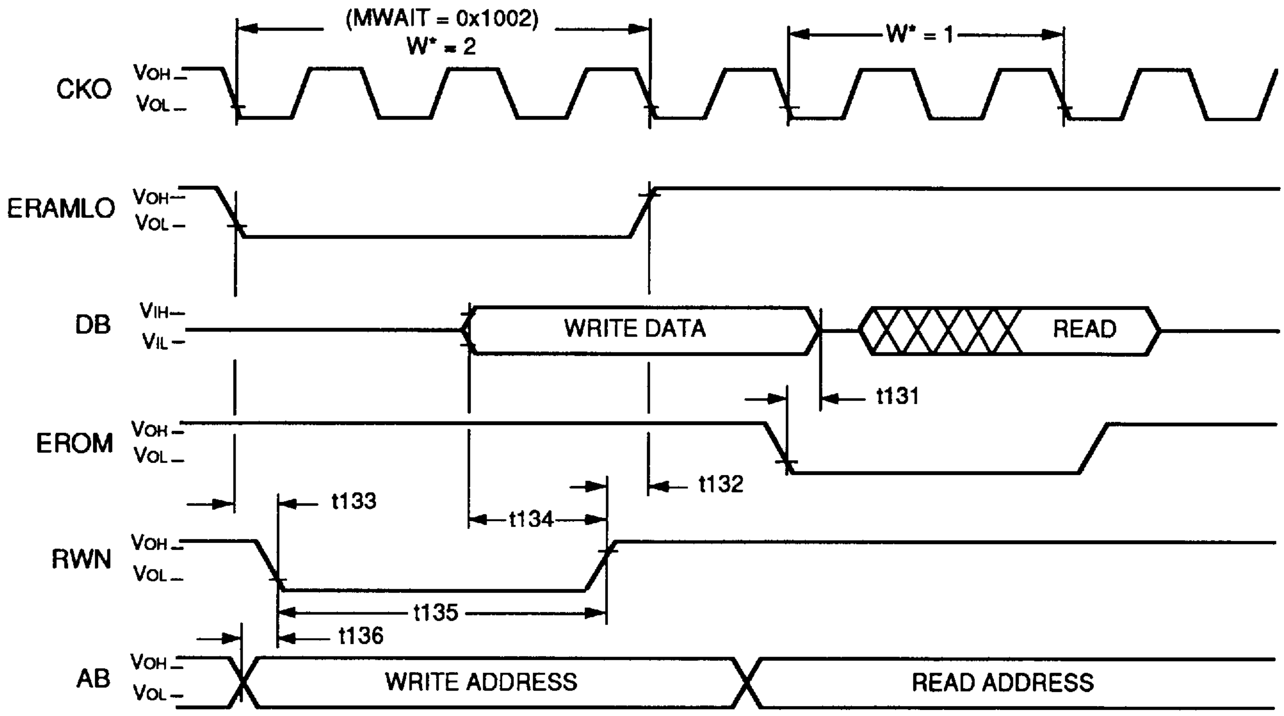
Abbreviated Reference	Parameter	Min	Max	Unit
t129	Read Data Setup (valid to enable high)	20	—	ns
t130	Read Data Hold (enable high to hold)	0	—	ns

Table 83. Timing Requirements for Device Read (DSEL[3:0])

Abbreviated Reference	Parameter	Min	Max	Unit
t129	Read Data Setup (valid to enable high)	21	—	ns
t130	Read Data Hold (enable high to hold)	0	—	ns

Timing Characteristics (continued)

External Memory Interface



* W = number of wait-states

Figure 17. External Memory Data Write Timing Diagram

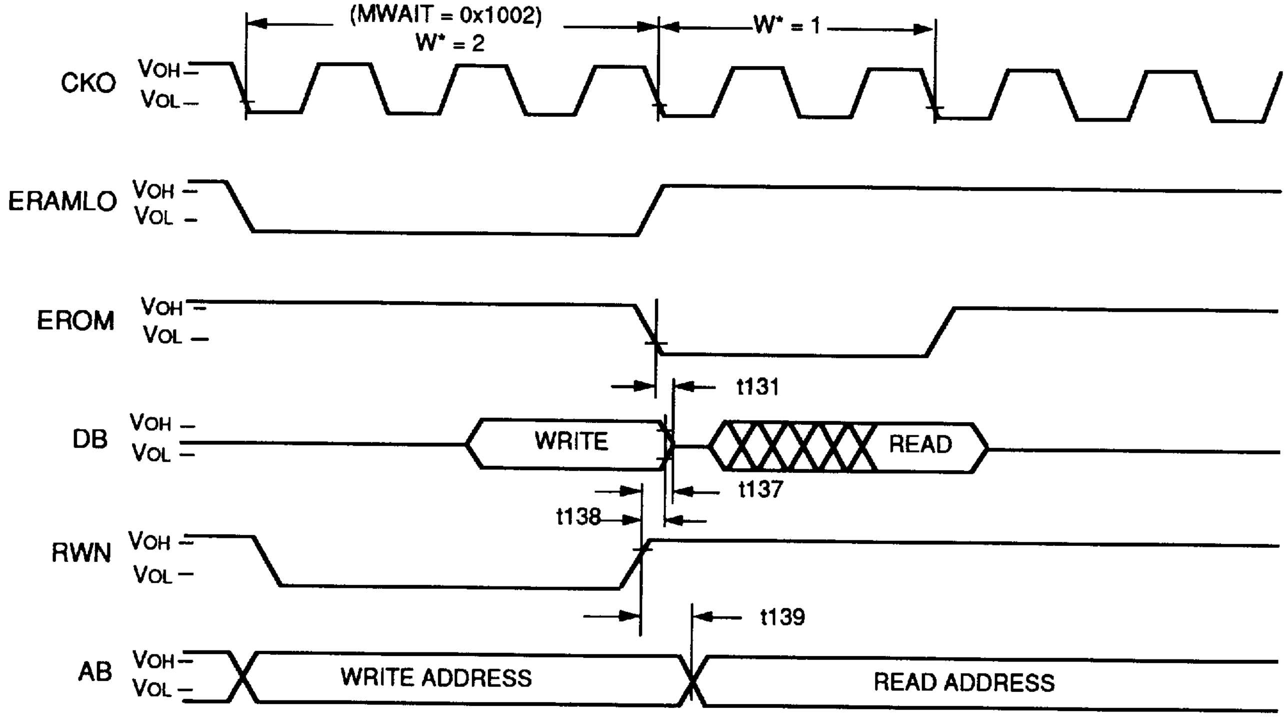
Table 84. Timing Characteristics for External Memory Data Write (All Enables)

Abbreviated Reference	Parameter	Min	Max	Unit
t131	Write Overlap (enable low to 3-state)	—	0	ns
t132	RWN Advance (RWN high to enable high)	0	—	ns
t133	RWN Delay (enable low to RWN low)	0	—	ns
t134	Write Data Setup (data valid to RWN high)	T*(1 + W)/2 – 10	—	ns
t135	RWN Width (low to high)	T*(1 + W) – 8	—	ns
t136	Write Address Setup (address valid to RWN low)	0	—	ns

* T= free-running CKO period (i.e., T = 2 x t1 where t1 = CKI period, see Table 65).

Timing Characteristics (continued)

External Memory Interface



* W = number of wait-states

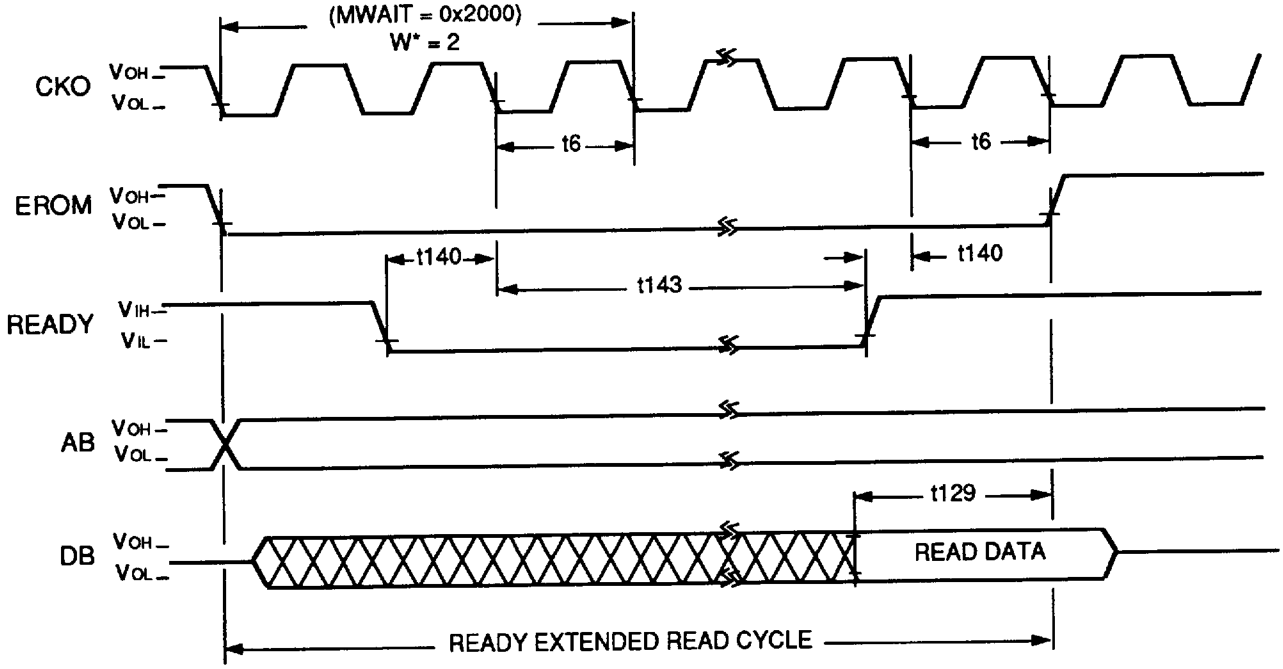
Figure 18. Write Cycle Followed by Read Cycle

Table 85. Timing Characteristics for Write Cycle Followed by Read Cycle

Abbreviated Reference	Parameter	Min	Max	Unit
t137	Write Data 3-State (RWN high to 3-state)	—	2	ns
t138	Write Data Hold (RWN high to data hold)	0	—	ns
t139	Write Address Hold (RWN high to address hold)	0	—	ns

Timing Characteristics (continued)

External Memory Interface



* W = number of wait-states

Figure 19. READY[†] Extended Read Cycle Timing

Table 86. Timing Requirements for READY Extended Read Cycle Timing

Abbreviated Reference	Parameter	Min	Max	Unit
t140	Ready Setup (valid to CKO low)	15	—	ns
t143	Ready Hold (CKO low to invalid)	0	—	ns

[†] There is a design exception that affects versions F10, F11, F12, and F13 of the DSP1610 involving the READY pin. READY is an input used to stall the DSP. It is typically used when an external memory access requires >15 wait-states. If the next instruction following the READY stretched access requires wait-states, program execution or the next access can be corrupted. Use of READY with these versions of the DSP1610 must be avoided.

Timing Characteristics (continued)

10.8 PIO Specifications

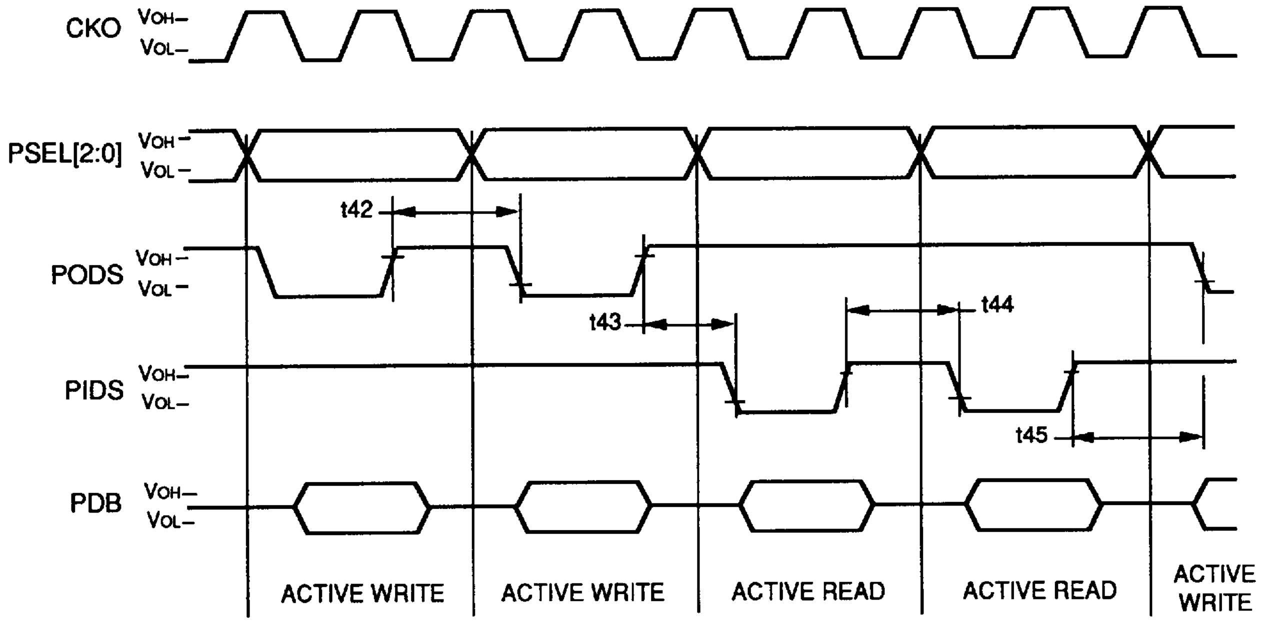


Figure 20. PIO Active Mode Interaccess Timing

Table 87. Timing Requirements for PIO Active Mode Interaccess

Abbreviated Reference	Parameter	Min	Max	Unit
t42	PODS High to PODS Low	20	—	ns
t43	PODS High to PIDS Low	20	—	ns
t44	PIDS High to PIDS Low	20	—	ns
t45	PIDS High to PODS Low	20	—	ns

Timing Characteristics (continued)

PIO Specifications

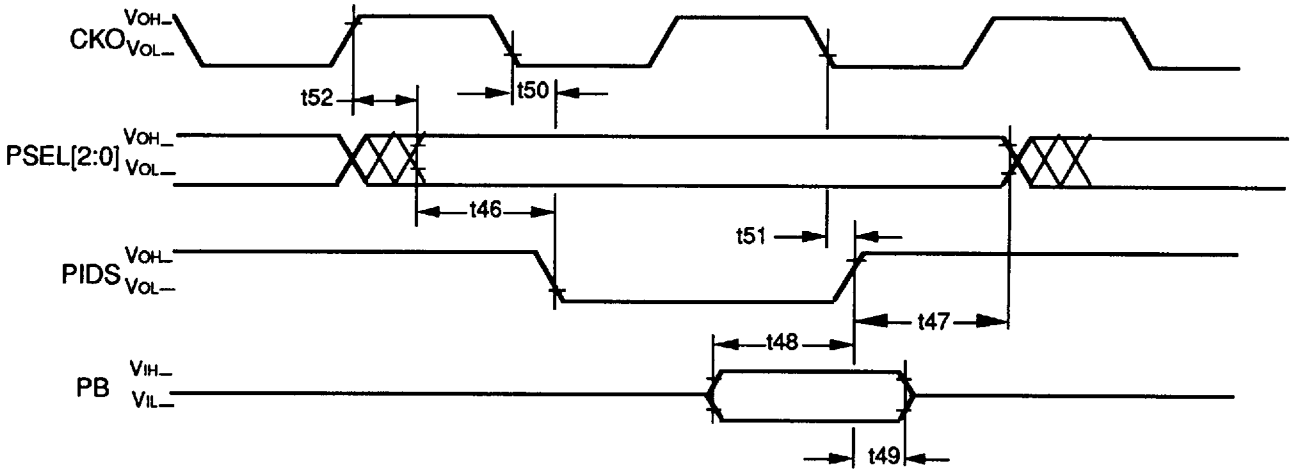


Figure 21. PIO Active Mode Input Timing Diagram

Table 88. Timing Requirements for PIO Active Mode Input

Abbreviated Reference	Parameter	Min	Max	Unit
t48	PB[7:0] Setup Time (valid to high)	15	—	ns
t49	PB[7:0] Hold Time (high to invalid)	0	—	ns

Table 89. Timing Characteristics for PIO Active Mode Input

Abbreviated Reference	Parameter	Min	Max	Unit
t46	PSEL[2:0] Valid to PIDS low	T*/2	—	ns
t47	PIDS High to PSEL[2:0] Invalid	T*/2	—	ns
t50	CKO Fall to PIDS Assertion (low to low)	—	12.5	ns
t51	CKO Fall to PIDS Negation (low to high)	—	10	ns
t52	PSEL Valid from CKO High (high to valid)	—	12.5	ns

* T = the period of the nonwait-stated CKO

Timing Characteristics (continued)

PIO Specifications

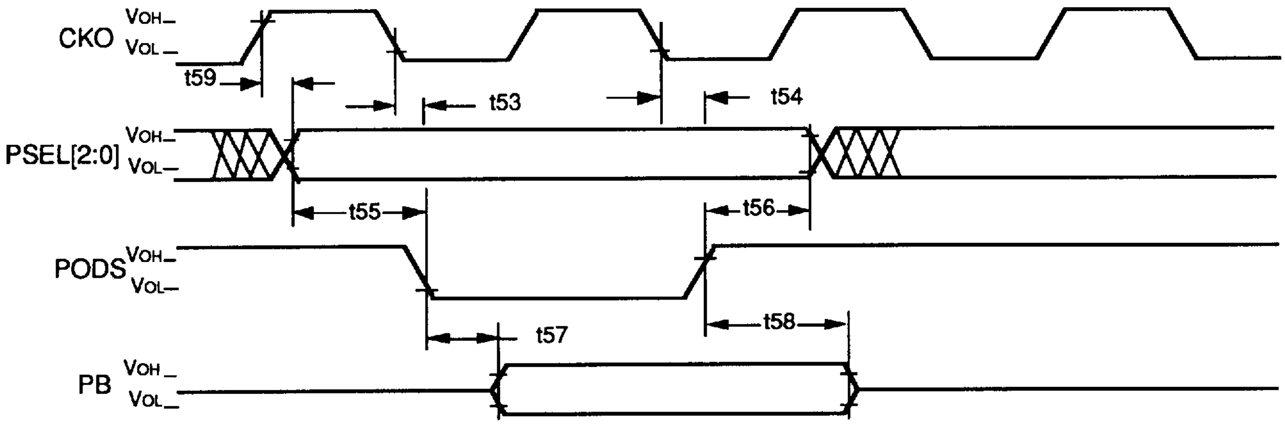


Figure 22. PIO Active Mode Output Timing Diagram

Table 90. Timing Characteristics for PIO Active Mode Output

Abbreviated Reference	Parameter	Min	Max	Unit
t53	CKO Low to PODS Assertion (low to low)	—	12.5	ns
t54	CKO Low to PODS Negation (low to high)	—	10	ns
t55	PSEL[2:0] Valid to PODS low	T*/2	—	ns
t56	PODS High to PSEL[2:0] Invalid	T*/2	—	ns
t57	PODS Low to PB Valid (low to valid)	—	15	ns
t58	PODS High to PB 3-State (high to invalid)	T*/2 – 8	—	ns
t59	PSEL Valid from CKO High (high to valid)	—	12.5	ns

* T= free-running CKO period (i.e., T = 2 x t1 where t1 = CKI period, see Table 65).

Timing Characteristics (continued)

PIO Specifications

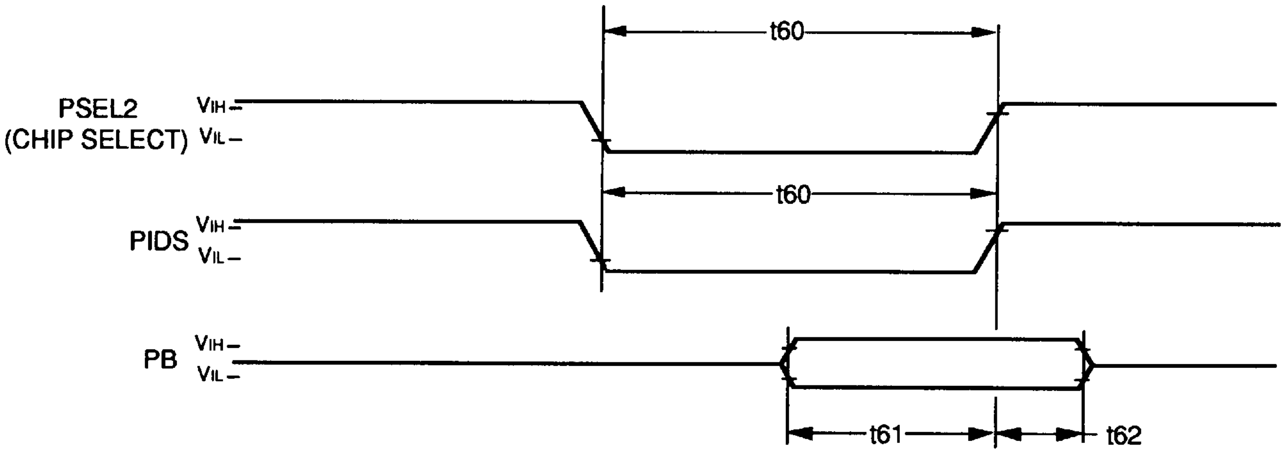


Figure 23. PIO Passive Mode Input Timing Diagram

Table 91. Timing Requirements for Passive Mode Input

Abbreviated Reference	Parameter	Min	Max	Unit
t60	PIDS or PSEL2 Pulse Width (low to high)*	T‡	—	ns
t61	PB[15:0] Setup Time† (valid to high)	15	—	ns
t62	PB[15:0] Hold Time† (high to invalid)	0	—	ns

* Data on PB is latched on the rising edge of PIDS or PSEL2, whichever occurs first.

† Setup and hold are measured from PSEL2 or PIDS, whichever occurs first.

‡ T = the period of the nonwait-stated CKO.

Timing Characteristics (continued)

PIO Specifications

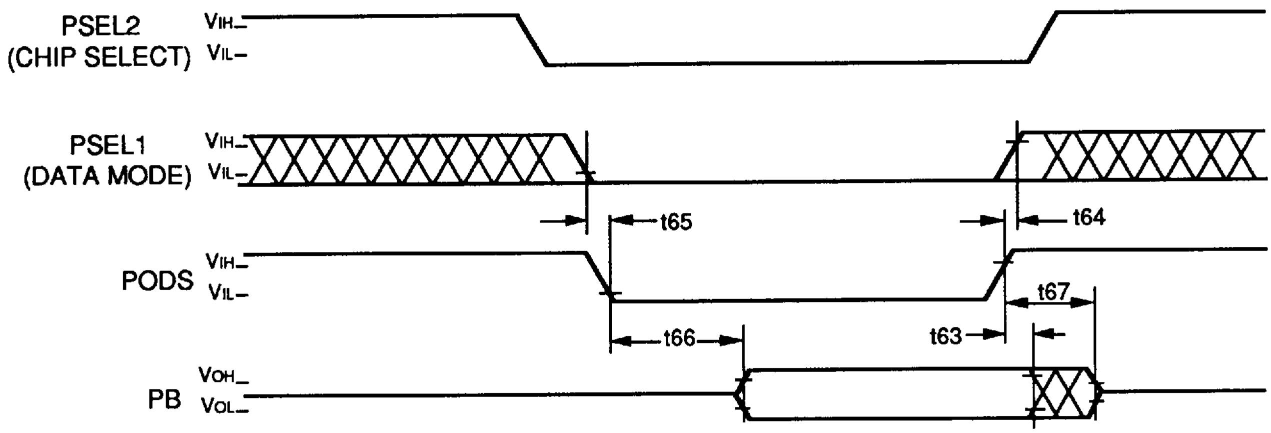


Figure 24. PIO Passive Mode Output Timing Diagram

Table 92. Timing Requirements for PIO Passive Mode Output

Abbreviated Reference	Parameter	Min	Max	Unit
t64	PSEL1 Hold * (high to invalid)	0	—	ns
t65	PSEL1 Setup† (valid to low)	0	—	ns

* Hold is measured from PODS high or PSEL2 high, whichever occurs first.

† Setup is measured from PODS low or PSEL2 low, whichever occurs last.

Note: If PSEL1 is low during the access, pdx data appears on PB. If PSEL1 is high during the access, PIO status appears on PB.

Table 93. Timing Characteristics for PIO Passive Mode Output

Abbreviated Reference	Parameter	Min	Max	Unit
t66	PODS Low‡ to PB Valid	—	18	ns
t67	PODS High§ to PB 3-state	—	18	ns
t63	PB Hold (PODS high to PB invalid)	0	—	ns

‡ Data valid delay is measured from PODS low or PSEL2 low, whichever occurs last.

§ Data 3-state delay is measured from PODS high or PSEL2 high, whichever occurs first.

Timing Characteristics (continued)

PIO Specifications

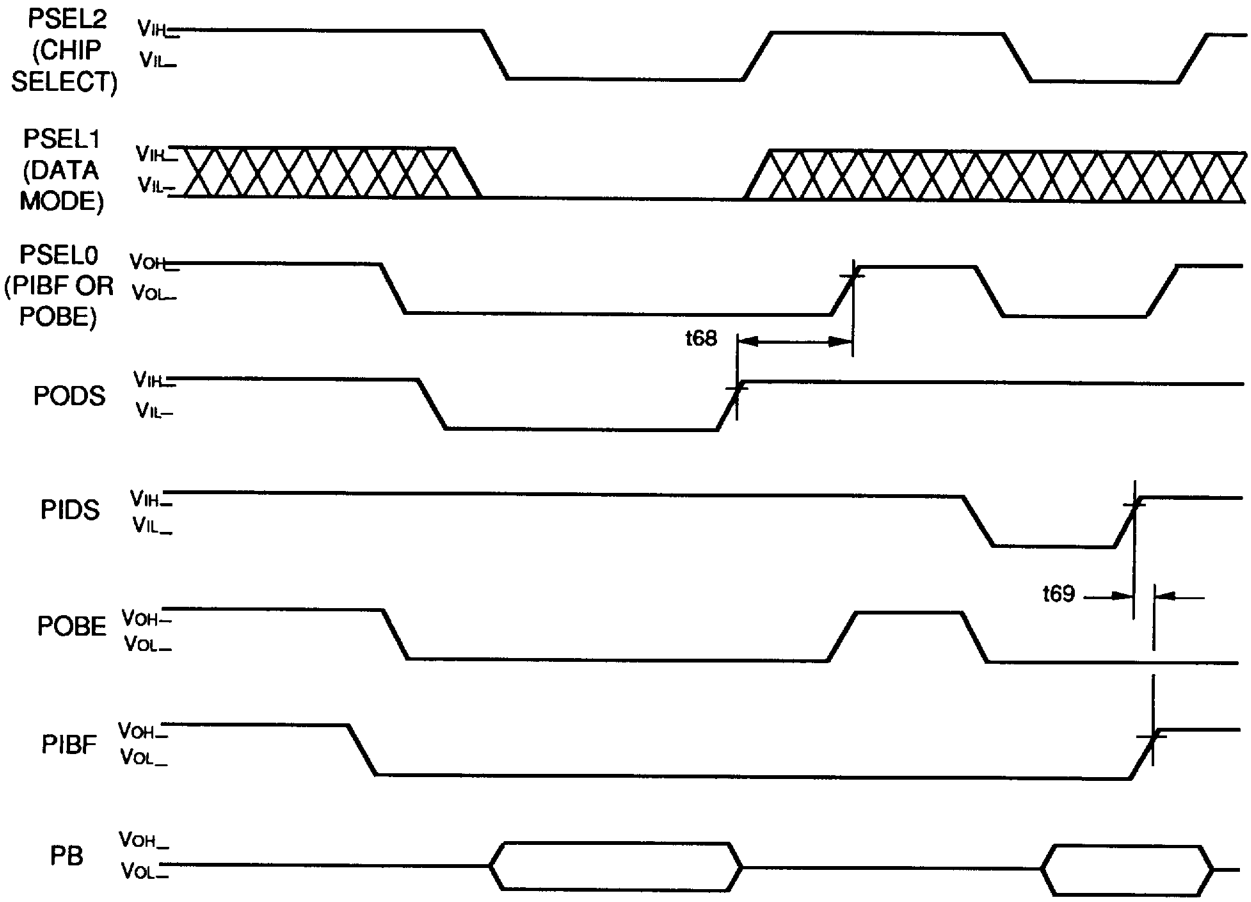


Figure 25. PIO Peripheral Mode Input/Output Timing Diagram

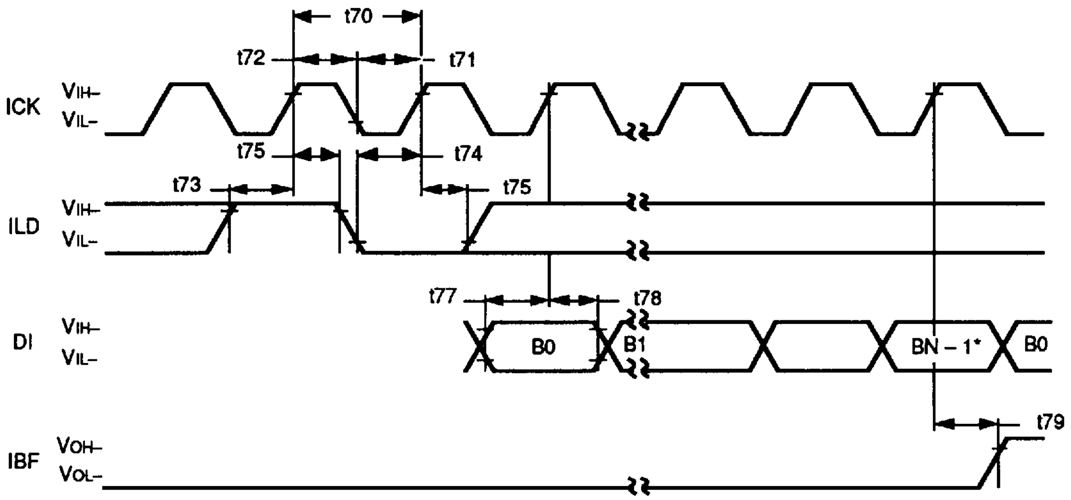
Table 94. Timing Characteristics for Peripheral Mode Input/Output

Abbreviated Reference	Parameter	Min	Max	Unit
t68	PSEL2/PODS High to POBE/PSEL0 High	—	2T*	ns
t69	PSEL2/PIDS High to PIBF/PSEL0 Low	—	2T*	ns

* T = 2 x t1 (t1 = CKI period, see Table 65).

Timing Characteristics (continued)

10.9 Serial I/O Specifications



* N = 16 or 8 bits

Figure 26. SIO Passive Mode Input Timing Diagram

Table 95. Timing Requirements for Serial Inputs

Abbreviated Reference	Parameter	33 ns		25 ns		Unit
		Min	Max	Min	Max	
t70	Clock Period (high to high)	66	—*	50	—*	ns
t71	Clock Low Time (low to high)	30	—	23	—	ns
t72	Clock High Time (high to low)	30	—	23	—	ns
t73	Load High Setup (high to high)	8	—	8	—	ns
t74	Load Low Setup (low to high)	8	—	8	—	ns
t75	Load High Hold (high to invalid)	2	—	2	—	ns
t77	Data Setup (valid to high)	7	—	7	—	ns
t78	Data Hold (high to invalid)	2	—	2	—	ns

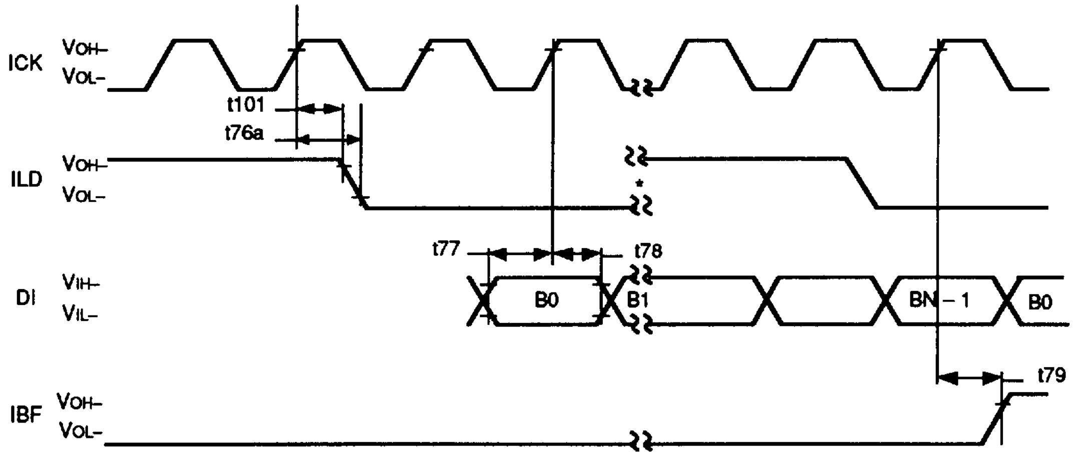
* Device is fully static; t70 is tested at 2000 ns.

Table 96. Timing Characteristics for Serial Outputs

Abbreviated Reference	Parameter	Min	Max	Unit
t79	IBF Delay (high to high)	—	20	ns

Timing Characteristics (continued)

Serial I/O Specifications



* ILD goes high during bit 6 (of 0—15), N = 8 or 16.

Figure 27. SIO Active Mode Input Timing Diagram

Table 97. Timing Requirements for Serial Inputs

Abbreviated Reference	Parameter	33 ns		25 ns		Unit
		Min	Max	Min	Max	
t77	Data Setup (valid to high)	7	—	7	—	ns
t78	Data Hold (high to invalid)	2	—	2	—	ns

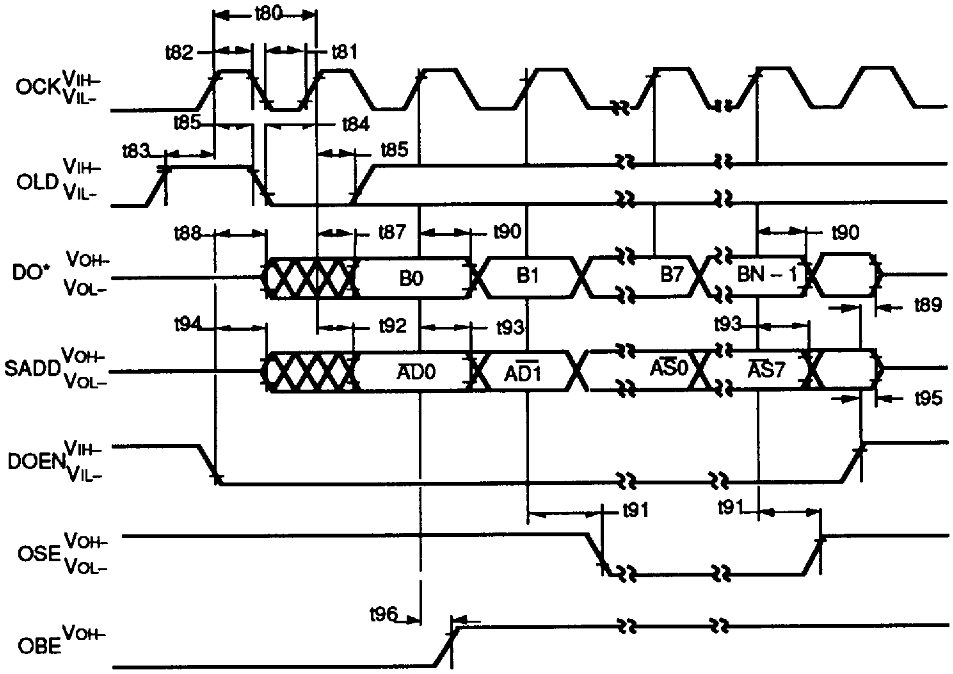
* Device is fully static; t70 is tested at 2000 ns.

Table 98. Timing Characteristics for Serial Outputs

Abbreviated Reference	Parameter	Min	Max	Unit
t76a	ILD Delay (high to low)	—	30	ns
t101	ILD Hold (high to invalid)	3	—	ns
t79	IBF Delay (high to high)	—	20	ns

Timing Characteristics (continued)

Serial I/O Specifications



* See SIOC register, MSB field to determine if B0 is the MSB or LSB. See SIOC register, ILEN field to determine if the DO word length is 8 or 16 bits.

Figure 28. SIO Passive Mode Output Timing Diagram

Table 99. Timing Requirements for Serial Inputs

Abbreviated Reference	Parameter	33 ns		25 ns		Unit
		Min	Max	Min	Max	
t80	Clock Period (high to high)	66	—*	50	—*	ns
t81	Clock Low Time (low to high)	30	—	23	—	ns
t82	Clock High Time (high to low)	30	—	23	—	ns
t83	Load High Setup (high to high)	7	—	7	—	ns
t84	Load Low Setup (low to high)	7	—	7	—	ns
t85	Load Hold	2	—	2	—	ns

* Device is fully static; t80 is tested at 2000 ns.

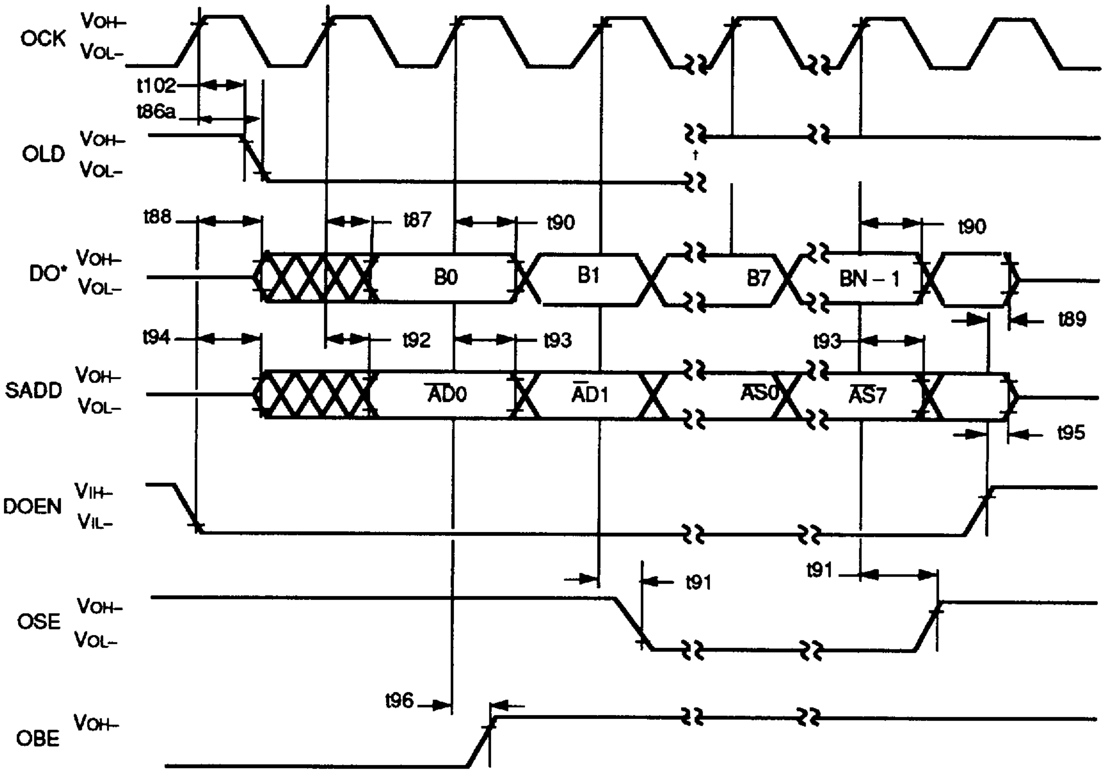
Table 100. Timing Characteristics for Serial Outputs*

Abbreviated Reference	Parameter	Min	Max	Unit
t87	Data Delay (high to valid)	—	35	ns
t88	Enable Data Delay (low to active)	—	35	ns
t89	Disable Data Delay (high to 3-state)	—	35	ns
t90	Data Hold (high to invalid)	5	—	ns
t91	OSE Delay (high to high)	—	25	ns
t92	Address Delay (high to valid)	—	30	ns
t93	Address Hold (high to invalid)	5	—	ns
t94	Enable Delay (low to active)	—	35	ns
t95	Disable Delay (high to 3-state)	—	35	ns
t96	OBE Delay (high to high)	—	25	ns

* Capacitance load on OCK and DO equals 100 pF.

Timing Characteristics (continued)

Serial I/O Specifications



* See SIOC register, MSB field to determine if B0 is the MSB or LSB. See SIOC register, ILEN field to determine if the DO word length is 8 or 16 bits.

† OLD goes high at the end of bit 6 of 0—15.

Figure 29. SIO Active Mode Output Timing Diagram

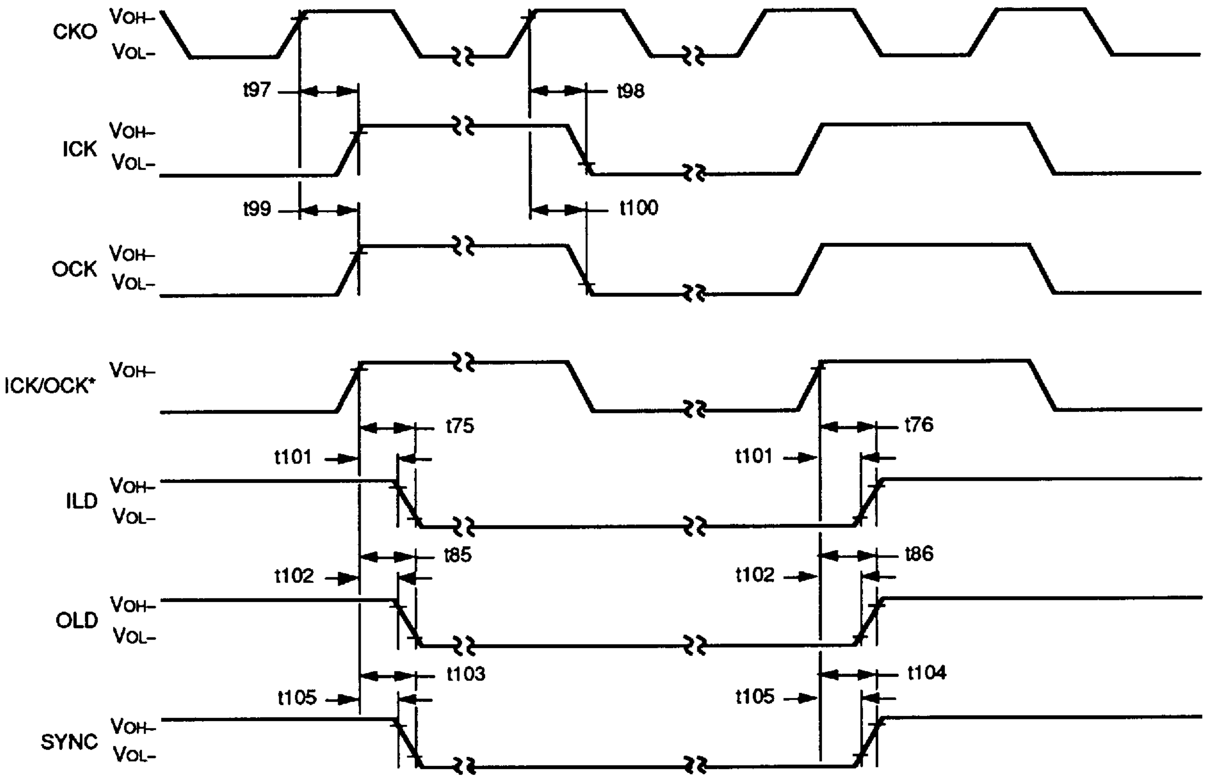
Table 101. Timing Characteristics for Serial Outputs*

Abbreviated Reference	Parameter	Min	Max	Unit
t86a	OLD Delay (high to low)	—	30	ns
t102	OLD Hold (high to invalid)	3	—	ns
t87	Data Delay (high to valid)	—	35	ns
t88	Enable Data Delay (low to active)	—	35	ns
t89	Disable Data Delay (high to 3-state)	—	35	ns
t90	Data Hold (high to invalid)	5	—	ns
t91	OSE Delay (high to high)	—	25	ns
t92	Address Delay (high to valid)	—	30	ns
t93	Address Hold (high to invalid)	5	—	ns
t94	Enable Delay (low to active)	—	35	ns
t95	Disable Delay (high to 3-state)	—	35	ns
t96	OBE Delay (high to high)	—	25	ns

* Capacitance load on OCK and DO equals 100 pF.

Timing Characteristics (continued)

Serial I/O Specifications



* See sioc register, LD field.

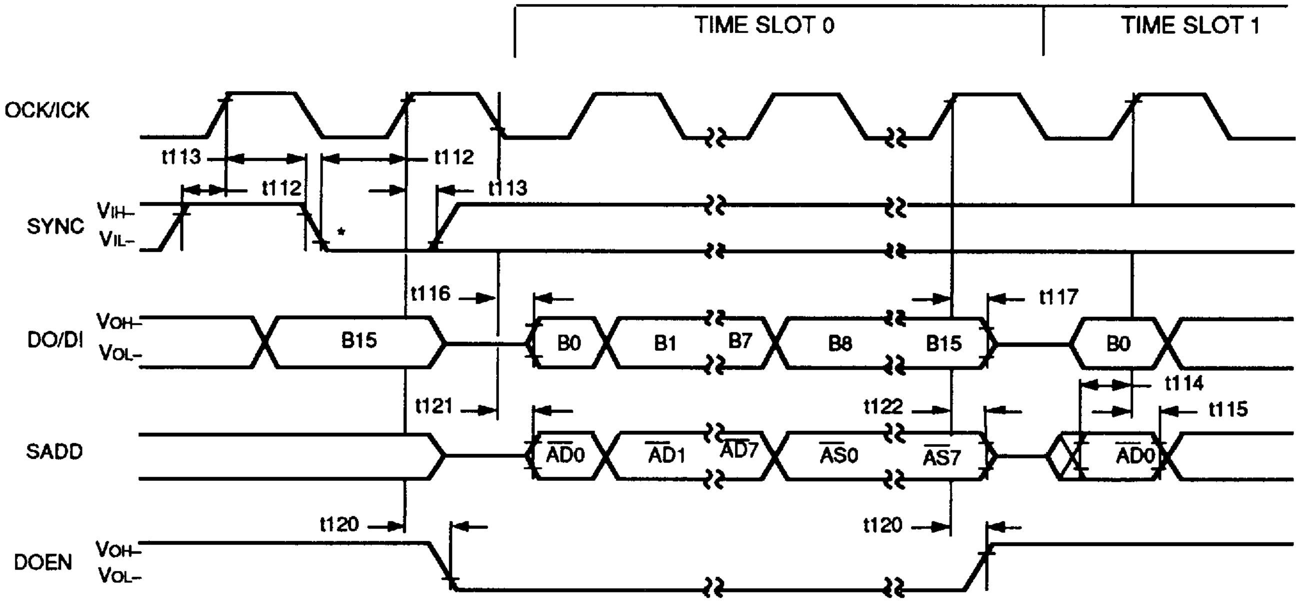
Figure 30. Serial I/O Active Mode Clock Timing

Table 102. Timing Characteristics for Signal Generation

Abbreviated Reference	Parameter	Min	Max	Unit
t97	ICK Delay (high to high)	—	15	ns
t98	ICK Delay (high to low)	—	15	ns
t99	OCK Delay (high to high)	—	15	ns
t100	OCK Delay (high to low)	—	15	ns
t175	ILD Delay (high to low)	—	30	ns
t176	ILD Delay (high to high)	—	30	ns
t101	ILD Hold (high to invalid)	3	—	ns
t185	OLD Delay (high to low)	—	30	ns
t186	OLD Delay (high to high)	—	30	ns
t102	OLD Hold (high to invalid)	3	—	ns
t103	SYNC Delay (high to low)	—	30	ns
t104	SYNC Delay (high to high)	—	30	ns
t105	SYNC Hold (high to invalid)	3	—	ns

Timing Characteristics (continued)

10.10 SIO Multiprocessor Communication



* Negative edge initiates time slot 0.

Figure 31. SIO Multiprocessor Timing Diagram

Table 103. Timing Requirements for SIO Multiprocessor Communication

Note: All serial I/O timing requirements and characteristics (except DOEN characteristics) still apply.

Abbreviated Reference	Parameter	Min	Max	Unit
t112	Sync Setup (high/low to high)	12	—	ns
t113	Sync Hold (high to high/low)	2	—	ns
t114	Address Setup (valid to high)	12	—	ns
t115	Address Hold (high to invalid)	2	—	ns

Table 104. Timing Characteristics for SIO Multiprocessor Communication

Note: All serial I/O timing requirements and characteristics (except DOEN characteristics) still apply.

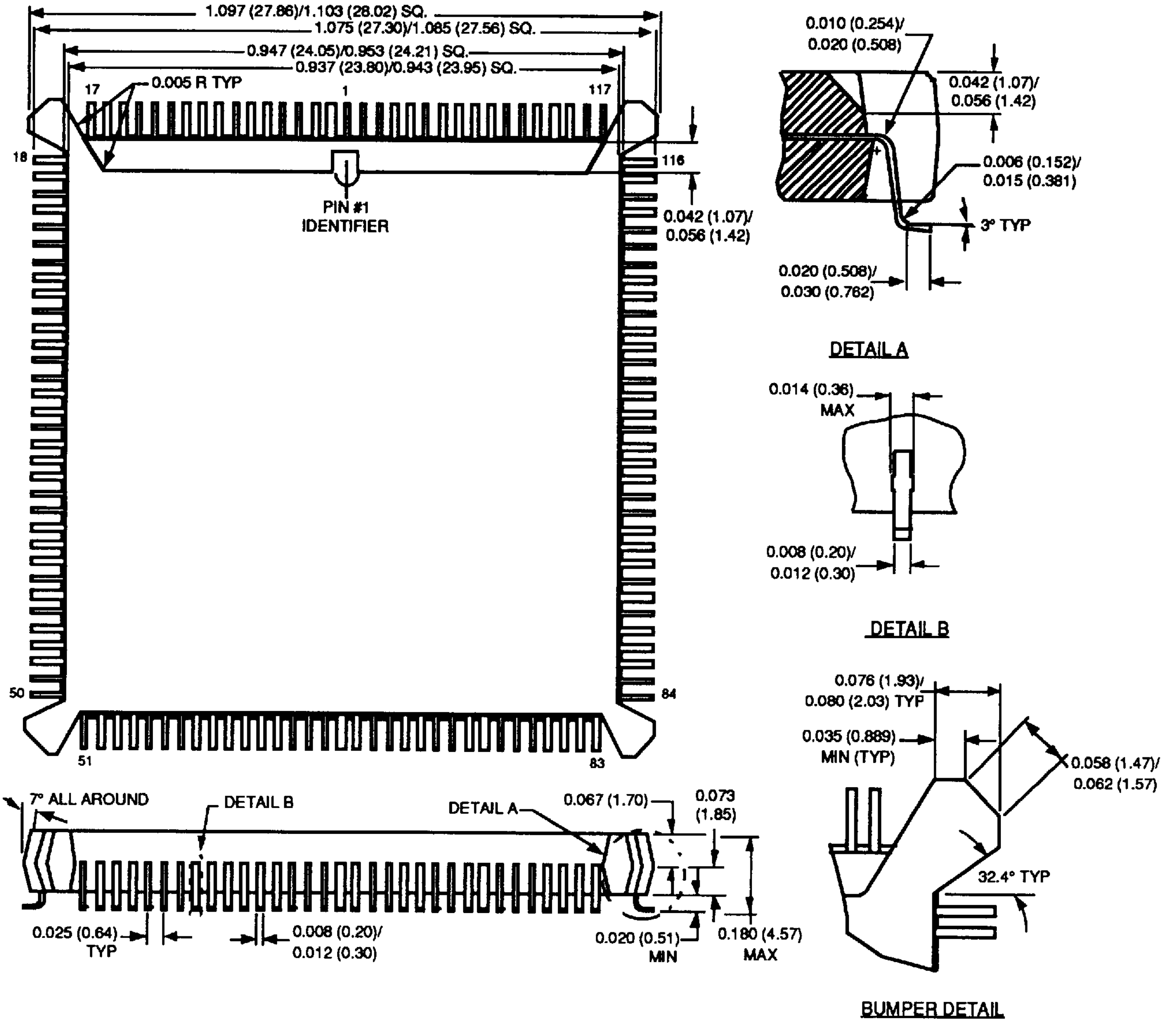
Abbreviated Reference*	Parameter	Min	Max	Unit
t116	Data Delay (bit 0 only) (low to valid)	—	33	ns
t117	Data Disable Delay (high to 3-state)	—	40	ns
t120	DOEN Valid Delay (high to valid)	—	25	ns
t121	Address Delay (bit 0 only) (low to valid)	—	33	ns
t122	Address Disable Delay (high to 3-state)	—	40	ns

* Capacitance load on ICK, OCK, DO, SYNC, and SADD = 100 pF.

11.0 Outline Diagram

132-Pin Plastic Quad Flat Pack

Dimensions are in inches and (millimeters).


1.097 (27.86)/1.103 (28.02) SQ.
1.075 (27.30)/1.085 (27.56) SQ.
0.947 (24.05)/0.953 (24.21) SQ.
0.937 (23.80)/0.943 (23.95) SQ.
0.005 R TYP
17
1
117
18
116
0.042 (1.07)/
0.056 (1.42)
PIN #1
IDENTIFIER
50
84
51
83
7° ALL AROUND
DETAIL B
DETAIL A
0.067 (1.70)
0.073
(1.85)
0.025 (0.64)
TYP
0.008 (0.20)/
0.012 (0.30)
0.020 (0.51)
MIN
0.180 (4.57)
MAX
0.010 (0.254)/
0.020 (0.508)
0.042 (1.07)/
0.056 (1.42)
0.006 (0.152)/
0.015 (0.381)
3° TYP
0.020 (0.508)/
0.030 (0.762)
DETAIL A
0.014 (0.36)
MAX
0.008 (0.20)/
0.012 (0.30)
DETAIL B
0.076 (1.93)/
0.080 (2.03) TYP
0.035 (0.889)
MIN (TYP)
0.058 (1.47)/
0.062 (1.57)
32.4° TYP
BUMPER DETAIL