Data Sheet
September 1995

AT&T Microelectronics

DSP1616-x11 Digital Signal Processor

1 Features

- Optimized for digital cellular applications with a bit manipulation unit for higher signal coding efficiency
- 38 ns and 33 ns instruction cycle
- Low power consumption:
  <15.0 mW/MIPS typical at 5 V
- Low-profile TQFP package (1.5 mm) available
- Pin and object code upward compatible with the DSP1616-x10; additional features available through target code reassembly
- Mask-programmable clock options: 1X, 2X, crystal oscillator, small signal, CMOS, and TTL
- Single-cycle squaring
- 16 x 16-bit multiplication and 36-bit accumulation in one instruction cycle
- Instruction cache for high-speed, program-efficient, zero-overhead looping
- Two external vectored interrupts and trap
- 12 Kword ROM (with secure option),
  2 Kword dual-port RAM
- Dual 15 Mbit/s serial I/O ports with multiprocessor capability—16-bit data channel, 8-bit protocol channel
- 8-bit PIO/host interface
- 8-bit control I/O interface
- 256 memory-mapped I/O ports, one internally decoded for glueless device interfacing
- Interrupt timer
- CMOS I/O levels
- *IEEE** P1149.1 test port (JTAG with boundary scan)
- Full-speed in-circuit emulation hardware development system on-chip
- Supported by DSP1616 software and hardware development tools

2 Description

Designed specifically for applications requiring low power dissipation in digital cellular systems, the DSP1616-x11 is a signal coding device that can be programmed to perform a wide variety of fixed-point signal processing functions. The device is based on the DSP1600 core with a bit manipulation unit for enhanced signal coding efficiency. The DSP1616 includes a mix of peripherals specifically intended to support processing-intensive but cost-sensitive applications in the area of digital mobile communications. In addition to 12 Kwords of ROM, the device contains 2 Kwords of dual-port RAM (DPRAM) that allow simultaneous access to two RAM locations in a single instruction cycle.

The DSP1616-x11, a pin-for-pin replacement to the DSP1616-x10, provides architectural enhancements including a single-cycle squaring feature for more efficient signal coding and an 8-bit control I/O for hardware flexibility. The DSP1616-x11 is also object code compatible with the DSP1616-x10.

The DSP1616 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 multiplication and 36-bit accumulation, or a 32-bit ALU operation, in one instruction cycle. A high-performance bit manipulation capability is provided, allowing single-cycle 36-bit barrel shifting, normalization, and bit-field insertion or extraction. These and other functions act to greatly enhance the efficiency of speech and channel coding operations as required in European, North American, and Japanese digital mobile systems.

The device is packaged in a 100-pin BQFP or a 100-pin TQFP and is available with 33 ns and 38 ns instruction cycle speeds.

* *IEEE* is a registered trademark of The Institute of Electrical and Electronics Engineers, Inc.

Table of Contents

Contents **Page**

Table of Contents (continued)

Contents **Page**

3 Pin Information

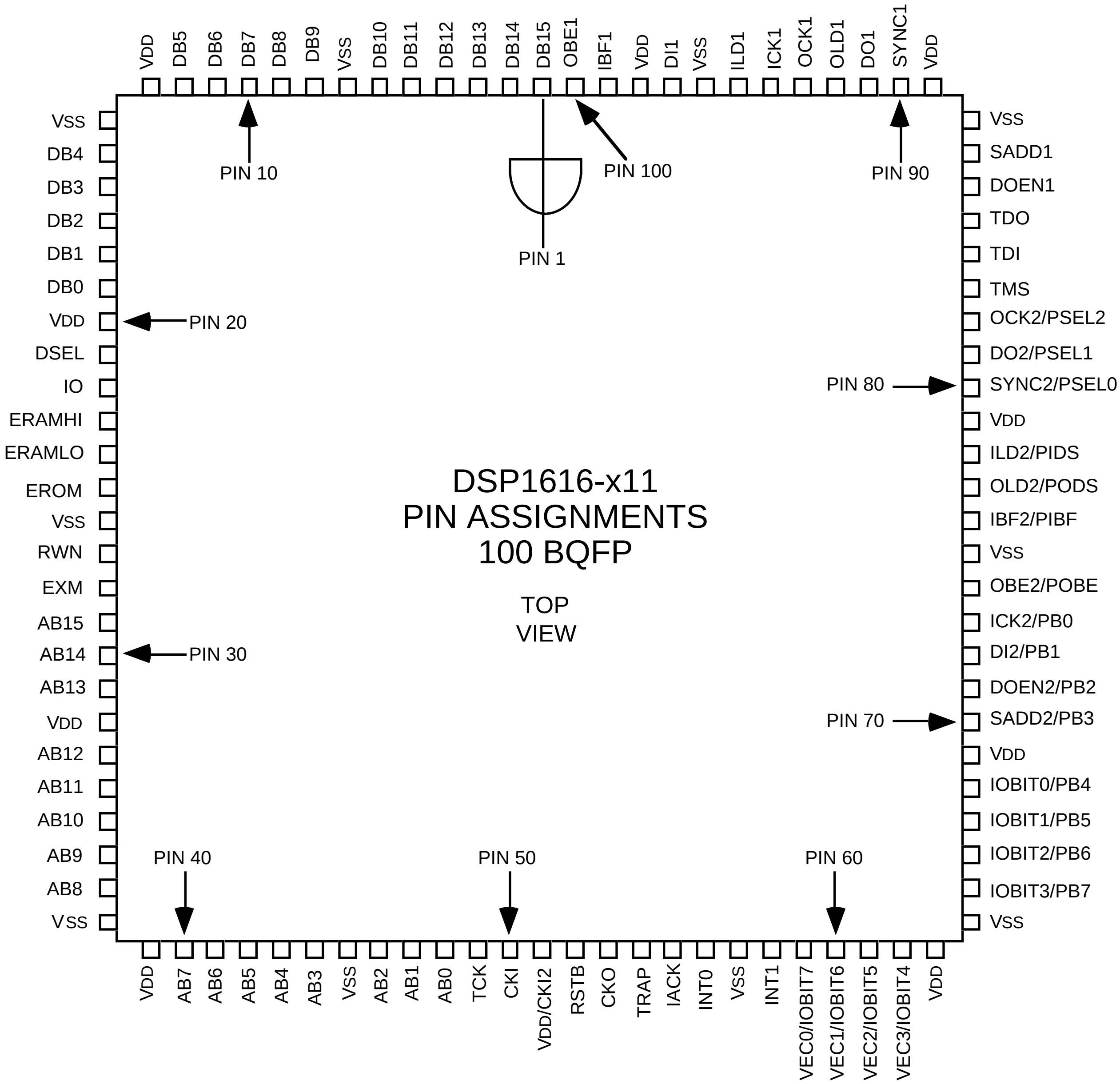


Figure 1. DSP1616 BQFP Pin Diagram

3 Pin Information (continued)

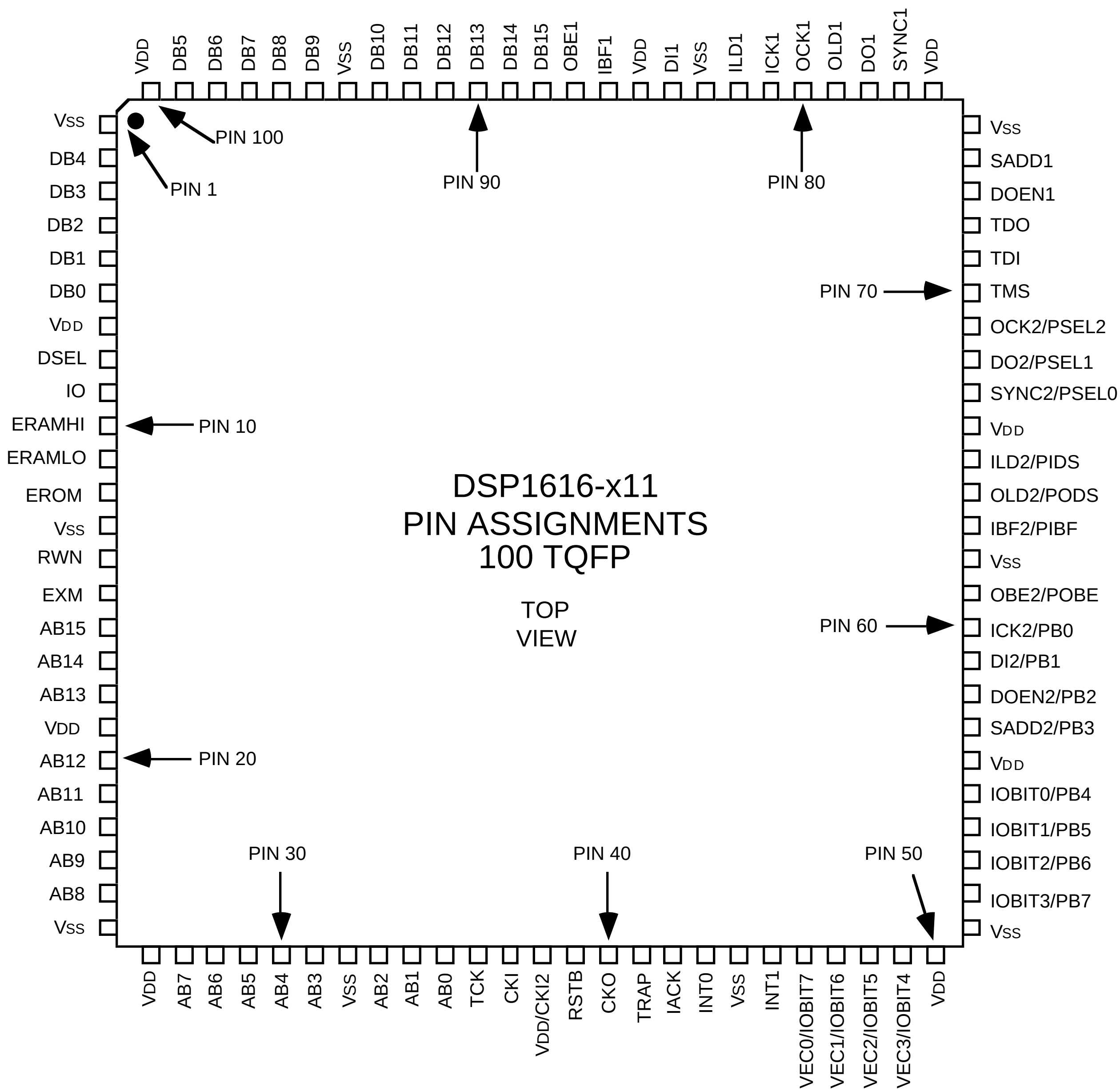


Figure 2. DSP1616-x11 TQFP Pin Diagram

Pin Information (continued)

Functional descriptions of pins 1—100 are found in Section 6, Signal Descriptions. The functionality of pins 50 and 51 (TQFP pins 37 and 38) are mask-programmable (see Section 7, Mask-Programmable Options).

Table 1. Pin Descriptions

<table>
<tr><th>BQFP Pin</th><th>TQFP Pin</th><th>Symbol</th><th>Type</th><th colspan="5">Name/Function</th></tr>
<tr><td>1, 2, 3, 4, 5, 6, 8, 9, 10, 11, 12, 15, 16, 17, 18, 19</td><td>88, 89, 90, 91, 92, 93, 95, 96, 97, 98, 99, 2, 3, 4, 5, 6</td><td>DB[15:0]</td><td>I/O*</td><td colspan="5">External Memory Data Bus DB[15:0]</td></tr>
<tr><td>21</td><td>8</td><td>DSEL</td><td>O†</td><td colspan="5">I/O Enable for Data Address 0x7F00.</td></tr>
<tr><td>22</td><td>9</td><td>IO</td><td>O†</td><td colspan="5">Data Address 0x7F00 to 0x7FFF I/O Enable.</td></tr>
<tr><td>23</td><td>10</td><td>ERAMHI</td><td>O†</td><td colspan="5">Data Address 0x8000 to 0xFFFF External RAM Enable.</td></tr>
<tr><td>24</td><td>11</td><td>ERAMLO</td><td>O†</td><td colspan="5">Data Address 0x2000 to 0x7EFF External RAM Enable.</td></tr>
<tr><td>25</td><td>12</td><td>EROM</td><td>O†</td><td colspan="5">Program Address External ROM Enable.</td></tr>
<tr><td>27</td><td>14</td><td>RWN</td><td>O†</td><td colspan="5">Read/Write Not.</td></tr>
<tr><td>28</td><td>15</td><td>EXM</td><td>I</td><td colspan="5">External ROM Enable.</td></tr>
<tr><td>29, 30, 31, 33, 34, 35, 36, 37, 40, 41, 42, 43, 44, 46, 47, 48</td><td>16, 17, 18, 20, 21, 22, 23, 24, 27, 28, 29, 30, 31, 33, 34, 35</td><td>AB[15:0]</td><td>O*</td><td colspan="5">External Memory Address Bus 15—0.</td></tr>
<tr><td>49</td><td>36</td><td>TCK</td><td>I</td><td colspan="5">JTAG Test Clock.</td></tr>
<tr><td colspan="3" rowspan="3"></td><td rowspan="3"></td><td colspan="5">Mask-Programmable Input Clock Option</td></tr>
<tr><td rowspan="2">TTL</td><td rowspan="2">CMOS</td><td rowspan="2">Small Signal</td><td colspan="2">Crystal</td></tr>
<tr><td>Oscillator</td><td>CMOS</td></tr>
<tr><td>50</td><td>37</td><td>CKI‡</td><td>I</td><td>CKI</td><td>CKI</td><td>V_{IN+}</td><td>XLO, 10 pF capacitor to V_{SS}</td><td>CKI</td></tr>
<tr><td>51</td><td>38</td><td>V_{DD}/CKI2‡</td><td>P or I</td><td>V_{DD}</td><td>Open</td><td>V_{IN-}</td><td>XHI, 10 pF capacitor to V_{SS}</td><td>Open</td></tr>
<tr><td>52</td><td>39</td><td>RSTB</td><td>I</td><td colspan="5">Reset Bar.</td></tr>
<tr><td>53</td><td>40</td><td>CKO</td><td>O†</td><td colspan="5">Processor Clock Output.</td></tr>
<tr><td>54</td><td>41</td><td>TRAP</td><td>I/O*</td><td colspan="5">Nonmaskable Program Trap/Breakpoint Indication.</td></tr>
<tr><td>55</td><td>42</td><td>IACK</td><td>O*</td><td colspan="5">Interrupt Acknowledge.</td></tr>
<tr><td>56</td><td>43</td><td>INT0</td><td>I</td><td colspan="5">Vectored Interrupt 0.</td></tr>
<tr><td>58</td><td>45</td><td>INT1</td><td>I</td><td colspan="5">Vectored Interrupt 1.</td></tr>
<tr><td>59</td><td>46</td><td>VEC0/IOBIT7</td><td>I/O*</td><td colspan="5">Vectored Interrupt Indication 0/Status/Control Bit 7.</td></tr>
<tr><td>60</td><td>47</td><td>VEC1/IOBIT6</td><td>I/O*</td><td colspan="5">Vectored Interrupt Indication 1/Status/Control Bit 6.</td></tr>
<tr><td>61</td><td>48</td><td>VEC2/IOBIT5</td><td>I/O*</td><td colspan="5">Vectored Interrupt Indication 2/Status/Control Bit 5.</td></tr>
<tr><td>62</td><td>49</td><td>VEC3/IOBIT4</td><td>I/O*</td><td colspan="5">Vectored Interrupt Indication 3/Status/Control Bit 4.</td></tr>
</table>

* 3-states when RSTB = 0, or by JTAG control.
† 3-states when RSTB = 0 and INT0 = 1. Output = 1 when RSTB = 0 and INT0 = 0.
‡ See Section 7, Mask-Programmable Options.

Pin Information (continued)

Functional descriptions of pins 1—100 are found in Section 6, Signal Descriptions.

Table 1. Pin Descriptions (continued)

BQFP Pin	TQFP Pin	Symbol	Type	Name/Function
65	52	IOBIT3/PB7	I/O*	**Status/Control Bit 3/PIO Data Bus Bit 7.**
66	53	IOBIT2/PB6	I/O*	**Status/Control Bit 2/PIO Data Bus Bit 6.**
67	54	IOBIT1/PB5	I/O*	**Status/Control Bit 1/PIO Data Bus Bit 5.**
68	55	IOBIT0/PB4	I/O*	**Status/Control Bit 0/PIO Data Bus Bit 4.**
70	57	SADD2/PB3††	I/O*	**SIO2 Multiprocessor Address/PIO Data Bus Bit 3.**
71	58	DOEN2/PB2	I/O*	**SIO2 Data Output Enable/PIO Data Bus Bit 2.**
72	59	DI2/PB1	I/O*	**SIO2 Data Input/PIO Data Bus Bit 1.**
73	60	ICK2/PB0	I/O*	**SIO2 Input Clock/PIO Data Bus Bit 0.**
74	61	OBE2/POBE	O*	**SIO2 Output Buffer Empty/PIO Output Buffer Empty.**
76	63	IBF2/PIBF	O*	**SIO2 Input Buffer Full/PIO Input Buffer Full.**
77	64	OLD2/PODS	I/O*	**SIO2 Output Load/PIO Output Data Strobe.**
78	65	ILD2/PIDS	I/O*	**SIO2 Input Load/PIO Input Data Strobe.**
80	67	SYNC2/PSEL0	I/O*	**SIO2 Multiprocessor Synchronization/Peripheral Select 0.**
81	68	DO2/PSEL1	I/O*	**SIO2 Data Output/Peripheral Select 1.**
82	69	OCK2/PSEL2	I/O*	**SIO2 Output Clock/Peripheral Select 2.**
83	70	TMS	I§	**JTAG Test Mode Select.**
84	71	TDI	I§	**JTAG Test Data Input.**
85	72	TDO	O**	**JTAG Test Data Output.**
86	73	DOEN1	I/O*	**SIO1 Data Output Enable.**
87	74	SADD1††	I/O*	**SIO1 Multiprocessor Address.**
90	77	SYNC1	I/O*	**SIO1 Multiprocessor Synchronization.**
91	78	DO1	O*	**SIO1 Data Output.**
92	79	OLD1	I/O*	**SIO1 Output Load.**
93	80	OCK1	I/O*	**SIO1 Output Clock.**
94	81	ICK1	I/O*	**SIO1 Input Clock.**
95	82	ILD1	I/O*	**SIO1 Input Load.**
97	84	DI1	I	**SIO1 Data Input.**
99	86	IBF1	O*	**SIO1 Input Buffer Full.**
100	87	OBE1	O*	**SIO1 Output Buffer Empty.**
7, 14, 26, 38, 45, 57, 64, 75, 88, 96	94, 1, 13, 25, 32, 44, 51, 62, 75, 83	VSS	P	**Ground.**
13, 20, 32, 39, 63, 69, 79, 89, 98	100, 7, 19, 26, 50, 56, 66, 76, 85	VDD	P	**5 V Supply.**

* 3-states when RSTB = 0, or by JTAG control.
† 3-states when RSTB = 0 and INT0 = 1. Output = 1 when RSTB = 0 and INT0 = 0.
‡ See Section 7, Mask-Programmable Options.
§ Pull-up devices on input.
** 3-states by JTAG control.
†† For SIO multiprocessor applications, add 5 kΩ external pull-up resistors to SADD1 and/or SADD2 for proper initialization.

4 Hardware Architecture

The DSP1616 device is a 16-bit fixed-point digital signal processor (DSP) that is upward object code compatible with the *WE*® DSP16A and DSP1610, except for specific I/O configurations (see Table 36, **ioc** Register). The DSP1616 consists of a DSP1600 core together with on-chip memory and peripherals. Many added architectural features give the DSP1616 high program efficiency for signal coding applications.

4.1 DSP1616 Architectural Overview

Figure 3 shows a block diagram of the DSP1616 which consists of a number of modules. The DSP1616 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 DSP16A. 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 two banks of zero wait-state memory. Each bank consists of 1K 16-bit words and has separate address and data ports to the instruction/coefficient and data memory spaces. A program can reference memory from either space at any time, transparently and without restrictions. The DSP1600 core automatically performs the required 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 port access first, followed by the instruction/coefficient port 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 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 DSP1616 contains 12K 16-bit words of zero wait-state mask-programmable ROM for program and fixed coefficients.

External Memory Multiplexer (EMUX)

The EMUX is used to connect the DSP1616 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 extends 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 shuffled with the data in the main accumulators. Flags returned by the BMU mesh seamlessly with the DSP1600 conditional instructions.

Timer

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

Bit Input/Output (BIO)

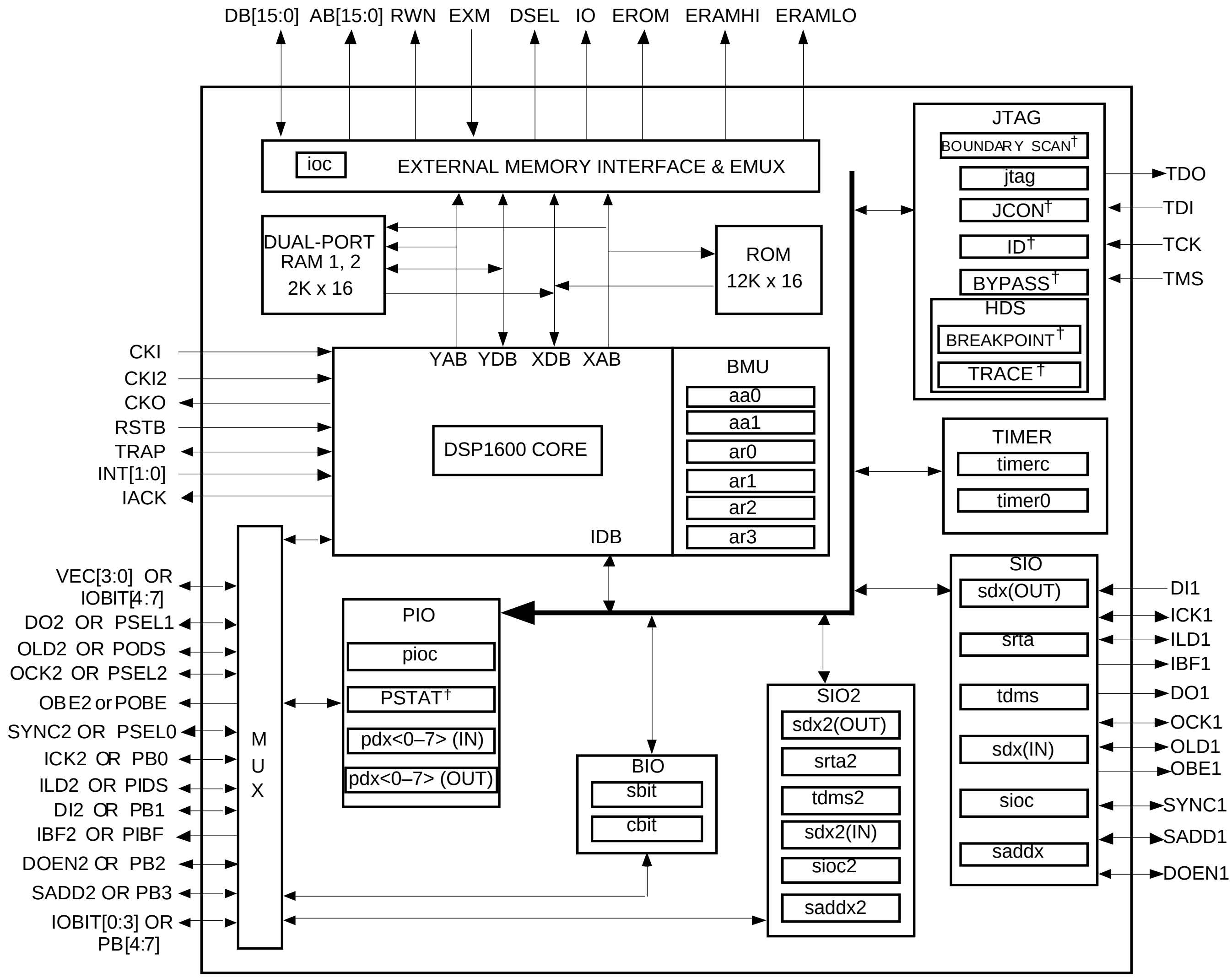
The BIO provides convenient and efficient monitoring and control of eight individually configurable pins. When configured as outputs, the pins can be individually set, cleared, or toggled. When configured as inputs, individual pins or combinations of pins can be tested for patterns. Flags returned by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

Serial Input/Output Units (SIO and SIO2)

SIO and SIO2 offer asynchronous, full-duplex, double-buffered channels that operate at up to 15 Mbits/s (for a 33 ns instruction cycle in a non-multiprocessor configuration) and easily interface 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 serial I/O ports with few, if any, additional components. SIO2 is identical to SIO, with the exception that SIO2 interrupts are not DSP16A compatible.

An 8-bit serial protocol channel may be transmitted in addition to the address of the called processor in multiprocessor mode. This feature is useful for transmitting high-level framing information or for error detection and correction. SIO2 and BIO are pin-multiplexed with the PIO.

Hardware Architecture (continued)



† These registers are accessible through the external pins only.

Figure 3. DSP1616 Block Diagram

Hardware Architecture (continued)

Table 2. DSP1616 Block Diagram Legend

Symbol	Name
aa<0—1>	Alternate Accumulators.
ar<0—3>	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[1, 2]	JTAG Configuration Registers.
jtag	16-bit Serial/Parallel Register.
pdx<0—7> (in)	Parallel I/O Data Transmit Input Registers 0—7.
pdx<0—7> (out)	Parallel I/O Data Transmit Output Registers 0—7.
PIO	Parallel Input/Output Unit.
pioc	Parallel I/O Control Register.
PSTAT	Parallel I/O Status Register.
ROM	Internal ROM 12 Kwords.
saddx	Multiprocessor Protocol Register.
saddx2	Multiprocessor Protocol Register for SIO2.
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.
srta2	Serial Receive/Transmit Address Register for SIO2.
tdms	Serial I/O Time-division Multiplex Signal Control Register.
tdms2	Serial I/O Time-division Multiplex Signal Control Register for SIO2.
TIMER	Programmable Timer.
timer0	Timer Running Count Register.
timerc	Timer Control Register.
TRACE	Program Discontinuity Trace Buffer.
XAB	Program Memory Address Bus.
XDB	Program Memory Data Bus.
YAB	Data Memory Address Bus.
YDB	Data Memory Data Bus.

Hardware Architecture (continued)

Parallel Input/Output Unit (PIO)

The PIO is an 8-bit parallel I/O unit which can interface to an 8-bit bus containing other AT&T DSPs (e.g., DSP16A or DSP1610), microprocessors, or peripheral I/O devices. The port data rate depends upon the instruction cycle rate. A 33 ns instruction cycle allows the PIO to support data rates up to 15 Mbytes/s.

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 DSP1616 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 (see Section 4.8, Parallel I/O Unit (PIO)).

Pin Multiplexing

In order to allow flexible device interfacing while maintaining a low package pin count, the DSP1616 multiplexes 16 package pins between BIO, PIO, VEC[3:0], and SIO2.

Upon reset, the vectored interrupt indication signals, VEC[3:0], are connected to the package pins while IOBIT[4:7] are disconnected. Setting bit 12, EBIOH, of the **ioc** register connects IOBIT[4:7] to the package pins and disconnects VEC[3:0].

Upon reset, the parallel I/O (PIO) is connected to the package pins while the second serial port (SIO2) and IOBIT[0:3] are disconnected. Setting bit 10, ESIO2, of the **ioc** register connects the SIO2 and IOBIT[0:3] and disconnects the PIO. The currently selected I/O unit(s) must be set to passive mode before changing the I/O pin configuration.

Powerdown

Many applications, such as portable cellular terminals, require low standby power. Setting the AWAIT bit in the **alf** register puts the processor into a power-saving standby mode until an interrupt occurs. The processor then resumes normal operation.

Hardware Development System (HDS) Module

The on-chip HDS performs instruction breakpointing and branch tracing at full speed without additional off-chip hardware. Using the JTAG port, the breakpointing is set up, and the trace history is read back. The port works in conjunction with the HDS code in the on-chip ROM and the hardware and software in a remote computer. The HDS code must be linked to the user's application code and reside in the first 4 Kwords of ROM. The on-chip HDS cannot be used with the secure ROM masking option (see Section 7.2, ROM Security Options).

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 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 4 shows a block diagram of the DSP1600 core.

System Cache and Control Section (SYS)

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

The cache loop iteration count can be specified at run time under program control as well as at assembly time.

Hardware Architecture (continued)

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 accumulator data can be directly loaded from, or stored to, memory in two 16-bit words with optional saturation on overflow. The arithmetic logic 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 of the DSP16A and is fully compatible with the DSP1610 instruction set. See Section 5.1 for more information on the instruction set.

The user also has access to two additional DAU registers. The **psw** register contains status information from the DAU (see Table 25, Processor Status Word Register). The arithmetic control register, **auc**, is used to configure some of the features of the DAU (see Table 26) including single-cycle squaring. The **auc** register is cleared by reset.

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 two 1 Kword banks of on-chip DPRAM or three external data memory segments. Up to 56 Kwords of off-chip RAM are addressable. One individual address in an external data memory segment (the IO segment) is decoded to provide the DSEL output.

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 memory space and also specifies one of 16 registers for the source or destination.

X Space Address Arithmetic Unit (XAAU)

The XAAU supports high-speed, register-indirect, instruction/coefficient memory addressing with postmodification of the register. The 16-bit **pt** register is used for addressing coefficients. The signed register **i** holds 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 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), two 1K banks of DPRAM, and external ROM.

The locations of these memory segments depend upon the memory map 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 ROM (see Section 7, Mask-Programmable Options).

4.3 Interrupts and Trap

The DSP1616 supports prioritized, vectored interrupts and a trap. The device has nine internal hardware sources of program interrupt and two 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. The **icall** instruction is reserved for use by the hardware development system. Each of these sources of interrupt and trap has a unique vector address and priority assigned to it. In addition, for compatibility with the DSP16A, four of the internal sources and one external pin can be assigned a common vector value.

Vectored interrupts are enabled in the **inc** register (see Table 28, Interrupt Control (**inc**) Register) and monitored in the **ins** register (see Table 29, Interrupt Status (**ins**) Register). When the DSP1616 goes into an interrupt or trap service routine, the IACK pin is asserted. In addition, pins VEC[3:0] encode which interrupt/trap is being serviced. Table 4 details the encoding used for VEC[3:0].

Hardware Architecture (continued)

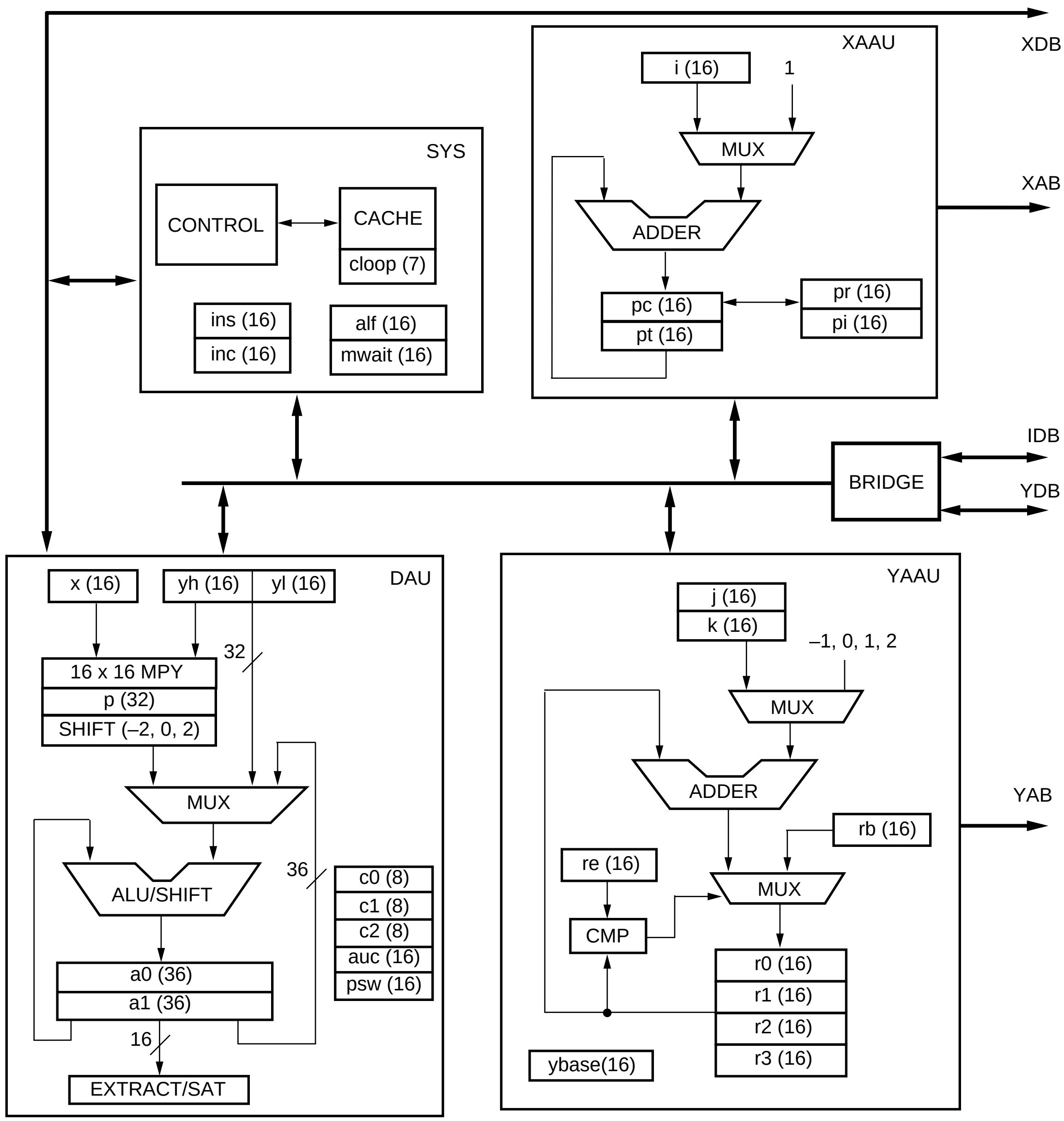


Figure 4. DSP1600 Core Block Diagram

Hardware Architecture (continued)

Table 3. DSP1600 Core Block Diagram Legend

Symbol	Name
16 x 16 MPY	16-bit x 16-bit Multiplier.
a0—a1	Accumulators 0 and 1 (16-bit halves specified as **a0**, **a0l**, **a1**, and **a1l**)*.
alf	Await, Lowpr, Flags.
ALU/SHIFT	Arithmetic Logic Unit/Shifter.
auc	Arithmetic Unit Control.
c0—c2	Counters 0—2.
cloop	Cache Loop Count.
CMP	Comparator.
DAU	Digital Arithmetic Unit.
i	Increment Register for the X Address Space.
IDB	Internal Data Bus.
inc	Interrupt Control.
ins	Interrupt Status.
j	Increment Register for the Y Address Space.
k	Increment Register for the Y Address Space.
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.
r0—r3	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	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 the execution of an interruptible instruction. 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 that 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 machine state 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) 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 DSP16A, 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 DSP16A programs, the 1616 **pioc** register contains enables and status bits for the INT0 pin and the PIDS, PODS, IBF, and OBE interrupts (see Table 27, Parallel I/O Control (**pioc**) Register). As in the DSP16A, 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 31 and 32 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. Note that neither software interrupts nor traps can be masked.

The occurrence of an interrupt that 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. If a masked interrupt occurs, it is latched in the **ins** register, but the interrupt is not taken. When unmasked, this latched interupt will initiate automatic processor interrupt action. The status of the 11 interrupt sources is readable in the **ins** register even if the interrupt is masked in the **inc** register. See the *DSP1616 Digital Signal Processor Information Manual* for a thorough description of the interrupts.

Signaling Interrupt Service Status

Five pins of the DSP1616 are devoted to signaling interrupt service status. The IACK pin goes high while any interrupt or user 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 interrupt or user trap is being serviced. Table 4 contains the encodings used by each interrupt.

Traps due to HDS breakpoints have no effect on either the IACK or VEC[3:0] pins. They, instead, show the interrupt state or interrupt source of the DSP when the trap occurred.

Clearing Interrupts

The PIO interrupts (PIDS and PODS) are cleared, as in the DSP16A, 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 DSP16A, 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.

Four of the vectored interrupts are cleared by writing to the **ins** register. Writing a 1 to the INT0, INT1, TIME, or EMUXBOTH bits in the **ins** will cause the corresponding interrupt status bit to be cleared to a logic 0. The status bit for these vectored interrupts is also cleared when the **ireturn** instruction is executed, leaving set any other vectored interrupts that are pending.

Traps

The TRAP pin of the DSP1616 is a bidirectional signal. At reset, it is configured as an input to the processor. Asserting the TRAP pin will force a user trap.

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
TIME	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	—*	breakpoint, jtag, or pin
TRAP from User	0x46	19 = highest	0x7	pin

* Traps due to HDS breakpoints have no effect on the VEC[3:0] pins.

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.

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 may be overwritten. Specifically, it is not possible to return to an interrupt service routine from a user trap (0x46) service routine. Continuing program execution when a trap occurs during a cache loop is also not possible.

Hardware breakpoints should not be set and single-stepping should be avoided while using the hardware development system with the X=Y= bit of the **auc** register set for single-cycle squaring operations.
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, if present, 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 or Sleep Mode)

Setting the AWAIT bit (bit 15) of the **alf** register (**alf = 0x8000**) causes the processor to go into a power-saving standby or sleep 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 **nop** instructions should be programmed after the AWAIT bit is set. 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.

The AWAIT bit should be set from within the cache if the code that is executing resides in external ROM where more than one wait state has been programmed. This insures that an interrupt will not disturb the device from completely entering the sleep state.

For additional power savings, set **ioc = 0x0180** and **timerc = 0x0040** in addition to setting **alf = 0x8000**. This will hold the CKO pin low and shut down the timer and prescaler (see Table 36 and Table 30).

4.4 Memory Maps and Wait-States

The DSP1600 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 DSP1616 contains 12 Kwords of ROM (IROM) and 2 Kwords 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 two 1K banks of dual-port RAM (i.e., RAM1 and RAM2). 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 IROM from 0x0 in MAP1 to 0x2000 in MAP3, and EROM from 0x0 in MAP2 to 0x2000 in MAP4.

The second parameter is the value at reset of the EXM pin (pin 28 or pin 15, depending upon the package type). EXM determines whether the internal 12 Kword 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 IROM starting at 0x0, and two 1 Kword banks of DPRAM starting at 0x3000. External ROM (EROM) is accessed for addresses above 0x5000. This map is used if DSP1616 has EXM 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 DSP1616 EXM is high at reset, the LOWPR parameter is programmed to zero, and an HDS trap is not in progress.

MAP3 has the two 1 Kword banks of DPRAM located starting at 0x0. The 12 Kword IROM starts at address 0x2000. This map is used if DSP1616 EXM is low at reset, the LOWPR bit is programmed to 1, 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 at reset. 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 DSP1616 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

<table>
<tr><th>Decimal Address</th><th>Address in PC, pt, pi, pr</th><th>MAP1*
EXM = 0
LOWPR† = 0</th><th>MAP2
EXM = 1
LOWPR = 0</th><th>MAP3‡
EXM = 0
LOWPR = 1</th><th>MAP4
EXM = 1
LOWPR = 1</th></tr>
<tr><td>0</td><td>0x0000</td><td rowspan="4">IROM</td><td rowspan="4">EROM</td><td>RAM1</td><td>RAM1</td></tr>
<tr><td>1K</td><td>0x0400</td><td>RAM2</td><td>RAM2</td></tr>
<tr><td>2K</td><td>0x0800
0x0C00
0x1000
0x1400
0x1800
0x1C00</td><td>Reserved</td><td>Reserved</td></tr>
<tr><td>8K</td><td>0x2000
0x2400
0x2800
0x2C00</td><td rowspan="4">IROM</td><td rowspan="5">EROM</td></tr>
<tr><td>12K</td><td>0x3000</td><td>RAM1</td><td>RAM1</td></tr>
<tr><td>13K</td><td>0x3400</td><td>RAM2</td><td>RAM2</td></tr>
<tr><td>14K</td><td>0x3800
0x3C00
0x4000
0x4400
0x4800
0x4C00</td><td>Reserved</td><td>Reserved</td></tr>
<tr><td>20K

64K – 1</td><td>0x5000
• • •
up to 0xFFFF</td><td>EROM</td><td>EROM</td><td>EROM</td></tr>
</table>

* 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.

Hardware Architecture (continued)

Data Memory Mapping

Table 6. Data Memory Map

Decimal Address	Address in r0, r1, r2, r3	Segment
0	0x0000 0x03FF	RAM1
1K	0x0400 0x07FF	RAM2
2K	0x0800 0x1FFF	Reserved
8K 32K – 257	0x2000 0x7EFF	ERAMLO
32K – 256 32K – 1	0x7F00 0x7FFF	IO
32K 64K – 1	0x8000 0xFFFF	ERAMHI

On the data memory side (Table 6), the two 1K banks of dual-port RAM (RAM1 and RAM2) are located starting at address 0. Addresses from 0x2000 to 0x7EFF reference the low external data RAM segment (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 34). 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.

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 DSP1616 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 the first address of these 256 external memory locations, 0x7F00, is addressed, the associated predecoded enable DSEL is also asserted. The assertion level of DSEL is programmable via the **ioc** register (see Table 36).

If a data memory location with an address greater than or equal to 0x8000 is addressed, ERAMHI is asserted low. When the external instruction/coefficient memory is addressed, EROM is asserted low.

The flexibility provided by the programmable options of the external memory interface (see Table 34, **mwait** Register and Table 36, **ioc** Register) allows the DSP1616 to interface gluelessly with a variety of commercial memory chips.

Hardware Architecture (continued)

Each of the four external memory segments, ERAMLO, IO, ERAMHI, and EROM, has a number of wait-states which are programmable (from 0 to 15) by writing to the **mwait** register. 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 in the appropriate field of the **mwait** register.

When writing to external memory, the RWN pin goes low for the external cycle. The external data bus, DB[15:0], is driven by the DSP1616 starting halfway through the cycle. The data driven on the external data bus is automatically held after the cycle unless an external read cycle immediately follows.

The DSP1616 allows writing into external instruction/coefficient memory. By setting bit 11, WEROM, of the **ioc** register, writing to (or reading from) data memory or memory-mapped I/O asserts the EROM strobe instead of ERAMLO, IO, or ERAMHI. Therefore, with WEROM set, EROM appears in both Y space (replacing ERAM) and X space, in its normal position. When WEROM is active, DSEL will not be asserted.

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 memory segment and device enable pins is inactive-high, active-low. If bit 6 of the **ioc** register is set, the definition of the device enable pin, DSEL, will be inactive-low, active-high.

The leading edge of the memory segment enables or the DSEL can be delayed by approximately one-half a CKO period by programming the **ioc** register (see Table 36). This is used to avoid a situation in which two devices drive the data bus simultaneously.

Bits 7 and 8 of the **ioc** register select whether the CKO pin provides the free-running unstretched cycle clock, the wait-stated clock, a high level, or a low level (see Table 36). The high-to-low transition of the wait-stated clock is synchronized with the high-to-low transition of the free-running clock (CKO).

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**. In one instruction cycle, 36-bit data can be shuffled, or swapped, between one of the main accumulators and one of the alternate accumulators. The **ar**<**0**—**3**> registers are 16-bit registers that control the operations of the BMU. They store a value that determines the amount of shift or the width and offset fields for bit extraction or insertion. Certain operations in the BMU set flags in the DAU **psw** register and the **alf** register (see Table 25, Processor Status Word (**psw**) Register and Table 33, **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 thorough description of the BMU, see the *DSP1616 Digital Signal Processor Information Manual*.

4.7 Serial I/O Units (SIOs)

The serial I/O ports on the DSP1616 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. SIO and SIO2 are identical. 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. 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 21, Serial I/O Control Registers (**sioc** and **sioc2**)). A multiprocessor I/O configuration is supported. This feature allows up to eight DSP1616 devices to be connected together on an SIO port without requiring external glue logic.

Hardware Architecture (continued)

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 (DO1 to DI1 and DO2 to DI2). To exercise loopback, the SIO clocks (ICK1, ICK2, OCK1, and OCK2) should either all be in the active mode, 16-bit condition, or each pair should be driven from one external source in passive mode. Similarly, pins ILD1 (ILD2) and OLD1 (OLD2) must both be in active mode or tied together and driven from one external frame clock in passive mode. During loopback, DO1, DO2, DI1, DI2, ICK1, ICK2, OCK1, OCK2, ILD1, ILD2, OLD1, OLD2, SADD1, SADD2, SYNC1, SYNC2, DOEN1, and DOEN2 are 3-stated.

Programmable Modes

Programmable modes of operation for the SIO and SIO2 are controlled by the serial I/O control registers (**sioc** and **sioc2**). These registers, shown in Table 21, 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 DSP1616 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. Input and output data formats can be independently configured.

Multiprocessor Mode

The multiprocessor mode allows up to eight processors (DSP1616, DSP1610, or DSP16A) to be connected together to provide data transmission among any of the DSPs in the system. Either SIO port (SIO or SIO2) may be independently used for the multiprocessor mode. The multiprocessor interface is a four-wire interface, consisting of a data channel, a address/protocol channel, a transmit/receive clock, and a sync signal (see Figure 5). The DI1 and DO1 pins of all the DSPs are connected to transmit and receive the data channel. The SADD1 pins of all the DSPs are connected to transmit and receive the address/protocol channel. ICK1 and OCK1 should be tied together and driven from one source. The SYNC1 pins of all the DSPs are connected.

In the configuration shown in Figure 5, the master DSP (DSP0) generates active SYNC1 and OCK1 signals while the slave DSPs use the SYNC1 and OCK1 signals in passive mode to synchronize operations. In addition, all DSPs must have their ILD1 and OLD1 signals in active mode.

While ILD1 and OLD1 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 SYNC1, ILD1, and OLD1 is derived from OCK1 (see Table 21). With this configuration, all DSPs should use ICK1 (tied to OCK1) in passive mode to avoid conflicts on the clock (CK) line (see the *DSP1616 Digital Signal Processor Information Manual* for more information).

Four registers (per SIO) configure the multiprocessor mode. They are the time-division multiplexed slot register (**tdms** or **tdms2**), the serial receive and transmit address register (**srta** or **srta2**), the serial data transmit register (**sdx** or **sdx2**), and the multiprocessor serial address/protocol register (**saddx** or **saddx2**).

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 24, 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 DSP1616 device to transmit in more than one time slot. This procedure is useful for multiprocessor systems with less than eight DSP1616 devices when a higher bandwidth is necessary between certain devices in that system. The DSP operating during time slot 0 also drives SYNC1.

In order to prevent multiple bus drivers, only one DSP can be programmed to transmit in a particular time slot. In addition, it is important to note that the address/protocol channel is 3-stated in any time slot which is not being driven. Therefore, to prevent spurious inputs, the address/protocol channel should be pulled up to V_{DD} with a 5 kΩ resistor, or it should be guaranteed that the bus is driven in every time slot. (If the SYNC1 signal is externally generated, then this pull-up is required for correct initialization.)

Hardware Architecture (continued)

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 0s.

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%.

Using SIO2

The SIO2 functions the same as the SIO. Please refer to Pin Multiplexing in Section 4.1 for a description of pin multiplexing of BIO, PIO, VEC[3:0], and SIO2.

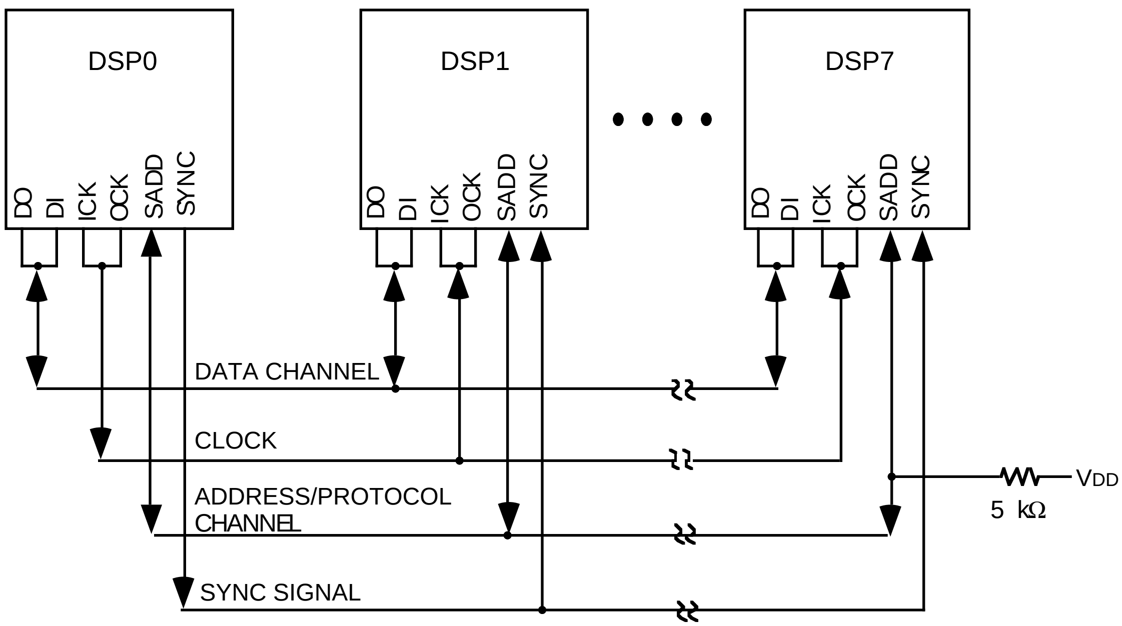


Figure 5. Multiprocessor Communication and Connections

Hardware Architecture (continued)

4.8 Parallel I/O Unit (PIO)

The DSP1616 has an 8-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 provided in the PIO unit. Although there is only one physical PIO port, there are eight logical PIO ports, **pdx**<**0**—**7**>. The eight logical ports are distinguished by the state of the peripheral select pins, PSEL[2:0] (see Table 7 for the functionality of PSEL[2:0]).

The data path of the PIO consists of an 8-bit input buffer, **pdx**(in), and an 8-bit output buffer, **pdx**(out). When PIDS and PODS are in passive mode, two pins, parallel input buffer full (PIBF) and parallel output buffer empty (POBE), indicate the state of the buffers. The **pdx**(in) register is shadowed in some modes to allow the PIO to accept data on an interrupt 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 register (**pioc**, see Table 27), and the PIO status register (PSTAT, see Table 8). 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 independently in either of these modes.

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 PD[7:0] pins as well as PODS and PIDS; PODS now drives PIDS internally. Thus, the PIO loopback feature allows a PIO write to automatically initiate a PIDS strobe. If the **pioc** register is configured for active output and passive input with PIOLBC set, **pdx**<**0**—**7**> can then be used as a temporary storage register.

Active Mode Operation

In active mode, PIDS is an output that indicates when data can be put on the bus during a read. Likewise, PODS is an output that 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. This allows 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 DSP1616, indicating which of the eight channels are being accessed. If either input or output is passive, some of these pins become inputs and serve different purposes.

Passive Mode Operation

In passive mode, the DSP1616 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** register, the assertion of PIDS and PODS by an external device causes the DSP1616 device to recognize an interrupt. This provides functional synchronization between the DSP1616 and an external device. The PIO interrupts can be configured in vectored mode (via the **inc** register) or DSP16A 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**<**0**—**3**> channel number (i.e., **pdx**<**4**—**7**> will alias into **pdx**<**0**—**3**>). PSEL2 is an input that, when low, enables PIDS.

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

When PIDS and PODS are both passive, PSEL2 is an input that, when low, enables PODS and PIDS. PSEL1 is an input that, when high, causes the low 8 bits of the PSTAT register (instead of data) to be output to PB[7:0]. PSEL0 is an output that 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 DSP1616 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 DSP1616 ignores any activity of the passive PIDS and/or PODS. If a DSP1616 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)
P	passive	Input (enable active-low)	Output (psel1)	Output (psel0)
Passive	Active	Input (enable active-low)	Input (status/data)	Output (psel0)
Passive	Passive	Input (enable active-low)	Input (status/data)	Output (pibf or pobe)

Table 8. PSTAT Register as Seen on PB[7:0]

Bit	7	6	5	4	3	2	1	0
Field	RESERVED					LPIDS*	PIBF†	POBE‡

* LPIDS = 1 indicates active mode input.
† PIBF = 1 indicates input buffer full.
‡ POBE = 1 indicates output buffer empty.

Pin Multiplexing

Please refer to Pin Multiplexing in Section 4.1 for a description of pin multiplexing of BIO, PIO, VEC[3:0], and SIO2.

4.9 Bit Input/Output Unit (BIO)

The BIO 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 current values (VALUE[7:0]) of the eight bidirectional pins IOBIT[7:0]. The upper half of the **sbit** register (DIREC[7:0]) controls the direction of each of the pins. A logic 1 configures the corresponding pin as an output; a logic 0 configures it as an input. The upper half of the **sbit** register 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/PAT[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 pin has been configured to be an output, the meanings are MODE and DATA. For an 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 that 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 33).

Table 9. BIO Operations

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

* $0 \leq n \leq 7$.

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

Pin Multiplexing

Please refer to Pin Multiplexing in Section 4.1 for a description of pin multiplexing of BIO, PIO, VEC[3:0], and SIO2.

Hardware Architecture (continued)

4.10 Timer

The interrupt timer is composed of the timer control register, the **timer0** register, the prescaler, and the counter 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 counter reload value (or period register) and to set the initial value of the counter. The prescaler slows the clock to the timer by a number of binary divisors to allow for a wide range of interrupt delay periods.

The counter is a 16-bit down counter that can be loaded with an arbitrary number from software. It counts down to 0 at the clock rate provided by the prescaler. Upon reaching 0 count, a vectored interrupt to program address 0x10 is issued to the DSP1616, provided that the interrupt is enabled (bit 8 of **inc** and **ins** registers). The counter will then either wait in an inactive 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.

When RELOAD is 0, the counter counts down from its initial value to 0, interrupts the DSP1616, and then stops, remaining inactive until another value is written to the **timer0** register. Writing to the **timer0** register causes both the counter and the period register to be written with the specified 16-bit number. When RELOAD is 1, the counter counts down from its initial value to 0, interrupts the DSP1616, automatically reloads the specified initial value from the period register into the counter, and repeats indefinitely. This provides for either a single timed interrupt event or a regular interrupt clock of arbitrary period.

The timer can be stopped and started by software, and can be reloaded with a new period at any time. Its count value, at the time of the read, can also be read by software. Due to pipeline stages, stopping and starting the timer may result in one inaccurate count or prescaled period. When the DSP1616 is reset, the bottom 6 bits of the **timerc** register and the **timer0** register and counter are initialized to 0. This sets the prescaler to CKO/2*, turns off the reload feature, disables timer counting, and initializes the timer to its inactive state. The act of resetting the chip does not cause a timer interrupt. Note that the period register is not initialized on reset.

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.

* Frequency of CKO/2 is equivalent to either CKI/2 for 1X input clock option or CKI/4 for 2X input clock option. See Section 7, Mask-Programmable Options.

The PRESCALE field of the **timerc** register selects one of 16 possible clock rates for the timer input clock (see Table 30, **timerc** Register).

Setting the DISABLE bit of the **timerc** register to a logic 1 shuts down the timer and the prescaler for power savings. The DISABLE bit is cleared by writing a 0 to the **timerc** register to restore the normal operating mode.

4.11 JTAG Test Port

The DSP1616 uses a JTAG/*IEEE* 1149.1 standard four-wire test port for self-test and hardware emulation. There is no separate TRST input pin.

An instruction register, a boundary-scan register, a bypass register, and a device identification register have been implemented. The device identification register coding for the DSP1616 is shown in Table 35. The instruction register (IR) is 4 bits long. The instruction for accessing the device ID is 0xE (1110).

The behavior of the instruction register is summarized in Table 10. Cell 0 is the LSB (closest to TDO).

Table 10. 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.

Boundary-Scan Register

All of the chip's inputs and outputs are incorporated in a JTAG scan path shown in Table 11. The types of boundary-scan cells are as follows:

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

Hardware Architecture (continued)

Table 11. JTAG Boundary-Scan Register

Note: The direction of shifting is from TDI to cell 105 to cell 104 . . . to cell 0 to TDO.

Cell	Type	Signal Name/Function
0—15	O	AB[0:15] (cell #0 is AB0, etc.)
16	I	EXM
17	O	RWN
18—21	O	EROM, ERAMLO, ERAMHI, IO
22	O	DSEL
23—29	B	DB[0:6]
30	DC	Controls cells 23—29, 31—39
31—39	B	DB[7:15]
40	O	OBE1
41	O	IBF1
42	I	DI1
43	DC	Controls cell 46
44	DC	Controls cell 47
45	DC	Controls cell 48
46	B	ILD1
47	B	ICK1
48	B	OCK1
49	B	OLD1
50	DC	Controls cell 49
51	DC	Controls cell 53
52	O	DO1
53	B	SYNC1
54	OE	Controls cell 52
55	DC	Controls cell 58
56	DC	Controls cell 59
57	I	Empty
58	B	SADD1
59	B	DOEN1
60	DC	Controls cell 63.
61	DC	Controls cell 62
62	B	OCK2/PSEL2*
63	B	DO2/PSEL1*
64	B	SYNC2/PSEL0*
65	DC	Controls cell 64
66	DC	Controls cell 69
67	DC	Controls cell 68
68	B	ILD2/PIDS*
69	B	OLD2/PODS*
70	O	IBF2/PIBF*
71	DC	Controls cell 75
72	DC	Controls cell 74
73	O	OBE2/POBE*
74	B	ICK2/PB0*
75	B	DI2/PB1*
76	B	DOEN2/PB2*
77	B	SADD2/PB3*
78	DC	Controls cell 77
79	DC	Controls cell 76
80	DC	Controls cell 85
81	DC	Controls cell 82
82	B	IOBIT0/PB4*
83	DC	Controls cell 87
84	DC	Controls cell 86
85	B	IOBIT1/PB5*
86	B	IOBIT2/PB6*
87	B	IOBIT3/PB7*
88	B	VEC3/IOBIT4*
89	B	VEC2/IOBIT5*
90	DC	Controls cell 88
91	DC	Controls cell 89
92	B	VEC1/IOBIT6*
93	B	VEC0/IOBIT7*
94	I	INT1
95	DC	Controls cell 92
96	DC	Controls cell 93
97	I	INT0
98	DC	Controls cell 101
99	OE	Controls cells 0—15, 40—41, 70, 73, 100
100	O	IACK
101	B	TRAP
102	O	CKO
103	OE	Controls cells 17—22,102
104	I	RSTB
105	I	CKI

* Please refer to Pin Multiplexing in Section 4.1 for a description of pin multiplexing of BIO, PIO, VEC[3:0], and SIO2.

5 Software Architecture

5.1 Instruction Set

The DSP1616 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 DSP1616 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 *DSP1616 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
~	1s 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.

Multiply/ALU Instructions

Note that the function statements and transfer statements in Table 12 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.

A single-cycle squaring function is now provided in DSP1616. By setting the x=y= bit in the **auc** register, any instruction that loads the **y** register also loads the **x** register with the same value. A subsequent instruction to multiply the **x** register and **y** register results in the square of the value being placed in the **p** register. The instruction **a0=p p=x*y   y=*r0++ x=*pt++** executed from the cache with the x=y= bit set will read the value pointed to by **r0**, load it to both **x** and **y**, square the previously fetched value, and transfer the previous square to **a0**. A table of values pointed to by **r0** can thus be squared in a pipeline with one instruction cycle per each value. Note that **pt** will be incremented even though **x** is not loaded from the value pointed to by **pt**. Also, any restrictions from reading the same bank of internal memory or reading from external memory apply as if the **x=*pt++** was actually implemented.

Software Architecture (continued)

Table 12. Multiply/ALU Instructions

Function Statements	
	p = x * y
aD = p	p = x * y
aD = aS + p	p = x * y
aD = aS – p	p = x * y
aD = p	
aD = aS + p	
aD = aS – p	
aD = y	
aD = aS + y	
aD = aS – y	
aD = aS & y	
aD = aS l y	
aD = aS ^ y	
aS - y	
aS & y	

Transfer Statement‡		Cycles (Out/In Cache)†
y = Y	x = X	2/1
y = aT	x = X	2/1
y[l] = Y		1/1
aT[l] = Y		1/1
x = Y		1/1
Y		1/1
Y = y[l]		2/2
Y = aT[l]		2/2
Z:y	x = X	2/2
Z:y[l]		2/2
Z:aT[l]		2/2

Note: For transfer statements when loading the upper half of an accumulator, the lower half is cleared if the corresponding CLR bit in the **auc** register is zero. **auc** is cleared by reset.

† Add cycles for:
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.

‡ The l in [] is an optional argument that specifies the low 16 bits of **aT** or **y.**

Table 13. Replacement Table for Multiply/ALU Instructions

Replace	Value	Meaning
aD, aS, aT	a0, a1	One of two DAU accumulators.
X	*pt++, *pt++i	X memory space 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 aD = aS >> 4 aD = aS >> 8 aD = aS >> 16	}	Arithmetic right shift (sign preserved) of 36-bit accumulators
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 aD = aS << 4 aD = aS << 8 aD = aS << 16	}	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)

† This function is not available on the DSP16A.

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 14. 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 16 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 **icall** instruction is reserved for development system use.

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

```
if le goto 0x0345
```

Table 15. 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 16 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 16 lists mnemonics used in conditional execution of special function and control instructions.

Table 16. DSP1616 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, unless the RAND field of the **auc** register is set. 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 (IM16). 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 in Table 17.

Table 17. F3 ALU 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.

F3 ALU Instructions*	
Cacheable (one-cycle)	Not Cacheable (two-cycle)†
aD = aS + aT	aD = aS**h** + IM16
aD = aS – aT	aD = aS**h** – IM16
aD = aS & aT	aD = aS**h** & IM16
aD = aS | aT	aD = aS**h** **|** IM16
aD = aS ^ aT	aD = aS**h** ^ IM16
aS – aT	aS**h** – IM16
aS & aT	aS**h** & IM16
aD = aS + p	aD = aS**l** + IM16
aD = aS – p	aD = aS**l** – IM16
aD = aS & p	aD = aS**l** & IM16
aD = aS | p	aD = aS**l** | IM16
aD = aS ^ p	aD = aS**l** ^ IM16
aS – p	aS**l** – IM16
aS & p	aS**l** & IM16

* If PC points to external memory, add programmed wait-states.
† The **h** and **l** are required notation in these instructions.

F4 BMU Instructions

The bit manipulation unit in the DSP1616 provides a set of efficient bit manipulation operations on accumulators. It contains four auxiliary registers, **ar**<**0**—**3**> (**arM**, **M** = **0**, **1**, **2**, **3**), two alternate accumulators (**aa**<**0**—**1**>), 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—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)
LLV = 1 if (a) 36-bit overflow, or if (b) illegal shift on field width/offset condition
LMV = 1 if bits 31—35 are not the same (32-bit overflow)

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 *DSP1616 Digital Signal Processor Information Manual* for further discussion of the BMU instructions.

Software Architecture (continued)

- **Barrel Shifter**

aD = aS >> IM16	Arithmetic right shift by immediate (36-bit, sign filled in); 2-cycle, 2-word.
aD = aS >> arM	Arithmetic right shift by **arM** (36-bit, sign filled in); 1-cycle.
aD = $\overline{\text{aS}}$ >> aS	Arithmetic right shift by **aS** (36-bit, sign filled in); 2-cycle.
aD = aS >>> IM16	Logical right shift by immediate (32-bit shift, 0s filled in); 2-cycle, 2-word.
aD = aS >>> arM	Logical right shift by **arM** (32-bit shift, 0s filled in); 1-cycle.
aD = $\overline{\text{aS}}$ >>> aS	Logical right shift by **aS** (32-bit shift, 0s filled in); 2-cycle.
aD = aS << IM16	Arithmetic left shift* by immediate (36-bit shift, 0s filled in); 2-cycle, 2-word.
aD = aS << arM	Arithmetic left shift* by **arM** (36-bit shift, 0s filled in); 1-cycle.
aD = $\overline{\text{aS}}$ << aS	Arithmetic left shift* by **aS** (36-bit shift, 0s filled in); 2-cycle.
aD = aS <<< IM16	Logical left shift by immediate (36-bit shift, 0s filled in); 2-cycle, 2-word.
aD = aS <<< arM	Logical left shift by **arM** (36-bit shift, 0s filled in); 1-cycle.
aD = $\overline{\text{aS}}$ <<< 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, arM)	Normalize **aS** with respect to bit 31, with exponent in **arM**; 1-cycle.

- **Bit Field Extraction and Insertion**

aD = extracts(aS, IM16)	Extraction with sign extension, field specified as immediate; 2-cycle, 2-word.
aD = extracts(aS, arM)	Extraction with sign extension, field specified in **arM**; 1-cycle.
aD = extractz(aS, IM16)	Extraction with zero extension, field specified as immediate; 2-cycle, 2-word.
aD = extractz(aS, arM)	Extraction with zero extension, field specified in **arM**; 1-cycle.
aD = insert(aS, IM16)	Bit field insertion, field specified as immediate; 2-cycle, 2-word.
aD = insert(aS, arM)	Bit field insertion, field specified in **arM**; 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 **arM**), 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 18. Replacement Table for F3 ALU Instructions and F4 BMU Instructions

Replace	Value	Meaning
aD, aT, aS	a0 or a1	One of the two accumulators.
IM16	immediate	16-bit data, sign-, zero-, or one-extended as appropriate.
arM	ar<0—3>	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, and 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 19. Replacement Table for Cache Instructions

Replace	Instruction Encoding	Meaning
K	cloop*	Number of times the instructions are to be executed taken from bits 0—6 of the cloop register.
	1 to 127	Number of times the instructions to be executed is encoded in the 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 timings of the instructions are 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 for the last instruction in the block of NI instructions. This instruction executes in two cycles.

2. During pass two through pass K – 1, each instruction is fetched from cache and the in-cache timings apply.

3. During the last (Kth) pass, the block of instructions is fetched from cache and the in-cache timings apply, 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 as follows:

R = IM16
aT[l] = R
SR = IM9
Y = R
R = Y
Z : R
R = aS[l]
DR = *(O)
*(O) = DR

Table 20. Replacement Table for Data Move Instructions

Replace	Value	Meaning
R	Any of the registers in Table 48	—
DR	r<0—3>, a0[l], a1[l], y[l], p, pl, 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.
IM16	16-bit value	Long immediate data.
IM9	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 11 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.

Notes:
sioc, **sioc2**, **tdms**, **tdms2**, **srta**, and **srta2** registers are not readable.

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**.

Software Architecture (continued)

5.2 Register Settings

Tables 21 through 37 describe the programmable registers of the DSP1616 device. Table 37 describes the register settings after reset.

Note that the following abbreviations are used in the tables:

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

Table 21. Serial I/O Control Registers

sioc

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	In active mode, ILD1 and/or OLD1 = ICK1/16, active SYNC1 = ICK1/[128/256*]. In active mode, ILD1 and/or OLD1 = OCK1/16, active SYNC1 = OCK1/[128/256*].
CLK	00 01 10 11	Active clock = CKI/4 (2X) or CKI/2 (1X). Active clock = CKI/12 (2X) or CKI/6 (1X). Active clock = CKI/16 (2X) or CKI/8 (1X). Active clock = CKI/20 (2X) or CKI/10 (1X).
MSB	0 1	LSB first. MSB first.
OLD	0 1	OLD1 is an input (passive mode). OLD1 is an output (active mode).
ILD	0 1	ILD1 is an input (passive mode). ILD1 is an output (active mode).
OCK	0 1	OCK1 is an input (passive mode). OCK1 is an output (active mode).
ICK	0 1	ICK1 is an input (passive mode). ICK1 is an output (active mode).
OLEN	0 1	16-bit output. 8-bit output.
ILEN	0 1	16-bit input. 8-bit input.

sioc2†

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

* See **tdms** register, SYNC field.
† The bit definitions of the **sioc2** register are identical to the **sioc** register bit definitions.

Software Architecture (continued)

Table 22. Time-Division Multiplex Slot Registers

tdms

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

Field	Value	Description
SYNCSP*	0†	SYNC1 = ICK1/128 if LD = 0.* SYNC1 = OCK1/128 if LD = 1.*
	1	SYNC1 = ICK1/256 if LD = 0.* SYNC1 = OCK1/256 if LD = 1.*
MODE	0	Multiprocessor mode off; DOEN1 is an input (passive mode).
	1	Multiprocessor mode on; DOEN1 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	Transmit slot 0, SYNC1 is an output (active mode).
	0	SYNC1 is an input (passive mode).

tdms2‡

Bit	9	8	7	6	5	4	3	2	1	0
Field	SYNCSP2†	MODE2	TRANSMIT SLOT2							SYNC2

* See **sioc** register, LD field.
† Select this mode when in multiprocessor mode.
‡ The **tdms2** register bit definitions are identical to the **tdms** register bit definitions.

Software Architecture (continued)

Table 23. Serial Receive/Transmit Address Registers

srta

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.

srta2*

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	RECEIVE ADDRESS2								TRANSMIT ADDRESS2							

* The **srta2** field definitions are identical to the **srta** register field definitions.

Table 24. Multiprocessor Protocol Registers

saddx

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

saddx2*

Bit Field	15—8	7—0
Write	X	Write Protocol2 Field [7:0]
Read	Read Protocol2 Field [7:0]	0

* The **saddx2** field definitions are identical to the **saddx** register field definitions.

Software Architecture (continued)

Table 25. 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) 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) 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 26. Arithmetic Unit Control (auc) Register†

Bit	8	7	6	5	4	3	2	1	0
Field	RAND	X=Y=	CLR			SAT		ALIGN	

Field	Value	Description
RAND	0 1	Pseudorandom sequence generator (PSG) reset by writing the pi register only outside an interrupt service routine. PSG never reset by writing the pi register.
X=Y=	0 1	Normal operation. y=Y transfer statements load both the x and the y registers, allowing single-cycle squaring with p = x * y.
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 x 4 (and zeros written to the two LSBs).
	11	Reserved.

† The **auc** is 9 bits [8:0]. The upper 7 bits [15:9] are always zero when read and should always be written with zeros to make the program compatible with future chip versions. The **auc** register is cleared at reset.

Software Architecture (continued)

Table 27. 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	Reserved	INTERRUPTS					STATUS				

Field	Value	Description
IBF	R	IBF interrupt status bit (same as bit 4).
STROBE		Strobe width of: PODS PIDS
	00	T* T
	01	2T 2T
	10	3T 3T
	11	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).
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 for 2X input clock option or T = t1 for 1X input clock option. See timing diagrams.
† The enable and status bits in the **pioc** affect only the SIO, not SIO2.

In active mode, reading or writing **pdx**<**0**—**7**> causes an encoded channel number (111 to 000) to appear on pins PSEL[2:0]. Note that PSEL0 then contains the same information that DSP16A PSEL pin has, for compatibility. In passive modes, the PSEL pins take on other functions shown in Table 7.

Table 28. Interrupt Control (inc) Register

Bit	15	14—12	11	10	9	8	7—6	5—4	3	2	1	0
Field	JINT*	Reserved	EMUX BOTH	OBE2	IBF2	TIME	Reserved	INT[1:0]	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 29. Interrupt Status (ins) Register

Bit	15	14—12	11	10	9	8	7—6	5—4	3	2	1	0
Field	JINT	Reserved	EMUX BOTH	OBE2	IBF2	TIME	Reserved	INT[1:0]	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 4, 5, 8, or 11 of **ins**, the corresponding interrupt is cleared.

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	Timer enabled.
	1	Timer and prescaler disabled. The period register and timer0 are not reset.
RELOAD	0	Timer stops after counting down to 0.
	1	Timer automatically reloads and repeats indefinitely.
T0EN	0	Timer holds current count.
	1	Timer counts down to 0.
PRESCALE	—	See table below.

PRESCALE Field

PRESCALE	Frequency of Timer Interrupts	
	1X	2X
0000	CKI/2	CKI/4
0001	CKI/4	CKI/8
0010	CKI/8	CKI/16
0011	CKI/16	CKI/32
0100	CKI/32	CKI/64
0101	CKI/64	CKI/128
0110	CKI/128	CKI/256
0111	CKI/256	CKI/512
1000	CKI/512	CKI/1024
1001	CKI/1024	CKI/2048
1010	CKI/2048	CKI/4096
1011	CKI/4096	CKI/8192
1100	CKI/8192	CKI/16384
1101	CKI/16384	CKI/32768
1110	CKI/32768	CKI/65536
1111	CKI/65536	CKI/131072

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	1xxxxxxx	IOBIT7 is an output (input when 0).
	x1xxxxxx	IOBIT6 is an output (input when 0).
	xx1xxxxx	IOBIT5 is an output (input when 0).
	xxx1xxxx	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[7:4]				MODE/MASK[3:0]				DATA/PAT[7:4]				DATA/PAT[3:0]			

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

* $0 \le n \le 7$.

Software Architecture (continued)

Table 33. alf Register

Bit	15	14	13—0
Field	AWAIT	LOWPR	FLAGS

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

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 34. 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 the INT1 is 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.

Table 35. DSP1616 32-bit JTAG ID Register

Bit	31	30	29—28	27—19	18—12	11—0
Field	PART	SECURE	CLOCK	ROMCODE	0 0 1 0 1 1 1	0x03B

Field	Value	Mask-Programmable Features
PART	0 1	DSP1616-x11. DSP1616-x30.
SECURE	0 1	Nonsecure ROM option. Secure ROM option.
CLOCK	00 01 10 11	TTL level input clock option. Small signal input clock option. Crystal oscillator input clock option. CMOS level input clock option.
ROMCODE	—	User's ROMCODE ID: The ROMCODE ID reads (in hexadecimal) the value calculated by the following formula: $[(20 \times \text{value for first letter}) + (\text{value of second letter})]_{16}$.

ROMCODE Letter	A	B	C	D	E	F	G	H	J	K	L	M	N	P	R	S	T	U	W	Y
Value	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19

Software Architecture (continued)

Table 36. ioc Register*

Bit	15—13	12	11	10	9	8—7	6	5	4	3—0
Field	Reserved	EBIOH	WEROM	ESIO2	SIOLBC	CKO[1:0]	DSELH	PIOLBC	DDSEL0	DENB[3:0]

* The field definitions for the **ioc** register are different from those of the DSP1610.

ioc Fields

ioc Field	Description
EBIOH	If 1, enables high half of BIO, IOBIT[4:7], and disables VEC[3:0] from pins.
WEROM	If 1, allows writing into external program (X) memory.
ESIO2	If 1, enables SIO2 and low half of BIO, and disables PIO from pins.
SIOLBC	If 1, DO1 and DO2 looped back to DI1 and DI2.
CKO[1:0]	CKO configuration (see below).
DSELH	If 1, DSEL active-high.
PIOLBC	If 1, PB[7:0] and PODS to PIDS internally looped back.
DDSEL0	If 1, delay DSEL.
DENB3	If 1, delay EROM.
DENB2	If 1, delay ERAMHI.
DENB1	If 1, delay IO.
DENB0	If 1, delay ERAMLO.

CKO1	CKO0	CKO Output		Description
		1X	2X	
0	0	CKI	CKI/2	Free-running clock.
0	1	CKI/(1 + W)	CKI/(2[1+W])	Wait-stated clock.
1	0	1	1	Held high.
1	1	0	0	Held low.

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

Software Architecture (continued)

Table 37. Register Settings after Reset

A • indicates that this bit is unknown on powerup reset and unaffected on subsequent reset. An S indicates that 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	••••••••••••••••	inc	0000000000000000
r1	••••••••••••••••	ins	0111010011000010
r2	••••••••••••••••	sdx2	••••••••••••••••
r3	••••••••••••••••	saddx	••••••••••••••••
j	••••••••••••••••	cloop	000000000•••••••
k	••••••••••••••••	mwait	0000000000000000†
rb	0000000000000000	saddx2	••••••••••••••••
re	0000000000000000	sioc2	••••••0000000000
pt	••••••••••••••••	cbit	••••••••••••••••
pr	••••••••••••••••	sbit	00000000PPPPPPPP
pi	SSSSSSSSSSSSSSSS	ioc	0000000000000000
i	••••••••••••••••	jtag	••••••••••••••••
p	••••••••••••••••	pdx4	00000000••••••••
pl	••••••••••••••••	pdx5	00000000••••••••
pdx2	00000000••••••••	pdx6	00000000••••••••
pdx3	00000000••••••••	pdx7	00000000••••••••
x	••••••••••••••••	a0	••••••••••••••••
y	••••••••••••••••	a0l	••••••••••••••••
yl	••••••••••••••••	a1	••••••••••••••••
auc	0000000000000000	a1l	••••••••••••••••
psw	••••00••••••••••	timerc	••••••••00000000
c0	••••••••••••••••	timer0	0000000000000000
c1	••••••••••••••••	tdms2	••••••0000000000
c2	••••••••••••••••	srta2	••••••••••••••••
sioc	••••••0000000000		
srta	••••••••••••••••		
sdx	••••••••••••••••	ar0	••••••••••••••••
tdms	••••••0000000000	ar1	••••••••••••••••
pioc	0000000000001000	ar2	••••••••••••••••
pdx0	00000000••••••••	ar3	••••••••••••••••
pdx1	00000000••••••••		
ybase	••••••••••••••••	alf	00000000••••••••

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

Software Architecture (continued)

5.3 Instruction Set Formats

This section defines the hardware-level encoding of the DSP1616 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 (IM16)															
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			
	Immediate Operand (IM16)															
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					$\overline{aT}$	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 (IM16)															
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 (IM9)								
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				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 38. 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	Bit 0 = 0, aT = R		7
01000	Bit 0 = 1, aTl = R		7
01001	Bit 10 = 0, R = a0		7
01001	Bit 10 = 1, R = a0l		7
01010	R = IM16		8
01011	Bit 10 = 0, R = a1		7
01011	Bit 10 = 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	Bit 0 = 0, branch indirect		5
11000	Bit 0 = 1, F3 ALU		3a
11001	y = a0 x = X	F1	1
11010	Cond. branch qualifier		6
11011	y = a1 x = X	F1	1
11100	Y = a0[l]	F1	1
11101	Z : y x = X	F1	2
11110	Bit 5 = 0, F4 ALU (BMU)		3b
11110	Bit 5 = 1, direct addressing		9a
11111	y = Y x = X	F1	1

Table 39. D Field

Specifies a destination accumulator.

D	Register
0	Accumulator 0
1	Accumulator 1

Table 40. $\overline{aT}$ Field

Specifies transfer accumulator.

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

Table 41. S Field

Specifies a source accumulator.

S	Register
0	Accumulator 0
1	Accumulator 1

Table 42. 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 43. 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 44. 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 45. 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 46. 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

Table 47. 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	—	—

Software Architecture (continued)

Table 48. 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	saddx2
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	tdms2
010111	c2	110111	srta2
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

Table 49. 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 50. DR Field

DR Value	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 51. 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 52. 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.

R/W Field

A zero specifies a write, *(O) = DR.
A one specifies a read, DR = *(O).

Table 53. F3 Field

Specifies the operation in an F3 ALU instruction.

F3	Operation		
1000	aD = aS[h, l]	|	{aT, IM16, p}
1001	aD = aS[h, l]	^	{aT, IM16, p}
1010	aS[h, l]	&	{aT, IM16, p}
1011	aS[h, l]	–	{aT, IM16, p}
1101	aD = aS[h, l]	+	{aT, IM16, p}
1110	aD = aS[h, l]	&	{aT, IM16, p}
1111	aD = aS[h, l]	–	{aT, IM16, p}

Table 54. SRC2 Field

Specifies operands in an F3 ALU instruction.

SRC2	Operands
00	aSl, IM16
10	aSh, IM16
01	aS, aT
11	aS, p

Table 55. BMU Encodings

F4	AR	Operation
0000	00xx	aD = aS >> arM
0001	00xx	aD = aS << arM
0000	10xx	aD = aS >>> arM
0001	10xx	aD = aS <<< arM
1000	0000	aD = $\overline{aS}$ >> aS
1001	0000	aD = $\overline{aS}$ << aS
1000	1000	aD = $\overline{aS}$ >>> aS
1001	1000	aD = $\overline{aS}$ <<< aS
1100	0000	aD = aS >> IM16
1101	0000	aD = aS << IM16
1100	1000	aD = aS >>> IM16
1101	1000	aD = aS <<< IM16
0000	1100	aD = exp(aS)
0001	11xx	aD = norm(aS, arM)
1110	0000	aD = extracts(aS, IM16)
0010	00xx	aD = extracts(aS, arM)
1110	0100	aD = extractz(aS, IM16)
0010	01xx	aD = extractz(aS, arM)
1110	1000	aD = insert(aS, IM16)
1010	10xx	aD = insert(aS, arM)
0111	0000	aD = aS:aa0
0111	0001	aD = aS:aa1

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

6 Signal Descriptions

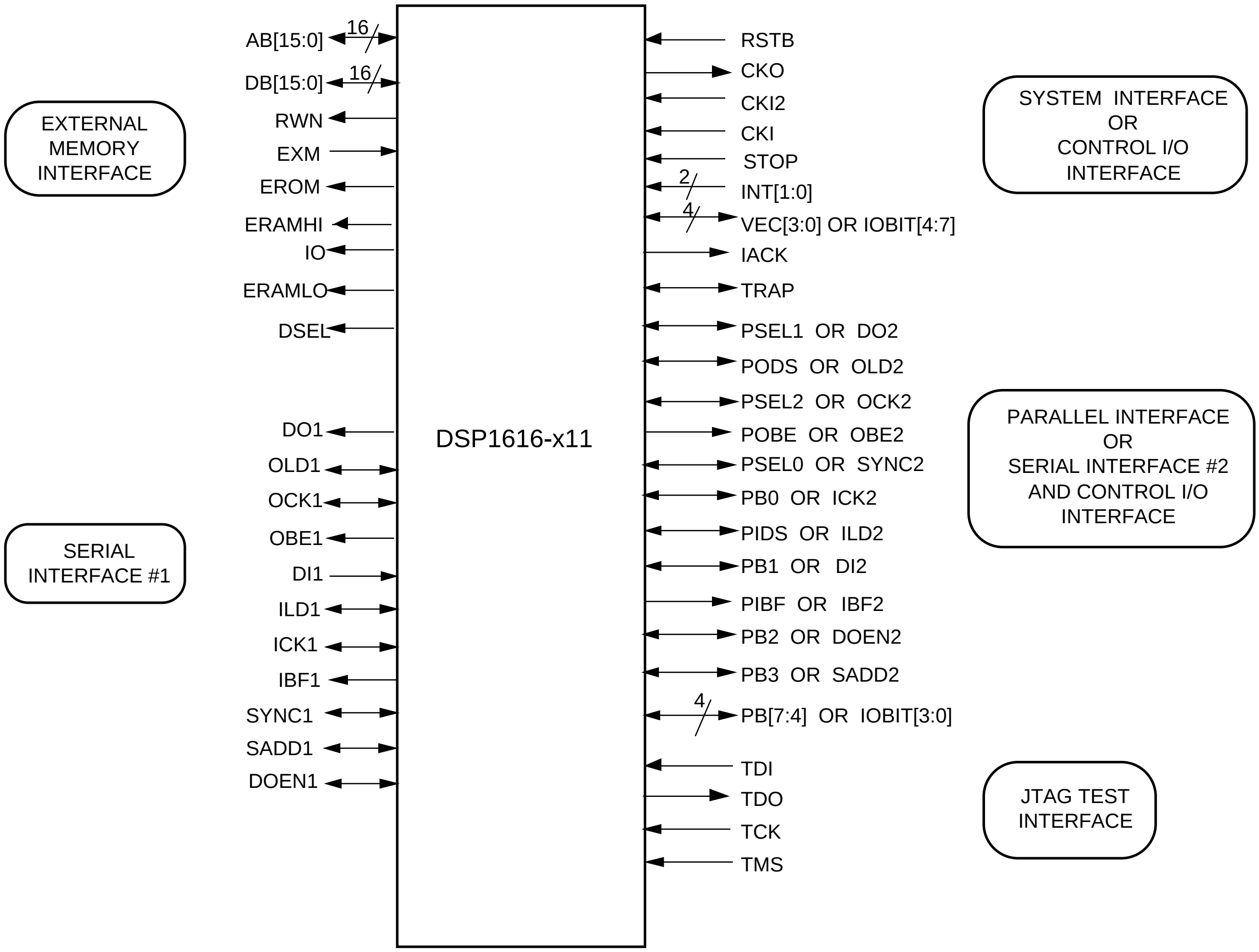


Figure 6. DSP1616 Pinout by Interface

Signal Descriptions (continued)

Figure 6 shows the pinout for the DSP1616. The signals can be separated into five 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 **auc**, **sioc**, **sioc2**, **pioc** (except OBE status bit set), **pdx**<**0**—**7**> (upper byte), **tdms**, **tdms2**, **timerc**, **timer0**, **sbit** (upper byte), **inc**, **ins** (except OBE and OBE2 status bits set), **alf** (upper 2 bits, AWAIT and LOWPR), **ioc**, **rb**, and **re** registers are cleared. 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 IACK, VEC[3:0]/IOBIT[4:7], IBF, and IBF2. The DAU condition flags are not affected by reset. IOBIT[7:0] are initialized as inputs. If any of the IOBIT pins are switched to outputs (by writing **sbit**), their initial value will be logic zero (see Table 37, Register Settings after Reset).

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

CKI

Input Clock: A mask-programmable option selects the input clock to processor clock ratio (1X or 2X). For a 2X clock selection, the input clock, CKI, runs at twice the frequency of internal operation (see Section 7, Mask-Programmable Options and Table 1, Pin Descriptions).

CKI2

Input Clock 2: Used with mask-programmable input clock options which require an external crystal or small signal differential across CKI and CKI2 (see Table 1, Pin Descriptions).

CKO

Clock Out: Buffered output clock with options programmable via the **ioc** register (see Table 36). The selectable CKO options (see Table 36) are as follows:

- A free-running output clock at half the frequency of CKI for the 2X input clock option
- A wait-stated clock
- A logic 0
- A logic 1

INT[1:0]

Processor Interrupts 0 and 1: Positive assertion. Hardware interrupt inputs to the DSP1616. Each is enabled via the **inc** register. When enabled and asserted, each cause the processor to vector to the memory location described in Table 4. INT0 can be enabled in the **pioc** for DSP16A compatibility. INT1 is used in conjunction with EXM to select the desired reset initialization of the **mwait** register (see Table 34). **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.**

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. VEC[3:0] are multiplexed with IOBIT[4:7].

IACK

Interrupt Acknowledge: Positive assertion. IACK signals when an interrupt is being serviced by the DSP1616. 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 pin can be used to signal an HDS breakpoint in a multiple processor environment.

Signal Descriptions (continued)

6.2 External Memory Interface

The external memory interface is used to interface the DSP1616 to external memory and I/O devices. It supports read/write operations from/to program and data memory spaces. The interface supports four external memory segments. Each external memory segment can have an independent number of software programmable wait-states. One hardware address is decoded, and an enable line is 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 the 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 the external memory or I/O.

RWN

Read/Write Not: 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 Select: Input only. This signal is latched into the device on the rising edge of RSTB. The value of EXM latched in determines whether the internal ROM is addressable in the instruction/coefficient memory map. 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: 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 36).

ERAMHI

External RAM High Enable Signal: 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 36).

ERAMLO

External RAM Low Enable Signal: 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 36).

IO

External I/O Enable Signal: Negative assertion. When asserted, the signal indicates an access to external data memory addresses 0x7F00 through 0x7FFF (see Table 6, Data Memory Map). This memory segment is intended for memory-mapped I/O. This signal's leading edge can be delayed via the **ioc** register (see Table 36).

DSEL

Device Select Line: Default negative assertion (positive assertion is selectable via the **ioc** register, see Table 36). This signal predecodes a specific memory address in the IO external memory segment. Access to location 0x7F00 asserts DSEL, as well as the external IO enable.

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 5, Multiprocessor Communications and Connections).

DI1

Data Input: Serial data is latched on the rising edge of ICK1, either LSB or MSB first, according to the **sioc** register MSB field (see Table 21).

ICK1

Input Clock: The clock for serial input data. In active mode, ICK1 is an output; in passive mode, ICK1 is an input, according to the **sioc** register ICK field (see Table 21).

ILD1

Input Load: The clock for loading the input buffer, **sdx**(in), from the input shift register isr. A falling edge of ILD1 indicates the beginning of a serial input word. In active mode, ILD1 is an output; in passive mode, ILD1 is an input, according to the **sioc** register ILD field (see Table 21).

IBF1

Input Buffer Full: Positive assertion. IBF1 is asserted when the input buffer, **sdx**(in), is filled. IBF1 is negated by a read of the buffer, as in **a0 = sdx**. IBF1 is also negated by asserting RSTB.

DO1

Data Output: The serial data output from the output shift register (osr), either LSB or MSB first (according to the **sioc** register MSB field). DO1 changes on the rising edges of OCK1. DO1 is 3-stated when DOEN1 is high.

DOEN1

Data Output Enable: Negative assertion. An input when not in the multiprocessor mode. DO1 and SADD1 are enabled only if DOEN1 is low. DOEN1 is bidirectional when in the multiprocessor mode (**tdms** register MODE field set). In the multiprocessor mode, DOEN1 indicates a valid time slot for a serial output.

OCK1

Output Clock: The clock for serial output data. In active mode, OCK1 is an output; in passive mode, OCK1 is an input, according to the **sioc** register OCK field (see Table 21).

OLD1

Output Load: The clock for loading the output shift register, osr, from the output buffer **sdx**(out). A falling edge of OLD1 indicates the beginning of a serial output word. In active mode, OLD1 is an output; in passive, OLD1 is an input, according to the **sioc** register OLD field (see Table 21).

OBE1

Output Buffer Empty: Positive assertion. OBE1 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**. OBE1 is also set by asserting RSTB.

SADD1

Serial Address: Negative assertion. A 16-bit serial bit stream typically used for addressing during multiprocessor communication between multiple DSP1616 devices. In multiprocessor mode, SADD1 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. SADD1 is always an output when not in multiprocessor mode and can be used as a second 16-bit serial output. See the *DSP1616 Digital Signal Processor Information Manual* for additional information. SADD1 is 3-stated when DOEN1 is high. When used on a bus, SADD1 should be pulled high through a 5 kΩ resistor.

SYNC1

Multiprocessor Synchronization: Typically used in the multiprocessor mode, a falling edge of SYNC1 indicates the first word of a TDM I/O stream and causes the resynchronization of the active ILD1 and OLD1 generators. SYNC1 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, SYNC1 must be tied low unless part of a TDM interface. When used as an output, SYNC1 = [ILD1/OLD1]/8 or 16, depending on the setting of the SYNCSP field of the **tdms** register. When configured as described above, SYNC1 can be used to generate a slow clock for SIO operations.

Signal Descriptions (continued)

6.4 Parallel Interface or Serial Interface #2 and Control I/O Interface

This interface pin multiplexes a parallel I/O interface with a second serial I/O interface and a 4-bit I/O interface. The interface selection is made by writing the ESIO2 bit in the **ioc** register (see Table 36 and Section 4.1). The signals for the second SIO correspond exactly with those in SIO #1. Therefore, the pin descriptions below discuss only PIO and BIO pin functionality.

PB[7:0]

Parallel I/O Data Bus: This 8-bit bidirectional bus is used to input data to, or output data from, the PIO. Note that PB[3:0] are pin multiplexed with SIO2 functionality, and PB[7:4] are pin multiplexed with BIO unit pins IOBIT[3:0] (see Section 4.1).

PSEL[2:0]

Peripheral Select 2—0 (see Table 7): When the PIO configuration for both input and output 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**<**0**—**7**>) is active.

If the PIO is configured with either PIDS or PODS passive, PSEL2 becomes an input that 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 is configured in passive mode, PSEL1 and PSEL0 form a 2-bit field selecting between four channels (**pdx**<**0**—**4**> alias into **pdx**<**0**—**3**>).

When PODS is passive, PSEL1 becomes an input. If PSEL1 is high, the PIO will output the contents of the pstat register on PB[7:0]. 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 is being written is odd or even (e.g., **pdx**<**3**,**5**,**7**> alias into 1; and **pdx**<**2**,**4**,**6**> alias into 0). 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 low, data can be placed onto the PB bus by an external device. PIDS is asserted by the DSP1616 device during active mode read transaction.

In passive mode, PIDS is an input. PIDS is pulled low by an external device to indicate that data is available on the PB bus.

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 DSP1616 device during an active mode write transaction.

In passive mode, PODS is an input. When PODS is pulled low by an external device, the DSP1616 places the contents of the parallel output register (**pdx<0**—**7>**) onto the PB bus.

PIBF

Parallel Input Buffer Full: 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 **pdx**(in) register. When the DSP1616 reads the contents of this register, emptying the buffer, the flag is cleared.

POBE

Parallel Output Buffer Empty: 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 **pdx**(out) has been driven onto the PB bus. When the DSP1616 writes to **pdx**(out), filling the buffer, this flag 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 DSP1616. It is pin multiplexed with the PIO and VEC[3:0] pins (see Section 4.1). Setting the ESIO2 and EBIOH bits in the **ioc** register will provide a full 8-bit BIO interface at the pins.

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.

6.6 JTAG Test Interface

The JTAG test interface has features that allow programs to be downloaded into the DSP via four pins. This provides extensive test and diagnostic capability. In addition, internal circuitry allows the device to be controlled through the JTAG port to provide on-chip in-circuit emulation. AT&T provides hardware and software tools to interface to the on-chip HDS via the JTAG port.

Note: The DSP1616 provides all JTAG/*IEEE* 1149.1 standard test capabilities including boundary-scan. See the *DSP1616 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, and out of the port through TDO, and controls the port by latching the TMS signal inside the state-machine controller.

7 Mask-Programmable Options

The DSP1616 contains a 12 Kword ROM that is mask-programmable. The selection of several programmable features is made when a custom ROM is encoded. These features select the input clock options and hardware emulation or ROM security option, as summarized in Table 56.

Table 56. DSP1616 ROM Options

Features	Options	Comments
Input Clock to Processor Clock Ratio	1X 2X	CKI ≤ 30 MHz CKI ≤ 60 MHz
Input Clock	TTL Level CMOS Level Small Signal Crystal	1X or 2X 1X or 2X 1X or 2X 1X only
ROM Security	Nonsecure Secure	Specify and link **hds.v#** *, allows emulation Specify and link **crc16.v#** †, no emulation capability

* hds.v# (# indicates the current version number) is the relocatable HDS object code. It uses approximately 140 words and must reside in the first 4 Kwords of ROM.

† crc16.v# is the cyclic redundancy check object code. It uses approximately 80 words and must reside in the first 4 Kwords of ROM.

7.1 Input Clock Options

For a 2X clock selection, the input clock CKI runs at twice the frequency of internal operation. If this option is selected, either TTL or CMOS levels may be applied at the CKI pin, or a small signal differential voltage may be applied between pins CKI and CKI2 .

For a 1X clock selection, the TTL, CMOS, or small signal input buffer may be chosen, or the internal oscillator may be used with an external crystal. If the option for using an external crystal is chosen, the internal oscillator may be used as a noninverting input buffer simply by supplying a CMOS level input to the CKI pin and leaving the CKI2 pin open.

7.2 ROM Security Options

The DSP1600 Hardware Development System (HDS) provides on-chip in-circuit emulation and requires that the relocatable HDS code be linked to the application code. This code's object file is called hds.v# (hds.v# occupies approximately 140 words of memory in the first 4 Kword page). If on-chip in-circuit emulation is desired, a nonsecure ROM must be chosen.

If ROM security is desired with the DSP1616, the HDS cannot be used. To provide testing of the internal ROM contents, a cyclic redundancy check (CRC) program is called by and linked with the user's source code. The CRC code (crc16.v#) resides in the first 4 Kwords of ROM (crc16.v# occupies approximately 80 words of memory).

8 Device Characteristics

8.1 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 V_{DD} with Respect to Ground .... –0.5 V to +6 V
Voltage Range on any Signal Pin with Respect to Ground .... V_{SS} – 0.5 V to V_{DD} + 0.5 V
Power Dissipation .... 1 W
Ambient Temperature Range .... –40 °C to +85 °C
Storage Temperature Range .... –65 °C to +150 °C

8.2 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 buildup, 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 DSP1616 is greater than 2000 V.

8.3 Recommended Operating Conditions

Table 57. Recommended Operating Conditions

Device Speed	Input Clock Ratio	Input Clock	Package	Supply Voltage V_{DD} (V)		Ambient Temperature T_A (°C)	
				Min	Max	Min	Max
33 ns	2x	CMOS, TTL, small-signal	BQFP or TQFP	4.75	5.25	–40	85
33 ns	1X	CMOS, TTL, small-signal, crystal	BQFP or TQFP	4.75	5.25	–40	85
33 ns	2X	CMOS, TTL, small-signal	BQFP or TQFP	4.75	5.25	0	70
33 ns	1X	CMOS, TTL, small-signal, crystal	BQFP or TQFP	4.75	5.25	0	70
38 ns	2X	CMOS, TTL, small-signal,	BQFP or TQFP	4.75	5.25	–40	85
38 ns	1X	CMOS, TTL, small-signal, crystal	BQFP or TQFP	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 equation describes the relationship between these parameters. If the applications' maximum power is less than the worst-case value, this relationship determines a higher maximum ambient temperature or maximum temperature measured at top dead center of the package.

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

Maximum Junction Temperature (T_J) in 100-Pin BQFP .... 125 °C
100-pin BQFP Maximum Thermal Resistance in Still-Air-Ambient (Θ_{JA}) .... 55 °C/W
100-pin BQFP Maximum Thermal Resistance, Junction to Top Dead Center of Package, in Still-Air-Ambient (Θ_{J-TDC}) .... 12 °C/W

Maximum Junction Temperature (T_J) in 100-Pin TQFP .... 125 °C
100-pin TQFP Maximum Thermal Resistance in Still-Air-Ambient (Θ_{JA}) .... 64 °C/W
100-pin TQFP Maximum Thermal Resistance, Junction to Top Dead Center of Package, in Still-Air-Ambient (Θ_{J-TDC}) .... 6 °C/W

9 Electrical Characteristics and Requirements

The following electrical characteristics are subject to change. Electrical characteristics refer to the behavior of the device under specified conditions. Electrical requirements refer to conditions imposed on the user for proper operation of the device. The parameters below are valid for the conditions described in Section 8.3, Recommended Operating Conditions.

Table 58. Electrical Characteristics and Requirements

Parameter	Symbol	Min	Max	Unit
Input Voltage:				
Low	V_{IL}	—	0.3 * V_{DD}	V
High	V_{IH}	0.7 * V_{DD}	—	V
Input Current (except TMS, TDI):				
Low (V_{IL} = 0 V, V_{DD} = 5.25 V)	I_{IL}	–5	—	µA
High (V_{IH} = 5.25 V, V_{DD} = 5.25 V)	I_{IH}	—	5	µA
Input Current (TMS, TDI):				
Low (V_{IL} = 0 V, V_{DD} = 5.25 V)	I_{IL}	–100	—	µA
High (V_{IH} = 5.25 V, V_{DD} = 5.25 V)	I_{IH}	—	5	µA
Output Low Voltage:				
Low (I_{OL} = 2.0 mA)	V_{OL}	—	0.4	V
Low (I_{OL} = 50 µA)	V_{OL}	—	0.2	V
Output High Voltage:				
High (I_{OH} = –2.0 mA)	V_{OH}	V_{DD} – 0.7	—	V
High (I_{OH} = –50 µA)	V_{OH}	V_{DD} – 0.2	—	V
Output 3-State Current:				
Low (V_{DD} = 5.25 V, V_{IL} = 0 V)	I_{OZL}	–10	—	µA
High (V_{DD} = 5.25 V, V_{IH} = 5.25 V)	I_{OZH}	—	10	µA
Input Capacitance	C_I	—	10	pF

Table 59. Electrical Requirements for Mask-Programmable Input Clock Options

1X Parameter	Symbol	5 V Operating Voltage 1X Min	1X Max	2X Min	2X Max	Unit
CKI TTL Level Input Voltage:						
Low	V_{IL}	—	0.8	—	0.8	V
High	V_{IH}	2.4	—	2.4	—	V
CKI CMOS Level Input Voltage:						
Low	V_{IL}	—	0.3 * V_{DD}	—	0.3 * V_{DD}	V
High	V_{IH}	0.7 * V_{DD}	—	0.7 * V_{DD}	—	V
Small-Signal Peak-to-Peak Voltage (across pins CKI and CKI2) †	Vpp	0.6	—	0.8	—	V
Small-Signal Input Voltage Range (pins: CKI, CKI2) ‡	V_{in}	1.0	V_{DD} – 1.0	1.5	V_{DD} – 1.2	V
Frequency Range of Fundamental Mode or Overtone Crystal	f_X	5	30	5	30	MHz
Series Resistance of Fundamental Mode or Overtone Crystal (pins: CKI, CKI2)	R_S	—	40	—	40	Ω
Mutual Capacitance of Crystal (includes board stray capacitance)	C_0	—	7	—	7	pF

† The dc components of the signals on CKI and CKI2 must be equal to each other and have a magnitude of at least 1.0 V.
‡ The voltage on the CKI and CKI2 pins cannot exceed these limits.

Additional Electrical Requirements with Crystal Option: See Section 11, Crystal Electrical Characteristics and Requirements.

Electrical Characteristics (continued)

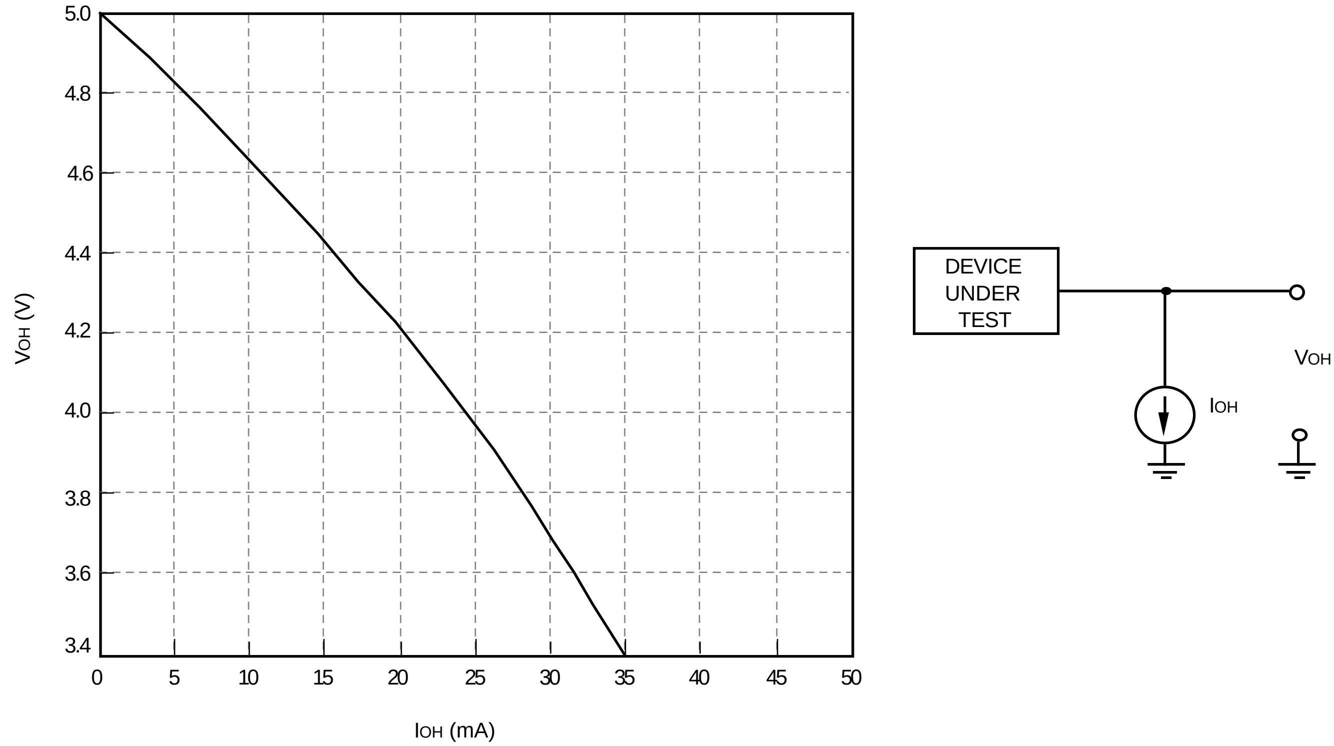


Figure 7. Plot of V_{OH} vs. I_{OH} Under Typical Operating Conditions

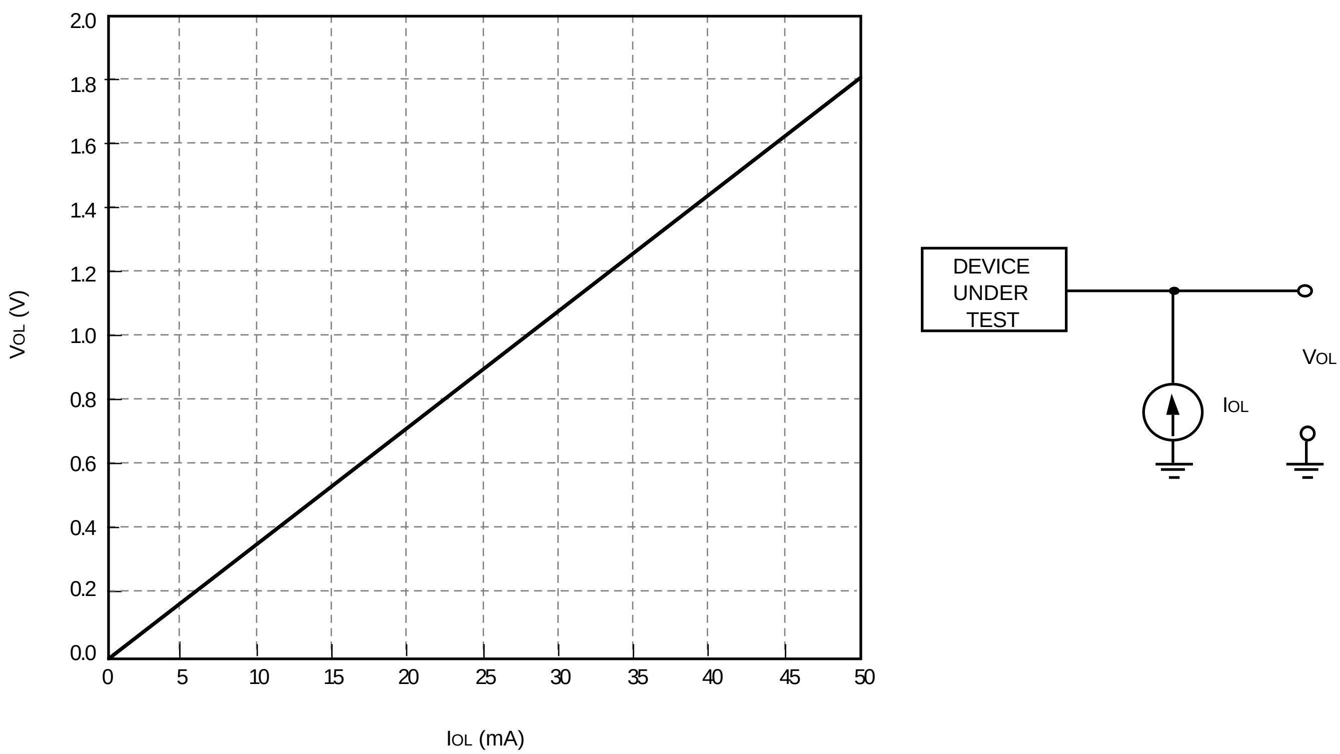


Figure 8. Plot of V_{OL} vs. I_{OL} Under Typical Operating Conditions

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 subject to change.

Table 60. Power Dissipation

Parameter	Symbol	Typical	Unit
Power Dissipation (V_{DD} = 5.0 V, inputs at V_{DD}, ioc = 0x0180):			
CKI(1X) = 30 MHz, CMOS/TTL	PD30	425	mW
Crystal Oscillator	—	455	mW
Small Signal	—	430	mW
CKI (1X) = 0 MHz, CMOS/TTL	PD0	0.55	mW
Small Signal	—	3.8	mW
Power Dissipation (V_{DD} = 5.0 V, sleep mode, ioc = 0x0180, alf = 0x8000, timerc = 0x0040):			
CKI(1X) = 30 MHz, CMOS/TTL	PDA30	45	mW
Crystal Oscillator	—	75	mW
Small Signal	—	50	mW

The power dissipation listed is for internal power dissipation only. 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 buffers is highly dependent on the input voltage level. At full CMOS levels, essentially no dc current is drawn. However, for levels between the power supply rails, especially at or near the threshold of $V_{DD}/2$, high currents can flow. Therefore, **all unused input pins should be tied to their inactive state**, either V_{DD} or V_{SS}. Although, I/O buffers may be left untied (since the input voltage levels of I/O buffers are designed to remain at full CMOS levels when not driven by the DSP), it is still recommended that unused I/O pins be tied to V_{SS} or V_{DD} through a 10 kΩ resistor to avoid application ambiguities. Further, if I/O pins are tied high or low, they should be pulled fully to V_{SS} or V_{DD}.

WARNING: The device needs to be clocked for at least six CKI cycles with the 1X option or 12 CKI cycles with the 2X option during reset after powerup. Otherwise, high currents may flow.

10 Timing Characteristics

The following timing characteristics and requirements are preliminary information and are subject to change. Timing characteristics refer to the behavior of the device under specified conditions. Timing requirements refer to conditions imposed on the user for proper operation of the device. All timing data is valid for the following conditions:

T_A = –40 °C to +85 °C (See Section 8.3.)

V_{DD} = 5 V ± 5%, V_{SS} = 0 V (See Section 8.3.)

Capacitance load on outputs (C_L) = 50 pF

Output characteristics can be derated as a function of load capacitance (C_L).

All outputs: $dt/dC_L \leq 0.06$ ns/pF for $0 \leq C_L \leq 100$ pF at V_{IH} for rising edge
$dt/dC_L \leq 0.05$ ns/pF for $0 \leq C_L \leq 100$ pF at V_{IL} for falling edge

For example, if the actual load capacitance is 30 pF instead of 50 pF, the derating for a rising edge is

(30 – 50) pF x 0.06 ns/pF = 1.2 ns **less** than the specified rise time or delay which includes a rise time.

Test conditions for inputs:

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

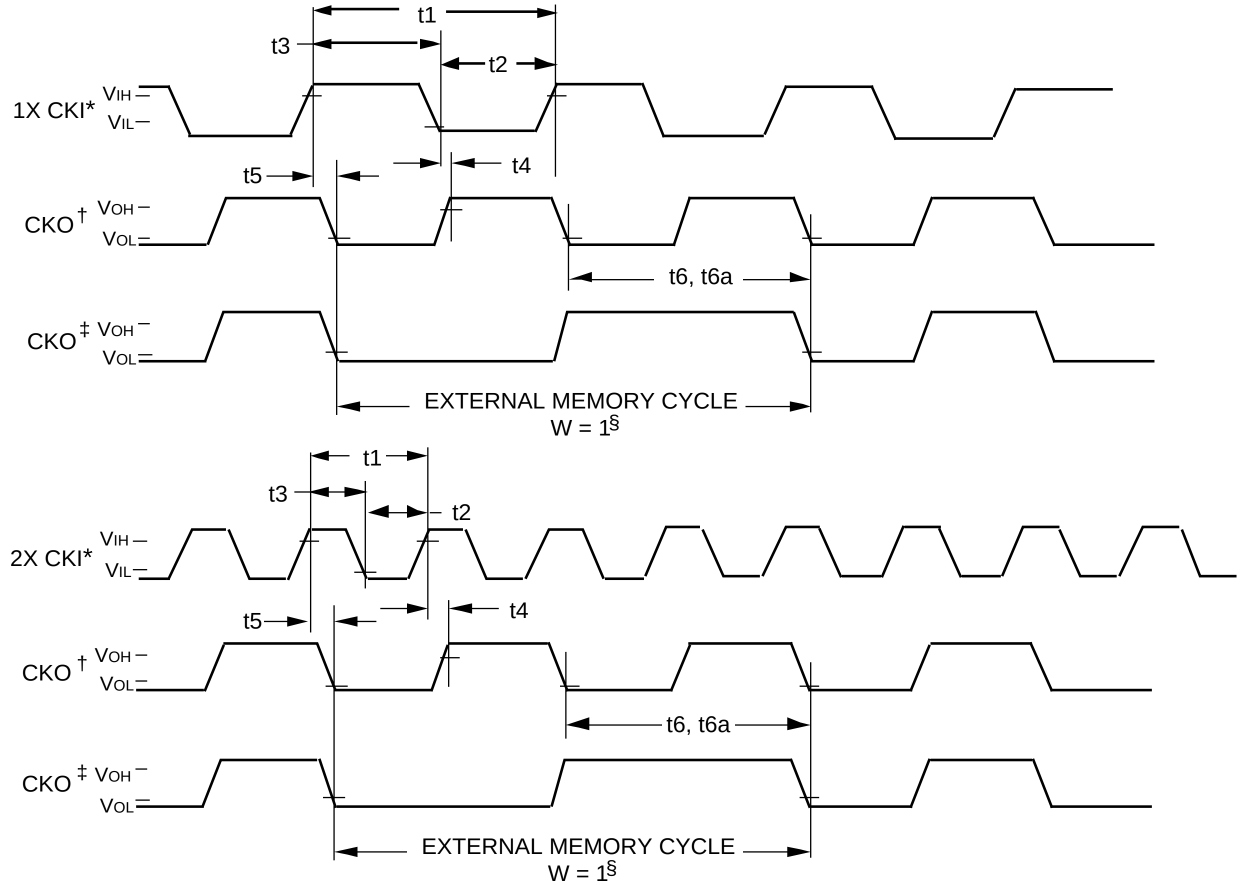
Test conditions for outputs (unless noted otherwise):

- C_{LOAD} = 50 pF
- Timing reference levels for delays = V_{IH}, V_{IL}
- 3-state delays measured to the high-impedance state of the output driver

For the timing diagrams, see Table 59 for input clock requirements with mask-programmable CKI options.

Timing Characteristics (continued)

10.1 DSP Clock Generation



* See Table 59 for input clock electrical requirements.
† Free-running clock.
‡ Wait-stated clock (see Table 34).
§ W = number of wait-states.

Figure 9. I/O Clock Timing Diagram

Table 61. Timing Requirements for Input Clock

Abbreviated Reference	Parameter	38 ns				33 ns				Unit
		2X		1X		2X		1X		
		Min	Max	Min	Max	Min	Max	Min	Max	
t1	Clock In Period (high to high)	19	—*	38	—*	16.5	—*	33	—*	ns
t2	Clock In Low Time (low to high)	6	—	17	—	6	—	15	—	ns
t3	Clock In High Time (high to low)	6	—	17	—	6	—	15	—	ns

* Device is fully static, t1 is tested at 100 ns for 1X input clock option or 200 ns for 2X input clock option, and memory hold time is tested at 0.1 s.

Table 62. Timing Characteristics for Input Clock and Output Clock

Abbreviated Reference	Parameter	Min	Max	Unit
t4	Clock Out High Delay	—	21	ns
t5	Clock Out Low Delay (high to low)	—	21	ns
t6	Clock Out Period (low to low)	T*	—	ns

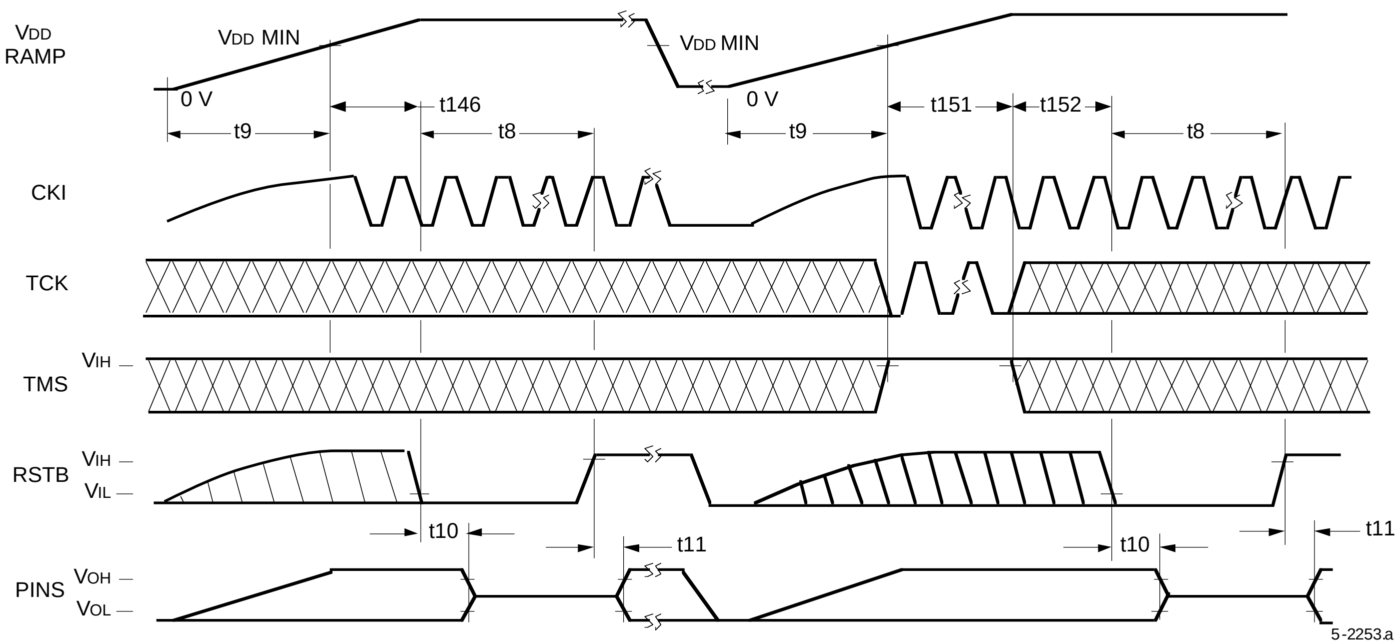
* T = 2 x t1 for 2X input clock option or T = t1 for 1X input clock option.

Timing Characteristics (continued)

10.2 Reset Circuit

The DSP1616 has a powerup reset circuit that automatically clears the JTAG controller upon powerup. If the supply voltage falls below VDD MIN* and a reset is required, the JTAG controller must be reset—even if the JTAG port isn't being used—by applying six low-to-high clock edges on TCK with TMS held high, followed by the usual RSTB and CKI reset sequence. Figure 10 shows two separate events: an initial powerup and a powerup following a drop in the power supply voltage.

* See Table 57, Recommended Operating Conditions.



Notes:
See Table 59 for CKI electrical requirements and Table 66 for TCK timing requirements.
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, EROM, ERAMHI, ERAMLO, IO, DSEL, and RWN outputs remain high, and CKO remains a free-running clock.
TMS and TCK signals have internal pull-up devices.

Figure 10. Powerup Reset and Chip Reset Timing Diagram

Table 63. Timing Requirements for Powerup Reset and Chip Reset

Abbreviated Reference	Parameter		Min	Max	Unit
t8	Reset Pulse (low to high)		6T	—	ns
t9	VDD Ramp		—	10	ms
t146	VDD MIN to RSTB Low	CMOS, TTL	2T	—	ns
		Crystal*	20	—	ms
		Small-Signal	20	—	µs
t151	TMS High		6 * TTCK†	—	ns
t152	JTAG Reset to RSTB Low	CMOS, TTL	2T	—	ns
		Crystal*	20 ms – 6 * TTCK if 6 * TTCK < 20 ms	—	
			0 if 6 * TTCK ≥ 20 ms	—	
		Small-Signal	20 µs – 6 * TTCK if 6 * TTCK < 20 µs	—	
			0 if 6 * TTCK ≥ 20 µs	—	

* With external components as specified in Table 59.
† TTCK = t12 = TCK period. See Table 66 for TCK timing requirements.

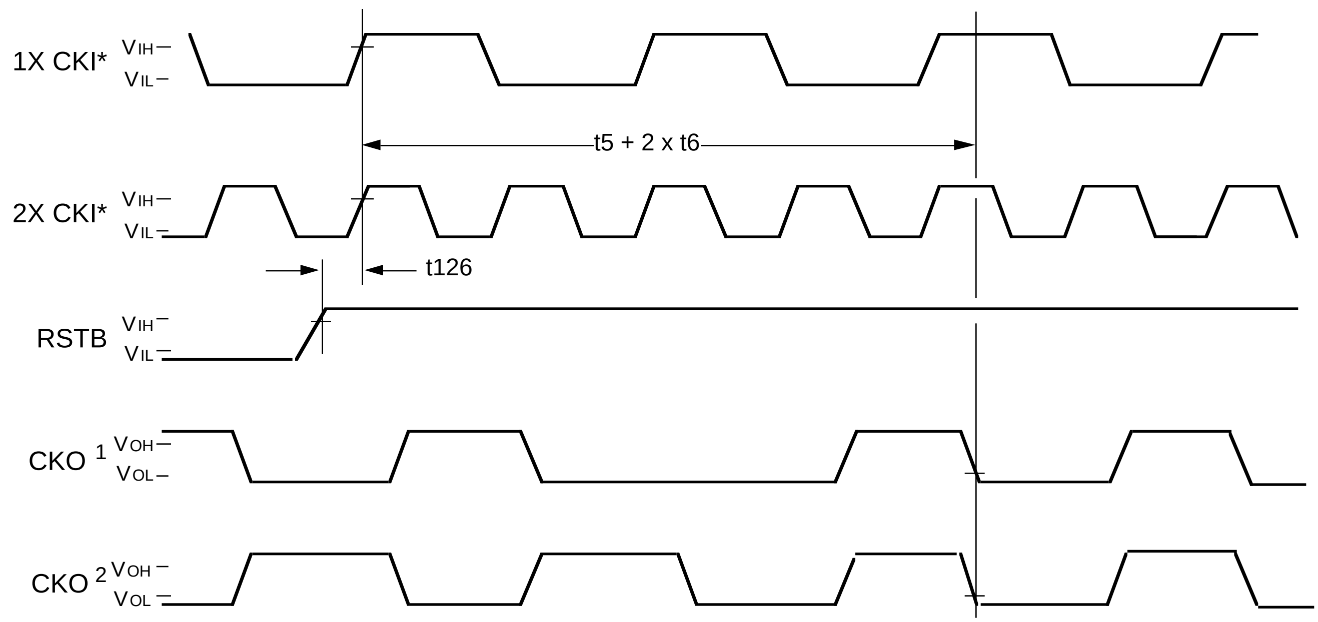
Timing Characteristics (continued)

Table 64. Timing Characteristics for Powerup 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 six CKI cycles with the 1X option or 12 CKI cycles with the 2X option during reset after powerup. Otherwise, high currents may flow.

10.3 Reset Synchronization



* See Table 59 for input clock electrical requirements.

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

Figure 11. Reset Synchronization Timing

Table 65. Timing Requirements for Reset Synchronization Timing

Abbreviated Reference	Parameter	38 ns		33 ns		Unit
		Min	Max	Min	Max	
t126	Reset Setup (high to high)	4	T/2 – 5	4	T/2 – 5	ns

Timing Characteristics (continued)

10.4 JTAG I/O Specifications

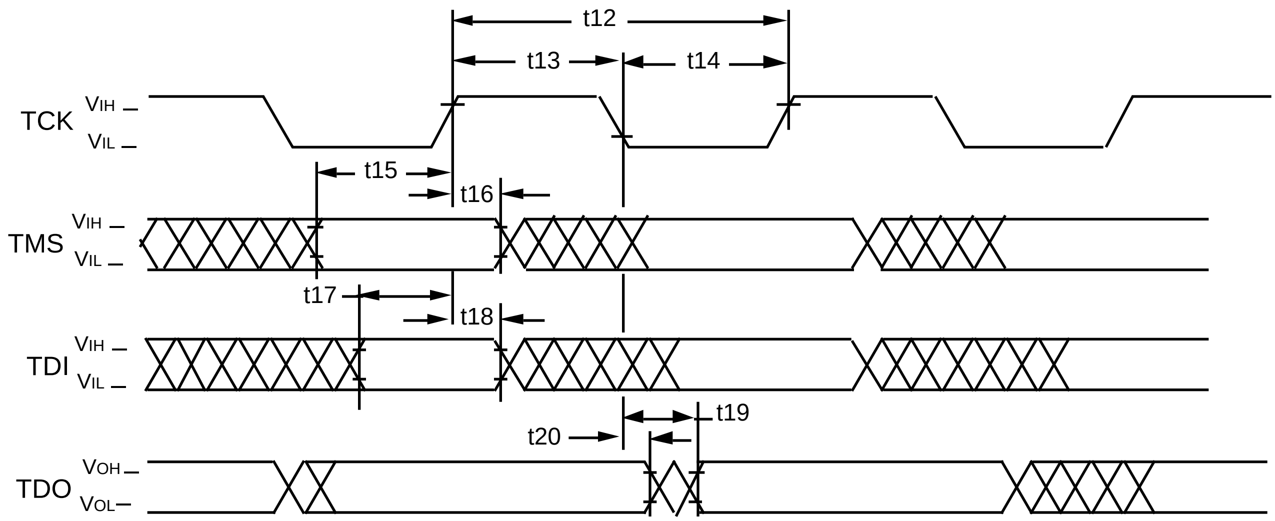


Figure 12. JTAG Timing Diagram

Table 66. Timing Requirements for JTAG Input/Output

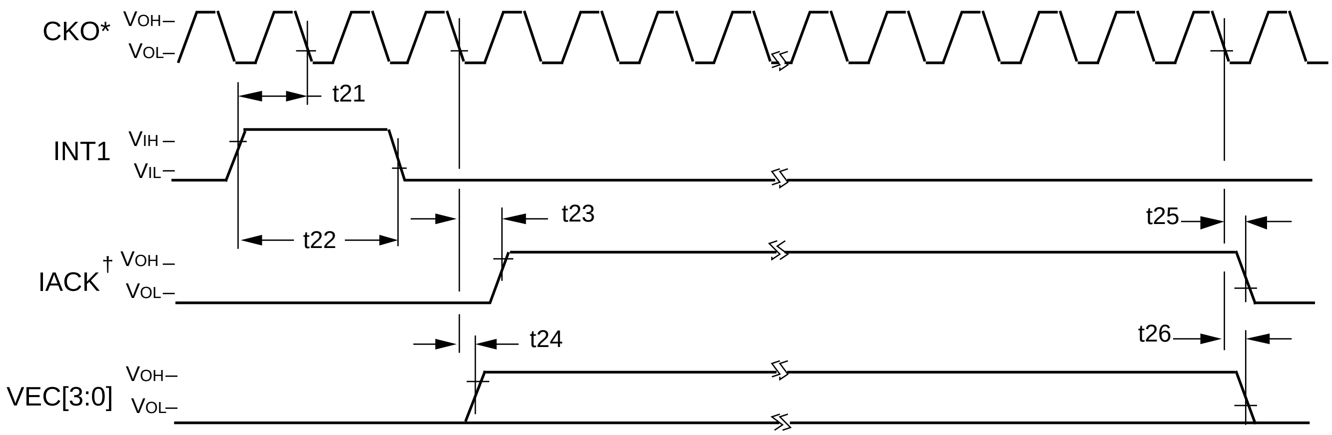
Abbreviated Reference	Parameter	Min	Max	Unit
t12	TCK Period (high to high)	66	—	ns
t13	TCK High Time (high to low)	30	—	ns
t14	TCK Low Time (low to high)	30	—	ns
t15	TMS Setup Time (valid to high)	10	—	ns
t16	TMS Hold Time (high to invalid)	3	—	ns
t17	TDI Setup Time (valid to high)	10	—	ns
t18	TDI Hold Time (high to invalid)	3	—	ns

Table 67. 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 assertion is guaranteed to be enclosed by VEC[3:0] assertion.

Figure 13. Interrupt Timing Diagram

Table 68. 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

Table 69. 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

Timing Characteristics (continued)

10.6 Bit Input/Output (BIO)

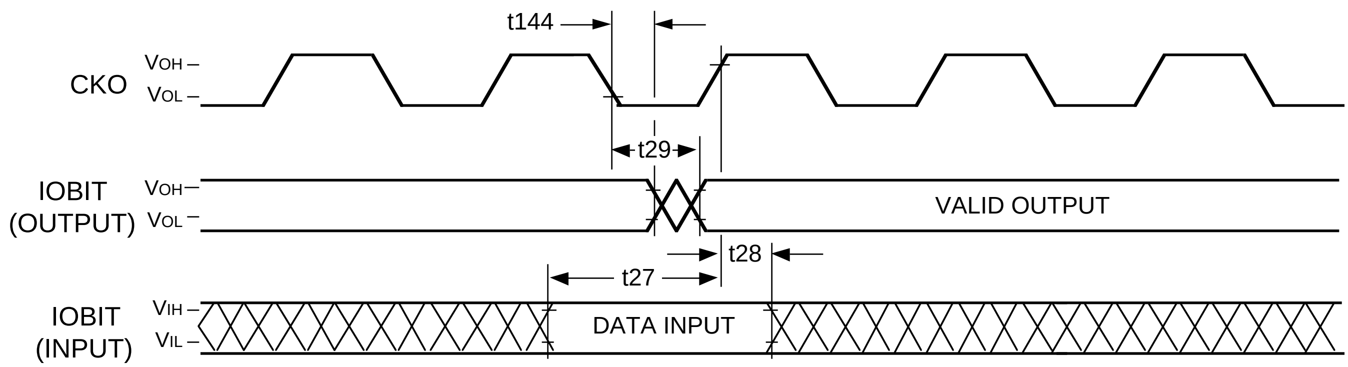


Figure 14. Write Outputs Followed by Read Inputs (cbit = Immediate; a1 = sbit)

Table 70. 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 71. Timing Characteristics for BIO Output

Abbreviated Reference	Parameter	Min	Max	Unit
t29	IOBIT Output Valid Time (low to valid)	—	10	ns
t144	IOBIT Output Hold Time (low to invalid)	0	—	ns

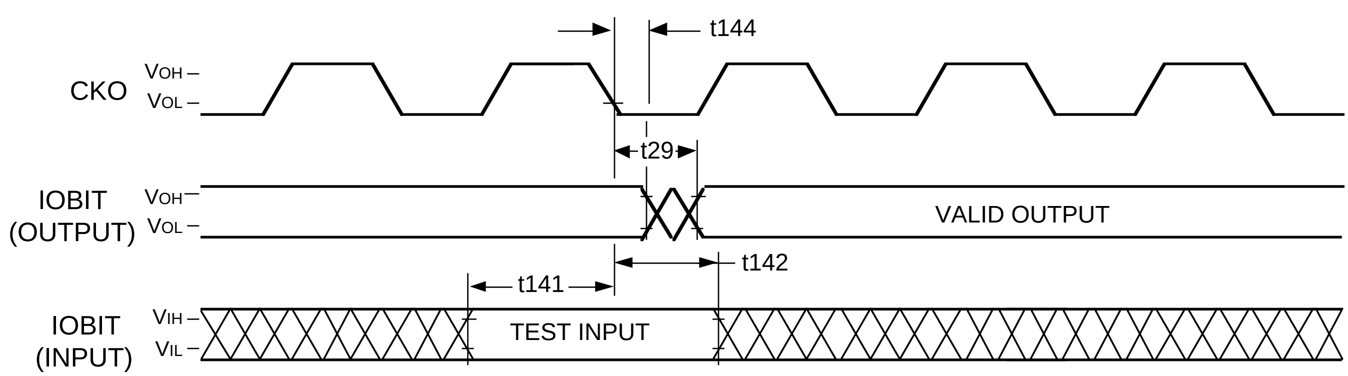


Figure 15. Write Outputs and Test Inputs (cbit = Immediate)

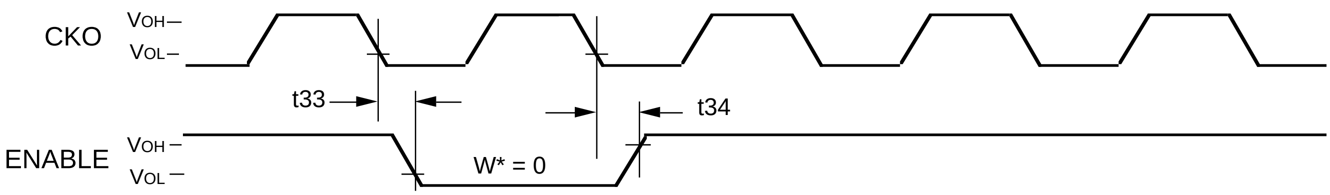
Table 72. Timing Requirements for BIO Input Test

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 *DSP1616 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 16. Enable Transition Timing

Table 73. Timing Characteristics for External Memory Enables (EROM, ERAMHI, IO, ERAMLO)

Abbreviated Reference	Parameter	Min	Max	Unit
t33	CKO to ENABLE Active (low to low)	0	5	ns
t34	CKO to ENABLE Inactive (low to high)	–2	5	ns

Table 74. Timing Characteristics for Device Enable (DSEL)

Abbreviated Reference	Parameter	Min	Max	Unit
t33	CKO to ENABLE Active (low to low)	0	6	ns
t34	CKO to ENABLE Inactive (low to high)	–1	6	ns

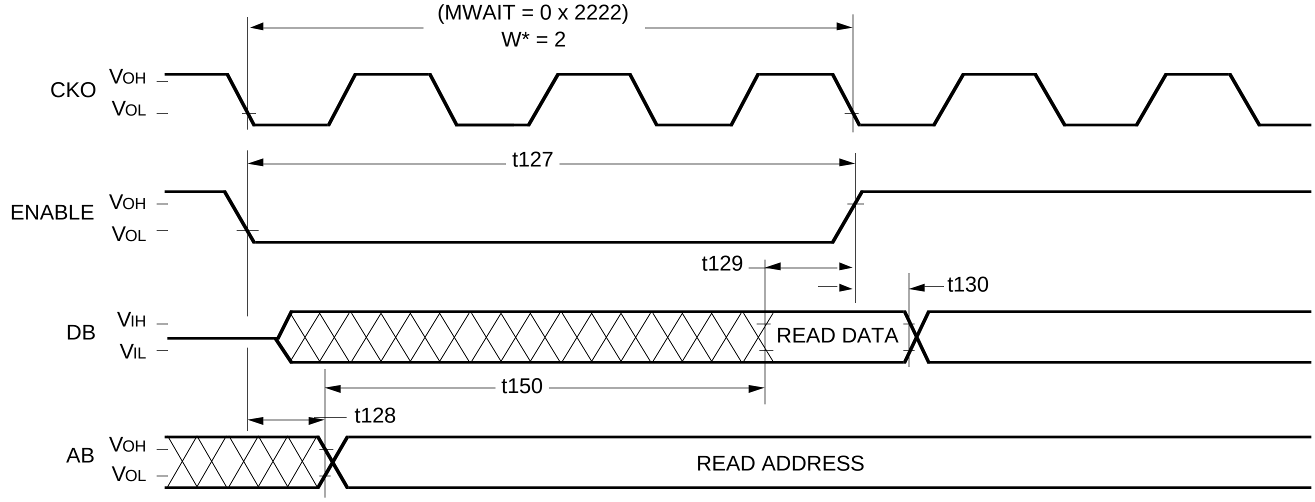
Table 75. Timing Characteristics for Delayed External Memory Enables (ioc = 0x000F)

Abbreviated Reference	Parameter	Min	Max	Unit
t33	CKO to Delayed ENABLE Active (low to low)	T/2 – 2	T/2 + 9	ns

Table 76. Timing Characteristics for Delayed Device Enable (ioc = 0x0010)

Abbreviated Reference	Parameter	Min	Max	Unit
t33	CKO to Delayed ENABLE Active (low to low)	T/2 – 1	T/2 + 10	ns

Timing Characteristics (continued)



* W = number of wait-states.

Figure 17. External Memory Data Read Timing Diagram

Table 77. Timing Characteristics for External Memory Access

Abbreviated Reference	Parameter	Min	Max	Unit
t127	Enable Width (low to high)	T(1 + W) – 2	—	ns
t128	Address Valid (enable low to valid)	—	3	ns

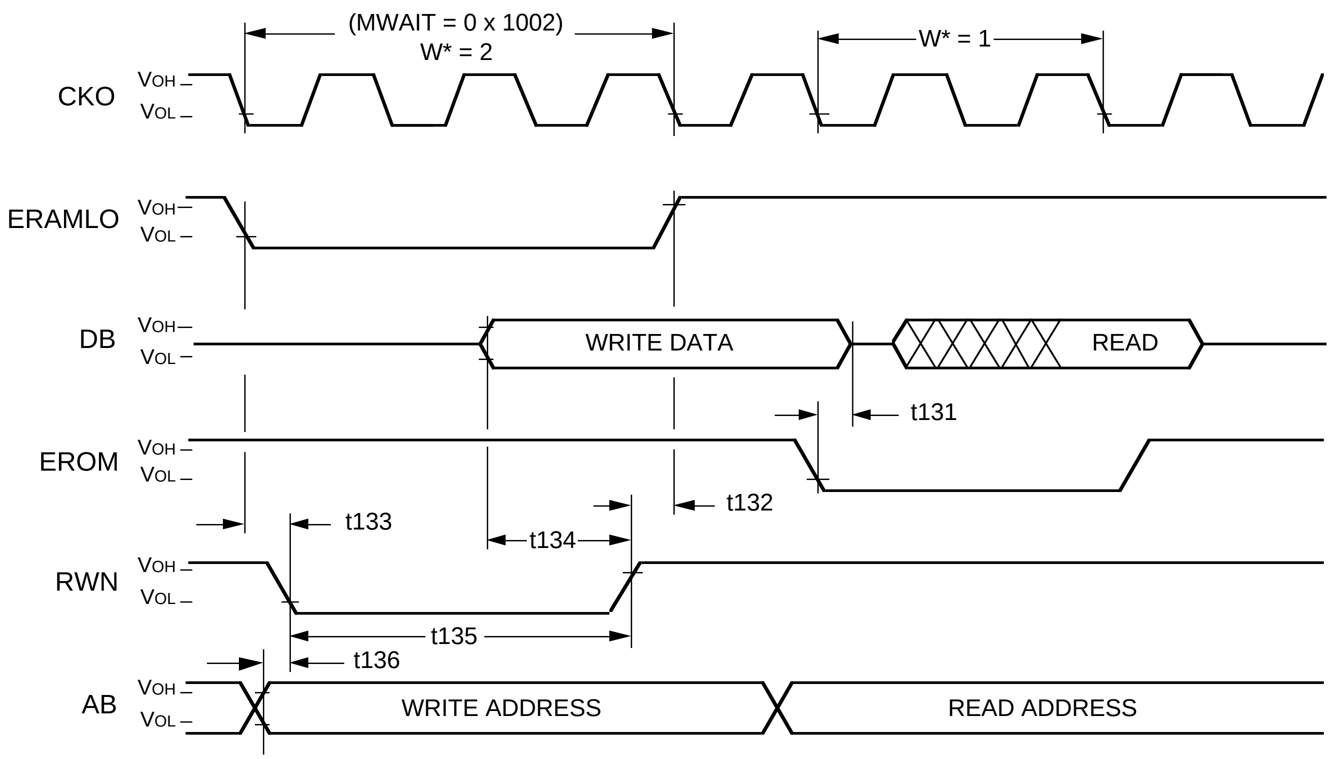
Table 78. Timing Requirements for External Memory Read (EROM, ERAMHI, IO, ERAMLO)

Abbreviated Reference	Parameter	Min	Max	Unit
t129	Read Data Setup (valid to enable high)	14	—	ns
t130	Read Data Hold (enable high to hold)	0	—	ns
t150	External Memory Access (valid to valid)	—	T(1 + W) – 17	ns

Table 79. Timing Requirements for Device Read (DSEL)

Abbreviated Reference	Parameter	Min	Max	Unit
t129	Read Data Setup (valid to enable high)	15	—	ns
t130	Read Data Hold (enable high to hold)	0	—	ns
t150	External Memory Access (valid to valid)	—	T(1 + W) – 18	ns

Timing Characteristics (continued)



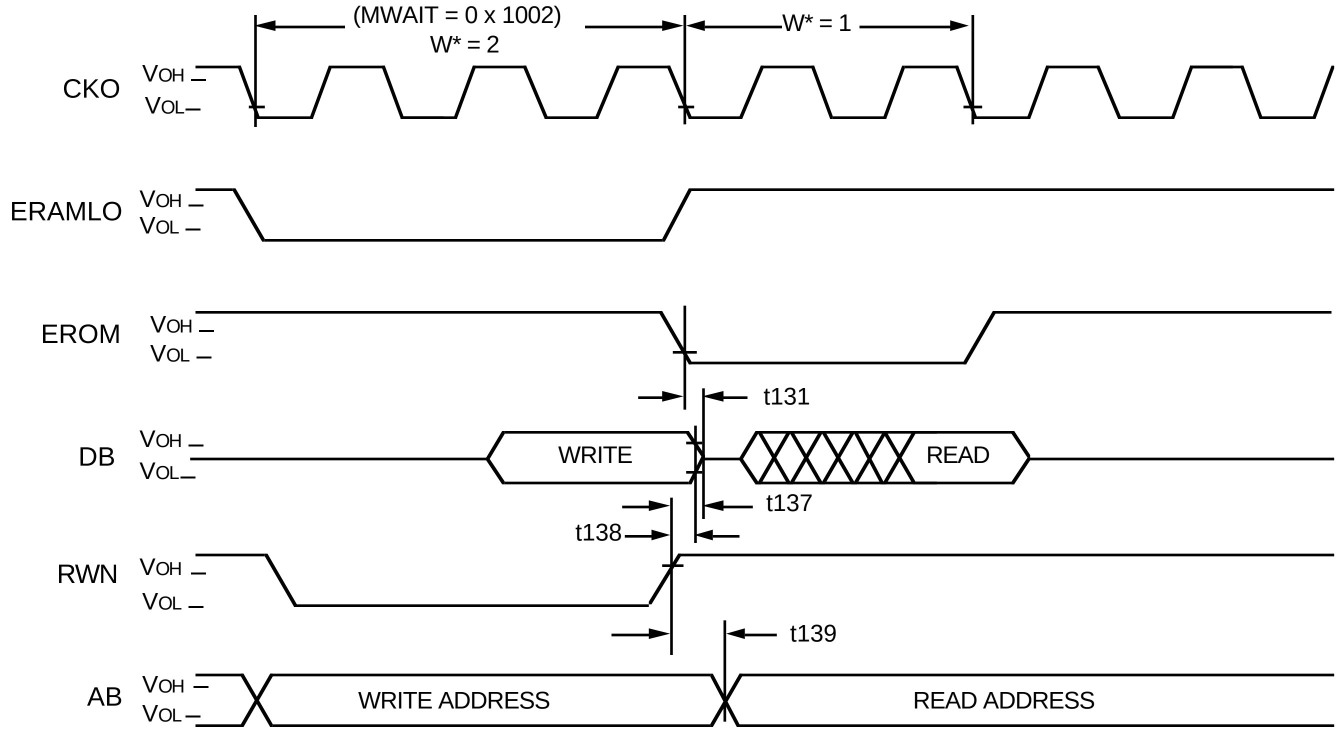
* W = number of wait-states.

Figure 18. External Memory Data Write Timing Diagram

Table 80. 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 – 6.5	—	ns
t135	RWN Width (low to high)	T(1 + W) – 5	—	ns
t136	Write Address Setup (address valid to RWN low)	0	—	ns

Timing Characteristics (continued)



* W = number of wait-states.

Figure 19. Write Cycle Followed by Read Cycle

Table 81. Timing Characteristics for Write Cycle Followed by Read Cycle

Abbreviated Reference	Parameter	Min	Max	Unit
t131	Write Overlap (enable low to 3-state)	—	0	ns
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)

10.8 PIO Specifications

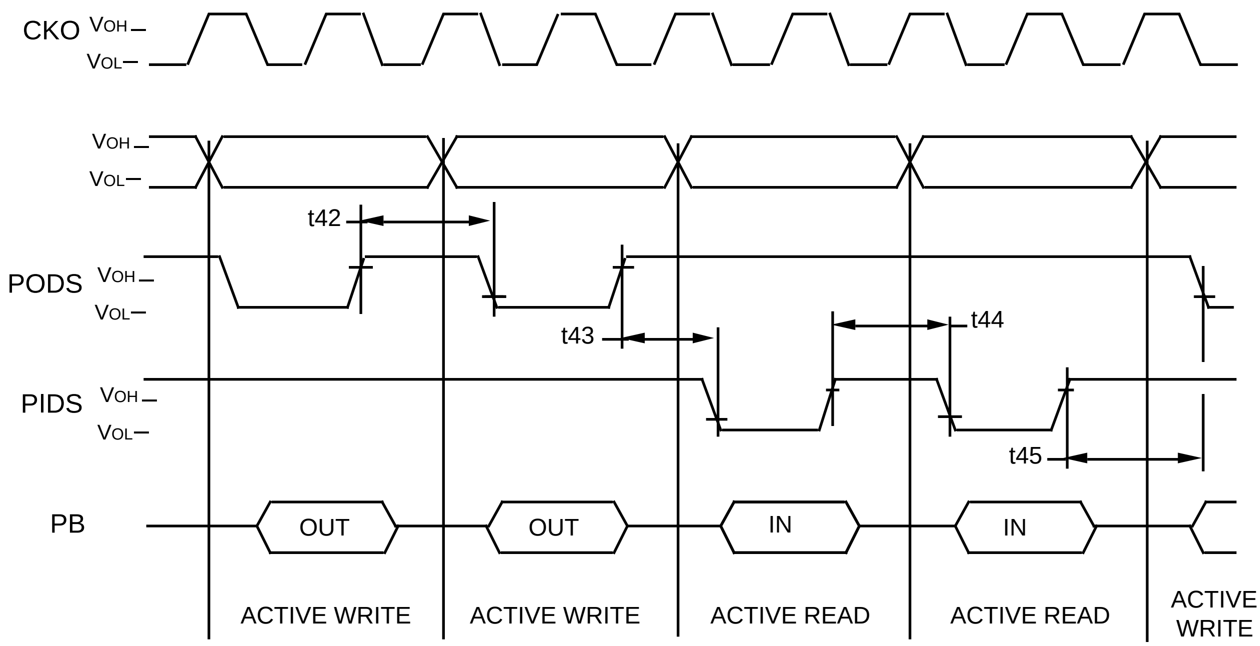


Figure 20. Active Mode Interaccess Timing

Table 82. Timing Requirements for PIO Active Mode Interaccess

Abbreviated Reference	Parameter	Min	Max	Unit
t42	PODS High to PODS Low	T – 4	—	ns
t43	PODS High to PIDS Low	T – 4	—	ns
t44	PIDS High to PIDS Low	T – 4	—	ns
t45	PIDS High to PODS Low	T – 4	—	ns

Timing Characteristics (continued)

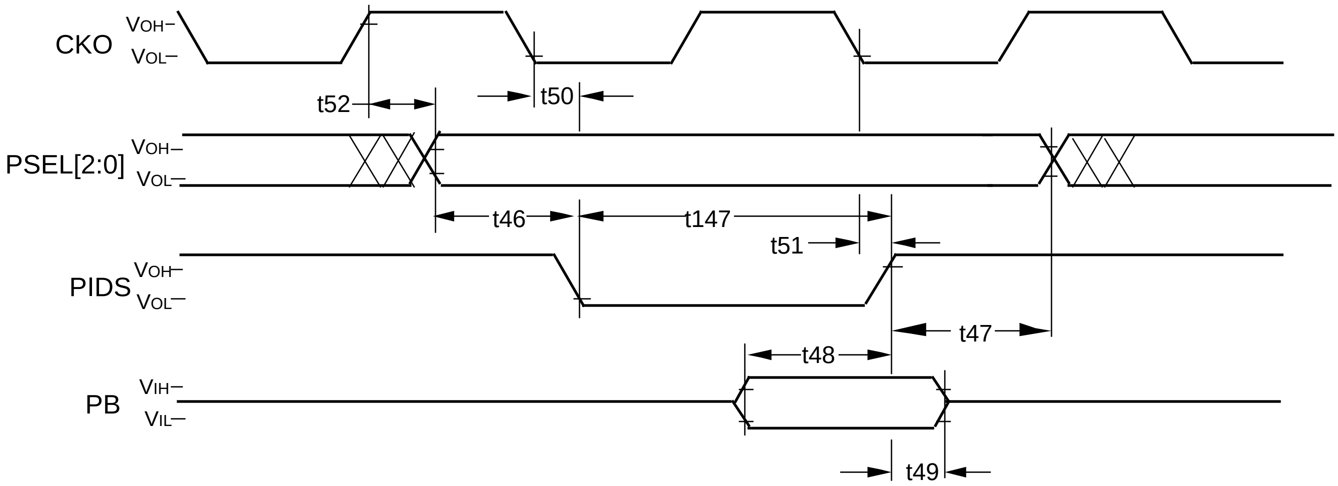


Figure 21. PIO Active Mode Input Timing Diagram

Table 83. Timing Requirements for PIO Active Mode Input

Abbreviated Reference	Parameter	Min	Max	Unit
t48	PB[7:0] Setup Time (valid to high)	8	—	ns
t49	PB[7:0] Hold Time (high to invalid)	0	—	ns

Table 84. Timing Characteristics for PIO Active Mode Input

Abbreviated Reference	Parameter	Min	Max	Unit
t46	PSEL[2:0] Valid to PIDS Low	T/2 – 2	—	ns
t47	PIDS High to PSEL[2:0] Invalid	T/2 – 2	—	ns
t147	PIDS Width (low to high)	T – 4		
t50	CKO Low to PIDS Assertion (low to low)	—	12	ns
t51	CKO Low to PIDS Negation (low to high)	—	10	ns
t52	CKO High to PSEL[2:0] Valid (high to valid)	—	10	ns

Timing Characteristics (continued)

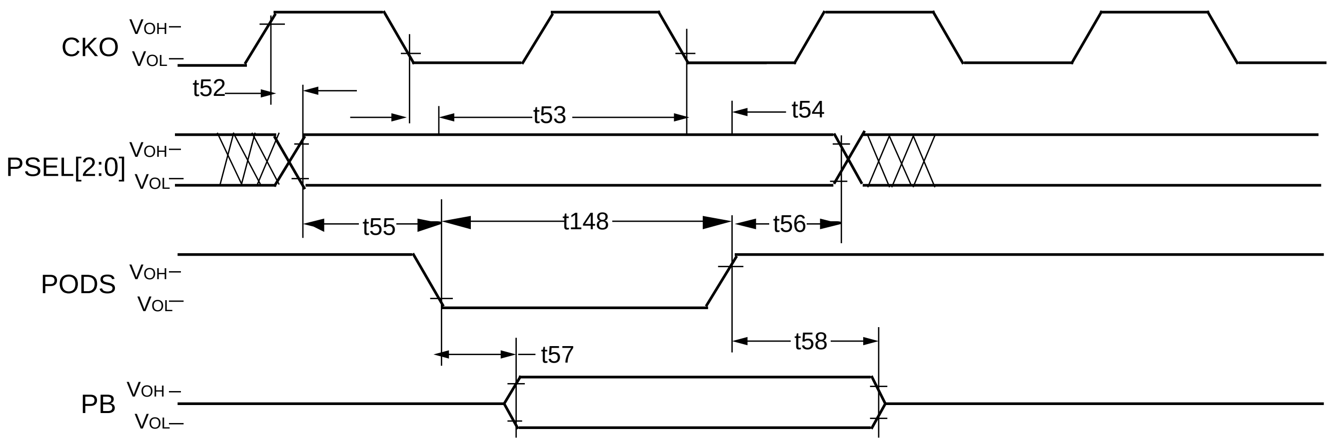


Figure 22. PIO Active Mode Output Timing Diagram

Table 85. Timing Characteristics for PIO Active Mode Output

Abbreviated Reference	Parameter	Min	Max	Unit
t53	CKO Low to PODS Assertion (low to low)	—	12	ns
t54	CKO Low to PODS Negation (low to high)	—	10	ns
t55	PSEL[2:0] Valid to PODS low	T/2 – 2	—	ns
t56	PODS High to PSEL[2:0] Invalid	T/2 – 2	—	ns
t57	PODS Low to PB Valid (low to valid)	—	12	ns
t58	PB Hold (high to invalid)	T/2 – 8	—	ns
t148	PODS Width (low to high)	T – 4		ns

Timing Characteristics (continued)

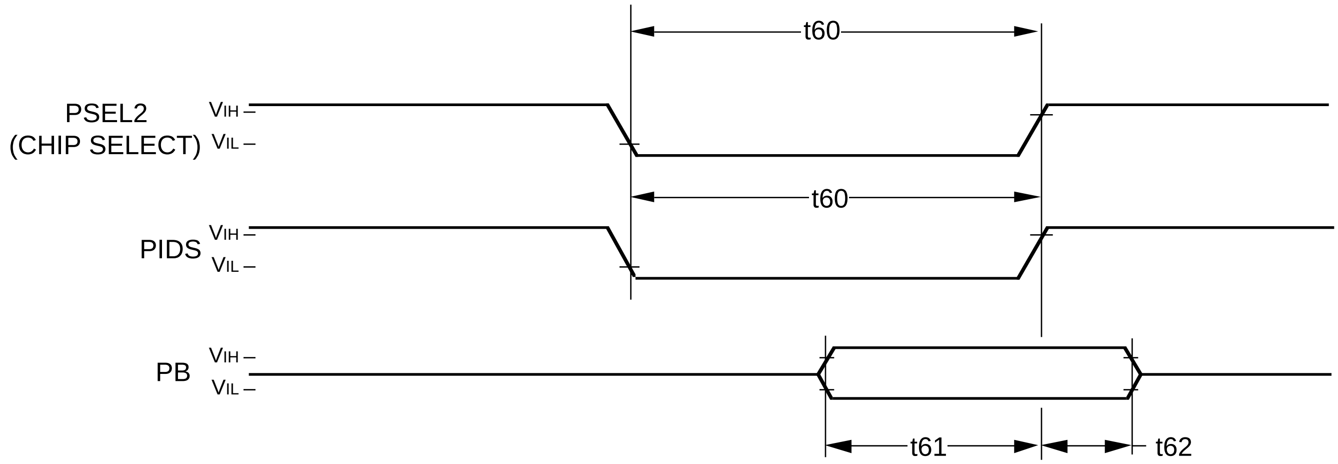


Figure 23. PIO Passive Mode Input Timing Diagram

Table 86. Timing Requirements for PIO Passive Mode Input

Abbreviated Reference	Parameter	Min	Max	Unit
t60	PIDS or PSEL2 Pulse Width (low to high)*	T	—	ns
t61	PB[7:0] Setup Time† (valid to high)	8	—	ns
t62	PB[7:0] 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.

Timing Characteristics (continued)

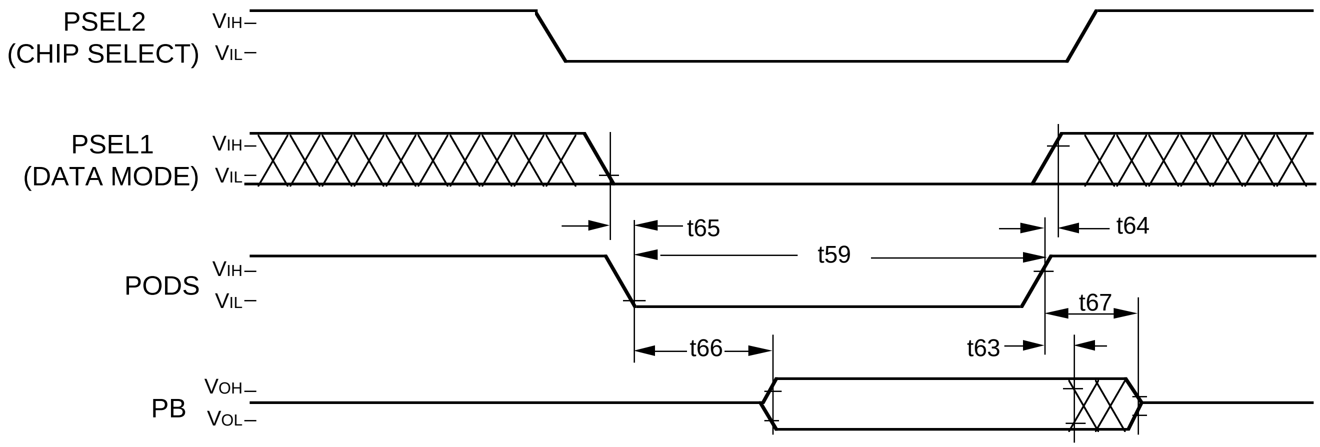


Figure 24. PIO Passive Mode Output Timing Diagram

Table 87. Timing Requirements for PIO Passive Mode Output

Abbreviated Reference	Parameter	Min	Max	Unit
t59	PSEL2, PSEL1, or PODS Pulse Width (low to high)	T		ns
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 88. Timing Characteristics for PIO Passive Mode Output

Abbreviated Reference	Parameter	Min	Max	Unit
t66	PODS Low* to PB Valid	—	28	ns
t67	PODS High† to PB 3-state	—	20	ns
t63	PB Hold (high to invalid)	6	—	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)

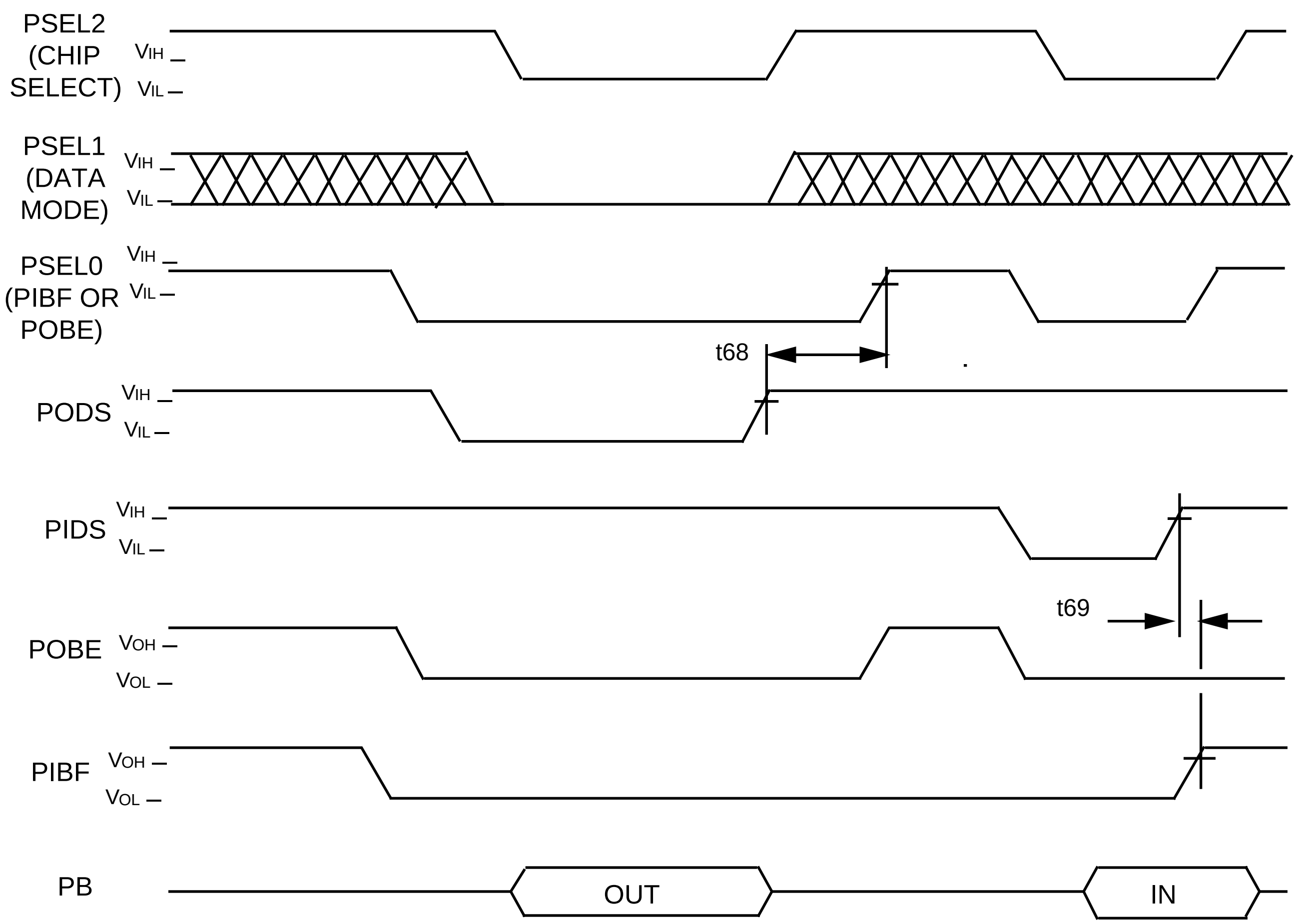


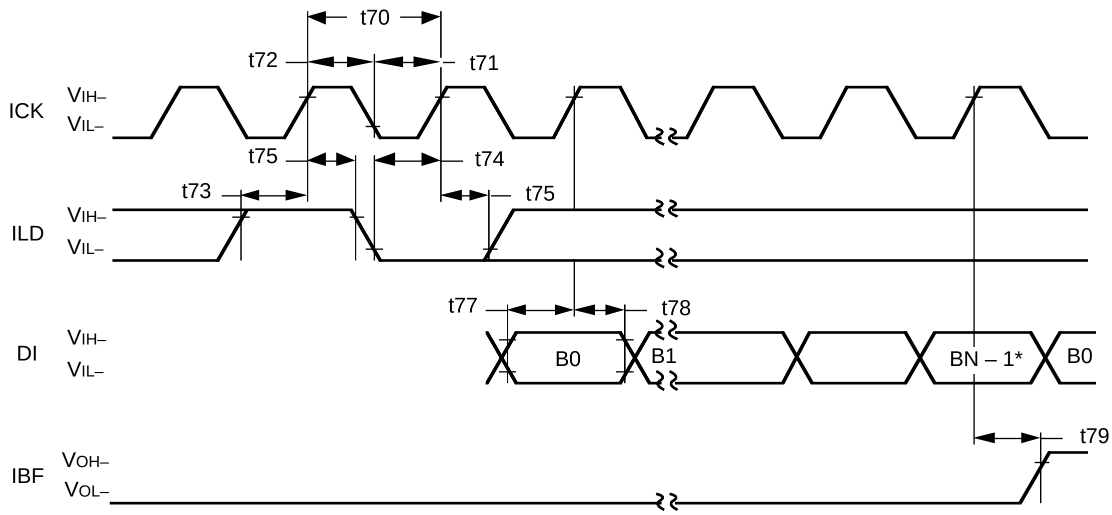
Figure 25. PIO Peripheral Mode Input/Output Timing Diagram

Table 89. Timing Characteristics for PIO 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 High	—	2T	ns

Timing Characteristics (continued)

10.9 Serial I/O Specifications



* N = 16 or 8 bits.

Figure 26. SIO Passive Mode Input Timing Diagram

Table 90. Timing Requirements for Serial Inputs

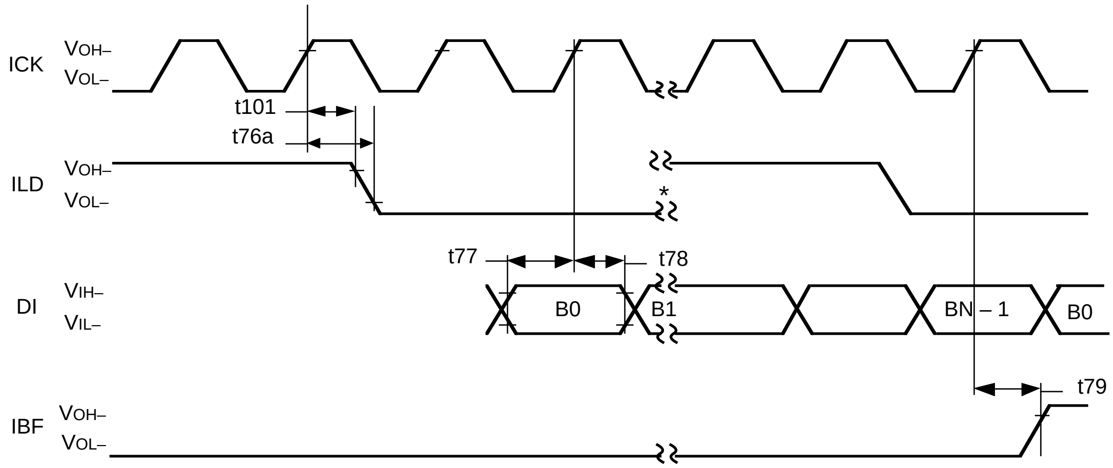
Abbreviated Reference	Parameter	38 ns		33 ns		Unit
		Min	Max	Min	Max	
t70	Clock Period (high to high)*	76	—†	66	—†	ns
t71	Clock Low Time (low to high)	35	—	30	—	ns
t72	Clock High Time (high to low)	35	—	30	—	ns
t73	Load High Setup (high to high)	6	—	6	—	ns
t74	Load Low Setup (low to high)	6	—	6	—	ns
t75	Load High Hold (high to invalid)	2	—	2	—	ns
t77	Data Setup (valid to high)	5	—	5	—	ns
t78	Data Hold (high to invalid)	5	—	5	—	ns

* For multiprocessor mode, see note in Section 10.10.
† Device is fully static; t70 is tested at 200 ns.

Table 91. Timing Characteristics for Serial Outputs

Abbreviated Reference	Parameter	Min	Max	Unit
t79	IBF Delay (high to high)	—	35	ns

Timing Characteristics (continued)



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

Figure 27. SIO Active Mode Input Timing Diagram

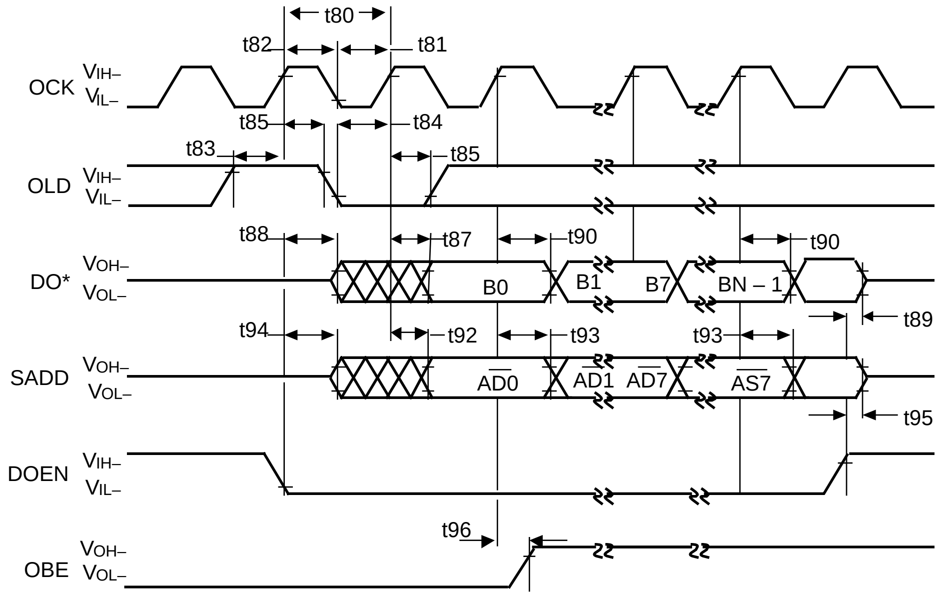
Table 92. Timing Requirements for Serial Inputs

Abbreviated Reference	Parameter	Min	Max	Unit
t77	Data Setup (valid to high)	5	—	ns
t78	Data Hold (high to invalid)	5	—	ns

Table 93. Timing Characteristics for Serial Outputs

Abbreviated Reference	Parameter	Min	Max	Unit
t76a	ILD Delay (high to low)	—	35	ns
t101	ILD Hold (high to invalid)	5	—	ns
t79	IBF Delay (high to high)	—	35	ns

Timing Characteristics (continued)



* 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 bits or 16 bits.

Figure 28. SIO Passive Mode Output Timing Diagram

Table 94. Timing Requirements for Serial Inputs

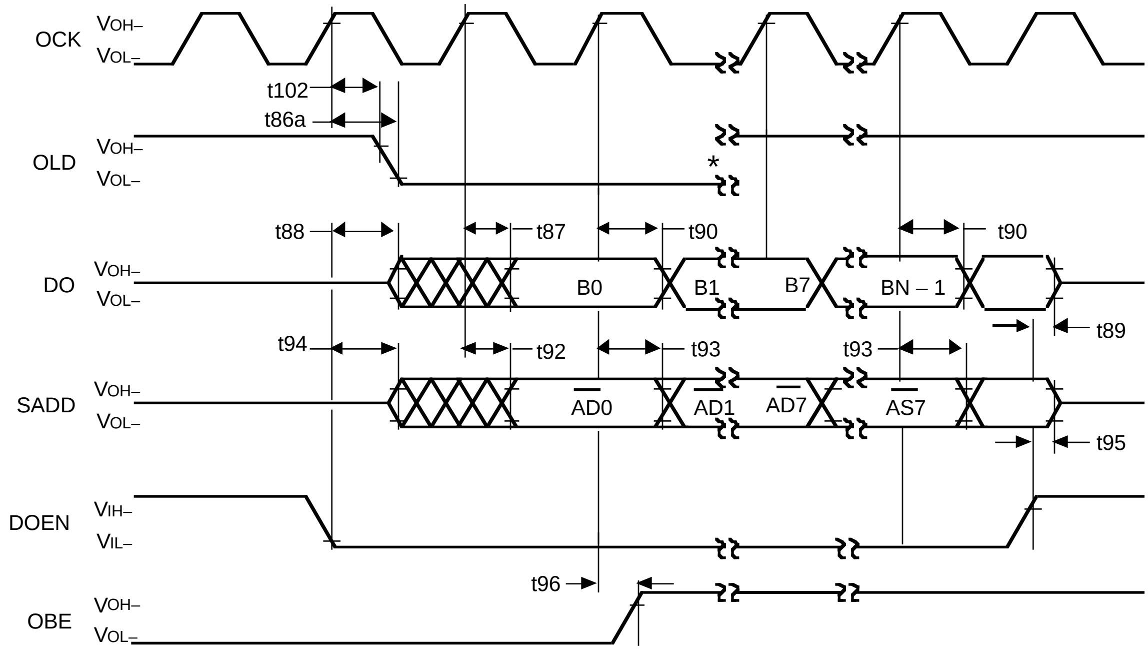
Abbreviated Reference	Parameter	38 ns		33 ns		Unit
		Min	Max	Min	Max	
t80	Clock Period (high to high)*	76	—†	66	—†	ns
t81	Clock Low Time (low to high)	35	—	30	—	ns
t82	Clock High Time (high to low)	35	—	30	—	ns
t83	Load High Setup (high to high)	6	—	6	—	ns
t84	Load Low Setup (low to high)	6	—	6	—	ns
t85	Load Hold (high to invalid)	2	—	2	—	ns

* For multiprocessor mode, see note in Section 10.10.
† Device is fully static; t80 is tested at 200 ns.

Table 95. 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)	6	—	ns
t92	Address Delay (high to valid)	—	35	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)	—	35	ns

Timing Characteristics (continued)



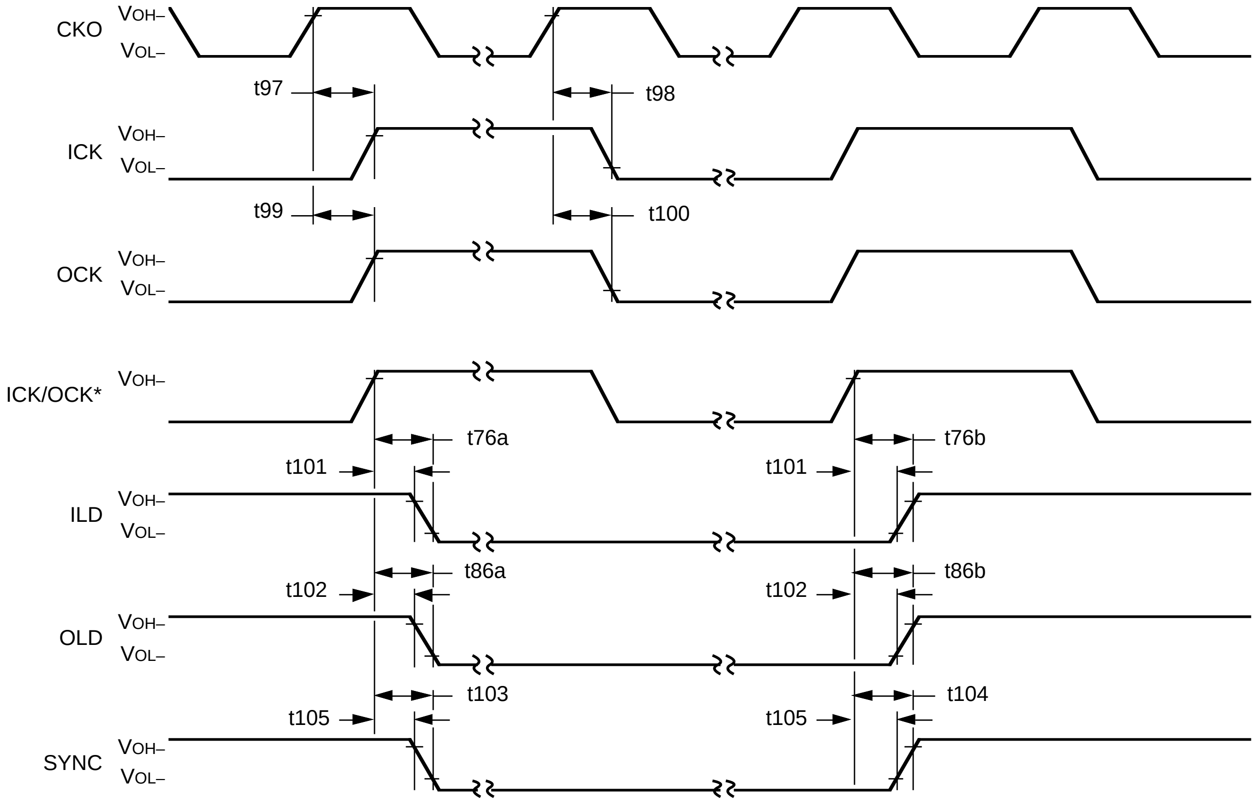
* OLD goes high at the end of bit 6 of 0:15.

Figure 29. SIO Active Mode Output Timing Diagram

Table 96. Timing Characteristics for Serial Outputs

Abbreviated Reference	Parameter	Min	Max	Unit
t86a	OLD Delay (high to low)	—	35	ns
t102	OLD Hold (high to invalid)	5	—	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)	6	—	ns
t92	Address Delay (high to valid)	—	35	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)	—	35	ns

Timing Characteristics (continued)



* See **sioc** register, LD field.

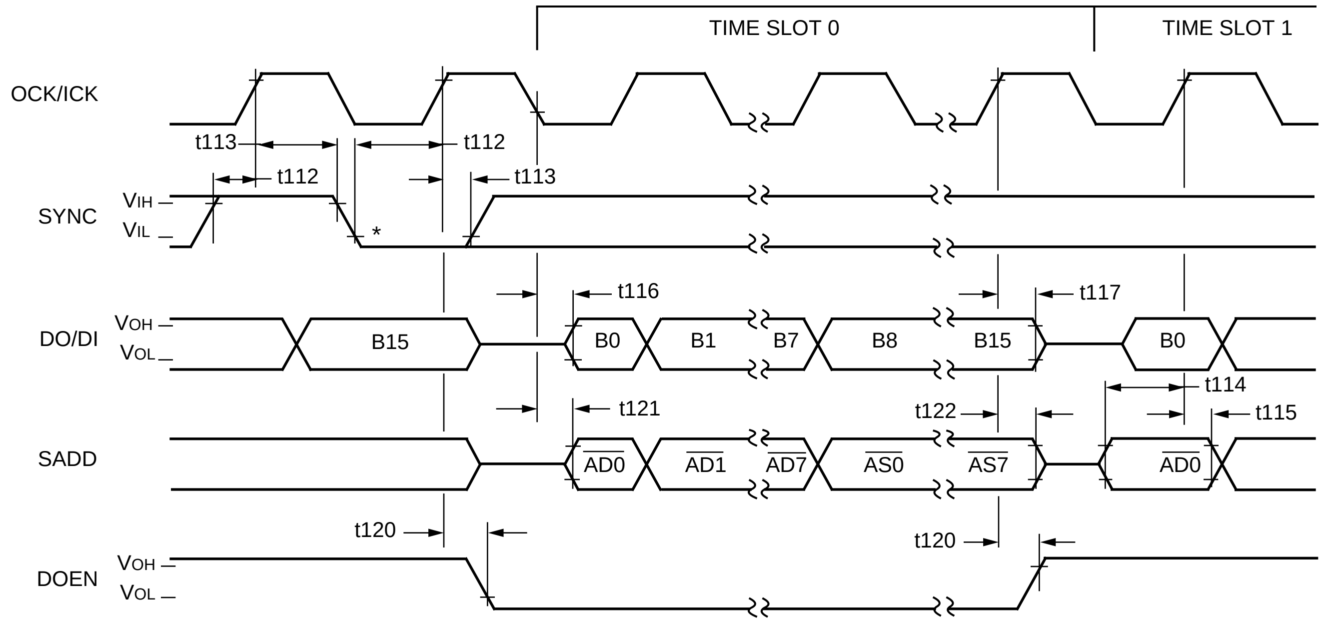
Figure 30. Serial I/O Active Mode Clock Timing

Table 97. Timing Characteristics for Signal Generation

Abbreviated Reference	Parameter	Min	Max	Unit
t97	ICK Delay (high to high)	—	18	ns
t98	ICK Delay (high to low)	—	18	ns
t99	OCK Delay (high to high)	—	18	ns
t100	OCK Delay (high to low)	—	18	ns
t76a	ILD Delay (high to low)	—	35	ns
t76b	ILD Delay (high to high)	—	35	ns
t101	ILD Hold (high to invalid)	5	—	ns
t86a	OLD Delay (high to low)	—	35	ns
t86b	OLD Delay (high to high)	—	35	ns
t102	OLD Hold (high to invalid)	5	—	ns
t103	SYNC Delay (high to low)	—	35	ns
t104	SYNC Delay (high to high)	—	35	ns
t105	SYNC Hold (high to invalid)	5	—	ns

Timing Characteristics (continued)

10.10 Multiprocessor Communication



* Negative edge initiates time slot 0.

Figure 31. SIO Multiprocessor Timing Diagram

Note: All serial I/O timing requirements and characteristics still apply, except the minimum clock period in passive multiprocessor mode, assuming 50% duty cycle, is calculated as (t77 + t116) * 2.

Table 98. Timing Requirements for SIO Multiprocessor Communication

Abbreviated Reference	Parameter	Min	Max	Unit
t112	Sync Setup (high/low to high)	30	—	ns
t113	Sync Hold (high to high/low)	9	—	ns
t114	Address Setup (valid to high)	10	—	ns
t115	Address Hold (high to invalid)	4	—	ns

Table 99. Timing Characteristics for SIO Multiprocessor Communication

Abbreviated Reference*	Parameter	Min	Max	Unit
t116	Data Delay (bit 0 only) (low to valid)	—	35	ns
t117	Data Disable Delay (high to 3-state)	—	30	ns
t120	DOEN Valid Delay (high to valid)	—	25	ns
t121	Address Delay (bit 0 only) (low to valid)	—	35	ns
t122	Address Disable Delay (high to 3-state)	—	30	ns

* With capacitance load on ICK, OCK, DO, SYNC, and SADD = 100 pF, add 4 ns to t116—t122.

11 Crystal Electrical Characteristics and Requirements

If the option for using the external crystal is chosen, the following electrical characteristics and requirements apply.

11.1 External Components for the Crystal Oscillator

The crystal oscillator is enabled by connecting a crystal across CKI and CKI2, along with one external capacitor from each of these pins to ground (see Figure 32). For most applications, 10 pF external capacitors are recommended; however, larger values may be necessary if precise frequency tolerance is required (see Section 11.3, Frequency Accuracy Considerations). The crystal should be either fundamental or overtone mode, parallel resonant, with a rated power of at least 1 mW and be specified at a load capacitance equal to the total capacitance seen by the crystal (including external capacitors and strays). The series resistance of the crystal should be specified to be less than **half** the absolute value of the negative resistance shown in Figure 33 for the crystal frequency. The frequency of both the internal processor clock with the 1X CKI input clock option and the output clock CKO will be equal to the crystal frequency.

The drive level, or power dissipated by the crystal itself, must be specified when ordering a crystal. This parameter depends heavily on the application circuit in which the crystal is used. Drive level is affected by supply voltage, crystal resistance, and load capacitance. Although crystals with a minimum specified drive level of at least 1 mW will work in most applications, this parameter should be measured in the actual circuit to make sure that the crystal is not over stressed.

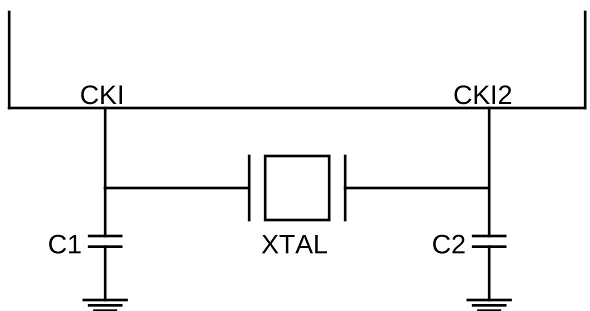


Figure 32. Fundamental Crystal Configuration

The following guidelines should be followed when designing the printed-circuit board layout for a crystal-based application:

1. Keep crystal and external capacitors as close to CKI and CKI2 pins as possible to minimize board stray capacitance.
2. Keep high-frequency digital signals such as CKO away from CKI and CKI2 traces to avoid coupling.

11.2 Power Dissipation

Figure 34 indicates the typical power dissipation of the on-chip crystal oscillator circuit versus frequency. Note that this curve shows the relative effects of load capacitance on supply current and that the actual supply current measured will depend on crystal resistance. For typical crystals, measured supply current at the V_{DDA} pin should be less than that shown in the figures.

Crystal Electrical Characteristics and Requirements (continued)

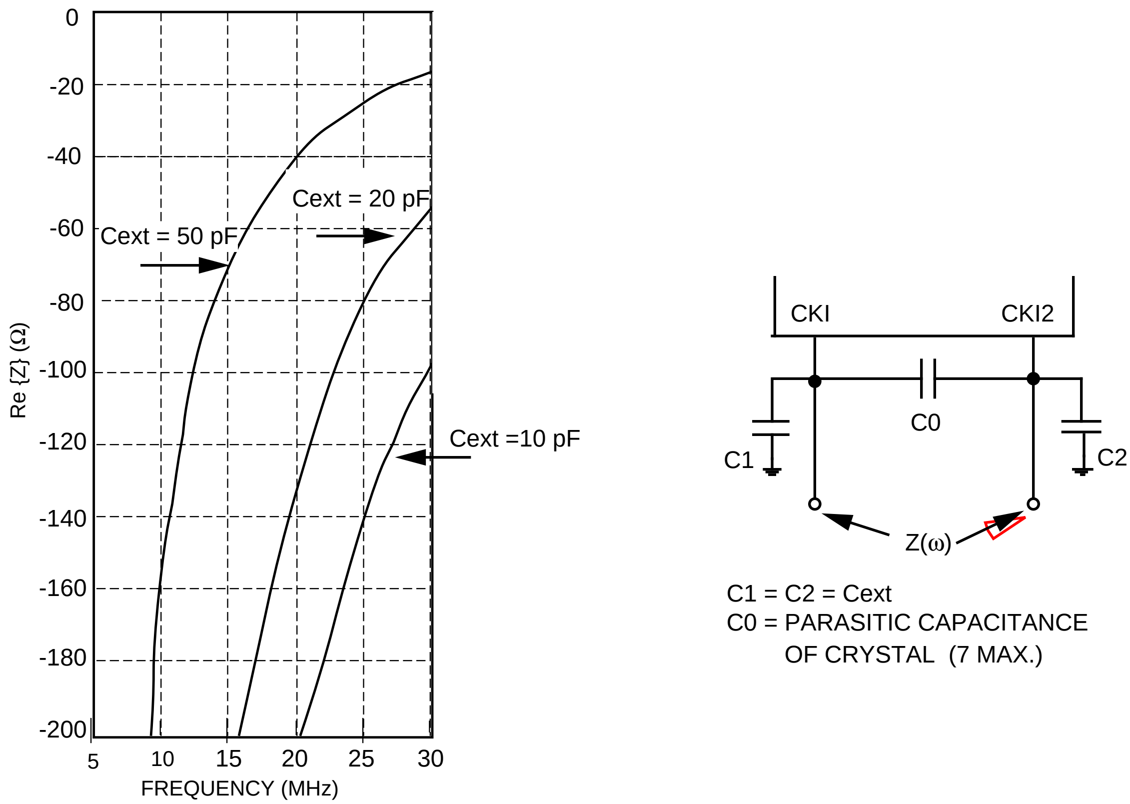


Figure 33. Negative Resistance of Crystal Oscillator Circuit, VDD = 4.75 V

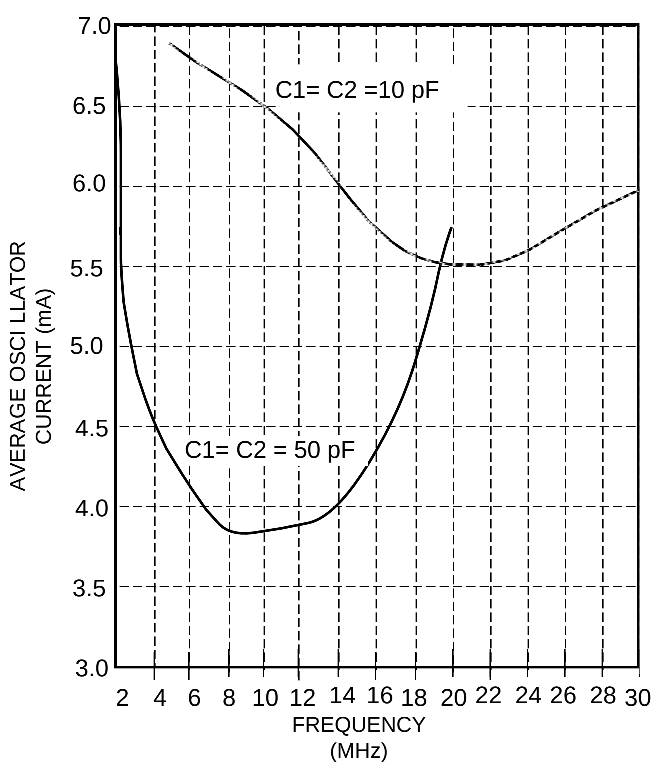


Figure 34. Typical Supply Current of Crystal Oscillator Circuit, VDD = 5.0 V, 25 °C

Crystal Electrical Characteristics and Requirements (continued)

11.3 LC Network Design for Third Overtone Crystal Circuits

For operating frequencies greater than 30 MHz, it is usually cheaper to use a third overtone crystal instead of a fundamental mode crystal. When using third overtone crystals, it is necessary, however, to filter out the fundamental frequency so that the circuit will oscillate only at the third overtone. There are several techniques that will accomplish this; one of these is described below. Figure 35 shows the basic setup for third overtone operation.

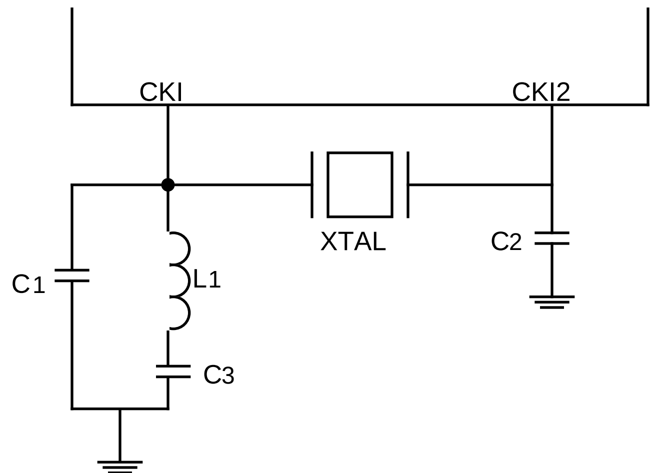


Figure 35. Third Overtone Crystal Configuration

The parallel combination of L_1 and C_1 forms a resonant circuit with a resonant frequency between the first and third harmonic of the crystal such that the LC network appears inductive at the fundamental frequency and capacitive at the third harmonic. This ensures that a 360° phase shift around the oscillator loop will occur at the third overtone frequency but not at the fundamental. The blocking capacitor, C_3, provides dc isolation for the trap circuit and should be chosen to be large compared to C_1.

For example, suppose it is desired to operate with a 40 MHz, third overtone, crystal:

Let:
- f_3 = operating frequency of third overtone crystal (40 MHz in this example)
- f_1 = fundamental frequency of third overtone crystal, or $f_3/3$ (13.3 MHz in this example)
- f_T = resonant frequency of trap = $\frac{1}{2\pi\sqrt{L_1C_1}}$
- C_2 = external load capacitor (10 pF in this example)
- C_3 = dc blocking capacitor (0.1 µF in this example)

Arbitrarily, set trap resonance to geometric mean of f_1 and f_3. Since $f_1 = f_3/3$, the geometric mean would be:

$$f_T = \frac{f_3}{\sqrt{3}} = \frac{40\ \text{MHz}}{\sqrt{3}} = 23\ \text{MHz}$$

Crystal Electrical Characteristics and Requirements (continued)

At the third overtone frequency, f_3, it is desirable to have the net impedance of the trap circuit (X_T) equal to the impedance of C_2 (X_{C2}), i.e.,

$$X_T = X_{C2} = X_{C1} \parallel (X_{C3} + X_{L1})$$

Selecting C_3 so that $X_{C3} << X_{L1}$ yields,

$$X_T = X_{C2} = X_{C1} \parallel X_{L1}$$

For a capacitor,

$$X_C = \frac{-j}{\omega C} \quad \text{where} \quad \omega = 2\pi f.$$

For an inductor,

$$X_L = j\omega L$$

Solving for C_1, and realizing that $L_1C_1 = 3/(\omega_3)^2$ yields,

$$C_1 = \frac{3}{2} C_2$$

Hence, for C_2 = 10 pF, C_1 = 15 pF. Since the impedance of the trap circuit in this example would be equal to the impedance of a 10 pF capacitor, the negative resistance and supply current curves for $C_1 = C_2$ = 10 pF at 40 MHz would apply to this example.

Finally, solving for the inductor value,

$$L_1 = \frac{1}{4\pi^2(f_T)^2 C_1}$$

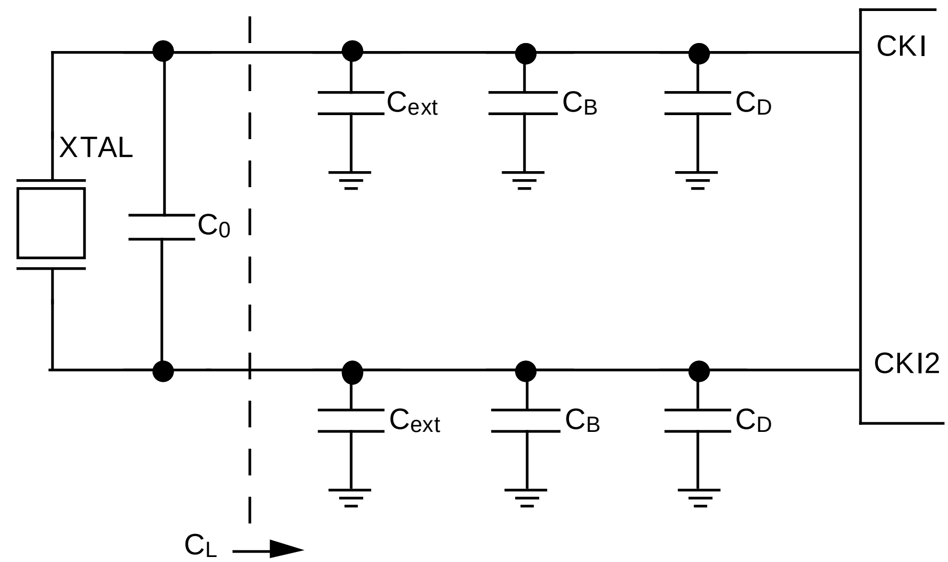
For the above example, L_1 is 3.2 µH.

Crystal Electrical Characteristics and Requirements (continued)

11.4 Frequency Accuracy Considerations

For most applications, clock frequency errors in the hundreds of parts per million can be tolerated with no adverse effects. However, for applications where precise frequency tolerance on the order of 100 ppm is required, care must be taken in the choice of external components (crystal and capacitors) as well as in the layout of the printed-circuit board. Several factors determine the frequency accuracy of a crystal-based oscillator circuit. Some of these factors are determined by the properties of the crystal itself. Generally, a low-cost, standard crystal will not be sufficient for a high-accuracy application, and a custom crystal must be specified. Most crystal manufacturers provide extensive information concerning the accuracy of their crystals, and an applications engineer from the crystal vendor should be consulted prior to specifying a crystal for a given application.

In addition to absolute, temperature, and aging tolerances of a crystal, the operating frequency of a crystal is also determined by the total load capacitance seen by the crystal. When ordering a crystal from a vendor, it is necessary to specify a load capacitance at which the operating frequency of the crystal will be measured. Variations in this load capacitance due to temperature and manufacturing variations will cause variations in the operating frequency of the oscillator. Figure 36 illustrates some of the sources of this variation.



Notes:
C_{ext} = external load capacitor (one each required for CKI and CKI2).
C_D = parasitic capacitance of the DSP1616 itself.
C_B = parasitic capacitance of the printed wiring board.
C_0 = parasitic capacitance of crystal (not part of C_L, but still a source of frequency variation).

Figure 36. Components of Load Capacitance for Crystal Oscillator

The load capacitance, C_L, must be specified to the crystal vendor. The crystal manufacturer will cut the crystal so that the frequency of oscillation will be correct when the crystal sees this load capacitance. Note that C_L refers to a capacitance seen across the crystal leads, meaning that for the circuit shown in Figure 36, C_L is the series combination of the two external capacitors ($C_{ext}/2$) plus the equivalent board and device strays ($C_B/2 + C_D/2$). For example, if 10 pF external capacitors were used and parasitic capacitance is neglected, then the crystal should be specified for a load capacitance of 5 pF. If the load capacitance deviates from this value due to the tolerance on the external capacitors or the presence of strays, then the frequency will also deviate. This change in frequency as function of load capacitance is known as "pullability" and is expressed in units of ppm/pF. For small deviations of a few pF, pullability can be determined by the equation below.

$$\text{pullability (ppm/pF)} = \frac{(C_1)(10^6)}{2(C_0 + C_L)^2}$$

where C_0 = parasitic capacitance of crystal in pF.
C_1 = motional capacitance of crystal in pF.
(usually between 1 fF to 25 fF, value available from crystal vendor).
C_L = total load capacitance in pF seen by crystal.

Crystal Electrical Characteristics and Requirements (continued)

Note that for a given crystal, the pullability can be reduced, and, hence, the frequency stability improved, by making C_L as large as possible while still maintaining sufficient negative resistance to ensure start-up per the curves shown in Figure 33.

Since it is not possible to know the exact values of the parasitic capacitance in a crystal-based oscillator system, the external capacitors are usually selected empirically to null out the frequency offset on a typical prototype board. Thus, if a crystal is specified to operate with a load capacitance of 10 pF, the external capacitors would have to be made slightly less than 20 pF each in order to account for strays. Suppose, for instance, that a crystal for which C_L = 10 pF is specified is plugged into the system and it is determined empirically that the best frequency accuracy occurs with C_{ext} = 18 pF. This would mean that the equivalent board and device strays from each lead to ground would be 2 pF.

As an example, suppose it is desired to design a 26 MHz system with ±100 ppm frequency accuracy. The parameters for a typical high-accuracy, custom, 26 MHz fundamental mode crystal are as follows:

Initial Tolerance	10 ppm
Temperature Tolerance	25 ppm
Aging Tolerance	6 ppm
Series Resistance	20 Ω max.
Motional Capacitance (C_1)	15 fF max.
Parasitic Capacitance (C_0)	7 pF max.

In order to ensure oscillator start-up, the negative resistance of the oscillator with load and parasitic capacitance must be at least twice the series resistance of the crystal, or 40 Ω. Interpolating from Figure 33, external capacitors plus strays can be made as large as 33 pF while still achieving 40 Ω of negative resistance. Assume for this example that external capacitors are chosen so that the total load capacitance including strays is 33 pF per lead, or 16.5 pF total. Thus, a load capacitance, C_L = 16.5 pF would be specified to the crystal manufacturer.

From the above equation, the pullability would be calculated as follows:

$$\text{pullability (ppm/pF)} = \frac{(C_1)(10^6)}{2(C_0 + C_L)^2} = \frac{(0.015)(10^6)}{2(7 + 16.5)^2} = 13.6 \text{ ppm/pF}$$

If 2% external capacitors are used, the frequency deviation due to capacitor tolerance is equal to

$$(0.02)(16.5 \text{ pF})(13.6 \text{ ppm/pF}) = 4.5 \text{ ppm}$$

Note: To simplify analysis, C_{ext} is considered to be 33 pF. In practice, it would be slightly less than this value to account for strays. Also, temperature and aging tolerances on the capacitors have been neglected.

Typical capacitance variation of the oscillator circuit in the DSP1616 itself across process, temperature, and supply voltage is ±1 pF. Thus, the expected frequency variation due to the DSP1616 is:

$$(1 \text{ pF})(13.6 \text{ ppm/pF}) = 13.6 \text{ ppm}$$

Approximate variation in parasitic capacitance of crystal = ±0.5 pF.
Frequency shift due to variation in C_0 = (0.5 pF)(13.6 ppm/pF) = 6.79 ppm

Approximate variation in parasitic capacitance of printed-circuit board = ±1.5 pF
Frequency shift due to variation in board capacitance = (1.5 pF)(13.6 ppm/pF) = 20.4 ppm

Thus, the contributions to frequency variation add up as follows:

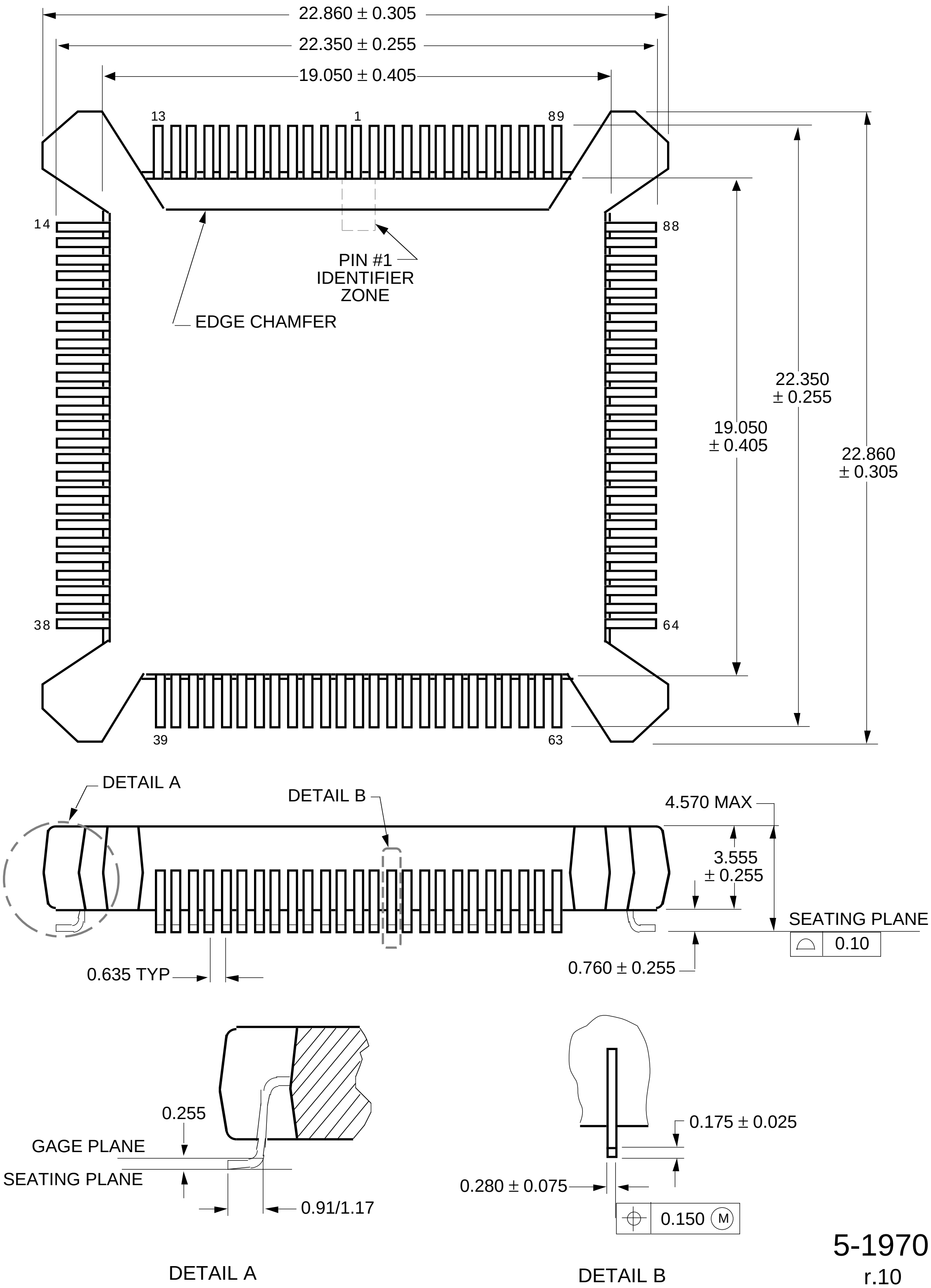
Initial Tolerance of Crystal	10.0 ppm
Temperature Tolerance of Crystal	25.0
Aging Tolerance of Crystal	6.0
Load Capacitor Variation	4.5
DSP1616 Circuit Variation	13.6
C_0 Variation	6.8
Board Variation	20.4
Total	86.3 ppm

This type of detailed analysis should be performed for any crystal-based application where frequency accuracy is critical.

12 Outline Diagrams

12.1 100-Pin BQFP (Bumpered Quad Flat Pack)

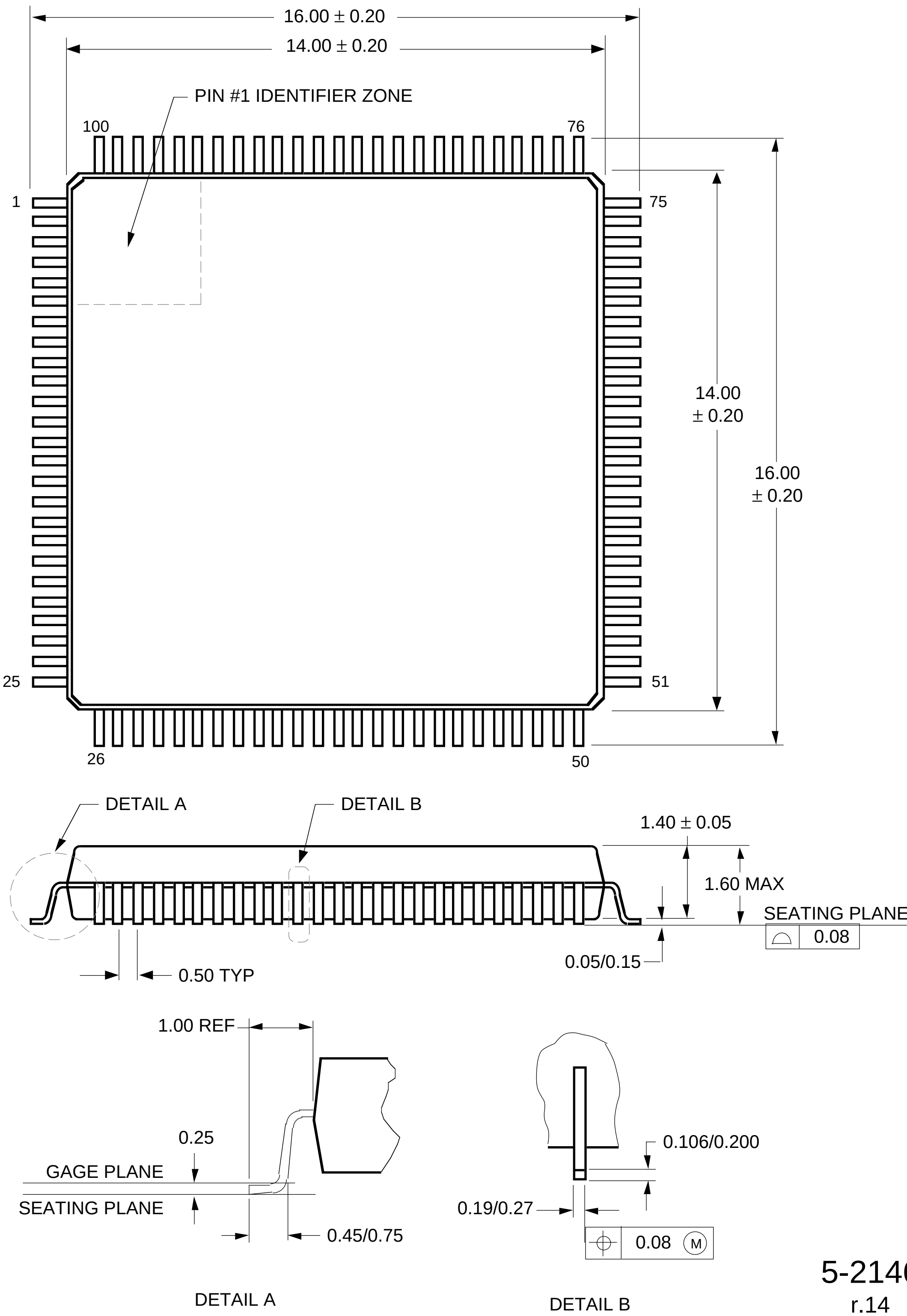
Dimensions are in millimeters.



Outline Diagrams (continued)

12.2 100-Pin TQFP (Thin Quad Flat Pack)

Dimensions are in millimeters.



Notes

Notes

For additional information, contact your AT&T Account Manager or the following:
U.S.A.: AT&T Microelectronics, 555 Union Boulevard, Room 21Q-133BA, Allentown, PA 18103
1-800-372-2447, FAX 610-712-4106 (In CANADA: **1-800-553-2448**, FAX 610-712-4106)
ASIA PACIFIC: AT&T Microelectronics Asia/Pacific, 14 Science Park Drive, #03-02A/04 The Maxwell, Singapore 0511
Tel. (65) 778-8833, FAX (65) 777-7495
JAPAN: AT&T Microelectronics, AT&T Japan Ltd., 7-18, Higashi-Gotanda 2-chome, Shinagawa-ku, Tokyo 141, Japan
Tel. (81) 3-5421-1600, FAX (81) 3-5421-1700
For data requests in Europe:
AT&T DATALINE: **Tel. (44) 1734 324 299**, FAX (44) 1734 328 148
For technical inquiries in Europe:
CENTRAL EUROPE: **(49) 89 95086 0** (Munich), NORTHERN EUROPE: **(44) 1344 487 111** (Bracknell UK),
FRANCE: **(33) 1 47 67 47 67** (Paris), SOUTHERN EUROPE: **(39) 2 6601 1800** (Milan) or **(34) 1 807 1700** (Madrid)

September 1995
DS94-189WDSP (Replaces DS93-110DSP)

AT&T
Microelectronics